

**PENGEMBANGAN APLIKASI ARSIP DIGITAL BERBASIS
OCR DAN *RULE-BASED SYSTEM***

SKRIPSI



**ECI WULANDARI
2022230032**

**PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH
2026**

PENGEMBANGAN APLIKASI ARSIP DIGITAL BERBASIS OCR DAN *RULE-BASED SYSTEM*

Untuk Memenuhi Salah Satu Syarat Memperoleh Gelar Sarjana
Program Studi Informatika Universitas Prabumulih



ECI WULANDARI
2022230032

**FAKULTAS ILMU KOMPUTER
PROGRAM STUDI INFORMATIKA
UNIVERSITAS PRABUMULIH
2026**

HALAMAN PENGESAHAN

PENGEMBANGAN APLIKASI ARSIP DIGITAL BERBASIS OCR DAN RULE-BASED SYSTEM

SKRIPSI

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana
Program Studi Informatika Universitas Prabumulih

Oleh :

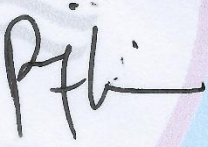
Eci Wulandari
2022230032

Prabumulih, 2 Juni 2026

Pembimbing I

Pembimbing II


Iwan Setiawan, S.Kom., M.Kom.
NUPTK. 1541769670130303


Riza Kartina, S.Kom., M.Kom.
NUPTK. 0549767668231073

Ketua Program Studi
Informatika


Iwan Setiawan, S.Kom., M.Kom.
NUPTK. 1541769670130303

HALAMAN PERSETUJUAN

Skripsi ini dengan judul “Pengembangan Aplikasi Arsip Digital Berbasis OCR dan *Rule-Based System*” telah dipertahankan dihadapan Tim Penguji Skripsi Program Studi Informatika Fakultas Ilmu Komputer Universitas Prabumulih pada tanggal 11 Juni 2026.

Prabumulih, 11 Juni 2026

Tim Penguji Tugas Akhir

Ketua:

Iwan Setiawan, S.Kom., M.Kom
NUPTK. 1541769670130303

()

Anggota:

1. Riza Kartina, S.Kom., M.Kom
NUPTK. 0549767668231073

()

2. Khana Wijaya, S.Kom., M.Kom
NUPTK. 7146768669130363

()

**Mengetahui,
Dekan Fakultas Ilmu Komputer**



(Khana Wijaya, S.Kom., M.Kom)
NUPTK. 7146768669130363

**Ketua Program Studi
Informatika**



(Iwan Setiawan, S.Kom., M.Kom)
NUPTK. 1541769670130303

PERNYATAAN INTEGRITAS

Saya yang bertanda tangan di bawah ini

Nama : Eci Wulandari

NIM : 2022230032

Program Studi : Informatika

Menyatakan dengan sebenarnya bahwa, skripsi ini benar-benar merupakan hasil karya saya dan tidak terdapat karya orang lain, apabila di kemudian hari terbukti atau dapat dibuktikan bahwa tugas akhir ini hasil *plagiasi*, maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku atas perbuatan tersebut.



Prabumulih, 2 Juni 2026

Penulis



Eci Wulandari

NIM. 2022230032

ABSTRAK

Pengembangan Aplikasi Arsip Digital Berbasis OCR Dan Rule-Based System

Eci Wulandari, 2022230032, 2026

Administrasi pengarsipan surat merupakan kegiatan penting dalam tata kelola dokumen di lingkungan perguruan tinggi. Proses yang masih dilakukan secara manual sering menyebabkan keterlambatan, penumpukan arsip, dan risiko kehilangan dokumen. Penelitian ini bertujuan untuk mengembangkan sistem arsip digital berbasis *Optical Character Recognition (OCR)* dan *Rule-Based System*. Teknologi *OCR* digunakan untuk mengekstraksi teks dari dokumen digital seperti surat keluar/masuk, sedangkan *Rule-Based System* berperan dalam menentukan validitas dan klasifikasi dokumen berdasarkan aturan administrasi kampus. Hasil penelitian diharapkan dapat mendukung proses pengarsipan dan validasi surat di Universitas Prabumulih secara lebih efisien, akurat, dan terintegrasi, sekaligus menjadi model penerapan AI ringan dalam pengelolaan dokumen institusi pendidikan.

Kata Kunci : Pengarsipan Digital, *Optical Character Recognition*, *Rule-Based System*

ABSTRACT

Development of a Digital Archiving Application Based on OCR and Rule-Based System

Eci Wulandari, 2022230032, 2026

Letter archiving administration is an essential activity in document management within higher education institutions. The manual process that is still commonly used often leads to delays, document accumulation, and a higher risk of document loss. This study aims to develop a digital archiving system based on Optical Character Recognition (OCR) and a Rule-Based System. The OCR technology is utilized to extract text from digital documents such as incoming and outgoing letters, while the Rule-Based System functions to determine the validity and classification of documents according to the university's administrative rules. The results of this research are expected to support the archiving and validation process of letters at the University of Prabumulih more efficiently, accurately, and in an integrated manner, as well as to serve as a model for lightweight AI implementation in document management for educational institutions.

Keywords: Digital Archiving, Optical Character Recognition, Rule-Based System

KATA PENGANTAR

Puji dan syukur penulis ucapkan kehadirat Allah SWT atas limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan Skripsi yang berjudul *Pengembangan Aplikasi Arsip Digital Berbasis OCR Dan Rule-Based System*, Skripsi ini merupakan salah satu syarat untuk mendapatkan gelar Sarjana di Program Studi S1 Informatika Fakultas Ilmu Komputer Universitas Prabumulih. Penulis banyak menerima bantuan, arahan, dan bimbingan dari berbagai pihak. Penulis mengucapkan terima kasih kepada:

1. Ibu Dr. Yuniar Pratiwi, S.Si., M.Si selaku Rektor Universitas Prabumulih yang telah memberikan izin untuk melakukan penelitian di Universitas Prabumulih.
2. Bapak Andi Christian, S.Kom., M.Kom selaku Wakil Rektor Universitas Prabumulih.
3. Bapak Khana Wijaya, S.Kom., M.Kom selaku Dekan Fakultas Ilmu Komputer Universitas Prabumulih.
4. Ibu Suhartini, S.Kom., M.Kom selaku Wakil Dekan Bidang Keuangan Fakultas Ilmu Komputer Universitas Prabumulih.
5. Ibu Nur Aini Hutagalung, S.Kom., M.Si., M.Kom selaku Wakil Dekan Bidang Kemahasiswaan Fakultas Ilmu Komputer Universitas Prabumulih.
6. Bapak Iwan Setiawan, S.Kom., M.Kom selaku Ketua Program Studi Informatika sekaligus Pembimbing I yang telah memberikan arahan, bimbingan, serta kepercayaan dan kesempatan kepada penulis dalam proses penyusunan Skripsi dan dalam masa perkuliahan.
7. Ibu Riza Kartina, S.Kom., M.Kom selaku Pembimbing II yang telah memberikan bimbingan, arahan, serta kepercayaan dan dukungan kepada penulis dalam proses penyusunan Skripsi dan dalam masa perkuliahan.
8. Bapak dan Ibu Dosen Program Studi Informatika Fakultas Ilmu Komputer Universitas Prabumulih.
9. Bapak/Ibu staf administrasi Fakultas Ilmu Komputer Universitas Prabumulih.
10. Staf Administrasi Rektorat Universitas Prabumulih yang telah memberikan izin dan membantu dalam penelitian ini.

11. Kepada kedua orang tua, terutama ibunda tercinta yang selalu memberikan doa, dukungan, semangat, serta motivasi kepada penulis selama proses perkuliahan hingga penyusunan skripsi ini. Terimakasih telah memberi kepercayaan bahwa penulis akan mampu melalui perjalanan sejauh ini. Terimakasih telah terus mengusahakan yang terbaik, bahkan dalam keadaan sesempit apa pun. Terimakasih atas semua pengorbanan dan lelah yang dikerahkan demi menghadirkan pendidikan terbaik kepada penulis.
12. Kepada pribadi yang telah bertahan melewati berbagai letih, keraguan, dan tekanan selama perjalanan perkuliahan ini, serta mau untuk terus belajar menguatkan diri dan melangkah dengan kemampuannya sendiri hingga mampu menuntaskan perjuangan ini sampai akhir, yang tak lain adalah diri penulis sendiri.
13. Kepada dua orang teman yang kutemui dalam satu tahun terakhir masa perkuliahan ini, Ajeng dan Sinta, terima kasih atas segala bantuan, waktu, dan momen-momen kecil yang telah dibagikan selama proses penyusunan skripsi ini. Terima kasih karena telah bersedia saling membantu dan saling menguatkan hingga sampai di tahap ini. Semoga kebaikan selalu menyertai langkah kalian, dan semoga kita dipertemukan dengan banyak hal baik di masa yang akan datang.

Penulis menyadari bahwa Skripsi ini masih terdapat kekurangan sehingga membutuhkan kritik dan saran yang bersifat membangun untuk kesempurnaan Skripsi ini.

Prabumulih, 2 Juni 2026

Penulis



Eci Wulandari

2022230032

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
HALAMAN PERSETUJUAN	iii
PERNYATAAN INTEGRITAS	iv
ABSTRAK.....	v
ABSTRACT	vi
KATA PENGANTAR.....	vii
DAFTAR ISI	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR.....	xii
DAFTAR LAMPIRAN	xv
DAFTAR SINGKATAN	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	5
1.6 Sistematika Penulisan	5
BAB II TINJAUAN PUSTAKA	7
2.1 Pengertian Pengembangan	7
2.2 Pengertian Aplikasi.....	8
2.3 Pengertian Arsip	9
2.4 Pengertian Digital	9
2.5 Pengertian Berbasis.....	10
2.6 Pengertian <i>Optical Character Recognition</i>	11
2.7 Pengertian <i>Rule-Based System</i>	12
2.8 Pengertian <i>Python</i>	13
2.9 Pengertian <i>Flutter</i>	14
2.10 Pengertian <i>MySQL</i>	14
2.11 Penelitian Terdahulu	15

2.12 Kerangka Berpikir Penelitian	18
BAB III METODE PENELITIAN	21
3.1 Objek dan Jadwal Penelitian	21
3.2 Metode Penelitian	27
3.3 Sumber Data	27
3.4 Teknik Pengumpulan Data	28
3.5 Metode Pengembangan Sistem.....	30
3.6 Alat Bantu Perancangan.....	32
3.7 Pengujian Sistem	37
BAB IV ANALISA DAN PERANCANGAN.....	39
4.1 Analisa Sistem	39
4.2 Perancangan Sistem yang Diusulkan	45
BAB V HASIL DAN PEMBAHASAN	90
5.1 Spesifikasi Perangkat	90
5.2 Implementasi <i>Database</i>	91
5.3 Tampilan Antarmuka	97
5.4 Hasil Pengujian Sistem	115
BAB VI KESIMPULAN DAN SARAN.....	178
6.1 Kesimpulan.....	178
6.2 Saran	178
DAFTAR PUSTAKA.....	180
LAMPIRAN	185

DAFTAR TABEL

	Halaman
Tabel 2.1 Penelitian Terdahulu.....	15
Tabel 3.2 Simbol <i>Usecase</i>	33
Tabel 3.3 Simbol <i>Activity Diagram</i>	35
Tabel 3.4 Simbol <i>ERD</i>	36
Tabel 4.1 <i>User</i>	67
Tabel 4.2 <i>Role</i>	68
Tabel 4.3 Unit.....	69
Tabel 4.4 Surat.....	70
Tabel 4.5 Disposisi.....	71
Tabel 4.6 <i>Reset Password</i>	72
Tabel 4.7 <i>Test case</i> Fungsi <i>Upload</i>	134
Tabel 5.1 Identifikasi <i>Statement</i> Fungsi <i>Login</i>	118
Tabel 5.2 <i>Test case</i> Fungsi <i>Login</i>	124
Tabel 5.3 Identifikasi <i>Statement</i> Fungsi <i>Upload</i>	127
Tabel 5.4 <i>Statement</i> Fungsi <i>Preprocessing</i> Dokumen.....	137
Tabel 5.5 <i>Test case</i> Fungsi <i>Preprocessing</i> Dokumen.....	144
Tabel 5.6 Identifikasi <i>Statement</i> Fungsi Ekstraksi Teks dengan <i>OCR</i>	146
Tabel 5.7 <i>Test case</i> Fungsi Ekstraksi Teks dengan <i>OCR</i>	150
Tabel 5.8 Identifikasi <i>Statement</i> Fungsi <i>Rule-Based System</i>	156
Tabel 5.9 <i>Test case</i> Fungsi <i>Rule-Based System</i>	164
Tabel 5.10 Fungsi Simpan Arsip	169
Tabel 5.11 <i>Test case</i> Fungsi Simpan Arsip	176

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Kerangka Berpikir Penelitian	19
Gambar 3.2 Struktur Organisasi	24
Gambar 3.3 Metode Pengembangan Sistem <i>Design Thinking</i>	30
Gambar 4.1 <i>Use case</i>	46
Gambar 4.2 <i>Login Activity</i>	48
Gambar 4.3 Tambah <i>User Activity</i>	49
Gambar 4.4 <i>Upload Surat Masuk Operator Activity</i>	51
Gambar 4.5 Arsip Surat Masuk Operator Activity	53
Gambar 4.6 <i>Upload Surat Keluar Operator Activity</i>	55
Gambar 4.7 <i>Activity</i> Buat Surat Keluar.....	57
Gambar 4.8 Arsip Surat Keluar Operator Activity.....	59
Gambar 4.9 Arsip Surat Masuk Pimpinan <i>Activity</i>	61
Gambar 4.10 Arsip Surat Keluar Pimpinan <i>Activity</i>	63
Gambar 4.11 ERD.....	65
Gambar 4.12 Halaman <i>Login</i>	73
Gambar 4.13 Halaman <i>Dashboard</i> Operator	74
Gambar 4.14 Halaman <i>Upload</i> Surat Masuk Operator.....	75
Gambar 4.15 Halaman Hasil Ekstraksi Surat Masuk.....	76
Gambar 4.16 Halaman <i>Upload</i> Surat Keluar Operator.....	77
Gambar 4.17 Halaman Hasil Ekstraksi Surat Keluar.....	78
Gambar 4.18 Halaman Editor Surat Keluar	79
Gambar 4.19 Halaman Daftar Arsip Surat Masuk.....	80
Gambar 4.20 Halaman Daftar Arsip Surat Keluar.....	81
Gambar 4.21 Halaman <i>Dashboard</i> Pimpinan	82
Gambar 4.22 Halaman Daftar Arsip Surat Masuk Pimpinan.....	83
Gambar 4.23 Tampilan Form Disposisi	84
Gambar 4.24 Tampilan Lihat Disposisi	85
Gambar 4.25 Tampilan <i>Inbox</i> Disposisi.....	86

Gambar 4.26 Halaman Daftar Arsip Surat Keluar Pimpinan.....	87
Gambar 4.27 Halaman Dashboard Administrator	88
Gambar 4.28 Halaman Manajemen User	89
Gambar 5.1 Tabel <i>Users</i>	92
Gambar 5.2 Tabel <i>Role</i>	93
Gambar 5.3 Tabel Unit.....	93
Gambar 5.4 Tabel Surat	94
Gambar 5.5 Tabel Disposisi	95
Gambar 5.6 Tabel Reset Password	96
Gambar 5.7 Halaman <i>Login</i>	97
Gambar 5.8 Halaman Dashboard.....	98
Gambar 5.9 Halaman <i>Upload</i> Surat Masuk	99
Gambar 5.10 Halaman Form <i>Upload</i> Surat Masuk	100
Gambar 5.11 Halaman <i>Upload</i> Surat Keluar	101
Gambar 5.12 Halaman Form <i>Upload</i> Surat Keluar	102
Gambar 5.13 Halaman Editor Surat Keluar	103
Gambar 5.14 Halaman Arsip Surat Masuk Sisi Operator	104
Gambar 5.15 Halaman Arsip Surat Keluar Sisi Operator	105
Gambar 5.16 halaman Dashboard Pimpinan.....	106
Gambar 5.17 Halaman Arsip Surat Masuk Sisi Pimpinan	107
Gambar 5.18 Form Disposisi.....	108
Gambar 5.19 Tampilan Lihat Disposisi	109
Gambar 5.20 Halaman Inbox Disposisi.....	110
Gambar 5.21 Halaman Daftar Arsip Surat Keluar Sisi Pimpinan	111
Gambar 5.22 Halaman Dashboard Administrator	112
Gambar 5.23 Halaman Manajemen User	113
Gambar 5.24 halaman Manajemen unit	114
Gambar 5.25 <i>Flowgraph</i> Fungsi <i>Login</i>	121
Gambar 5.26 Flowchart Fungsi <i>Upload</i>	129
Gambar 5.27 <i>Flowgraph</i> Fungsi <i>Upload</i>	130
Gambar 5.28 <i>Flowchart</i> Fungsi <i>Preprocessing</i> Dokumen.....	139
Gambar 5.29 <i>Flowgraph</i> Fungsi <i>Preprocessing</i> Dokumen	141

Gambar 5.30 <i>Flowchart</i> Fungsi Ekstraksi Teks dengan <i>OCR</i>	147
Gambar 5.31 <i>Flowgraph</i> Fungsi Ekstraksi Teks dengan <i>OCR</i>	148
Gambar 5.32 <i>Flowchart</i> Fungsi <i>Rule-Based System</i>	158
Gambar 5.33 <i>Flowgraph</i> Fungsi <i>Rule-Based System</i>	160
Gambar 5.34 <i>Flowchart</i> Fungsi Simpan Arsip	171
Gambar 5.35 <i>Flowgraph</i> Fungsi Simpan Arsip.....	173



DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Surat Permohonan Penelitian.....	185
Lampiran 2 Surat Balasan Permohonan Penelitian.....	186
Lampiran 3 Dokumentasi Wawancara & Observasi.....	187
Lampiran 4 Daftar Pertanyaan Wawancara.....	190
Lampiran 5 Berita Acara Demonstrasi Aplikasi.....	194



DAFTAR SINGKATAN

AI	: <i>Artificial intelligence</i>
ERD	: <i>Entity Relation Diagram</i>
FEB	: Fakultas Ekonomi dan Bisnis
FT	: Fakultas Teknik
FASILKOM	: Fakultas Ilmu Komputer
OCR	: <i>Optical Character Recognition</i>
PDF	: <i>Portable Document Format</i>
RBAC	: <i>Role-Based Access Control</i>
STMIK	: Sekolah Tinggi Manajemen Informatika Dan Komputer
STIE	: Sekolah Tinggi Ilmu Ekonomi
STIT	: Sekolah Tinggi Ilmu Teknik
UNPRA	: Universitas Prabumulih



BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi telah membawa perubahan besar terhadap sistem administrasi di berbagai lembaga, termasuk institusi pendidikan tinggi. Digitalisasi kini menjadi tuntutan agar proses akademik dan tata kelola kampus dapat berjalan lebih efisien, transparan, dan mudah diakses. Transformasi digital di lingkungan universitas tidak hanya berfokus pada penerapan teknologi baru, tetapi juga mencakup perubahan paradigma kepemimpinan, budaya organisasi, serta sistem manajemen yang mendukung efektivitas kerja di era digital. Dengan demikian, penerapan sistem berbasis teknologi pada pengelolaan dokumen akademik dan administrasi menjadi langkah penting untuk meningkatkan efisiensi, akurasi, serta keamanan arsip dibandingkan dengan praktik manual tradisional. Berbagai studi juga menunjukkan bahwa integrasi teknologi digital mampu mengubah cara institusi mengelola, menyimpan, dan menyediakan akses terhadap arsip maupun data administratifnya (Sulkowski, 2023).

Universitas Prabumulih merupakan salah satu institusi pendidikan tinggi di Kota Prabumulih, Sumatera Selatan, yang memiliki tiga fakultas, yaitu Fakultas Ekonomi dan Bisnis (FEB), Fakultas Teknik (FT), dan Fakultas Ilmu Komputer (FASILKOM), dengan delapan program studi di dalamnya, antara lain; Manajemen, Akuntansi, Bisnis Digital, Teknik Pertambangan, Teknik Lingkungan, Sistem Informasi, Komputerisasi Akuntansi, dan Informatika. Sebagai universitas yang terus berkembang, institusi ini berkomitmen untuk menerapkan teknologi informasi dalam mendukung kegiatan akademik maupun administrasi. Penerapan sistem administrasi berbasis teknologi diharapkan tidak hanya mempercepat proses pelayanan internal, tetapi juga memperkuat transparansi dan akurasi data antar unit kerja. Dengan demikian, Universitas Prabumulih memiliki peluang besar untuk mengimplementasikan sistem digital yang mampu mendukung efektifitas kegiatan

operasional kampus dan mendukung agenda transformasi diital di sektor pendidikan tinggi.

Dalam konteks sistem administrasi modern, penerapan teknologi digital menjadi bagian penting dalam meningkatkan efisiensi dan ketepatan pengelolaan dokumen. Salah satu teknologi yang banyak dimanfaatkan adalah *Optical Character Recognition (OCR)*, yaitu sistem yang mampu mengenali dan mengekstraksi teks dari dokumen hasil pindai agar dapat diolah secara digital (Panjaitan et al., 2024). Teknologi ini membantu proses konversi dokumen fisik ke bentuk digital dengan tingkat akurasi yang cukup baik. Seiring dengan kemajuan *deep learning*, *OCR* modern dapat mengenali teks dari dokumen dengan berbagai kondisi pencahayaan, tata letak, dan format. Implementasinya pada aplikasi *mobile* juga memberikan kemudahan bagi pengguna dalam melakukan ekstraksi teks secara langsung melalui kamera perangkat, sehingga proses digitalisasi dokumen menjadi lebih praktis dan efisien.

Selain *OCR*, *Rule-Based System* merupakan pendekatan yang digunakan untuk membantu proses otomatisasi pengambilan keputusan berdasarkan seperangkat aturan logis. *Rule-Based System* bekerja dengan prinsip *if-then rules* yang mengatur bagaimana sistem menafsirkan data dan menentukan keluaran tertentu. Dalam konteks administrasi digital, *Rule-Based System* cocok untuk diterapkan dalam mengelompokkan dokumen, menyeleksi informasi relevan, serta menyesuaikan data dengan kebijakan organisasi. Penerapan *Rule-Based System* dalam sistem informasi berpotensi meningkatkan konsistensi dan efisiensi dalam proses klasifikasi serta validasi data tanpa banyak memerlukan intervensi manual yang berlebihan. Melalui penerapan tersebut, sistem administrasi digital dapat berjalan lebih teratur dan mendukung keakuratan pengelolaan dokumen di lingkungan kampus (Pramono & Soekamto, 2024).

Meskipun sistem berbasis *web* telah banyak digunakan untuk mendukung administrasi kampus, penerapan sistem arsip digital yang bersifat cerdas masih jarang dikembangkan. Padahal, kebutuhan terhadap fleksibilitas akses dan kecepatan pencarian dokumen arsip semakin meningkat. Pengembangan *smart web* berbasis *OCR* dan *Rule-Based System* diharapkan mampu menjawab tantangan tersebut dengan menyediakan sistem yang mampu mengekstraksi, mengenali, dan

mengklasifikasikan dokumen arsip secara otomatis untuk mempermudah proses pencarian serta pengarsipan data.

Proses administrasi seperti pengelolaan surat masuk dan keluar di Universitas Prabumulih khususnya di unit rektorat masih dilakukan secara manual dan terpisah antar unit. Kondisi ini menyebabkan penumpukan dokumen fisik yang memerlukan ruang penyimpanan semakin besar, sehingga menyulitkan proses pencarian arsip, memperlambat pelayanan administrasi, serta meningkatkan risiko duplikasi dan hilangnya data. Selain itu berdasarkan hasil wawancara, arsip disimpan dengan masa simpan minimal lima (5) tahun. Setelah melewati masa simpan tersebut, dokumen fisik perlu dilakukan penyusutan melalui pemindahan dokumen ke gudang atau dilakukan pemusnahan sesuai ketentuan kearsipan, sementara informasi yang terkandung di dalam arsip tersebut masih berpotensi diperlukan di masa mendatang. Oleh karena itu, penerapan sistem arsip digital menjadi solusi yang relevan untuk mengurangi ketergantungan terhadap penyimpanan fisik, mendukung proses alih media arsip sebelum pemusnahan, serta menyediakan media penyimpanan arsip yang lebih fleksibel dan terintegrasi antar unit kerja di lingkungan Universitas Prabumulih.

Berdasarkan latar belakang tersebut, penulis mengusulkan pengembangan aplikasi arsip digital berbasis *Optical Character Recognition (OCR)* dan *Rule-Based System*. Pengembangan aplikasi ini diharapkan dapat membantu staf administrasi rektorat dalam mengelola arsip surat masuk dan keluar secara digital, efisien, dan terintegrasi. Melalui penerapan sistem ini, proses administrasi di unit rektorat Universitas Prabumulih diharapkan menjadi lebih cepat, akurat, serta mendukung implementasi transformasi digital di lingkungan kampus.

1.2 Rumusan Masalah

Untuk mencapai tujuan penelitian dan memastikan pengembangan sistem dilakukan secara terarah, maka diperlukan perumusan masalah yang menjadi dasar dari proses penelitian ini.

1. Bagaimana merancang sistem arsip digital berbasis *OCR* dan *Rule-Based System* yang dapat meningkatkan efisiensi dan mengotomatisasi

proses penyimpanan arsip surat masuk dan surat keluar di unit rektorat Universitas Prabumulih?

2. Bagaimana sistem dapat menjamin integrasi dan kemudahan akses arsip staf administrasi melalui *platform* digital?
3. Bagaimana efektivitas sistem dalam mendukung transformasi pengelolaan arsip manual berbasis dokumen fisik menjadi arsip digital terpusat di unit rektorat Universitas Prabumulih?

1.3 Batasan Masalah

Agar penelitian ini lebih terfokus dan tidak melebar dari konteks yang ditetapkan, maka diperlukan pembatasan ruang lingkup penelitian yang dijelaskan sebagai berikut.

1. Penelitian difokuskan pada pengembangan aplikasi arsip digital di lingkungan Universitas Prabumulih tepatnya pada unit rektorat yang mengelola pengarsipan dokumen seperti surat masuk dan surat keluar.
2. Penelitian ini menggunakan teknologi *OCR* untuk ekstraksi teks dari dokumen digital dan *Rule-Based System* untuk klasifikasi arsip berdasarkan aturan pengarsipan, yang dikembangkan dalam bentuk aplikasi berbasis *smart web*.
3. Data dan dokumen yang digunakan bersumber dari arsip administrasi nyata di unit rektorat Universitas Prabumulih, dengan penyesuaian untuk menjaga kerahasiaan informasi.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah disusun, maka tujuan dari penelitian ini adalah sebagai berikut:

1. Mengembangkan sistem arsip digital terintegrasi berbasis *OCR* dan *Rule-Based System* untuk mengotomatisasi proses digitalisasi, klasifikasi, dan pengelolaan dokumen arsip di Universitas Prabumulih.
2. Mewujudkan sistem arsip digital yang efisien, tertata, dan mudah diakses sebagai bentuk transformasi pengelolaan arsip manual berbasis dokumen fisik ke dalam sistem digital.

1.5 Manfaat Penelitian

Penelitian ini diharapkan bermanfaat bagi institusi, mahasiswa, dan akademisi di lingkungan pendidikan tinggi.

1. Bagi Institusi, penelitian ini diharapkan meningkatkan efisiensi penyimpanan arsip melalui penerapan sistem digital berbasis *OCR* dan *Rule-Based System*.
2. Bagi mahasiswa, penelitian ini menjadi sarana penerapan ilmu di bidang kecerdasan buatan, serta pengalaman langsung dalam merancang sistem arsip digital.
3. Bagi akademisi dan pengembang sistem, penelitian ini dapat menjadi referensi pengembangan sistem arsip digital berbasis *AI* di sektor pendidikan, yang mendukung transparansi dan transformasi digital di universitas.

1.6 Sistematika Penulisan

Penulisan skripsi ini disusun dalam beberapa bab sebagai berikut:

1. BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, serta sistematika penulisan.

2. BAB II TINJAUAN PUSTAKA

Bab ini memuat teori-teori yang relevan dengan penelitian, meliputi konsep arsip digital, teknologi *Optical Character Recognition (OCR)*, *Rule-Based System*, serta hasil penelitian terdahulu yang berkaitan dengan topik penelitian.

3. BAB III METODE PENELITIAN

Bab ini menjelaskan metode yang digunakan dalam penelitian dan pengembangan sistem, yang meliputi teknik pengumpulan data, metode pengembangan sistem, serta alat bantu yang digunakan dalam proses perancangan dan implementasi.

4. BAB IV ANALISA DAN PERANCANGAN

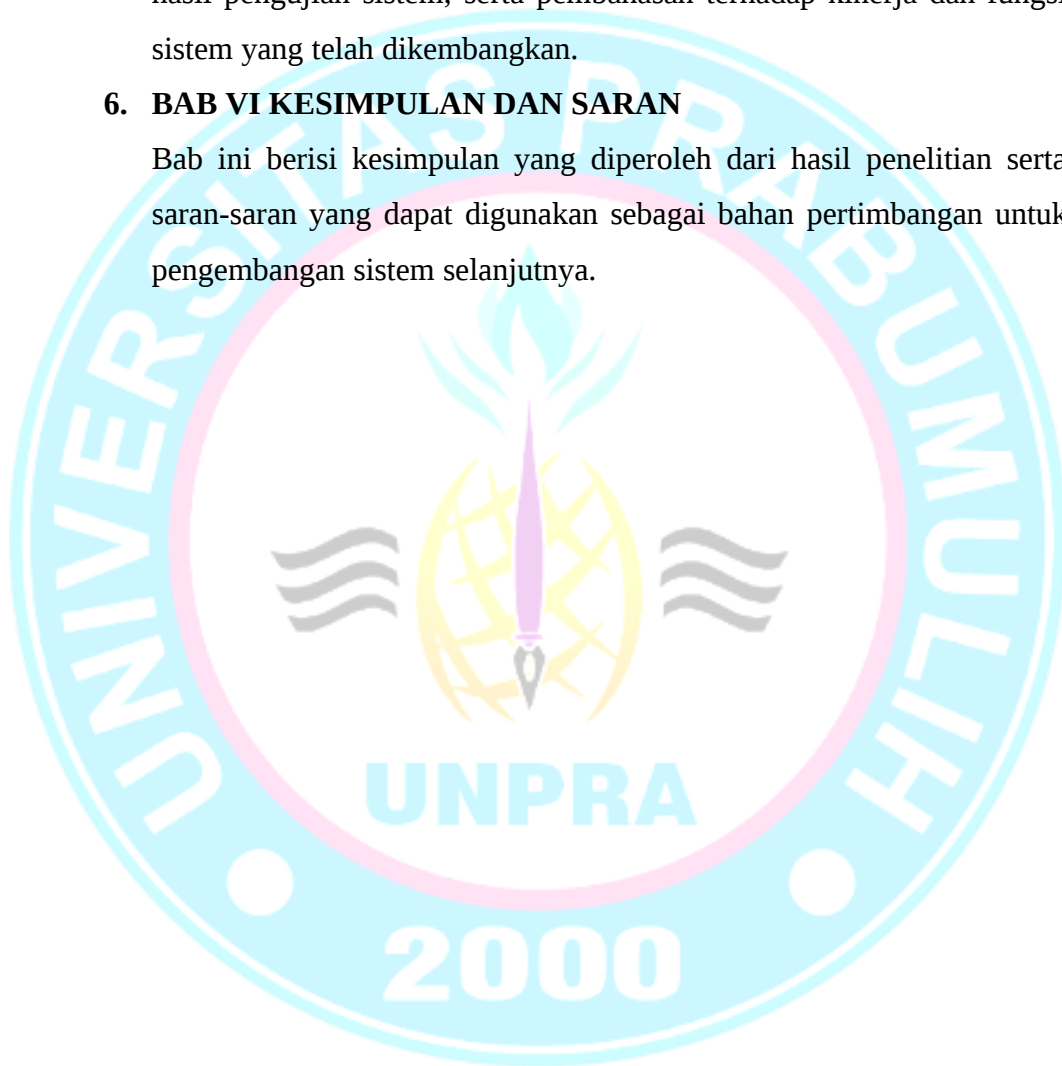
Bab ini membahas hasil analisis sistem yang sedang berjalan, evaluasi sistem, analisis kebutuhan, serta perancangan sistem yang diusulkan, termasuk perancangan basis data dan antarmuka pengguna.

5. BAB V HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil implementasi sistem, tampilan antarmuka, hasil pengujian sistem, serta pembahasan terhadap kinerja dan fungsi sistem yang telah dikembangkan.

6. BAB VI KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang diperoleh dari hasil penelitian serta saran-saran yang dapat digunakan sebagai bahan pertimbangan untuk pengembangan sistem selanjutnya.



BAB II

TINJAUAN PUSTAKA

2.1 Pengertian Pengembangan

Pengembangan perangkat lunak (*software development*) adalah proses yang sistematis dan bertahap, meliputi beberapa fase seperti perencanaan, analisis kebutuhan, perancangan (*design*), implementasi (penulisan kode), pengujian, hingga pemeliharaan sistem. Dalam studi *A Comprehensive Review of Software development Life Cycle methodologies: Pros, Cons, and Future Directions*, disebutkan “*the software development process consists of several sequential steps. Typically, these steps refer to all basic processes such as requirements analysis, design, implementation, testing, and maintenance*” (ALazzawi et al., 2023). Dengan demikian, pengembangan perangkat lunak tidak hanya sekedar ‘membuat program’, tetapi mencakup keseluruhan rangkaian aktivitas yang diperlukan agar perangkat lunak dapat berfungsi sesuai kebutuhan, terlacak, dan terpelihara.

Pengembangan perangkat lunak juga dapat dipandang sebagai proses yang berkelanjutan dan adaptif, di mana organisasi dan tim pengembang harus menyesuaikan diri dengan perubahan persyaratan, lingkungan, dan kebutuhan pengguna. Dalam artikel *Improving Software development with Continuous Integration and Deployment for Agile DevOps in Engineering Practices*, disebutkan bahwa “*software development in engineering practices is evolving rapidly, driven by the demands for efficiency, scalability, and adaptability*” (Kolawole & Fakokunde, 2024). Ini menunjukkan bahwa pengembangan *software* bukan hanya proses linear, melainkan juga membutuhkan fleksibilitas dan kesiapan perubahan agar perangkat lunak tetap relevan dan mampu memenuhi tuntutan pengguna serta organisasi.

Berdasarkan pemahaman tersebut, dapat disimpulkan bahwa pengembangan aplikasi merupakan suatu proses sistematis dan berkelanjutan yang tidak hanya berfokus pada pembuatan perangkat lunak, tetapi juga pada upaya memastikan sistem mampu beradaptasi terhadap kebutuhan pengguna dan

perubahan lingkungan teknologi. Proses ini menuntut pendekatan yang fleksibel, terstruktur, serta melibatkan evaluasi berulang agar aplikasi yang dihasilkan tidak hanya berfungsi sesuai rancangan awal, tetapi juga efektif, efisien, dan relevan dengan tujuan organisasi.

2.2 Pengertian Aplikasi

Aplikasi merupakan bagian dari perangkat lunak yang dirancang untuk membantu pengguna dalam melaksanakan tugas-tugas tertentu sesuai kebutuhan operasional. Aplikasi adalah perangkat lunak (*software*) atau program komputer yang dikembangkan untuk melakukan perintah tertentu. Istilah *application* berasal dari bahasa Inggris yang berarti penerapan atau penggunaan, sehingga secara harfiah aplikasi dapat dipahami sebagai penerapan perangkat lunak yang dibuat untuk menjalankan fungsi atau tugas tertentu (Nasution et al., 2023).

Dalam pengertian lain, aplikasi didefinisikan sebagai sebuah transformasi dari permasalahan atau pekerjaan yang awalnya sulit dipahami menjadi bentuk yang lebih sederhana, mudah diproses, dan dapat dimengerti oleh pengguna. Definisi ini menegaskan bahwa aplikasi tidak hanya berfungsi sebagai alat bantu, melainkan juga sebagai mekanisme penyederhanaan proses sehingga berbagai aktivitas, pengelolaan data, hingga layanan tertentu dapat dilaksanakan dengan lebih efektif melalui dukungan sistem komputasi (Yunita et al., 2022).

Berdasarkan kedua definisi tersebut, aplikasi dapat dipahami sebagai perangkat lunak yang dirancang untuk menjalankan fungsi tertentu sesuai kebutuhan pengguna sekaligus menyederhanakan proses yang kompleks menjadi lebih mudah dipahami dan dioperasikan. Aplikasi tidak hanya berperan sebagai program pelaksana tugas, tetapi juga sebagai media yang menghubungkan pengguna dengan sistem komputasi melalui mekanisme kerja yang terstruktur. Dengan demikian, dalam konteks pengembangan sistem informasi, aplikasi tidak hanya berfokus pada tampilan antarmuka, melainkan juga pada logika proses, struktur data, serta alur kerja internal yang memungkinkan otomatisasi, peningkatan efisiensi, dan dukungan terhadap pencapaian tujuan operasional secara optimal.

2.3 Pengertian Arsip

Arsip dipahami sebagai rekaman kegiatan atau peristiwa dalam berbagai bentuk dan media yang dibuat dan diterima oleh lembaga negara, pemerintah daerah, lembaga pendidikan, perusahaan, organisasi politik, maupun perorangan sesuai perkembangan teknologi informasi dan komunikasi. Arsip berfungsi sebagai memori atau pusat ingatan organisasi yang membantu pertanggungjawaban, akuntabilitas, dan pengambilan keputusan, melindungi hak karyawan dan warga negara serta melestarikan kesinambungan ilmu pengetahuan sepanjang masa, sehingga arsip perlu disimpan dan dikelola secara sistematis (Risparyanto, 2021).

Selain itu arsip juga didefinisikan sebagai informasi tentang aktivitas yang direkam (disimpan) dalam berbagai bentuk media mulai dari kertas, foto, hingga media elektronik yang berperan sebagai bukti sah aktivitas organisasi dan sumber informasi penting bagi administrasi serta pelestarian kontinuitas pengetahuan; definisi ini menekankan bahwa arsip bukan sekadar tempat penyimpanan tetapi rekaman aktivitas yang menuntut pengelolaan untuk efisiensi dan akuntabilitas (Pranata, 2024).

Berdasarkan kedua sumber di atas, arsip dapat diartikan sebagai rekaman aktivitas atau peristiwa yang tersimpan dalam berbagai bentuk media dan berfungsi sebagai memori institusi serta bukti yang mendukung akuntabilitas, pengambilan keputusan, dan pelestarian pengetahuan. Karena itulah arsip memerlukan pengelolaan yang sistematis baik penyusunan, penyimpanan, maupun aksesibilitas agar nilai informatif dan hukumnya tetap terjaga.

2.4 Pengertian Digital

Dalam kajian tentang transformasi dan pengelolaan informasi, istilah digital didefinisikan sebagai rangkaian teknologi dan proses yang memungkinkan konversi, penyimpanan, pengolahan, dan transmisi data secara elektronik meliputi artefak digital, *platform* digital, dan infrastruktur digital yang bukan hanya soal mengganti format analog ke digital tetapi juga perubahan proses dan model kerja organisasi yang memanfaatkan teknologi tersebut untuk efisiensi, kolaborasi *online*, dan pengelolaan informasi (Zulvikri et al., 2024).

Sementara itu, istilah digital dalam konteks kearsipan merujuk pada arsip yang dikelola dan disimpan dalam media digital, yaitu data yang direpresentasikan dalam bentuk kode biner dan dikelola menggunakan perangkat komputasi. Dengan demikian, arsip digital merupakan arsip yang diciptakan, disimpan diolah dan diakses melalui sistem digital, namun tetap memiliki karakteristik arsip sebagai bukti kegiatan dan informasi yang memiliki nilai guna di dalamnya (Pranata, 2024).

Berdasarkan kedua sumber tersebut, digital dapat disimpulkan sebagai ekosistem teknologi dan praktik yang melibatkan perangkat, *platform*, dan proses elektronik dalam kegiatan konversi, penyimpanan, pengolahan, dan transmisi data. Dalam konteks pengelolaan informasi, digital tidak hanya dipahami sebagai perubahan format dari analog ke digital, tetapi juga mencakup perubahan proses kerja dan sistem pengelolaan yang memanfaatkan teknologi digital untuk meningkatkan efisiensi, aksesibilitas, dan pemanfaatan informasi, termasuk dalam pengelolaan arsip.

2.5 Pengertian Berbasis

Dalam literatur sistem informasi kearsipan, istilah *berbasis* dipahami sebagai kondisi di mana sistem arsip dibangun dengan menjadikan teknologi tertentu sebagai fondasi utama operasional sistem. Hal ini terlihat pada pengembangan sistem informasi pengelolaan arsip digital berbasis *web*, di mana seluruh proses penyimpanan, pengelolaan, dan pencarian arsip dirancang sesuai dengan karakteristik teknologi *web* yang digunakan (I. P. Sari et al., 2022)

Dalam penelitian tentang aplikasi manajemen arsip surat yang menerapkan *Optical Character Recognition (OCR)*, istilah *berbasis* digunakan untuk menunjukkan bahwa teknologi *OCR* menjadi landasan utama dalam proses pengelolaan arsip digital. *OCR* tidak hanya berfungsi sebagai alat bantu, tetapi menentukan cara sistem membaca, mengubah, dan mengelola arsip dari dokumen fisik ke bentuk digital yang dapat diolah lebih lanjut. Dengan demikian, penggunaan istilah *berbasis OCR* dalam konteks kearsipan menandakan adanya ketergantungan langsung antara fungsi utama sistem arsip dan teknologi *OCR* sebagai fondasi operasionalnya (Iskandar et al., 2025).

Berdasarkan dua tinjauan di atas, istilah *berbasis* dalam konteks penelitian ini dapat disimpulkan sebagai penanda bahwa suatu sistem atau pendekatan dibangun dengan menjadikan teknologi tertentu sebagai fondasi konseptual dan operasional utama yang mengarahkan tujuan, mekanisme kerja, dan implementasi sistem. Dengan kata lain, penggunaan frasa *berbasis + X* menandakan bahwa *X* bukan sekadar alat pendukung, melainkan komponen inti yang memengaruhi cara proses berjalan dan hasil yang dicapai. Dalam penelitian ini, digitalisasi dan penyimpanan arsip dilakukan melalui aplikasi *web* yang berbasis teknologi, dengan fokus pada penerapan *Optical Character Recognition (OCR)* dan *Rule-Based System* sebagai landasan utama dalam proses pengelolaan arsip digital.

2.6 Pengertian *Optical Character Recognition*

Optical Character Recognition (OCR) merupakan teknologi yang digunakan untuk mengenali dan mengonversi karakter teks dari dokumen hasil pemindaian ke dalam bentuk teks digital yang dapat diproses oleh sistem komputer. Dalam penelitian *Implementasi Aplikasi Manajemen Arsip Surat Berbasis Optical Character Recognition Pada Badan Pusat Statistik Banyuwangi*, *OCR* diterapkan sebagai komponen inti dalam sistem pengelolaan arsip surat untuk mengekstraksi informasi teks dari dokumen fisik sehingga arsip dapat disimpan, dicari, dan dikelola secara digital. Penerapan *OCR* pada penelitian tersebut menunjukkan bahwa teknologi ini berperan penting dalam mempercepat proses digitalisasi arsip serta meningkatkan efisiensi pengelolaan dokumen surat yang sebelumnya dilakukan secara manual (Iskandar et al., 2025).

Dalam penelitian *Penerapan Teknologi Optical Character Recognition Pada Pengarsipan Dokumen (Studi Kasus: PT Pertamina Geothermal Energy Area Lahendong)*, di mana *OCR* digunakan sebagai dasar pengolahan arsip dokumen perusahaan. Dalam penelitian tersebut, *OCR* tidak hanya berfungsi untuk mengubah dokumen hasil pemindaian menjadi teks digital, tetapi juga menjadi landasan dalam proses pengindeksan dan pencarian arsip. Dengan adanya *OCR*, isi dokumen yang semula tidak dapat diolah secara langsung oleh sistem informasi arsip dapat diubah menjadi data tekstual yang terstruktur dan siap digunakan dalam sistem pengarsipan digital (Gosal & Rompas, 2023).

Berdasarkan kedua penelitian tersebut, *Optical Character Recognition (OCR)* dapat dipahami sebagai teknologi fundamental dalam digitalisasi arsip yang berfungsi sebagai penghubung antara dokumen fisik dan sistem arsip digital. Dalam konteks kearsipan, *OCR* tidak sekadar melakukan pengenalan karakter, tetapi menjadi fondasi operasional yang menentukan bagaimana dokumen diproses, disimpan, dan diakses kembali secara digital. Dengan demikian, penggunaan istilah *berbasis OCR* dalam sistem pengarsipan menunjukkan bahwa seluruh mekanisme pengelolaan arsip sangat bergantung pada kemampuan *OCR* dalam mengekstraksi dan mengubah informasi teks dari dokumen fisik ke dalam bentuk data digital yang dapat dikelola secara sistematis.

2.7 Pengertian *Rule-Based System*

Rule-Based System adalah sistem kecerdasan buatan (*artificial intelligence*) yang mengambil keputusan atau menyelesaikan masalah dengan mengikuti satu set aturan (*rules*) yang telah ditentukan sebelumnya. Aturan tersebut biasanya berbentuk kondisi (“*if-kondisi*”) dan aksi (“*then-aksi*”) yang dieksekusi ketika kondisi tercapai. Keunggulan *Rule-Based System* adalah kemampuan menjelaskan keputusan (*explainability*), karena setiap keputusan bisa dirunut ke aturan spesifik yang dilaksanakan (Kostopoulos et al., 2024).

Dalam konteks sistem pengarsipan digital, *Rule-Based System* digunakan sebagai dasar dalam proses klasifikasi dan pengelolaan arsip berbasis *web*. Pada penelitian mengenai implementasi algoritma *rule-based classification* pada aplikasi arsip *web*, sistem dirancang untuk mengelompokkan dan mengelola arsip berdasarkan aturan yang telah ditentukan sebelumnya. Aturan-aturan tersebut menjadi landasan utama dalam menentukan kategori arsip, alur pengelolaan, serta konsistensi proses pengarsipan, sehingga sistem dapat berjalan sesuai dengan kebijakan dan kebutuhan organisasi (Runtu et al., 2025).

Berdasarkan uraian tersebut, *Rule-Based System* dapat disimpulkan sebagai pendekatan kecerdasan buatan yang memformalisasi logika dan kebijakan ke dalam bentuk aturan terstruktur guna mendukung proses pengambilan keputusan secara sistematis. Dalam pengarsipan digital, *Rule-Based System* berperan sebagai fondasi logis yang mengatur proses klasifikasi, validasi, dan pengelolaan arsip berdasarkan

kriteria tertentu. Dengan demikian, penerapan *Rule-Based System* pada sistem arsip digital memungkinkan pengelolaan arsip dilakukan secara konsisten, terkontrol, dan mudah ditelusuri sesuai dengan aturan yang telah ditetapkan.

2.8 Pengertian Python

Python adalah bahasa pemrograman yang populer dan banyak digunakan di berbagai bidang, mulai dari pengembangan *web*, analisis data, kecerdasan buatan, hingga pengembangan perangkat lunak umum. *Python* diciptakan oleh Guido van Rossum dan pertama kali dirilis pada tahun 1991. *Python* dikenal dengan sintaks yang mudah dibaca dan dipahami, serta dukungan komunitas yang luas (Ardiansah & Permana, 2023).

Python adalah bahasa pemrograman yang menggunakan interpreter untuk menjalankan kode programnya. Interpreter tersebut dapat menerjemahkan kode secara langsung, dan *Python* dapat dijalankan di berbagai *platform* seperti Windows, Linux, dan lain-lain. *Python* menggabungkan berbagai paradigma pemrograman yang diambil dari beberapa bahasa lain, seperti paradigma prosedural yang mirip dengan C, paradigma berorientasi objek seperti pada *Java*, serta paradigma fungsional sebagaimana pada *Lisp*. Perpaduan pendekatan ini membuat *Python* fleksibel dan memudahkan *programmer* dalam membangun beragam jenis aplikasi dan proyek (Rahman et al., 2023).

Berdasarkan uraian tersebut, dapat disimpulkan bahwa *Python* merupakan bahasa pemrograman yang populer, fleksibel, dan mudah digunakan, dengan sintaks yang sederhana serta dukungan komunitas yang luas. *Python* dijalankan menggunakan interpreter sehingga kode dapat dieksekusi secara langsung di berbagai *platform*, serta mengadopsi berbagai paradigma pemrograman seperti prosedural, berorientasi objek, dan fungsional. Kombinasi karakteristik ini menjadikan *Python* sangat adaptif dan efektif untuk digunakan dalam berbagai bidang, mulai dari pengembangan *web*, analisis data, hingga kecerdasan buatan dan pengembangan perangkat lunak secara umum.

2.9 Pengertian Flutter

Flutter adalah sebuah *framework open-source* yang dikembangkan oleh *Google* untuk membangun antar muka (*user interface*) aplikasi *android* dan *iOS*. *Flutter* mendukung *cross platform*, atau dapat dijalankan pada beberapa *platform* yang berbeda. Dengan *Flutter*, kita dapat membuat aplikasi *Android* dan *iOS* sekaligus. Selain *mobile*, dapat juga membuat aplikasi *web* dan desktop. Tentunya ini akan menghemat waktu, dikarenakan tidak perlu mempelajari bahasa *native* masing – masing *platform* (P. R. Setiawan et al., 2022).

Flutter adalah perangkat pengembangan perangkat lunak antarmuka pengguna (*UI*) bersumber terbuka yang dibuat oleh *Google*, dirancang untuk membangun aplikasi yang dikompilasi secara *native* untuk *platform mobile*, *web*, dan desktop dari satu kode sumber. *Flutter* menggunakan bahasa pemrograman *Dart* dan menyediakan sejumlah *widget* pra-dibangun, memberikan fleksibilitas kepada para pengembang untuk menciptakan antarmuka pengguna yang menarik secara visual dan interaktif (Harahap et al., 2023).

Berdasarkan uraian tersebut, dapat disimpulkan bahwa *Flutter* merupakan *framework open-source* dari *Google* yang dirancang untuk membangun antarmuka pengguna aplikasi secara lintas *platform*, baik untuk *mobile*, *web*, maupun desktop, hanya dengan satu basis kode menggunakan bahasa *Dart*. Dukungan terhadap kompilasi *native* serta ketersediaan berbagai *widget* bawaan menjadikan *Flutter* fleksibel dan efisien dalam pengembangan aplikasi, sekaligus memungkinkan pengembang menciptakan tampilan yang menarik, interaktif, dan konsisten di berbagai *platform* tanpa harus mempelajari bahasa pemrograman khusus untuk masing-masing sistem.

2.10 Pengertian MySQL

MySQL adalah *DBMS* atau *database management system* yang menggunakan bahasa *SQL* sebagai bahasa penghubung antara perangkat lunak aplikasi dengan *database* pada sistem. *MySQL* merupakan salah satu *DBMS* yang sangat populer digunakan (Silalahi, 2022). *MySQL* adalah suatu *database management system (DBMS)* dengan perintah dasar *Structured Query Language*

(SQL) yang sering dimanfaatkan dalam pembuatan aplikasi dan *website* (Universitas Cakrawala, 2023).

Berdasarkan uraian tersebut, dapat disimpulkan bahwa *MySQL* merupakan salah satu sistem manajemen basis data (*DBMS*) yang populer dan banyak digunakan dalam pengembangan aplikasi maupun *website*. *MySQL* memanfaatkan bahasa *Structured Query Language (SQL)* sebagai penghubung antara aplikasi dan basis data, sehingga memudahkan proses pengelolaan, penyimpanan, dan pengambilan data secara terstruktur. Dengan kemudahan penggunaan serta dukungan yang luas, *MySQL* menjadi pilihan yang efektif dan andal sebagai media penyimpanan data dalam berbagai sistem informasi.

2.11 Penelitian Terdahulu

Penelitian terdahulu menjadi landasan penting dalam penyusunan penelitian ini karena memberikan gambaran mengenai sejauh mana topik yang sejenis telah dikaji oleh peneliti lain. Melalui telaah terhadap berbagai penelitian sebelumnya, penulis dapat mengetahui pendekatan, metode, serta hasil yang telah dicapai, sehingga dapat menghindari pengulangan dan menemukan celah penelitian yang masih belum terisi. Adapun beberapa penelitian yang relevan dan dijadikan acuan dalam penyusunan penelitian ini dapat dilihat pada tabel berikut.

Tabel 2.1 Penelitian Terdahulu

No	Judul Artikel	Penulis	Tahun Terbit	Rangkuman
1	Implementasi Arsip Digital untuk Penyimpanan Dokumen Digital	Ghifari Aminudin Fad'li, dkk	2023	Artikel ini membahas implementasi sistem arsip digital sebagai solusi penyimpanan dokumen organisasi. Penelitian menyoroti proses digitalisasi dokumen, pengelompokan arsip, serta manfaat penggunaan arsip digital dalam meningkatkan efisiensi penyimpanan dan pengelolaan data. Hasil penelitian menunjukkan bahwa sistem arsip digital lebih efektif dibandingkan arsip fisik dalam mendukung pengelolaan dokumen jangka panjang.

2	Perancangan Aplikasi Arsip Elektronik Fakultas Komputer Universitas Prabumulih	Yuntari Purbasari, Suhartini, Essy Malays Sari Sakti.	2023	Artikel ini membahas pembuatan aplikasi e-arsip di Fakultas Ilmu Komputer Universitas Prabumulih. Sistem dibangun dengan metode <i>prototyping</i> , dilengkapi diagram UML (<i>use case, activity</i>) dan struktur <i>database</i> . Fitur aplikasi mencakup manajemen arsip (menambah, mengubah, menghapus) dan laporan rekapitulasi arsip berdasarkan waktu. Kesimpulannya: aplikasi ini membantu mempercepat proses pengarsipan, mengurangi kesalahan pendataan, dan membutuhkan pelatihan pengguna agar dipakai dengan tepat.
3	Penerapan Teknologi <i>Optical Character Recognition</i> Pada Pengarsipan Dokumen (Studi Kasus: PT Pertamina Geothermal Energy Area Lahendong)	Faldy Andreas Gosal, Parabelem Tinno Dolf Rompas	2023	Penelitian ini mengimplementasikan teknologi <i>OCR (Optical Character Recognition)</i> untuk digitalisasi arsip dokumen di PT Pertamina Geothermal Energy Area Lahendong. Prosesnya mencakup pemindaian dokumen fisik menjadi gambar, lalu ekstraksi teks dengan <i>OCR</i> untuk menyimpan konten arsip dalam bentuk digital yang dapat dicari dan dikelola lebih efisien. Sistem ini membantu mengurangi pekerjaan manual dan mempercepat proses pengarsipan dokumen perusahaan.
4	Digital Archive Management in the Organization Section of the Regional Secretariat of Tuban Regency	Setiorini, Dwi Ridho Aulianto	2024	Penelitian ini membahas penerapan manajemen arsip digital pada bagian organisasi Sekretariat Daerah Kabupaten Tuban. Arsip yang sebelumnya dikelola secara konvensional dialihkan ke sistem digital untuk meningkatkan efisiensi penyimpanan dan kemudahan akses informasi. Hasil penelitian menunjukkan bahwa arsip digital membantu mempercepat temu kembali dokumen, meningkatkan kerapian pengelolaan arsip, serta mendukung kinerja administrasi pemerintahan.

5	Implementasi Aplikasi Manajemen Arsip Surat Berbasis <i>Optical Character Recognition</i> Pada Badan Pusat Statistik Banyuwangi	Moh. Syahrul Iskandar, Akhlis Munazilin, Adi Susanto	2025	Penelitian mengembangkan sistem manajemen arsip surat di BPS Banyuwangi menggunakan <i>OCR</i> agar dokumen surat bisa diubah jadi teks digital, memudahkan pencarian. Metode pengembangan yang digunakan adalah <i>Waterfall</i> , dan pengujian memakai <i>black-box</i> . Hasilnya: entri data surat jadi lebih cepat (~2,65 detik per dokumen), manajemen surat masuk/keluar lebih efisien. Namun, akurasi <i>OCR</i> tergantung kualitas citra dokumen, jadi peneliti menyarankan penambahan modul pra-pemrosesan gambar dan klasifikasi otomatis untuk pengembangan selanjutnya.
6	Implementasi Algoritma <i>Rule-Based Classification</i> pada Aplikasi Arsip Web Menggunakan Metode RAD dan Evaluasi ISO/IEC 25010	<u>Rafael Revi Runtu, Glenn D. P. Maramis, Alfiansyah Hasibuan.</u>	2025	Penelitian ini membuat aplikasi arsip berbasis web yang dapat mengklasifikasikan surat secara otomatis menggunakan algoritma <i>rule-based classification</i> . Metode pengembangan menggunakan RAD (<i>Rapid Application Development</i>), dan kualitas perangkat lunak dievaluasi berdasarkan standar ISO/IEC 25010 (fokus pada <i>functional suitability</i> , <i>usability</i> , dan <i>reliability</i>). Hasil evaluasi menunjukkan aplikasi sangat layak pakai dengan nilai fungsionalitas sekitar 95%, <i>usability</i> ~90%, dan keandalan ~85%.
7	Perancangan Aplikasi Database Sistem Pengelolaan Arsip Digital Berbasis <i>Web Mobile</i> Menggunakan Algoritma Levenshtein (Studi Kasus: SMA PLUS Taruna Akterlis Medan)	Rizky Ananda Fradana, Leni Marlina	2025	Artikel ini menjelaskan desain dan implementasi sistem arsip digital berbasis <i>web mobile</i> untuk SMA PLUS Taruna Akterlis Medan. Sistem dikembangkan dengan model <i>Waterfall</i> , menggunakan algoritma Levenshtein untuk meningkatkan akurasi pencarian arsip berdasarkan kesamaan teks. Pengujian dilakukan dengan <i>Black Box Testing</i> untuk memastikan fungsionalitas seperti penyimpanan, pencarian, dan manajemen arsip berjalan baik. Hasil menunjukkan sistem mampu memudahkan pengelolaan arsip, mempercepat pencarian dokumen, dan meningkatkan efisiensi operasional administrasi sekolah.

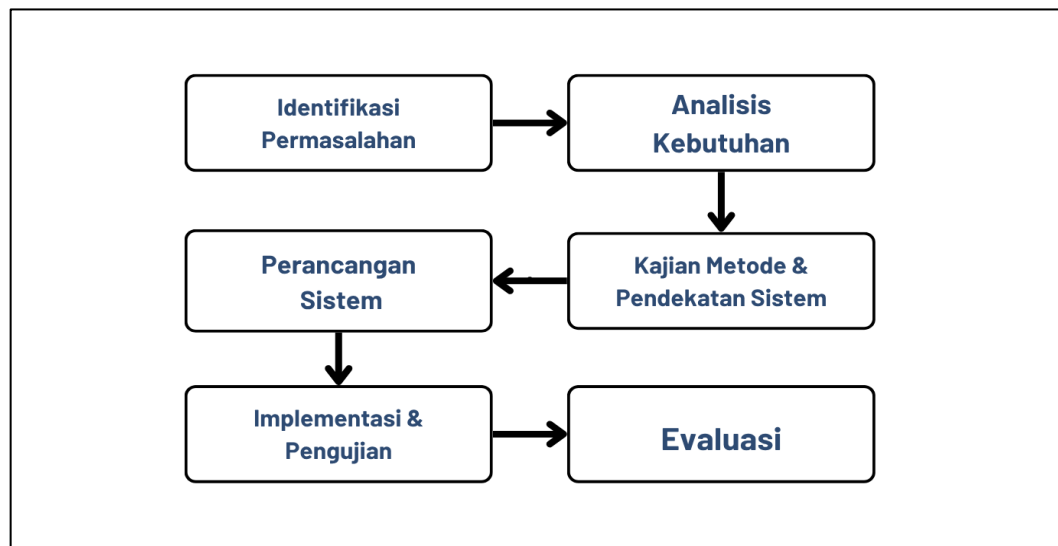
8	Perancangan <i>E-Office</i> Pelayanan dan Pengarsipan Digital Menggunakan Metode <i>OCR</i> Berbasis <i>Web</i>	Alvian Tri Putra Darti Akhsa, Muhammad Agus, Rosmiati, Andi M. Bahrul Ulum	2024	Penelitian ini merancang sistem <i>e-office</i> berbasis <i>web</i> yang mengintegrasikan teknologi <i>Optical Character Recognition (OCR)</i> untuk pengarsipan digital dokumen pelayanan publik. <i>OCR</i> digunakan untuk mengekstraksi teks dari dokumen hasil pemindaian sehingga arsip dapat disimpan dan dicari secara digital. Hasil penelitian menunjukkan bahwa sistem mampu mempercepat proses pengarsipan dan meningkatkan efisiensi pelayanan administrasi.
9	<i>Digital Archives Management System for Archives Security at BPIPI</i>	Ciptari Queen Rachmad Oktavia Lestanti, Joni Dwi Pribadi	2025	Penelitian ini mengembangkan sistem manajemen arsip digital untuk meningkatkan keamanan arsip di BPIPI. Sistem dirancang untuk mengelola penyimpanan, akses, dan pengamanan arsip secara terstruktur. Hasil penelitian menunjukkan bahwa penerapan sistem arsip digital dapat mengurangi risiko kehilangan arsip, meningkatkan keamanan data, serta mendukung pengelolaan arsip yang lebih tertib dan efisien.
10	Preservasi Digital sebagai Upaya Akuntabilitas Tata Kelola Arsip di Lembaga Kearsipan di Indonesia	Aulia Nurdiansyah, Rahmat Alfin Valentino, Salaisyah Amalia Yussuf	2025	Penelitian ini membahas penerapan preservasi digital arsip pada lembaga kearsipan di Indonesia sebagai upaya menjaga akuntabilitas dan keberlanjutan arsip. Metode preservasi yang dikaji meliputi migrasi, konversi, replikasi, dan enkapsulasi arsip digital. Hasil penelitian menunjukkan bahwa preservasi digital berperan penting dalam memastikan arsip tetap dapat diakses, terjaga keutuhannya, serta mendukung tata kelola arsip digital yang baik dan berkelanjutan.

Sumber : data diolah oleh penulis (2026)

2.12 Kerangka Berpikir Penelitian

Kerangka berpikir penelitian disusun untuk menjelaskan alur logika yang menghubungkan antara teori, hasil penelitian terdahulu, serta fokus penelitian yang sedang dilakukan. Bagian ini berfungsi sebagai dasar pemikiran yang menggambarkan proses berpikir peneliti dalam merumuskan langkah-langkah penelitian secara sistematis dan terarah. Melalui kerangka berpikir ini, dapat terlihat bagaimana konsep dan teori yang relevan diintegrasikan untuk menjawab rumusan

masalah, hingga akhirnya menghasilkan solusi berupa pengembangan aplikasi pengajuan dinas dan arsip digital terpadu berbasis *OCR* dan *Rule-Based System*. Adapun kerangka berpikir penelitian ini diilustrasikan pada Gambar 2.1 berikut yang menunjukkan alur pemikiran peneliti dalam menghubungkan teori, hasil penelitian terdahulu, serta proses pengembangan aplikasi yang dilakukan.



Sumber : data diolah oleh penulis: (2025)

Gambar 2.1 Kerangka Berpikir Penelitian

Kerangka berpikir penelitian ini menjelaskan alur logis dalam proses pengembangan arsip digital berbasis *OCR* dan *Rule-Based System*. Tahap pertama adalah identifikasi permasalahan, di mana hasil observasi dan wawancara menunjukkan bahwa proses pengarsipan dokumen di rektorat saat ini berjalan cukup baik dan jarang menimbulkan kendala seperti kehilangan arsip atau kesulitan pencarian, karena dikelola langsung oleh satu staf yang memahami sepenuhnya alur dan lokasi penyimpanan dokumen. Namun, kondisi ini sangat bergantung pada individu dan sulit dijaga konsistensinya apabila terjadi pergantian staf atau peningkatan volume dokumen. Berdasarkan hal tersebut, penelitian ini menawarkan solusi berupa sistem arsip digital yang tidak hanya mendigitalkan dokumen, tetapi juga menerapkan teknologi *OCR* untuk melakukan ekstraksi teks secara otomatis dan *Rule-Based System* sebagai mekanisme validasi dokumen yang terstruktur.

Selanjutnya, pada tahap kajian metode dan pendekatan sistem, ditentukan penerapan *OCR* untuk mengonversi teks dari hasil pindai dokumen, serta

penggunaan *Rule-Based System* untuk mengidentifikasi dan menentukan letak elemen-elemen teks yang terekstrak berdasarkan pola dan jenis dokumennya. Tahap berikutnya adalah perancangan sistem yang mencakup penyusunan arsitektur aplikasi, perancangan basis data, dan integrasi antara komponen *OCR* dan *Rule-Based System* agar sistem dapat berfungsi secara terpadu. Setelah tahap perancangan selesai, dilakukan implementasi dan pengujian guna memastikan sistem bekerja sesuai fungsi yang diharapkan, termasuk pengujian terhadap akurasi pembacaan teks dan efektivitas validasi dokumen. Terakhir, dilakukan evaluasi terhadap efisiensi, kecepatan, serta kepuasan pengguna untuk menilai sejauh mana sistem ini mampu meningkatkan keberlanjutan dan kualitas pengarsipan digital dibanding metode manual sebelumnya



BAB III

METODE PENELITIAN

3.1 Objek dan Jadwal Penelitian

Penelitian ini dilaksanakan dengan tujuan untuk mengembangkan sistem arsip digital berbasis *OCR* dan *Rule-Based System* yang diterapkan di lingkungan Universitas Prabumulih. Kegiatan penelitian ini dilakukan sesuai jadwal yang telah ditetapkan oleh peneliti, dengan mempertimbangkan waktu pelaksanaan setiap tahapan mulai dari pengumpulan data, analisis kebutuhan, perancangan sistem, hingga tahap pengujian dan evaluasi. Jadwal penelitian disusun agar seluruh proses dapat berjalan secara terarah dan sesuai dengan rencana yang telah ditentukan.

Objek pada penelitian ini adalah Universitas Prabumulih, yang berlokasi di Jalan Patra No.50 RT.01 RW.03 Kelurahan Sukaraja, Kecamatan Prabumulih Selatan, Kota Prabumulih, Sumatera Selatan.

3.1.1 Sejarah

Universitas Prabumulih di singkat UNPRA adalah perguruan tinggi swasta di Prabumulih, Sumatera Selatan, Indonesia, yang berdiri pada tahun 2001 dibawah naungan Yayasan Pendidikan Prabumulih. Universitas Prabumulih merupakan penggabungan dari Sekolah Tinggi Ilmu Ekonomi Prabumulih (STIE Prabumulih), Sekolah Tinggi Ilmu Teknik Prabumulih (STIT Prabumulih) dan Sekolah Tinggi Manajemen Informatika Dan Komputer Prabumulih (STMIK Prabumulih) berdasarkan surat Keputusan Menteri Pendidikan, Kebudayaan, Riset dan Teknologi Republik Indonesia nomor : 254/e/o/2022 pada 07 April 2022.

Pada awalnya yayasan pendidikan Prabumulih memiliki 3 perguruan tinggi dalam bentuk sekolah tinggi,yaitu:

1. Sekolah Tinggi Ilmu Ekonomi Prabumulih (STIE Prabumulih)
2. Sekolah Tinggi Manajemen Informatika Dan Komputer Prabumulih (STMIK Prabumulih)

3. Sekolah Tinggi Ilmu Teknik Prabumulih (STIT Prabumulih)

Pada tahun 2022 atas penggabungan 3 sekolah tinggi yang ada di universitas Prabumulih menjadi Fakultas Ilmu Ekonomi, Fakultas Ilmu Komputer, Fakultas Teknik Universitas Prabumulih. Sekolah Tinggi Ilmu Ekonomi Prabumulih (STIE Prabumulih) Menjadi Fakultas Ekonomi Dan Bisnis (FEB), Sekolah Tinggi Ilmu Teknik Prabumulih (STIT Prabumulih) Menjadi Fakultas Teknik (FT) Dan Sekolah Tinggi Manajemen Informatika Dan Komputer Prabumulih (STMIK Prabumulih) Menjadi Fakultas Ilmu Komputer (FASILKOM).

3.1.2 Visi, Misi, dan Tujuan Universitas Prabumulih

Visi, misi, dan tujuan Universitas Prabumulih merupakan dasar utama dalam pelaksanaan kegiatan pendidikan, penelitian, dan pengabdian kepada masyarakat. Ketiganya menjadi arah dan pedoman bagi seluruh sivitas akademika dalam mewujudkan tujuan institusional serta mendukung pencapaian mutu pendidikan yang berkelanjutan. Oleh karena itu, penting untuk memahami visi, misi, dan tujuan Universitas Prabumulih sebagai landasan dalam pelaksanaan penelitian ini.

a. Visi

Visi Universitas Prabumulih adalah terwujudnya Universitas Prabumulih menjadi Perguruan Tinggi yang terkemuka dalam menghasilkan sumber daya manusia yang berkualitas dan bertanggungjawab dalam Penguasaan dan Pengembangan Ilmu Pengetahuan, Teknologi dan Seni pada tahun 2029.

b. Misi

1. Menyelenggarakan kegiatan pendidikan dan pengajaran yang bermutu untuk menghasilkan sumber daya manusia yang berkualitas;
2. Menghasilkan sumber daya manusia yang memiliki kompetensi sesuai dengan kebutuhan dunia kerja;
3. Menghasilkan lulusan yang memiliki daya saing tinggi dalam menghadapi kompetensi global dan professional dalam penguasaan dan pengembangan ilmu pengetahuan, teknologi, dan seni;

4. Meningkatkan kualitas pelaksanaan kegiatan tridharma perguruan tinggi;
5. Mengembangkan kemitraan dengan badan dan/atau instansi lain dalam melaksanakan kegiatan tridharma perguruan tinggi;
6. Mendayagunakan sarana dan prasarana pendidikan secara dinamis, efektif, efisien, dan produktif.

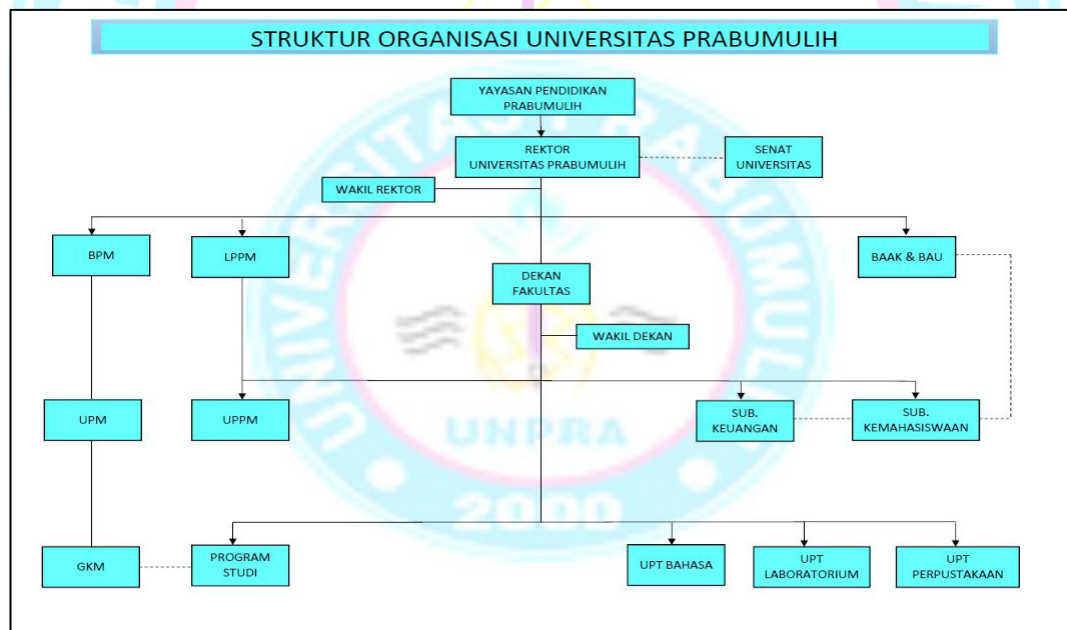
c. Tujuan

1. Membentuk dan menghasilkan lulusan yang mampu menjunjung tinggi almamaternya, berkepribadian Indonesia, bermoral tinggi, berbudi luhur, berwawasan kebangsaan serta tanggap terhadap keadaan tantangan dan perubahan yang timbul dan mampu menemukan solusi yang terbaik untuk mengatasi tantangan dan perubahan dalam kerangka turut bertanggung jawab atas kelangsungan hidup Bangsa dan Negara Kesatuan Republik Indonesia berdasarkan Pancasila dan UUD 1945;
2. Membentuk dan menghasilkan tenaga ahli yang mampu berpikir ilmiah, profesional, berbudi luhur, terampil, berjiwa wiraswasta serta berperan aktif dalam pembangunan nasional dengan berpegang teguh pada etika profesi;
3. Membentuk dan menghasilkan sarjana yang memiliki integritas tinggi, yang mampu memelihara, mengembangkan ilmu pengetahuan dan yang mampu berperan aktif dalam pengembangan budaya, ilmu pengetahuan, teknologi dan seni;
4. Mengembangkan dan menyebarluaskan ilmu pengetahuan teknologi dan seni serta mengupayakan penggunaannya untuk meningkatkan taraf hidup masyarakat dan memperkaya kebudayaan nasional;
5. Menghasilkan penelitian yang memiliki karakteristik sebagai berikut:
 - a. Bermanfaat bagi perkembangan iptek dan mendukung peningkatan daya saing bangsa;
 - b. Bermanfaat bagi kesejahteraan masyarakat; serta
 - c. Bernilai publikasi, baik di tingkat nasional maupun internasional.

6. Melaksanakan kegiatan pengabdian kepada masyarakat dengan karakteristik sebagai berikut:
 - a. Bermanfaat dalam memberdayakan potensi masyarakat;
 - b. Mampu menumbuhkan partisipasi aktif masyarakat dalam upaya memecahkan masalah; serta
 - c. Mampu meningkatkan kesejahteraan masyarakat

3.1.3 Struktur Organisasi

Struktur organisasi merupakan komponen penting dalam suatu institusi karena menggambarkan pembagian tugas, tanggung jawab, serta hubungan kerja antarbagian yang ada di dalamnya. Dengan memahami struktur organisasi, peneliti dapat mengetahui alur koordinasi dan pihak-pihak yang berperan dalam kegiatan administrasi, termasuk dalam pengelolaan arsip digital di lingkungan Universitas Prabumulih. Adapun struktur organisasi Universitas Prabumulih dapat dilihat sebagai berikut:



Sumber : Universitas Prabumulih (2022)

Gambar 3.2 Struktur Organisasi

Dari gambar struktur organisasi di atas, dapat dijelaskan bahwa setiap bagian memiliki peran dan tanggung jawab tertentu dalam mendukung jalannya kegiatan organisasi. Adapun penjelasan dari masing-masing bagian adalah sebagai berikut.

1. Yayasan Pendidikan Prabumulih, Merupakan lembaga tertinggi yang menaungi Universitas Prabumulih. Yayasan ini bertanggung jawab terhadap pembinaan, pengawasan, serta pengembangan universitas agar berjalan sesuai visi, misi, dan tujuan pendidikan nasional.
2. Rektor Universitas Prabumulih, Rektor adalah pimpinan tertinggi universitas yang memiliki tanggung jawab penuh atas seluruh kegiatan akademik, penelitian, pengabdian kepada masyarakat, serta administrasi universitas. Rektor juga berperan sebagai penentu kebijakan dan pengambil keputusan strategis dalam pengelolaan universitas.
3. Wakil Rektor membantu Rektor dalam bidang-bidang tertentu, antara lain: akademik, kemahasiswaan, keuangan, dan kerja sama. Setiap Wakil Rektor memiliki tugas sesuai bidang tanggung jawabnya dan melapor langsung kepada Rektor
4. Senat Universitas, Senat merupakan lembaga normatif dan perwakilan akademik tertinggi di universitas. Tugasnya mencakup penetapan kebijakan akademik, pengembangan kurikulum, serta pemberian pertimbangan terhadap kebijakan penting universitas. Anggotanya terdiri atas pimpinan universitas, dekan, dan perwakilan dosen
5. Badan Penjamin Mutu (BPM), BPM berfungsi sebagai pengendali mutu internal universitas. Lembaga ini memastikan bahwa seluruh kegiatan akademik dan non-akademik berjalan sesuai standar mutu yang telah ditetapkan, termasuk evaluasi terhadap kinerja fakultas dan program studi.
6. Lembaga Penelitian dan Pengabdian kepada Masyarakat (LPPM), LPPM bertugas mengelola dan mengembangkan kegiatan penelitian serta pengabdian masyarakat yang dilakukan oleh dosen dan mahasiswa. LPPM menjadi penghubung antara universitas dan masyarakat dalam hal inovasi dan penerapan ilmu pengetahuan.
7. Unit Penjaminan Mutu (UPM), UPM berada di bawah BPM dan berperan dalam pelaksanaan sistem penjaminan mutu di tingkat fakultas. UPM memastikan bahwa kegiatan akademik dan administrasi di setiap fakultas sesuai dengan standar universitas.

8. Unit Penjaminan Mutu (UPM), UPM berada di bawah BPM dan berperan dalam pelaksanaan sistem penjaminan mutu di tingkat fakultas. UPM memastikan bahwa kegiatan akademik dan administrasi di setiap fakultas sesuai dengan standar universitas.
9. Gugus Kendali Mutu (GKM), GKM adalah unit pelaksana mutu di tingkat program studi. Fungsinya memastikan kegiatan akademik, penilaian, serta pelayanan mahasiswa di prodi berjalan sesuai dengan standar mutu yang telah ditetapkan universitas.
10. Program Studi, merupakan pelaksana utama kegiatan akademik di bidang tertentu. Setiap prodi bertanggung jawab terhadap penyusunan kurikulum, proses pembelajaran, penelitian, dan kegiatan kemahasiswaan di lingkungannya masing-masing.
11. BAAK (Biro Administrasi Akademik dan Kemahasiswaan) dan BAU (Biro Administrasi Umum), Unit ini bertugas dalam pelayanan administrasi akademik (seperti pendaftaran mahasiswa, jadwal kuliah, dan nilai) serta administrasi umum dan kepegawaian. BAAK & BAU juga menjadi pusat pengelolaan arsip dan surat-menyurat universitas.
12. Sub. Keuangan, Sub bagian Keuangan bertanggung jawab atas seluruh kegiatan yang berkaitan dengan pengelolaan keuangan universitas, termasuk perencanaan anggaran, pembukuan, dan pelaporan keuangan.
13. Sub. Kemahasiswaan, Subbagian Kemahasiswaan menangani berbagai kegiatan non-akademik mahasiswa seperti organisasi kemahasiswaan, kegiatan sosial, prestasi, serta pembinaan karakter mahasiswa.
14. UPT Bahasa, Unit Pelaksana Teknis Bahasa berperan memberikan layanan peningkatan kemampuan bahasa bagi mahasiswa dan dosen, termasuk pelatihan *TOEFL/TOEIC* serta penerjemahan dokumen akademik.
15. UPT Laboratorium, UPT Laboratorium menyediakan sarana dan layanan praktikum untuk mendukung kegiatan pembelajaran dan penelitian mahasiswa maupun dosen di berbagai fakultas.
16. UPT Perpustakaan, UPT Perpustakaan merupakan pusat sumber belajar yang menyediakan koleksi buku, jurnal, dan referensi ilmiah, baik dalam bentuk cetak maupun digital, guna mendukung kegiatan akademik universitas.

3.1.4 Jadwal Penelitian

Penelitian ini dijadwalkan berlangsung selama 6 bulan, terhitung mulai tanggal 28 Oktober 2025 hingga 28 April 2026. Rentang waktu tersebut mencakup seluruh tahapan kegiatan penelitian, mulai dari perencanaan, pengumpulan data, analisis kebutuhan sistem, perancangan dan pengembangan sistem, hingga tahap pengujian dan penyusunan laporan akhir.

3.2 Metode Penelitian

Metode penelitian merupakan cara ilmiah yang digunakan untuk memperoleh data dengan tujuan dan kegunaan tertentu. Berdasarkan pandangan tersebut, penelitian harus memenuhi unsur ilmiah, data, tujuan, dan kegunaan agar hasilnya valid dan relevan (I. Setiawan et al., 2020). Sejalan tujuan penelitian ini yang berfokus pada pengembangan sistem, pendekatan yang digunakan adalah pendekatan deskriptif kualitatif. Pendekatan ini dipilih karena mampu menggambarkan secara mendalam proses dan hasil dari pengembangan aplikasi arsip digital berbasis *OCR* dan *Rule-Based System* secara kontekstual.

Penelitian kualitatif sendiri merupakan suatu pendekatan yang menekankan eksplorasi mendalam terhadap suatu fenomena sosial dengan mempertimbangkan konteks alaminya (Evi et al., 2025). Dalam konteks penelitian ini, peneliti berupaya memahami makna di balik data yang diperoleh secara induktif. Penelitian kualitatif berangkat dari data, memanfaatkan teori yang ada sebagai bahan penjelas, dan berakhir dengan penguatan atau pembentukan teori baru.

3.3 Sumber Data

Dalam penelitian ini, sumber data merupakan komponen penting yang digunakan untuk memperoleh informasi yang relevan dengan permasalahan penelitian. Sumber data diperoleh dari dua jenis, yaitu data primer dan data sekunder, yang keduanya saling melengkapi dalam mendukung hasil penelitian.

3.3.1 Data Primer

Data primer merupakan sumber data yang diperoleh secara langsung dari sumber asli atau pihak pertama (Ulwani et al., 2021). Dalam penelitian ini, data primer diperoleh dari hasil wawancara dan pengamatan langsung terhadap proses penyimpanan arsip di Unit Rektorat Universitas Prabumulih. Data ini digunakan untuk memahami kebutuhan pengguna, alur administrasi, serta kendala yang terjadi dalam sistem konvensional sebelum diterapkannya aplikasi berbasis *OCR* dan *Rule-Based System*.

3.3.2 Data Sekunder

Data sekunder dalam penelitian adalah data yang telah dikumpulkan, diolah, dan dipublikasikan oleh pihak lain sebelum digunakan oleh peneliti untuk tujuan penelitian mereka (Nuraeni et al., 2025). Data sekunder diperoleh dari sumber-sumber tidak langsung, seperti buku, jurnal, artikel ilmiah, laporan penelitian terdahulu, dan dokumen resmi universitas yang berkaitan dengan topik penelitian. Dalam penelitian ini, data sekunder digunakan sebagai landasan teoritis untuk memperkuat konsep pengembangan sistem, penerapan *Optical Character Recognition (OCR)*, dan *Rule-Based System*, serta mendukung analisis hasil dari data primer.

3.4 Teknik Pengumpulan Data

Pengumpulan data merupakan langkah penting dalam penelitian ini, karena kualitas data yang diperoleh akan sangat memengaruhi ketepatan perancangan sistem yang dikembangkan. Untuk itu, digunakan beberapa teknik pengumpulan data, yaitu observasi, wawancara, dokumentasi, dan studi literatur.

3.4.1 Observasi

Observasi dilakukan dengan mengamati langsung proses pengelolaan arsip manual di unit terkait (misalnya bagian administrasi). Melalui teknik ini, penulis dapat memahami alur kerja, mulai dari penerimaan dokumen, proses klasifikasi, pencatatan, hingga penyimpanan. Observasi juga membantu mengidentifikasi hambatan yang muncul, seperti waktu pencarian dokumen yang lama, potensi

kesalahan manusia dalam pencatatan, atau ketidakkonsistenan format dokumen. Dengan demikian, observasi memberikan gambaran nyata kondisi aktual sehingga kebutuhan sistem dapat dirumuskan berdasarkan permasalahan riil.

3.4.2 Wawancara

Wawancara dilakukan pada tanggal 30 Oktober 2025 dengan staf administrasi rektorat sebagai informan utama. Wawancara ini bertujuan untuk menggali informasi terkait prosedur pengarsipan, kendala yang sering dihadapi, kebutuhan fitur sistem untuk mempermudah pekerjaan, serta harapan terhadap implementasi arsip digital. Dari wawancara ini, penulis memperoleh data yang membantu memastikan bahwa rancangan sistem berbasis *OCR* dan *Rule-Based* dapat menyelesaikan masalah nyata yang terjadi di unit administrasi.

3.4.3 Dokumentasi

Dokumentasi dilakukan dengan mengumpulkan berbagai dokumen arsip yang ada di rektorat, seperti surat resmi, laporan akademik, memo, dan dokumen pendukung lainnya. Dokumen ini dianalisis untuk memahami pola data, jenis informasi yang harus diidentifikasi dan divalidasi oleh sistem, serta sebagai dataset awal untuk pengujian *OCR* dan *Rule-Based System*. Teknik dokumentasi juga membantu mengenali variasi format dokumen yang nantinya akan mempengaruhi akurasi sistem dalam membaca dan memproses data.

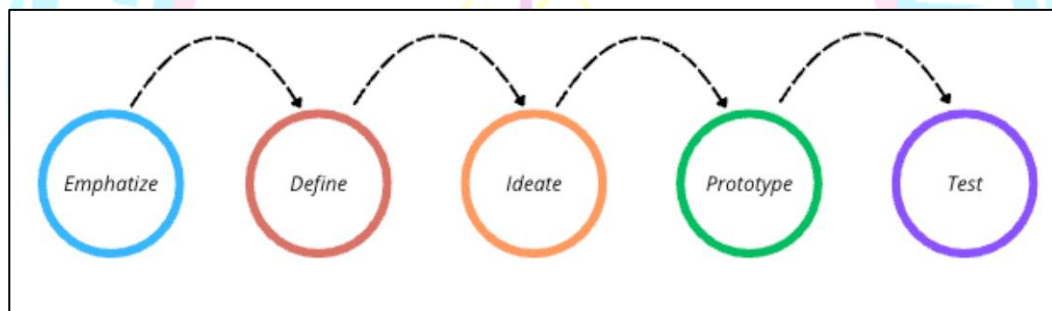
3.4.4 Studi Literatur

Studi literatur dilakukan dengan menelaah sumber akademik maupun non-akademik, termasuk jurnal ilmiah, buku, prosiding, dan penelitian terdahulu yang berkaitan dengan pengelolaan arsip digital, *OCR*, dan *Rule-Based System*. Studi ini bertujuan memperoleh landasan teori yang kuat, memahami perkembangan teknologi terkini, serta menentukan metode pengembangan sistem, mekanisme validasi data, dan evaluasi sistem yang tepat. Dengan demikian, penelitian ini memiliki dasar ilmiah yang kokoh sekaligus aplikatif di lingkungan Universitas Prabumulih.

3.5 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah *Design Thinking*. *Design Thinking* adalah pendekatan pemecahan masalah yang berpusat pada manusia dengan menggabungkan pemahaman kebutuhan pengguna, kreativitas, dan proses iteratif dalam pengembangan solusi. Pendekatan ini menekankan empati, kolaborasi lintas disiplin, serta pengujian berulang untuk menghasilkan solusi yang inovatif dan relevan melalui lima tahapan utama, yaitu *Empathize*, *Define*, *Ideate*, *Prototype*, dan *Test* (Lantana & Fauziah, 2025).

Metode ini dipilih karena berfokus pada pemahaman kebutuhan pengguna serta penyelesaian masalah melalui proses yang kreatif dan iteratif. Pendekatan ini dianggap paling sesuai dengan penelitian ini yang berfokus pada pengembangan sistem arsip digital berbasis *OCR* dan *Rule-Based System* di lingkungan Universitas Prabumulih, karena melibatkan pengguna secara langsung untuk menemukan solusi yang relevan dengan kondisi nyata di lapangan.



Sumber : Lantana & Fauziah (2025)

Gambar 3.3 Metode Pengembangan Sistem *Design Thinking*

Tahapan dari metode *Design Thinking* yang dilakukan dalam penelitian ini adalah sebagai berikut:

3.5.1 *Empathize* (Empati)

Tahap ini merupakan proses memahami kebutuhan dan permasalahan pengguna sistem, yaitu staf administrasi rektorat Universitas Prabumulih yang mengelola arsip secara manual. Pada tahap ini peneliti telah melakukan observasi langsung terhadap proses pengarsipan dokumen fisik serta melakukan wawancara dengan staf administrasi rektorat untuk mengetahui kendala yang sering muncul, seperti lamanya pencarian dokumen, tumpang tindih data, dan kesulitan validasi

arsip. Tujuan dari tahap ini adalah agar peneliti dapat merasakan permasalahan pengguna secara nyata sebelum merancang solusi sistem yang tepat.

3.5.2 *Define* (Perumusan Masalah)

Tahap ini dilakukan setelah tahap empati untuk menganalisis dan merumuskan inti permasalahan yang ditemukan di lapangan. Berdasarkan hasil observasi awal, peneliti telah melakukan analisis kebutuhan sistem dengan memetakan proses kerja lama dan mengidentifikasi titik masalah, seperti proses input data yang tidak terstruktur serta tidak adanya sistem otomatisasi penyimpanan dan pembacaan dokumen. Hasil dari tahap ini berupa perumusan masalah yang menjadi dasar rancangan sistem arsip digital berbasis *OCR* dan *Rule-Based System*.

3.5.3 *Ideate* (Pengembangan Ide)

Pada tahap ini peneliti telah mengembangkan ide dan alternatif solusi dari permasalahan yang telah didefinisikan sebelumnya. Proses ini dilakukan melalui kegiatan *brainstorming* untuk menentukan fitur utama yang akan dikembangkan, seperti modul pemindaian dokumen otomatis, pengenalan teks dengan *OCR*, serta pengelompokan dan penyimpanan arsip secara otomatis menggunakan *Rule-Based System*. Hasil dari tahap ini berupa rancangan konsep solusi yang siap diwujudkan dalam bentuk desain sistem.

3.5.4 *Prototype* (Pembuatan Prototype)

Tahap ini merupakan proses mengimplementasikan ide solusi ke dalam bentuk rancangan sistem. Pada tahap ini peneliti telah membuat prototipe sistem yang menggambarkan alur kerja, tampilan antarmuka, serta integrasi fungsi *OCR* dan *Rule-Based*. Prototipe ini dikembangkan menggunakan perangkat keras berupa laptop dan smartphone, serta perangkat lunak pendukung seperti Figma. *Prototipe* ini akan menjadi acuan dalam pembuatan sistem akhir.

3.5.5 *Test* (Pengujian)

Tahap terakhir adalah pengujian sistem untuk memastikan bahwa semua fungsi dalam sistem berjalan sesuai dengan rancangan dan kebutuhan pengguna.

Pada tahap ini akan dilakukan pengujian menggunakan metode *White box testing*, yaitu dengan menelusuri alur logika program untuk memastikan tidak terdapat kesalahan (*error*) pada setiap fungsi utama, seperti proses pembacaan dokumen, validasi data, dan penyimpanan hasil arsip digital. Hasil pengujian digunakan untuk memperbaiki kekurangan sistem sebelum tahap implementasi penuh.

Dengan penerapan metode *Design Thinking*, penelitian ini diharapkan dapat menghasilkan sistem arsip digital yang tidak hanya berfungsi dengan baik secara teknis, tetapi juga benar-benar sesuai dengan kebutuhan pengguna di Universitas Prabumulih.

3.6 Alat Bantu Perancangan

Alat bantu perancangan sistem digunakan untuk membantu pengembang dalam membuat rancangan, implementasi, serta pengujian sistem berbasis *web* secara efektif dan efisien. Dalam penelitian ini, alat bantu yang digunakan terdiri atas beberapa komponen utama, yaitu *development environment*, bahasa pemodelan visual, alat bantu perancangan basis data, desain antarmuka pengguna (*UI/UX*), serta alat debugging dan pengembangan. Penggunaan alat bantu ini bertujuan untuk memastikan proses pengembangan sistem *Smart Web: Arsip Digital Berbasis OCR dan Rule-Based System* dapat berjalan secara terstruktur, efisien, dan mudah dipahami.

3.6.1 Alat Bantu Utama

Visual Studio Code merupakan *Integrated Development Environment (IDE)* yang digunakan dalam proses pengembangan aplikasi. IDE ini menyediakan berbagai fitur pendukung seperti *syntax highlighting*, *auto-completion*, *extension support*, serta integrasi dengan *version control system* seperti *Git*. Fitur-fitur tersebut sangat membantu dalam proses penulisan, pengujian, serta pengelolaan kode program agar lebih efisien dan terstruktur. Pemilihan *Visual Studio Code* dilakukan karena sifatnya yang ringan, fleksibel, dan mendukung berbagai bahasa pemrograman maupun *framework* modern, sehingga sesuai dengan kebutuhan pengembangan aplikasi arsip digital yang berbasis *web* maupun *mobile*.

Flutter berperan sebagai *framework* utama dalam pengembangan aplikasi. *Framework* ini digunakan untuk membangun tampilan antarmuka pengguna (*User*

interface) sekaligus mengatur interaksi antara pengguna dan sistem. *Flutter* memungkinkan pengembang untuk membuat aplikasi dengan desain yang responsif dan performa tinggi melalui satu basis kode (*single codebase*) yang dapat dijalankan di berbagai *platform*, seperti *Android*, *iOS*, dan *web*. Selain itu, fitur *hot reload* yang dimilikinya memudahkan proses pengujian serta mempercepat iterasi pengembangan, sehingga efisiensi waktu dapat tercapai dalam proses implementasi sistem.

3.6.2 Alat Bantu Pemodelan Visual

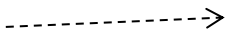
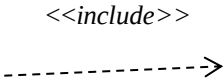
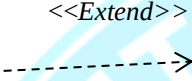
Bahasa pemodelan visual digunakan untuk membantu menggambarkan struktur dan perilaku sistem secara konseptual sebelum tahap implementasi dilakukan. Dengan adanya pemodelan visual, pengembang dapat memahami hubungan antar komponen sistem, alur proses yang terjadi, serta interaksi antara pengguna dan sistem secara lebih komprehensif. Selain itu, diagram visual juga berfungsi sebagai media komunikasi antara pengembang, pembimbing, dan pihak terkait lainnya agar rancangan sistem dapat dipahami dengan jelas dan seragam sebelum masuk ke tahap pengkodean. Dalam penelitian ini, bahasa pemodelan visual yang digunakan meliputi *Use Case Diagram*, *Activity Diagram*, dan *ERD*.

a. *Use Case Diagram*

Use Case Diagram digunakan untuk memetakan hubungan antara aktor (pengguna) dengan sistem, seperti proses *Login*, unggah dokumen, pemrosesan OCR, hingga validasi arsip digital menggunakan *Rule-Based System*. Diagram ini memberikan gambaran umum mengenai fungsionalitas utama yang disediakan oleh sistem. Adapun simbol *use case diagram* antara lain:

Tabel 3.2 Simbol *Usecase*

No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi itu sendiri.

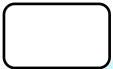
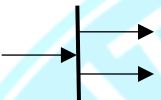
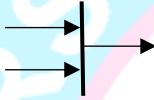
2		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (independent).
3		Generalization	Hubungan generalisasi dan spesialisasi (umum-khusus) antar dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya.
4		include	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya.
5		Extend	Relasi use case tambahan ke sebuah use case, dimana use case yang ditambahkan dapat berdiri sendiri
6		Association	Komunikasi antar aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
7		System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		Usecase	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor

Sumber : Suharni et al. (2023)

b. *Activity Diagram*

Activity Diagram digunakan untuk menggambarkan alur aktivitas dan proses bisnis yang terjadi di dalam sistem. Diagram ini memvisualisasikan rangkaian aktivitas yang saling berkaitan dari awal hingga akhir proses, sehingga pengembang dapat memahami bagaimana sistem bekerja secara menyeluruh. Fokus utama *Activity Diagram* adalah pada aktivitas sistem dan alur proses yang dijalankan, bukan pada tindakan aktor secara detail. Dengan adanya *Activity Diagram*, pengembang dapat melihat hubungan antar aktivitas, jalur proses, serta alur kerja sistem dari level umum secara jelas dan sistematis. Adapun simbol *Activity Diagram* antara lain:

Tabel 3.3 Simbol Activity Diagram

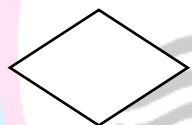
No	Simbol	Nama	Keterangan
1		<i>Initial</i>	Awal dari kumpulan aktifitas atau aksi
2		<i>Final</i>	Menghentikan seluruh aliran control atau aliran obyek pada sebuah aksi atau aktifitas
3		<i>Activities</i>	Kumpulan aktifitas atau aksi
4		<i>Fork</i> (Percabangan)	Memecah sebuah perilaku menjadi aksi atau aktifitas paralel
5		<i>Join</i> (Penggabungan)	Menggabungkan Kembali menjadi satu aliran perilaku dari aksi atau aktifitas paralel
6		<i>Decision</i>	Mewakili suatu pengujian yang bertujuan untuk memastikan aliran kontrol dan aliran obyek menuju ke satu arah
7		<i>Swimlane</i>	Kolom yang berguna untuk mengelompokkan aktifitas tertentu berdasarkan Actor

Sumber : (Andriyanto, 2022)

c. *Entity Relational Diagram (ERD)*

ERD atau *Entity Relationship Diagram* adalah diagram yang menggambarkan hubungan antar entitas (objek data) di dalam sistem, lengkap dengan atribut-atribut dan jenis relasinya. Entitas di sini bisa berupa *orang*, *tempat*, *objek*, atau *peristiwa* yang datanya ingin disimpan di *database*.

Tabel 3.4 Simbol ERD

No.	Simbol	Nama Simbol	Fungsi / Penjelasan
1		Atribut	Simbol yang menjelaskan karakteristik suatu entitas dan juga relasinya. Digambarkan dengan elips.
2		Entitas	Merupakan suatu simbol untuk mewakili suatu objek dengan karakteristik sama yang dilengkapi oleh atribut.
3		Relasi	Simbol yang digunakan untuk menghubungkan beberapa entitas berdasarkan fakta pada suatu lingkungan. Digambarkan dengan simbol belah ketupat.
4		Connection	Simbol untuk merangkai keterkaitan antara notasi-notasi yang berupa garis penghubung yang digunakan seperti entitas, relasi dan atribut

Sumber: L. Sari & Siregar (2021)

3.6.3 Alat Bantu Desain

Dalam proses pengembangan sistem, tahap desain memiliki peran penting untuk menerjemahkan kebutuhan pengguna menjadi rancangan visual dan fungsional dari aplikasi yang akan dibangun. Proses ini mencakup pembuatan tata letak halaman, penentuan elemen antarmuka, warna, ikon, serta alur interaksi antara pengguna dan sistem. Tahapan desain dilakukan sebelum proses implementasi agar pengembang memiliki acuan yang jelas dalam membangun tampilan dan fitur aplikasi sesuai dengan hasil analisis kebutuhan.

Figma digunakan sebagai alat bantu perancangan antarmuka pengguna (*User interface Design Tool*) dalam proses tersebut. Melalui Figma, peneliti dapat membuat rancangan visual berupa tata letak halaman, posisi elemen, ikon, warna, serta alur navigasi aplikasi sebelum tahap implementasi dilakukan. Penggunaan Figma bertujuan untuk mempermudah proses perancangan tampilan agar hasil desain sesuai dengan kebutuhan pengguna dan prinsip *user experience* yang baik. Selain itu, Figma memungkinkan kolaborasi secara *real-time* dan penyimpanan berbasis *cloud*, sehingga proses revisi desain dapat dilakukan dengan lebih cepat dan efisien sebelum diimplementasikan ke dalam kode menggunakan *Flutter*.

3.6.4 Alat Bantu Debugging

Dalam proses pengembangan aplikasi, tahap debugging merupakan langkah penting untuk memastikan bahwa sistem berjalan sesuai dengan logika dan bebas dari kesalahan (*error*). Debugging dilakukan untuk mendeteksi, menganalisis, serta memperbaiki kesalahan kode program yang dapat memengaruhi tampilan maupun fungsi aplikasi.

Pada penelitian ini, proses debugging dilakukan melalui *Visual Studio Code* dengan memanfaatkan fitur *debugger* bawaan yang terintegrasi dengan *Flutter*. Aplikasi dijalankan (*run*) menggunakan mode pengujian *web*, di mana hasil implementasi kode dapat langsung dilihat melalui *web browser* seperti Google Chrome. Dengan cara ini, pengembang dapat memantau alur eksekusi program secara *real-time*, menelusuri bagian kode yang bermasalah, serta melakukan perbaikan secara cepat melalui fitur *hot reload* yang disediakan *Flutter*.

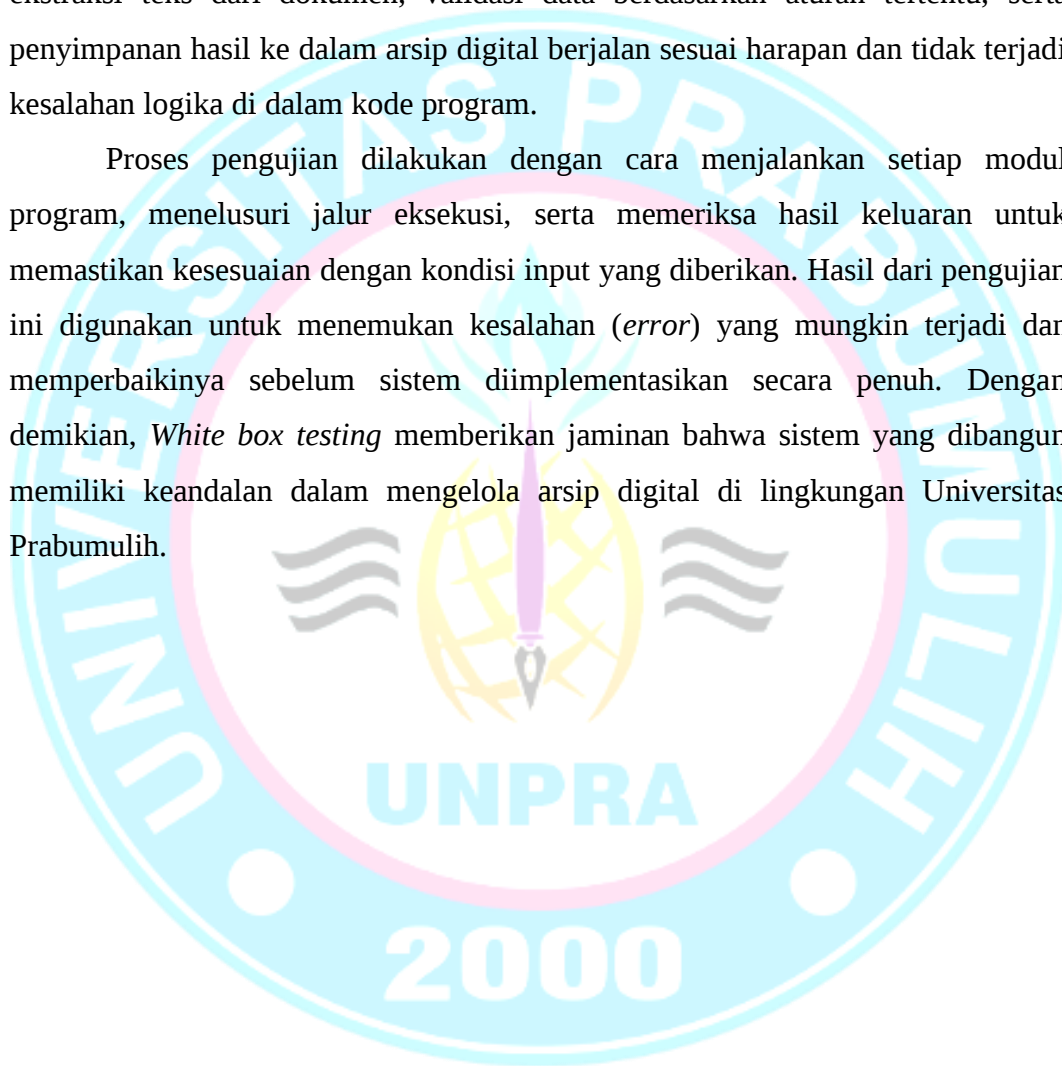
3.7 Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa aplikasi arsip digital berbasis *OCR* dan *Rule-Based System* telah berfungsi sesuai dengan rancangan yang dibuat. Metode pengujian yang digunakan dalam penelitian ini adalah *White box testing*. *White box testing* adalah metode pengujian perangkat lunak yang menitikberatkan pada pemeriksaan struktur internal sistem, di mana penguji memiliki akses penuh terhadap kode sumber dan logika program. Melalui metode ini, pengujian dilakukan secara menyeluruh terhadap alur kontrol, eksekusi

pernyataan program, serta pengolahan data untuk menemukan potensi kesalahan yang tidak terlihat dari sisi pengguna (Fauzi & Yunial, 2025).

Metode *White box* dipilih karena sistem yang dikembangkan memiliki komponen logika yang kompleks, terutama pada proses pembacaan teks menggunakan *OCR* dan penerapan aturan pada *Rule-Based System*. Dengan pengujian ini, penulis dapat memastikan bahwa setiap fungsi utama seperti proses ekstraksi teks dari dokumen, validasi data berdasarkan aturan tertentu, serta penyimpanan hasil ke dalam arsip digital berjalan sesuai harapan dan tidak terjadi kesalahan logika di dalam kode program.

Proses pengujian dilakukan dengan cara menjalankan setiap modul program, menelusuri jalur eksekusi, serta memeriksa hasil keluaran untuk memastikan kesesuaian dengan kondisi input yang diberikan. Hasil dari pengujian ini digunakan untuk menemukan kesalahan (*error*) yang mungkin terjadi dan memperbaikinya sebelum sistem diimplementasikan secara penuh. Dengan demikian, *White box testing* memberikan jaminan bahwa sistem yang dibangun memiliki keandalan dalam mengelola arsip digital di lingkungan Universitas Prabumulih.



BAB IV

ANALISA DAN PERANCANGAN

4.1 Analisa Sistem

4.1.1 Analisa dan Evaluasi Sistem Berjalan

a. Analisa Sistem Berjalan

Proses pengelolaan arsip surat masuk dan keluar di Unit Rektorat Universitas Prabumulih saat ini masih dilakukan secara manual dan belum menggunakan sistem berbasis aplikasi. Surat masuk dicatat terlebih dahulu dalam buku agenda, kemudian dilampirkan lembar disposisi dan diteruskan kepada pimpinan untuk mendapatkan arahan. Setelah diproses oleh bagian terkait, dokumen fisik disimpan dalam map atau ordner sesuai kategori dan diletakkan pada rak arsip. Surat keluar juga dicatat secara manual dalam buku agenda lalu dibuat salinan dan diarsipkan sebelum dokumen asli dikirim ke pihak tujuan. Arsip disimpan dalam bentuk fisik dengan masa simpan maksimal lima (5) tahun. Setelah melewati masa simpan tersebut, dokumen dipindahkan ke gudang arsip atau dimusnahkan sesuai ketentuan kearsipan yang berlaku.

Sistem pengarsipan manual tersebut memiliki beberapa kelemahan. Proses pencarian arsip membutuhkan waktu yang cukup lama, terutama ketika jumlah dokumen semakin banyak dan arsip yang dicari merupakan arsip lama. Ketelitian diperlukan dalam pencatatan dan penempatan dokumen agar arsip dapat ditemukan kembali, namun tetap terdapat risiko kesalahan pencatatan, duplikasi data, kehilangan atau kerusakan fisik dokumen. Selain itu, penyimpanan arsip dalam bentuk fisik dengan masa simpan yang panjang menyebabkan penumpukan dokumen dan memerlukan ruang penyimpanan yang besar. Kondisi ini mengakibatkan pengelolaan arsip menjadi kurang efisien dan berpotensi menghambat proses penelusuran informasi ketika arsip dibutuhkan kembali.

b. Evaluasi Sistem Berjalan

Berdasarkan analisa terhadap sistem pengarsipan yang berjalan saat ini, diperlukan solusi yang mampu mengatasi kelemahan dalam pencarian,

penyimpanan, dan pengelolaan arsip. Sistem yang diusulkan adalah sistem arsip digital berbasis *OCR* dan *Rule-Based System* yang memungkinkan dokumen fisik dialihmediakan menjadi arsip digital. Melalui proses pemindaian dan pengenalan karakter (*OCR*), informasi pada dokumen dapat diekstraksi secara otomatis sehingga memudahkan proses pencatatan dan penelusuran arsip.

Sistem arsip digital yang diusulkan diharapkan mampu mempercepat proses pencarian dokumen melalui fitur pencarian berbasis kata kunci, mengurangi risiko kehilangan dan kerusakan arsip fisik, serta mengoptimalkan penggunaan ruang penyimpanan. Selain itu, penerapan *Rule-Based System* memungkinkan pengelompokan dan validasi data arsip secara lebih terstruktur sesuai dengan aturan yang telah ditetapkan. Dengan demikian, sistem usulan ini dapat meningkatkan efisiensi, akurasi, dan keandalan pengelolaan arsip dibandingkan dengan sistem manual yang digunakan sebelumnya.

4.1.2 Analisa Kebutuhan

a. Kebutuhan Pengguna (*User*)

Sistem aplikasi arsip digital berbasis *OCR* dan *Rule-Based System* ini digunakan oleh beberapa jenis pengguna (*role*) dengan hak akses berbeda, yaitu:

1. Operator

Operator yang merupakan staf administrasi rektorat ditunjuk untuk bertugas mengelola arsip surat masuk dan surat keluar dengan cara mengunggah dokumen surat masuk dan keluar, menulis surat keluar, melihat hasil ekstraksi *OCR*, melakukan validasi hasil ekstraksi, menyimpan arsip, melihat disposisi, serta melakukan pencarian arsip.

2. Pimpinan

Pimpinan dalam hal ini Rektor sebagai pimpinan unit Rektorat bertugas menerima surat masuk dan surat keluar dari Operator, membaca detail surat, serta membuat dan mengirim disposisi untuk ditujukan kepada unit terkait.

3. Administrator

Bertugas mengelola akun pengguna sistem, seperti menambah *user*, mengatur *role*, dan *reset password*.

b. Kebutuhan Data

Data yang dibutuhkan dalam Aplikasi arsip di meliputi:

1. Data *User*

Data *user* merupakan data yang digunakan untuk mengelola akun pengguna dalam sistem Aplikasi Arsip Digital Berbasis *OCR* dan *Rule-Based System*. Data ini meliputi nama pengguna, *username* sebagai identitas *Login*, *password* untuk autentikasi, serta *role* atau peran pengguna dalam sistem, seperti Operator, Pimpinan dan Administrator yang mengelola akun staf dan pimpinan. Data *user* berfungsi untuk mengatur hak akses dan memastikan setiap pengguna hanya dapat mengakses fitur sesuai dengan tugas dan kewenangannya.

2. Data Surat Masuk

Data surat masuk adalah data yang berkaitan dengan dokumen yang diterima oleh instansi dan dikelola melalui sistem Aplikasi Arsip Digital Berbasis *OCR* dan *Rule-Based System*. Data ini mencakup nomor surat, tanggal surat, nama atau instansi pengirim, perihal surat, serta *File* surat dalam bentuk digital seperti *PDF*, *JPG*, *JPEG*, atau *PNG*. Data surat masuk digunakan sebagai arsip digital yang dapat dicari, ditampilkan, dan diproses lebih lanjut dalam sistem, termasuk untuk keperluan disposisi.

3. Data Surat Keluar

Data surat keluar merupakan data yang memuat informasi terkait surat yang dikeluarkan oleh instansi kepada pihak atau unit di luar organisasi. Data ini meliputi nomor surat, tanggal surat, tujuan pengiriman, perihal surat. *File* surat dalam bentuk format digital seperti *PDF*, *JPG*, *JPEG*, atau *PNG*. Data surat keluar berfungsi sebagai arsip resmi instansi yang dapat ditelusuri kembali apabila diperlukan, serta digunakan sebagai bahan pendukung dalam pembuatan laporan dan dokumentasi administrasi.

c. Kebutuhan fungsional

Kebutuhan fungsional merupakan penguraian dari suatu sistem secara utuh ke dalam bagian-bagian fungsionalnya dengan tujuan untuk mengidentifikasi dan mengevaluasi permasalahan, peluang, hambatan, serta kebutuhan yang diharapkan, sehingga dapat diusulkan perbaikan-perbaikan yang sesuai. Kebutuhan fungsional dari Sistem Arsip Digital Aplikasi Arsip Digital Berbasis *OCR* dan *Rule-Based System* yang akan dibangun adalah sebagai berikut:

1. Sistem dapat melakukan proses *Login* dengan cara memverifikasi username dan password pengguna yang telah terdaftar sesuai dengan peran atau *role* masing-masing, seperti Operator, Pimpinan, dan Administrator.
2. Sistem dapat mengelola data *user*, meliputi proses penambahan *user* baru, pengubahan *role* pengguna, serta *reset password* oleh Administrator.
3. Sistem dapat menerima unggahan (*upload*) surat masuk dalam bentuk *File* digital seperti *PDF*, *JPG*, *JPEG*, atau *PNG*.
4. Sistem dapat melakukan proses *Optical Character Recognition (OCR)* secara otomatis terhadap *File* surat yang diunggah tanpa perlu perintah manual dari pengguna.
5. Sistem dapat menampilkan hasil teks dari proses *OCR* untuk diverifikasi oleh Operator sebelum disimpan ke dalam arsip.
6. Sistem dapat mengelola data surat masuk, meliputi proses penambahan metadata surat, pengubahan data, dan penyimpanan surat ke dalam arsip digital.
7. Sistem dapat menampilkan daftar surat masuk kepada pimpinan.
8. Sistem dapat memfasilitasi pimpinan untuk membaca detail surat dan *File* lampiran serta mengisi disposisi secara langsung melalui sistem.
9. Sistem dapat mengelola proses disposisi surat, meliputi pengisian tujuan unit, instruksi, serta jenis surat berdasarkan urgensinya.
10. Sistem dapat menyimpan dan menampilkan riwayat disposisi surat.

11. Sistem dapat mengelola data surat keluar, meliputi penulisan surat keluar, penambahan atau *upload*, penyimpanan, dan pengarsipan surat keluar secara digital.
12. Sistem dapat melakukan pencarian arsip surat berdasarkan kata kunci tertentu.

d. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan kebutuhan yang berkaitan dengan kualitas sistem, bukan pada fungsi utama sistem. Kebutuhan ini mencakup aspek keamanan, kinerja, kemudahan penggunaan, keandalan, dan kompatibilitas sistem. Kebutuhan non-fungsional pada Sistem Arsip Digital Berbasis OCR dan *Rule-Based System* adalah sebagai berikut:

1. Keamanan (*Security*)

Sistem harus memiliki mekanisme autentikasi berupa *username* dan *password* untuk membatasi akses pengguna. Setiap pengguna hanya dapat mengakses fitur sesuai dengan peran (*role*) yang dimilikinya. Data surat dan arsip digital harus tersimpan dengan aman di server dan tidak dapat diakses oleh pihak yang tidak berwenang.

2. Kinerja (*Performance*)

Sistem harus mampu memproses unggahan dokumen dan menjalankan OCR secara otomatis dengan waktu respon yang cepat. Proses pencarian arsip surat juga harus berjalan efisien meskipun jumlah data semakin bertambah.

3. Kemudahan Penggunaan (*Usability*)

Antarmuka sistem harus dirancang sederhana, jelas, dan mudah dipahami oleh pengguna dari berbagai latar belakang. Menu dan fitur ditampilkan secara terstruktur agar pengguna dapat mengoperasikan sistem tanpa memerlukan pelatihan khusus.

4. Keandalan (*Reliability*)

Sistem harus dapat berjalan secara stabil dan meminimalkan terjadinya kesalahan saat mengelola data surat. Data yang telah disimpan tidak boleh hilang dan harus dapat ditampilkan kembali dengan benar.

5. Ketersediaan (*Availability*)

Sistem diharapkan dapat diakses kapan saja selama terhubung dengan jaringan internet, sehingga pengguna dapat mengelola dan melihat arsip surat tanpa terikat waktu dan tempat.

6. Kompatibilitas (*Compatibility*)

Sistem harus dapat dijalankan pada berbagai perangkat dan sistem operasi, seperti *Windows*, *Linux*, dan *macOS* melalui *web browser* modern.

e. Analisa Perangkat Keras dan Perangkat lunak

Perangkat keras komputer merupakan komponen fisik yang berfungsi dalam proses pemasukan data (*input*), pengolahan data (*process*), hingga menghasilkan keluaran (*output*). Perangkat ini dapat dilihat dan disentuh secara langsung karena memiliki bentuk nyata. Secara fungsional, perangkat keras berperan dalam mengumpulkan, memasukkan, memproses, menyimpan, serta menyajikan hasil pengolahan data menjadi informasi yang dapat digunakan (Putri et al., 2023). Berikut kebutuhan Perangkat Keras (*Hardware*) yang diperlukan untuk menjalankan aplikasi arsip digital berbasis *OCR* dan *Rule-Based System*.

1. *Processor* : Minimal Intel Celeron / AMD Athlon
2. *RAM* : Minimal 4 GB
3. *Storage* : Minimal 128 GB
4. Monitor
5. Keyboard
6. Mouse

Selain perangkat keras, penelitian ini juga membutuhkan perangkat lunak. Perangkat lunak (*software*) merupakan sekumpulan program yang berisi instruksi untuk mengolah data. *Software* berperan sebagai perantara antara pengguna dengan perangkat keras komputer, yaitu dengan menerjemahkan perintah dari bahasa manusia ke dalam bahasa mesin. Dengan demikian, perangkat keras dapat memahami keinginan pengguna, menjalankan instruksi yang diberikan, serta menghasilkan keluaran sesuai tujuan yang diharapkan (Silaen et al., 2022). Berikut

kebutuhan Perangkat Lunak (*Software*) yang digunakan untuk membangun aplikasi arsip digital berbasis *OCR* dan *Rule-Based System*.

1. Sistem Operasi : Microsoft *Windows* 10
2. Bahasa Pemrograman *Backend* : *Python* 3.10
3. *Framework Backend* : *FastAPI*
4. Bahasa Pemrograman *Frontend* : *Dart* 3.3.0
5. *Framework Frontend* : *Flutter Web* 3.13.0
6. *OCR Engine* : *Tesseract OCR* 5.0
7. *Database* : *MySQL*
8. *Code Editor* : *Visual Studio Code*
9. *Browser* : *Google Chrome* versi 145

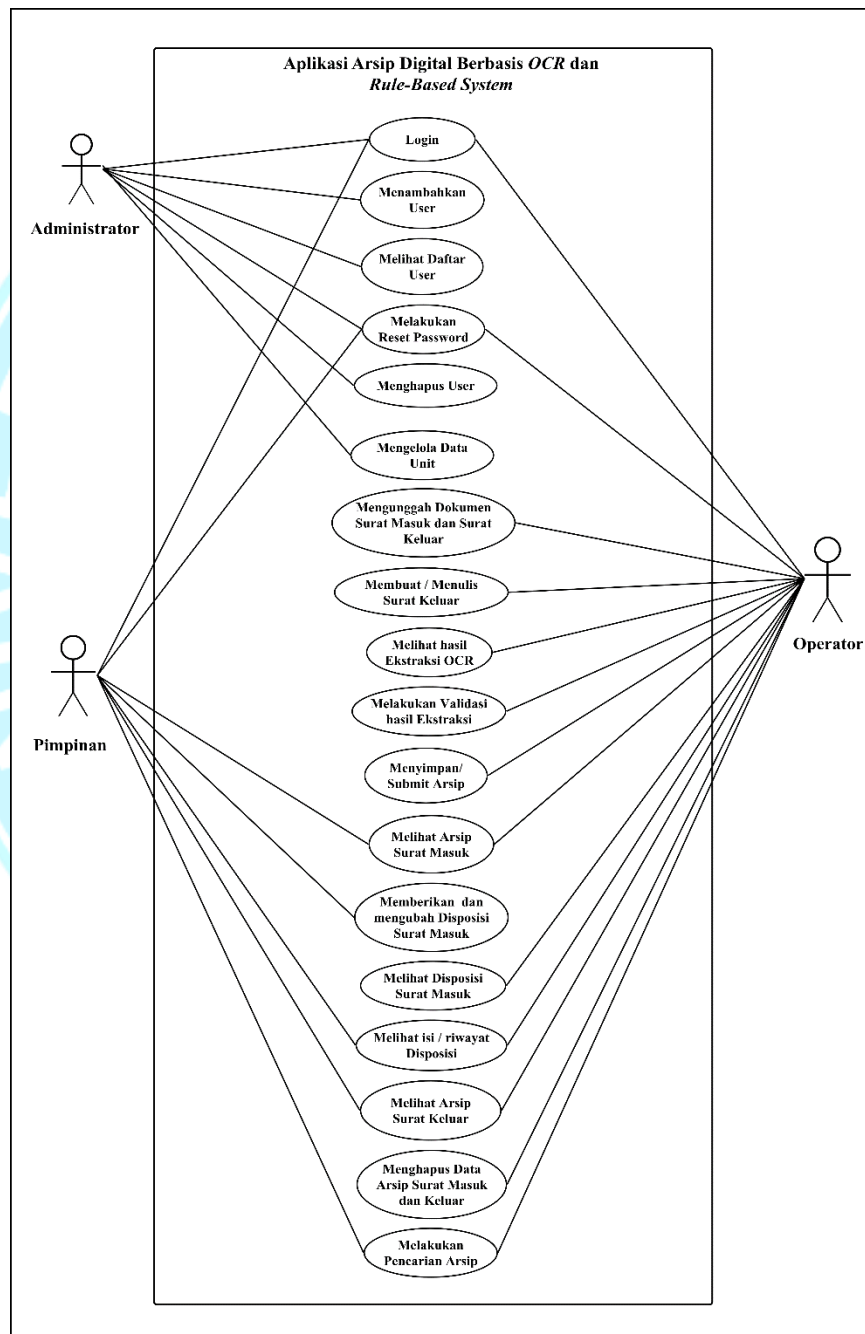
4.2 Perancangan Sistem yang Diusulkan

4.2.1 Perancangan Sistem (Logis)

Perancangan sistem secara logis merupakan tahap yang menjelaskan bagaimana sistem aplikasi arsip digital berbasis *OCR* dan *Rule-Based System* bekerja dari sisi fungsional tanpa memperhatikan implementasi teknis perangkat keras maupun perangkat lunak yang digunakan. Pada tahap ini, fokus utama adalah pada alur proses, interaksi antar pengguna dengan sistem, serta fungsi-fungsi utama yang harus dijalankan oleh sistem agar sesuai dengan kebutuhan yang telah dianalisis sebelumnya. Perancangan sistem logis ini digambarkan melalui model-model seperti *use case diagram*, *Activity diagram*, dan diagram alur proses yang menunjukkan hubungan antara aktor dan sistem secara konseptual.

a. Use case Diagram

Use case Diagram digunakan untuk memetakan hubungan antara aktor (pengguna) dengan sistem, seperti proses *login*, unggah dokumen, pemrosesan *OCR*, hingga validasi arsip digital menggunakan *Rule-Based System*. Diagram ini memberikan gambaran umum mengenai fungsionalitas utama yang disediakan oleh sistem.



Sumber : data diolah oleh penulis (2026)

Gambar 4.1 Use case

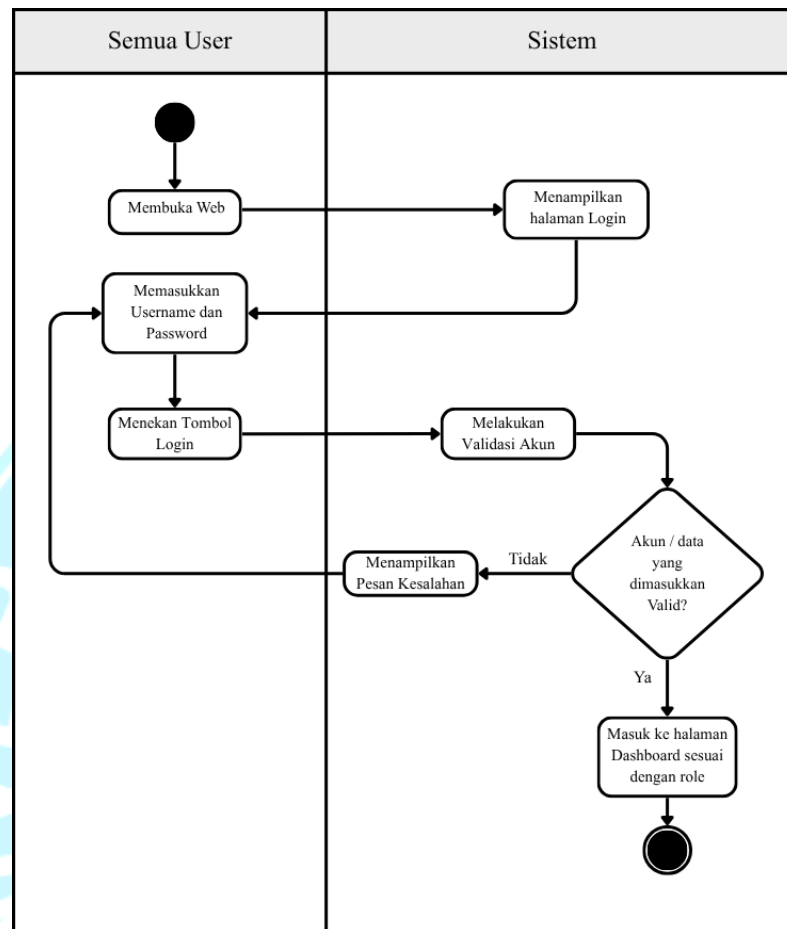
Use case Diagram menggambarkan interaksi antara aktor dengan sistem Aplikasi Arsip Digital Berbasis *OCR* dan *Rule-Based System*. Sistem ini memiliki tiga aktor utama yaitu;

1. Administrator bertugas mengelola pengguna sistem seperti menambahkan *user*, mereset *password*, dan melihat daftar *user*, meghapus *user*.
2. Operator bertugas mengelola arsip surat dengan cara mengunggah dokumen surat masuk dan keluar, membuat surat keluar, melihat hasil ekstraksi *OCR* , melakukan validasi hasil ekstraksi, menyimpan arsip, melihat disposisi, serta melakukan pencarian arsip.
3. Pimpinan berperan dalam pengambilan keputusan dengan melihat arsip surat masuk dan keluar, memberikan disposisi surat masuk, serta melihat isi dan riwayat disposisi.



b. Activity Diagram

1. Activity Login Umum User

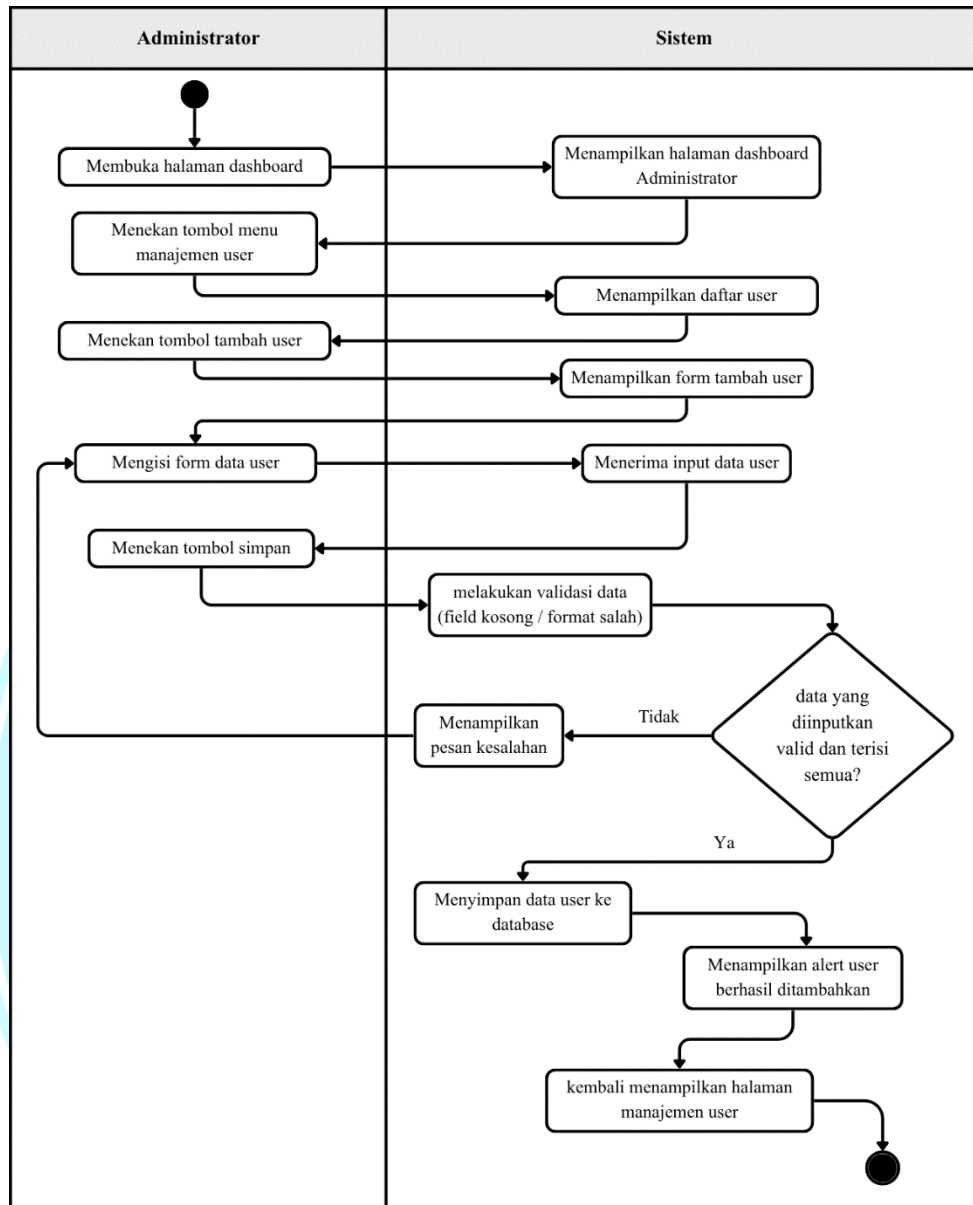


Sumber : data diolah oleh penulis (2026)

Gambar 4.2 Login Activity

Diagram *activity* pertama menggambarkan alur proses *login* yang dapat dilakukan oleh seluruh pengguna sistem. Proses dimulai ketika *user* membuka *website*, kemudian sistem menampilkan halaman *login*. Selanjutnya pengguna memasukkan *username* dan *password*, lalu menekan tombol *login*. Sistem melakukan validasi terhadap data akun yang diinputkan. Apabila data yang dimasukkan tidak valid, maka sistem akan menampilkan pesan kesalahan dan pengguna diminta untuk menginput ulang data *login*. Namun jika data valid, sistem akan mengarahkan pengguna ke halaman *dashboard* sesuai dengan *role* masing-masing. Proses berakhir ketika *dashboard* berhasil ditampilkan.

2. Activity Tambah User



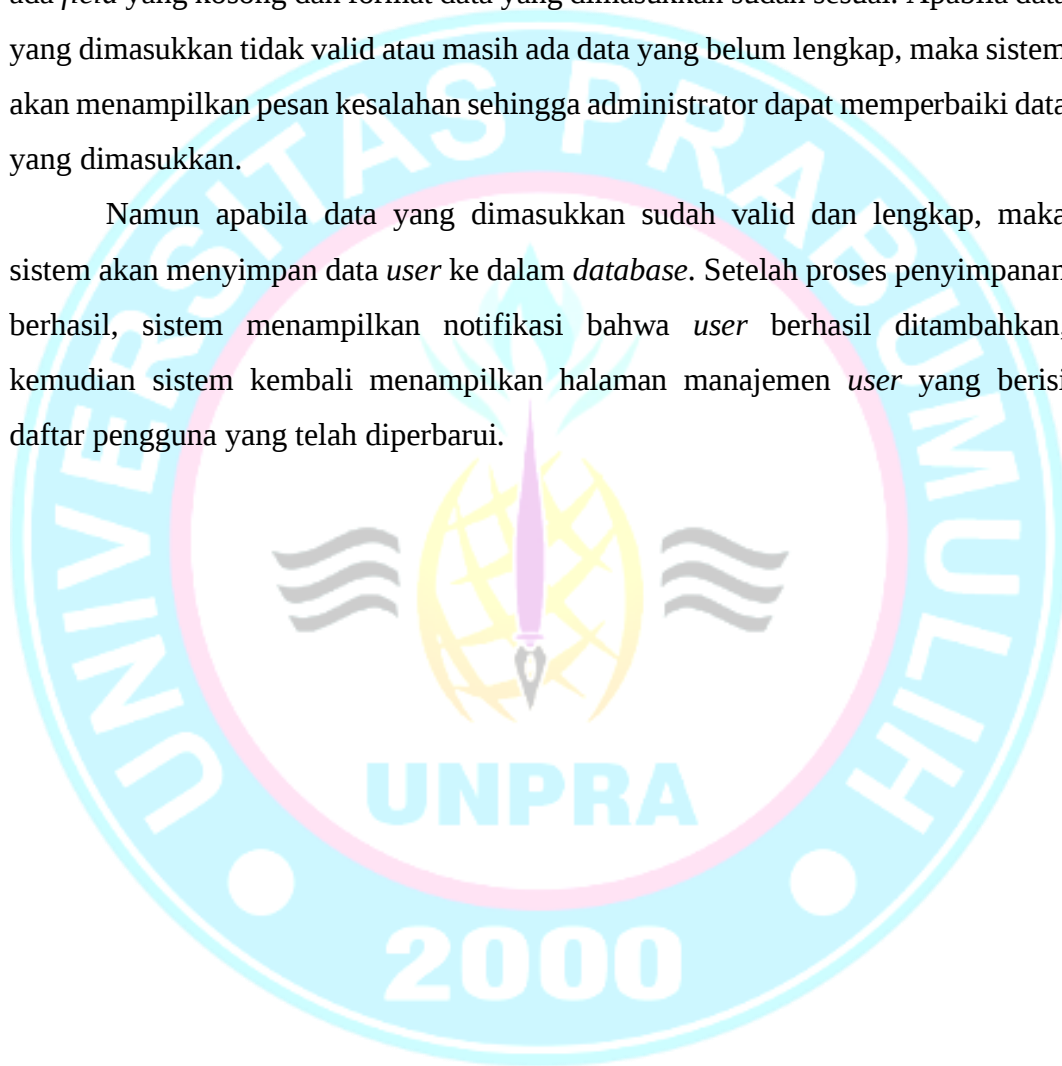
Sumber : data diolah oleh penulis (2026)

Gambar 4.3 Tambah User Activity

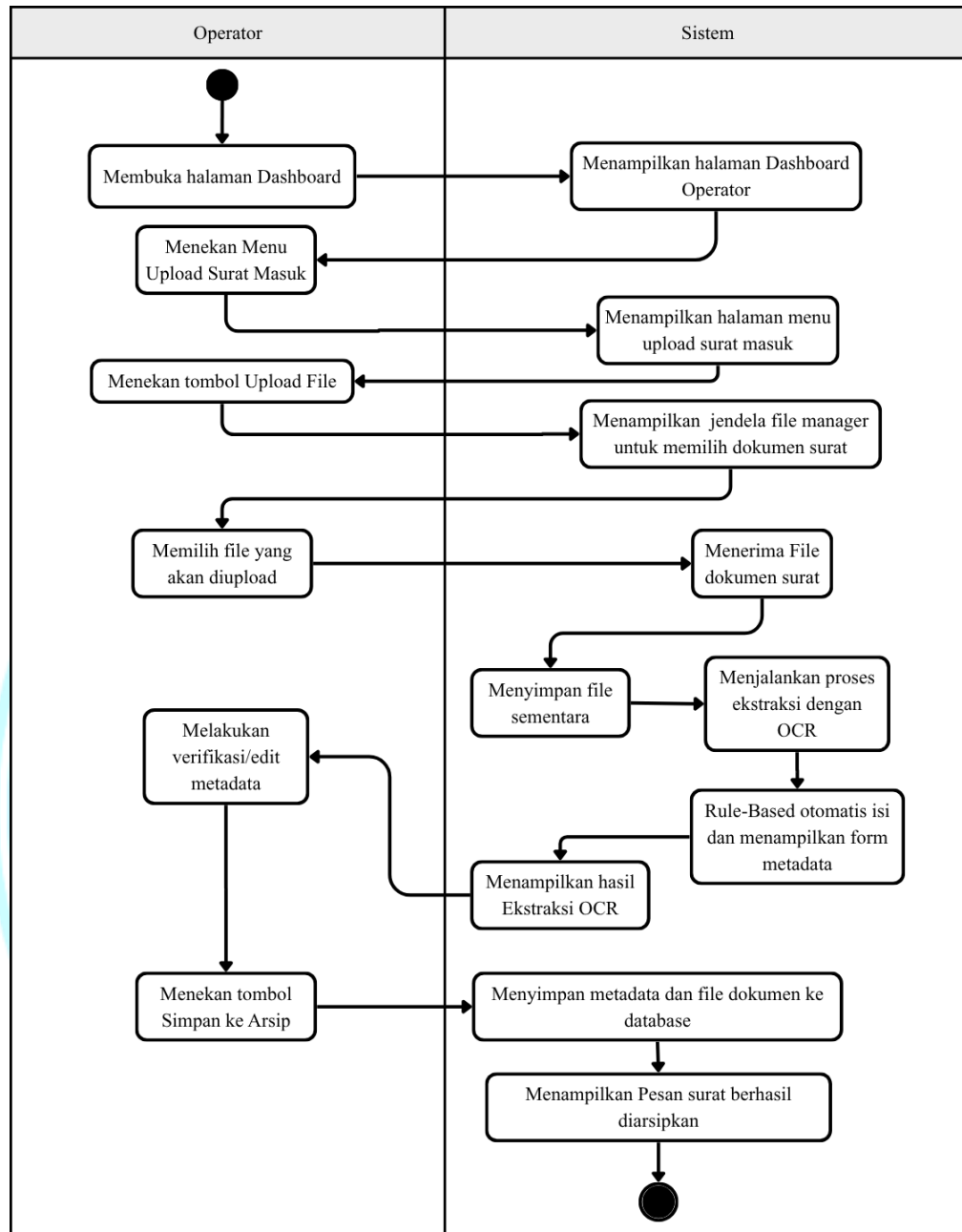
Activity diagram ini menggambarkan alur proses dimana administrator melakukan penambahan data pengguna baru pada sistem. Proses dimulai ketika administrator membuka halaman *dashboard*, kemudian sistem menampilkan halaman *dashboard* administrator. Setelah itu administrator menekan menu manajemen *user* sehingga sistem menampilkan daftar pengguna yang sudah terdaftar di dalam sistem.

Selanjutnya administrator menekan tombol tambah *user* untuk menambahkan pengguna baru. Sistem kemudian menampilkan form tambah *user* yang harus diisi oleh administrator. Pada tahap ini administrator mengisi data pengguna seperti nama, *e-mail*, *username*, *password*, *role*, dan unit. Setelah seluruh data diisi, administrator menekan tombol simpan sehingga sistem menerima input data tersebut. Sistem kemudian melakukan validasi data untuk memastikan tidak ada *field* yang kosong dan format data yang dimasukkan sudah sesuai. Apabila data yang dimasukkan tidak valid atau masih ada data yang belum lengkap, maka sistem akan menampilkan pesan kesalahan sehingga administrator dapat memperbaiki data yang dimasukkan.

Namun apabila data yang dimasukkan sudah valid dan lengkap, maka sistem akan menyimpan data *user* ke dalam *database*. Setelah proses penyimpanan berhasil, sistem menampilkan notifikasi bahwa *user* berhasil ditambahkan, kemudian sistem kembali menampilkan halaman manajemen *user* yang berisi daftar pengguna yang telah diperbarui.



3. Activity Upload Surat Masuk Operator



Sumber: data diolah oleh penulis (2026)

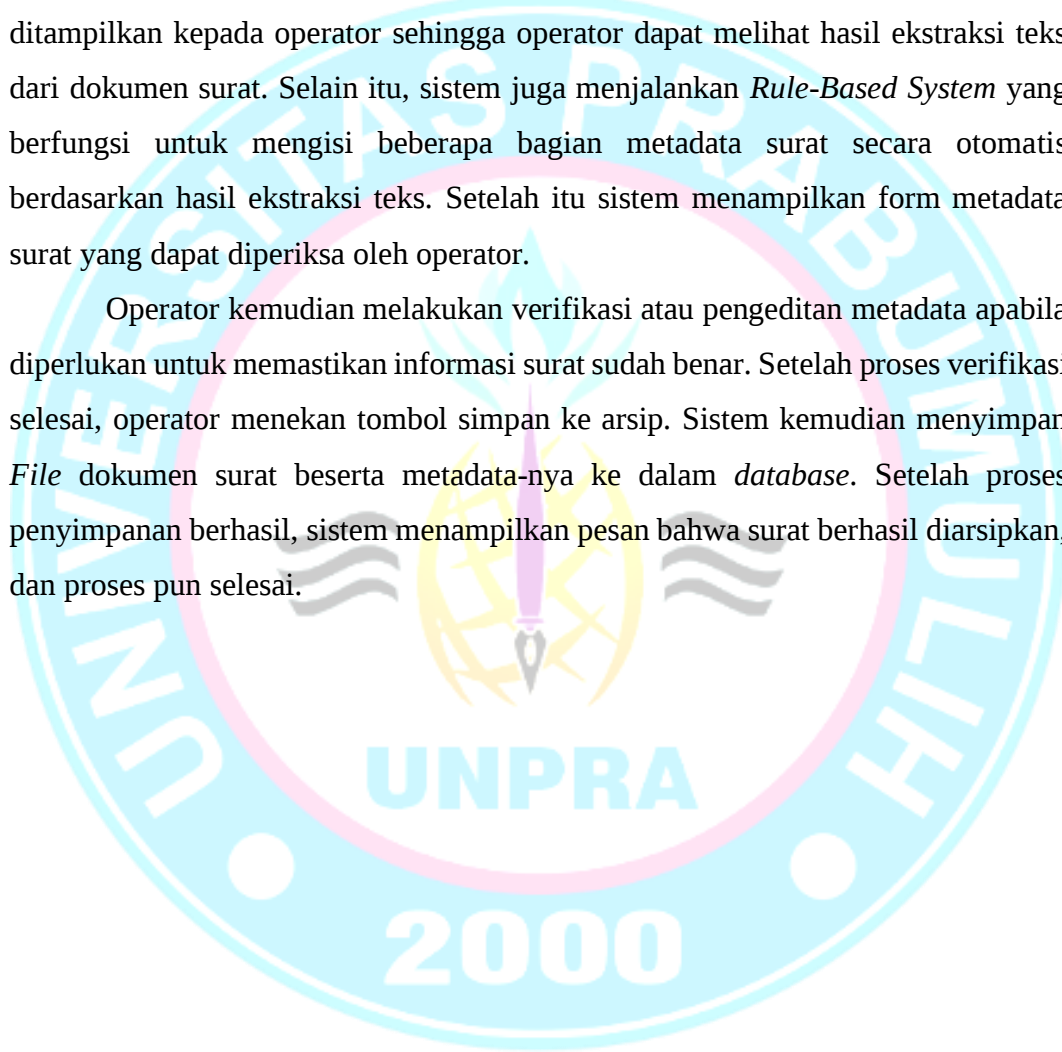
Gambar 4.4 Upload Surat Masuk Operator Activity

Activity diagram ini menggambarkan proses *upload* surat masuk oleh operator ke dalam sistem arsip digital. Proses dimulai ketika operator membuka halaman *dashboard*, kemudian sistem menampilkan halaman *dashboard* operator. Selanjutnya operator memilih menu *upload* surat masuk, sehingga sistem menampilkan halaman menu *upload* surat. Pada halaman ini operator menekan

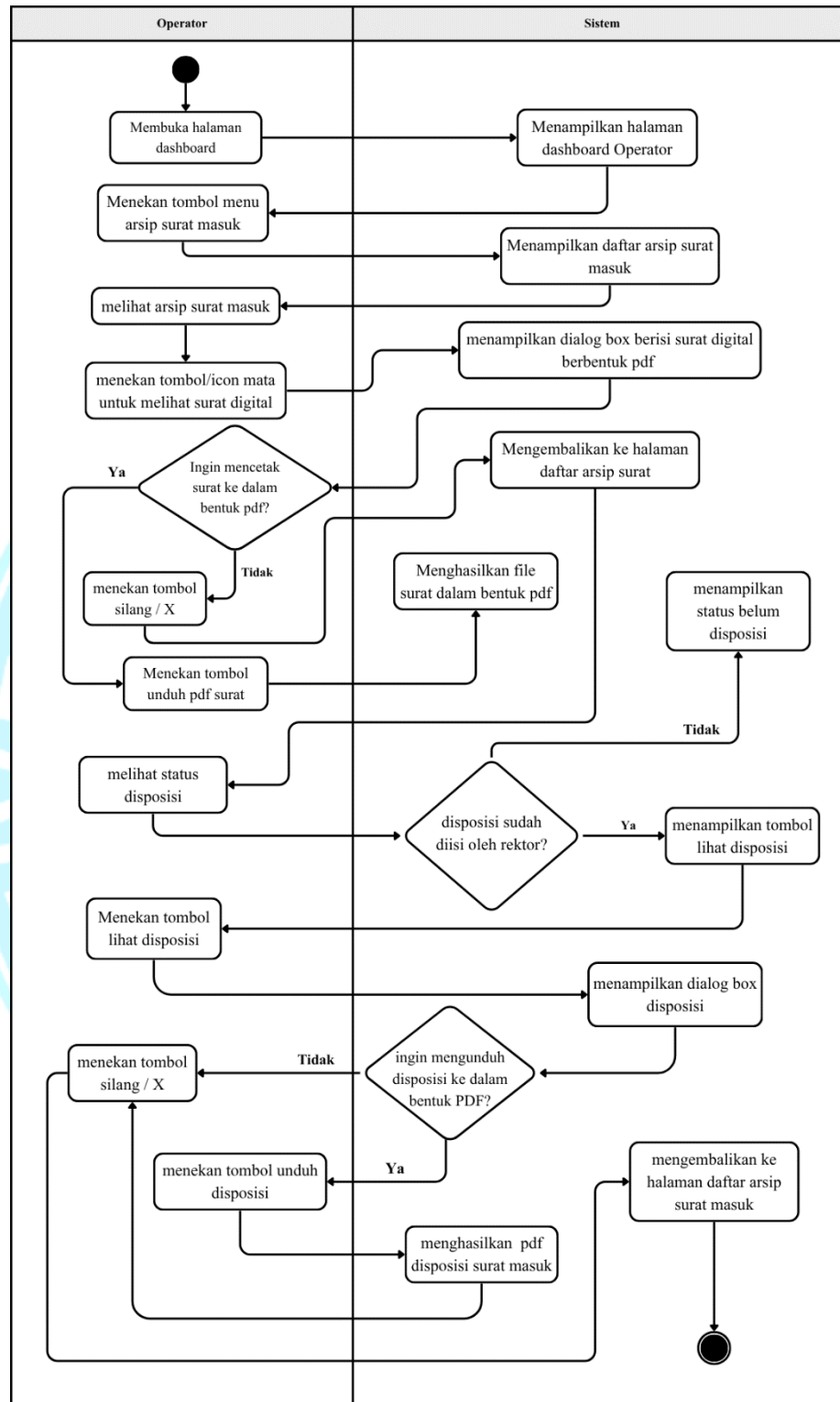
tombol *upload File* untuk memilih dokumen surat yang akan diunggah. Sistem kemudian menampilkan *File manager* yang memungkinkan operator memilih *File* surat dari perangkat.

Setelah operator memilih *File* yang akan diupload, sistem akan menerima *File* dokumen tersebut dan melakukan dua proses secara bersamaan, yaitu menyimpan *File* sementara dan menjalankan proses ekstraksi teks menggunakan *OCR (Optical Character Recognition)*. Hasil dari proses *OCR* kemudian ditampilkan kepada operator sehingga operator dapat melihat hasil ekstraksi teks dari dokumen surat. Selain itu, sistem juga menjalankan *Rule-Based System* yang berfungsi untuk mengisi beberapa bagian metadata surat secara otomatis berdasarkan hasil ekstraksi teks. Setelah itu sistem menampilkan form metadata surat yang dapat diperiksa oleh operator.

Operator kemudian melakukan verifikasi atau pengeditan metadata apabila diperlukan untuk memastikan informasi surat sudah benar. Setelah proses verifikasi selesai, operator menekan tombol simpan ke arsip. Sistem kemudian menyimpan *File* dokumen surat beserta metadata-nya ke dalam *database*. Setelah proses penyimpanan berhasil, sistem menampilkan pesan bahwa surat berhasil diarsipkan, dan proses pun selesai.



4. Activity Arsip Surat Masuk Operator



Sumber : data diolah oleh penulis (2026)

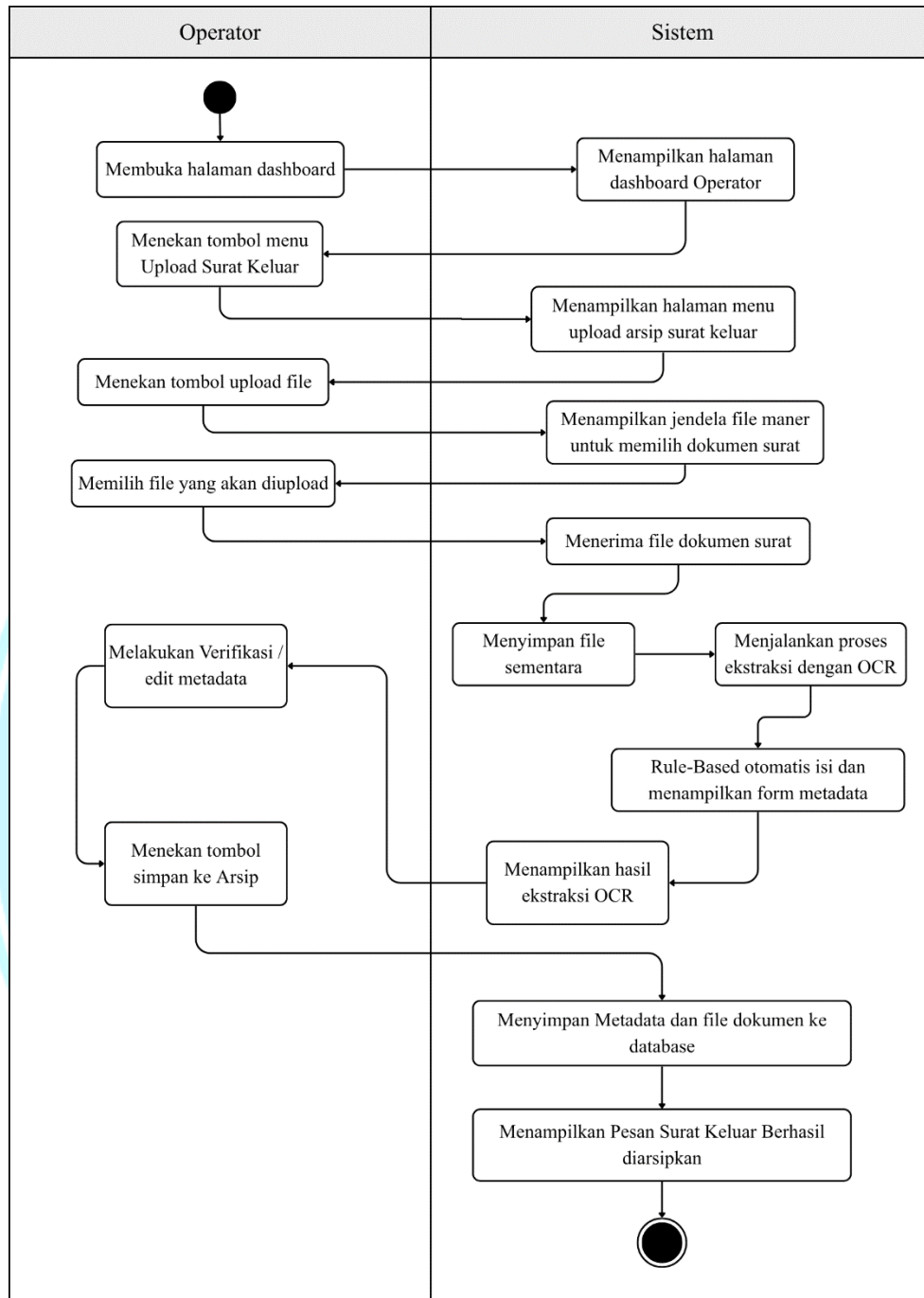
Gambar 4. 5 Arsip Surat Masuk Operator Activity

Activity diagram ini menggambarkan proses ketika operator melihat arsip surat masuk yang telah tersimpan di dalam sistem. Proses dimulai ketika operator membuka halaman *dashboard*, kemudian sistem menampilkan halaman *dashboard* operator. Selanjutnya operator menekan menu arsip surat masuk, sehingga sistem menampilkan daftar arsip surat masuk yang telah tersimpan di dalam sistem. Operator kemudian dapat melihat daftar surat yang tersedia dan memilih salah satu surat untuk dilihat secara lebih detail.

Untuk melihat dokumen surat, operator menekan ikon mata sehingga sistem menampilkan *dialog box* yang berisi surat digital dalam bentuk *PDF*. Pada tahap ini operator dapat membaca isi surat secara langsung melalui sistem. Apabila operator ingin mencetak atau mengunduh surat tersebut, operator dapat memilih opsi mengunduh surat dalam bentuk *PDF*. Sistem kemudian menghasilkan *File PDF* yang dapat diunduh oleh operator. Apabila operator tidak ingin mengunduh *File* tersebut, operator dapat menekan tombol tutup (X) sehingga sistem kembali menampilkan halaman daftar arsip surat masuk.

Selain itu operator juga dapat melihat status disposisi surat. Sistem akan memeriksa apakah disposisi surat sudah diisi oleh pimpinan atau belum. Apabila disposisi belum diisi, maka sistem akan menampilkan status belum disposisi. Namun apabila disposisi sudah tersedia, maka sistem akan menampilkan tombol lihat disposisi. Ketika operator menekan tombol tersebut, sistem akan menampilkan *dialog box* berisi disposisi surat. Operator juga dapat memilih untuk mengunduh disposisi dalam bentuk *PDF*. Jika operator memilih untuk mengunduh, maka sistem akan menghasilkan *File PDF* disposisi surat yang dapat disimpan oleh operator. Setelah proses selesai, sistem kembali menampilkan halaman daftar arsip surat masuk.

5. Activity Upload Surat Keluar Operator



Sumber : data diolah oleh penulis (2026)

Gambar 4.6 Upload Surat Keluar Operator Activity

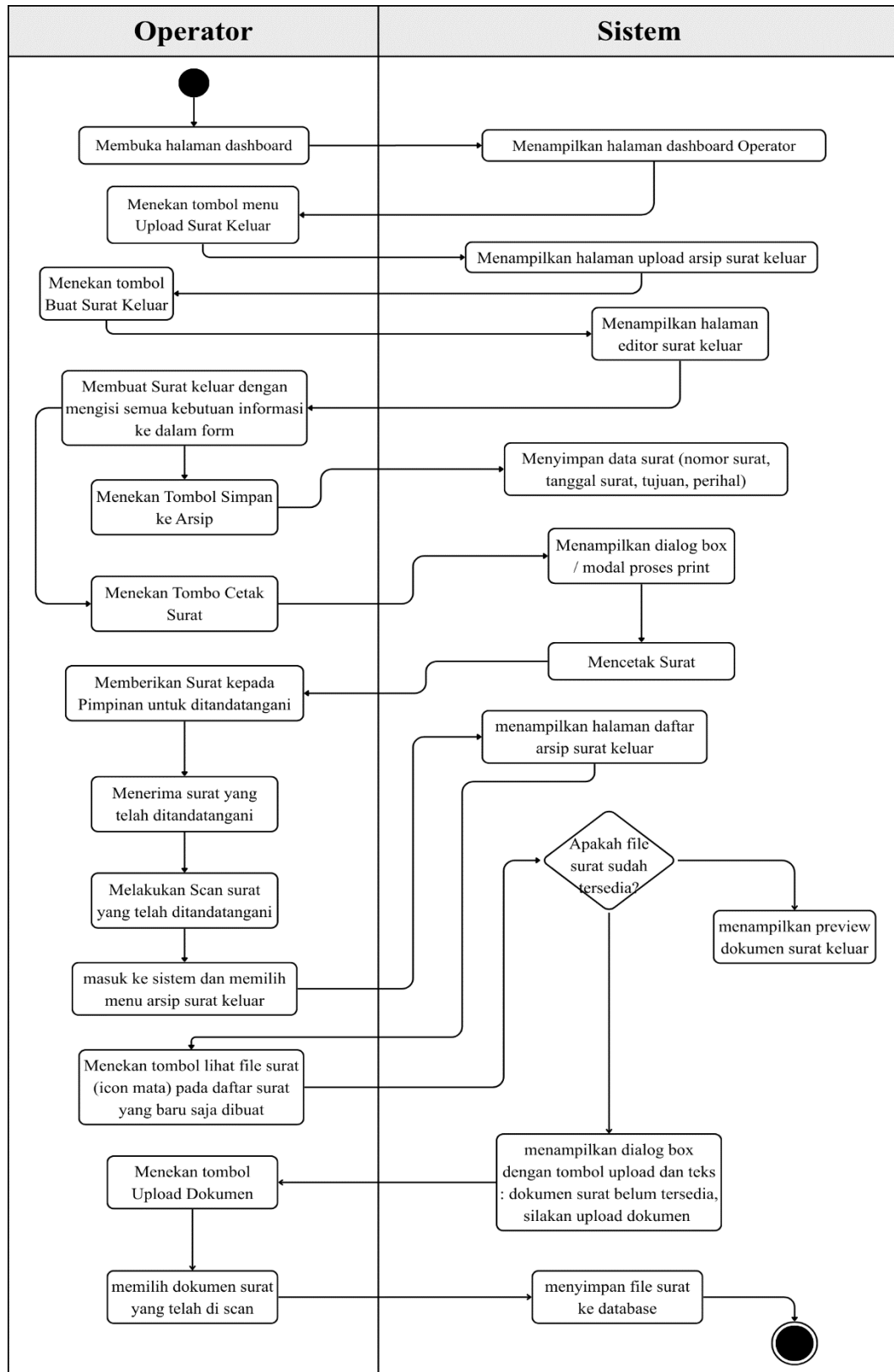
Activity diagram ini menggambarkan proses pengunggahan arsip surat keluar oleh operator ke dalam sistem arsip digital. Proses dimulai ketika operator membuka halaman *dashboard*, kemudian sistem menampilkan halaman *dashboard* operator. Selanjutnya operator menekan menu *upload* surat keluar, sehingga sistem

menampilkan halaman menu *upload* arsip surat keluar. Pada halaman tersebut operator menekan tombol *upload File* untuk memilih dokumen surat keluar yang akan diunggah ke sistem. Sistem kemudian menampilkan jendela *File manager* yang memungkinkan operator memilih *File* dari perangkat yang digunakan.

Setelah *File* dipilih, sistem akan menerima *File* dokumen surat keluar yang diunggah oleh operator. Pada tahap ini sistem melakukan dua proses secara bersamaan yaitu menyimpan *File* sementara serta menjalankan proses ekstraksi teks menggunakan teknologi *OCR (Optical Character Recognition)*. Hasil dari proses *OCR* kemudian ditampilkan oleh sistem sehingga operator dapat melihat teks yang berhasil diekstraksi dari dokumen surat. Selain itu sistem juga menjalankan *Rule-Based System* untuk membantu mengisi beberapa bagian metadata secara otomatis berdasarkan teks yang telah diekstraksi. Sistem kemudian menampilkan form metadata surat keluar yang dapat diperiksa oleh operator.

Operator selanjutnya melakukan verifikasi atau pengeditan metadata apabila terdapat data yang kurang sesuai. Setelah data dianggap benar, operator menekan tombol simpan ke arsip. Sistem kemudian menyimpan metadata dan *File* dokumen surat keluar ke dalam *database*. Setelah proses penyimpanan berhasil, sistem menampilkan pesan bahwa surat keluar berhasil diarsipkan, sehingga proses pengarsipan surat keluar selesai.

6. Activity Buat Surat Keluar Operator



Sumber : data diolah oleh penulis (2026)

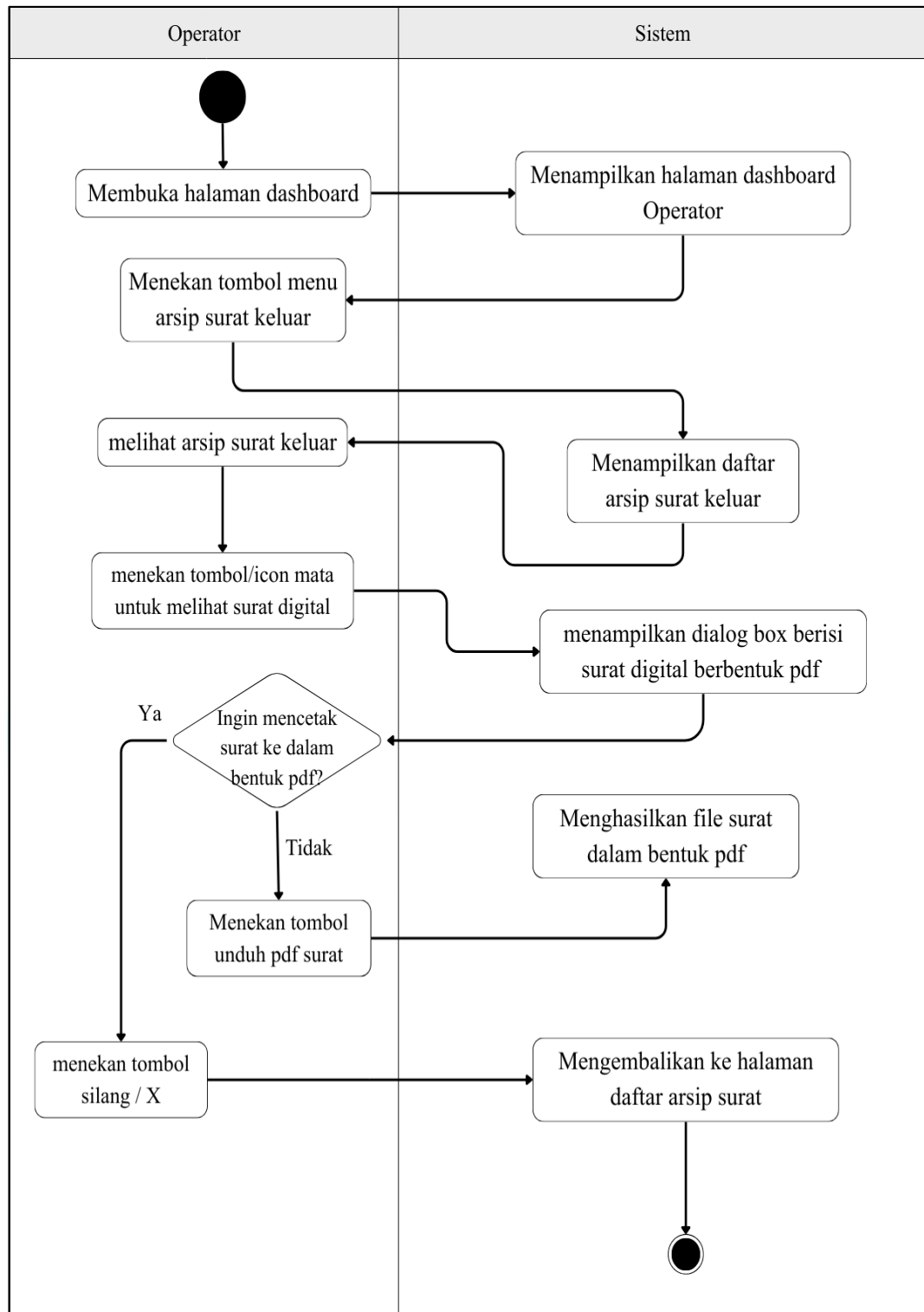
Gambar 4. 7 Activity Buat Surat Keluar

Activity diagram ini menggambarkan alur pengelolaan surat keluar yang dilakukan oleh Operator. Proses dimulai ketika operator membuka halaman dashboard dan sistem merespons dengan menampilkan antarmuka yang sesuai. Tahap pertama adalah pembuatan surat. Operator memilih menu "*Upload* Surat Keluar", lalu menekan tombol "Buat Surat Keluar" sehingga sistem menampilkan halaman editor. Operator mengisi seluruh informasi yang diperlukan ke dalam formulir, mencakup nomor surat, tanggal, tujuan, dan perihal, kemudian sistem menyimpan data tersebut ke dalam basis data.

Tahap kedua adalah pencetakan dan penandatanganan. Sistem menampilkan dialog *box* proses cetak, operator mencetak surat, lalu menyerahkannya kepada pimpinan untuk ditandatangani. Setelah surat diterima kembali, operator melakukan pemindaian (*scan*) untuk memperoleh berkas digital. Tahap ketiga adalah pengarsipan dokumen digital. Operator membuka menu arsip surat keluar dan menekan tombol lihat *file* pada entri surat yang dimaksud. Sistem kemudian melakukan pengecekan kondisional, jika berkas sudah tersedia, sistem menampilkan pratinjau dokumen, namun jika belum, sistem menampilkan notifikasi dan meminta operator untuk mengunggah dokumen. Operator memilih berkas hasil pindaian, dan sistem menyimpannya ke dalam basis data sebagai penanda akhir proses.

Secara keseluruhan, alur ini mencerminkan siklus lengkap pengelolaan surat keluar, mulai dari pembuatan, pencetakan, penandatanganan, hingga pengarsipan dokumen secara elektronik.

7. Activity Arsip Surat Keluar Operator



Sumber : data diolah oleh penulis (2026)

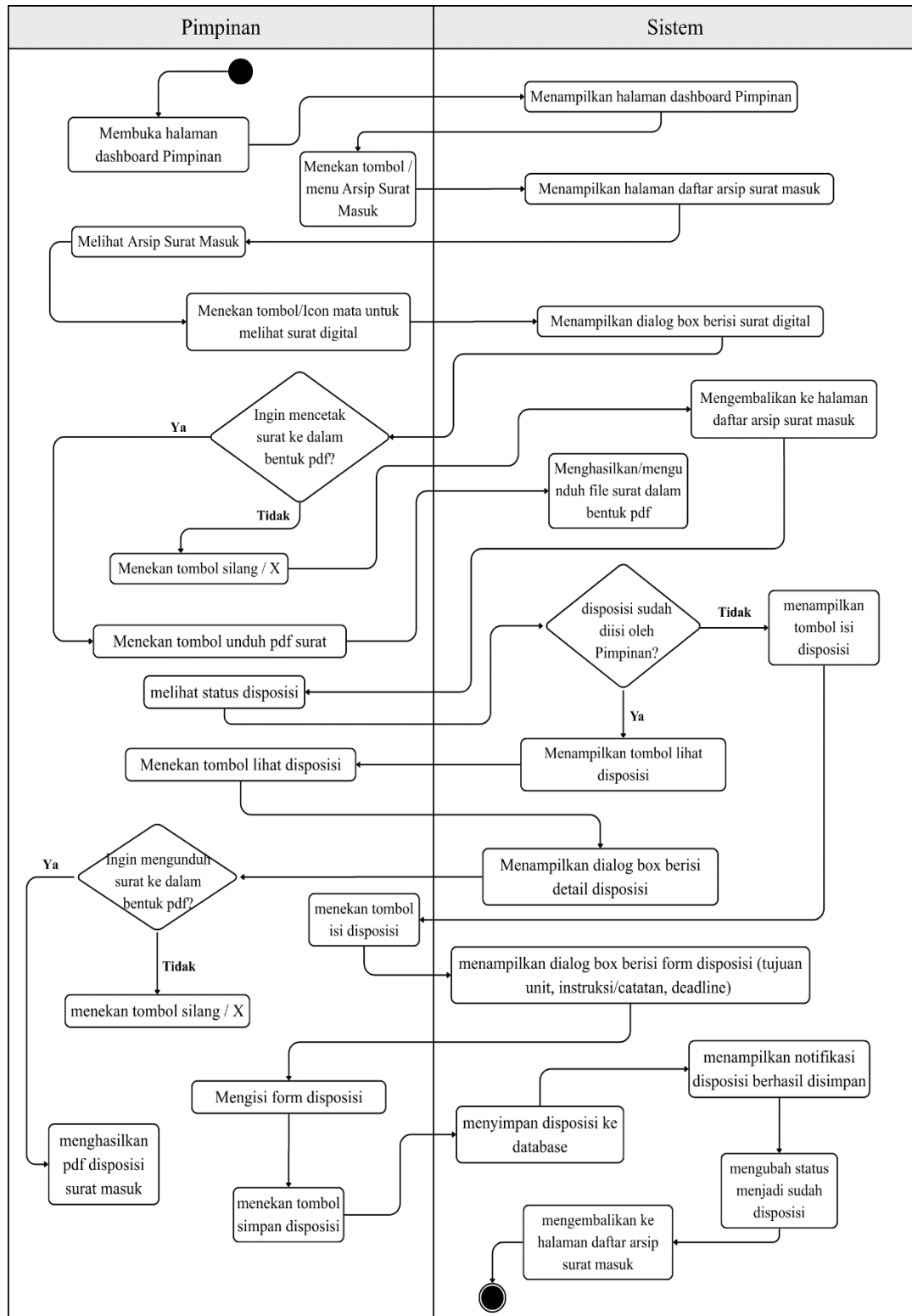
Gambar 4.8 Arsip Surat Keluar Operator Activity

Activity diagram ini menggambarkan proses ketika operator melihat arsip surat keluar yang tersimpan di dalam sistem. Proses dimulai ketika operator membuka halaman *dashboard*, kemudian sistem menampilkan halaman *dashboard* operator. Selanjutnya operator menekan menu arsip surat keluar, sehingga sistem menampilkan daftar arsip surat keluar yang tersedia di dalam sistem. Operator kemudian dapat melihat daftar surat yang tersimpan dan memilih salah satu surat untuk dilihat lebih lanjut. Untuk melihat isi surat, operator menekan ikon mata sehingga sistem menampilkan *dialog box* yang berisi surat digital dalam bentuk *PDF*. Pada tahap ini operator dapat membaca isi surat secara langsung melalui sistem

Apabila operator ingin menyimpan atau mencetak dokumen tersebut, operator dapat memilih opsi mengunduh surat dalam bentuk *PDF*. Sistem kemudian akan menghasilkan *file PDF* yang dapat diunduh oleh operator. Apabila operator tidak ingin mengunduh dokumen tersebut, operator dapat menekan tombol tutup (X) sehingga sistem akan kembali menampilkan halaman daftar arsip surat keluar.



8. Activity Arsip Surat Masuk Pimpinan



Sumber : data diolah oleh penulis (2026)

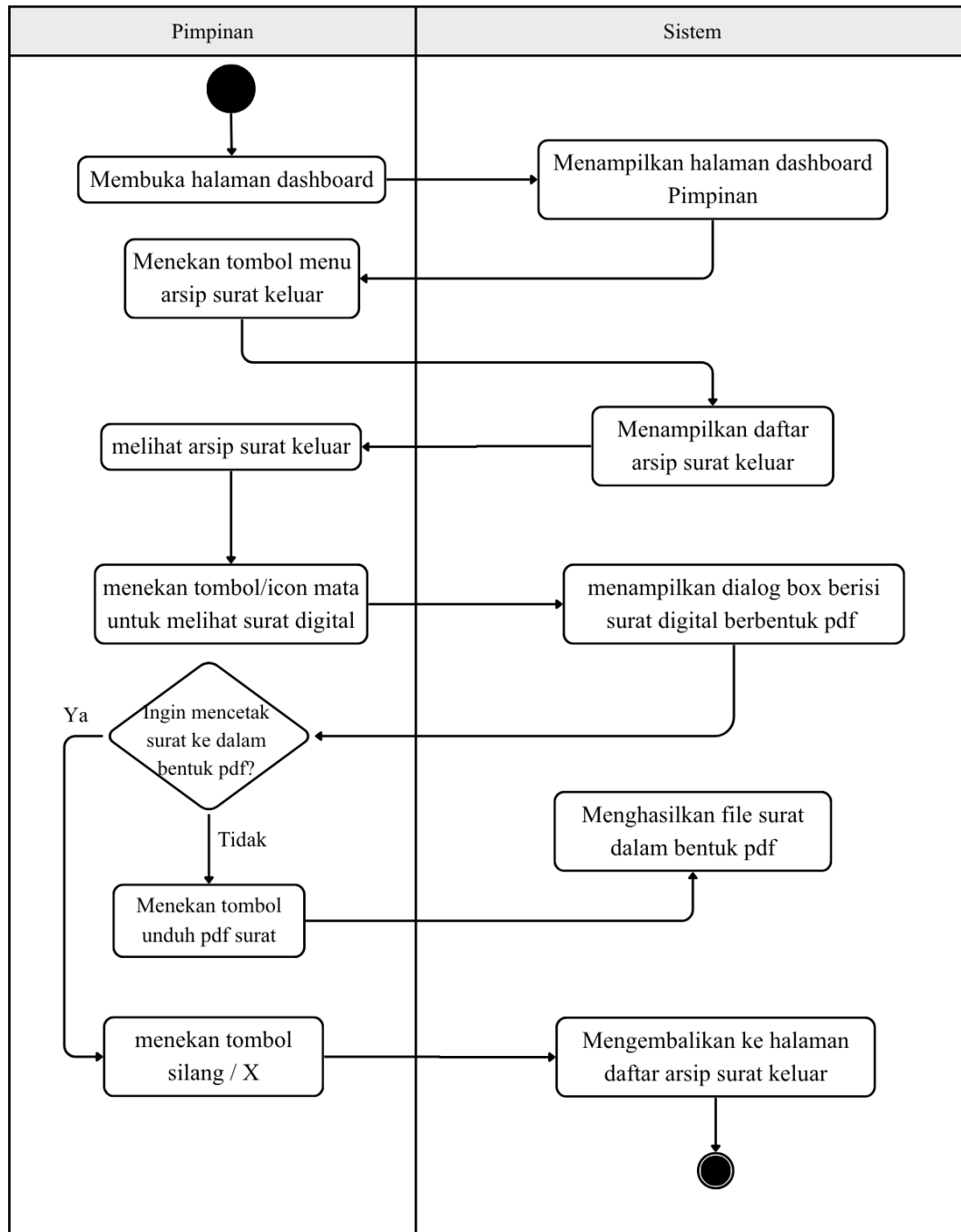
Gambar 4.9 Arsip Surat Masuk Pimpinan Activity

Activity diagram ini menggambarkan proses pimpinan dalam melihat arsip surat masuk dan mengelola disposisi surat melalui sistem. Proses dimulai ketika pimpinan membuka halaman *dashboard*, kemudian sistem menampilkan halaman *dashboard* pimpinan. Selanjutnya pimpinan menekan menu arsip surat masuk, sehingga sistem menampilkan daftar arsip surat masuk yang telah tersimpan di dalam sistem. Pimpinan kemudian dapat melihat daftar surat yang tersedia dan memilih salah satu surat untuk dilihat secara lebih detail.

Untuk melihat isi surat, pimpinan menekan ikon mata sehingga sistem menampilkan *dialog box* yang berisi surat digital. Pimpinan dapat membaca isi surat tersebut secara langsung melalui sistem. Apabila diperlukan, pimpinan juga dapat mengunduh surat dalam bentuk *PDF*. Setelah membaca surat, pimpinan dapat melihat status disposisi surat. Sistem kemudian memeriksa apakah disposisi surat sudah diisi sebelumnya atau belum. Apabila disposisi belum diisi, maka sistem akan menampilkan tombol beri/isi disposisi. Namun apabila disposisi sudah tersedia, sistem akan menampilkan tombol lihat disposisi.

Ketika pimpinan memilih untuk mengisi disposisi, sistem akan menampilkan form disposisi yang berisi beberapa informasi seperti tujuan unit, instruksi atau catatan, serta batas waktu penyelesaian. Pimpinan kemudian mengisi form disposisi tersebut sesuai dengan kebutuhan. Setelah form disposisi diisi, pimpinan menekan tombol simpan disposisi. Sistem kemudian menyimpan data disposisi ke dalam *database* dan menampilkan notifikasi bahwa disposisi berhasil disimpan. Selain itu sistem juga akan mengubah status surat menjadi sudah disposisi. Setelah proses selesai, sistem akan kembali menampilkan halaman daftar arsip surat masuk.

9. Activity Arsip Surat Keluar Pimpinan



Sumber : data diolah oleh penulis (2026)

Gambar 4.10 Arsip Surat Keluar Pimpinan Activity

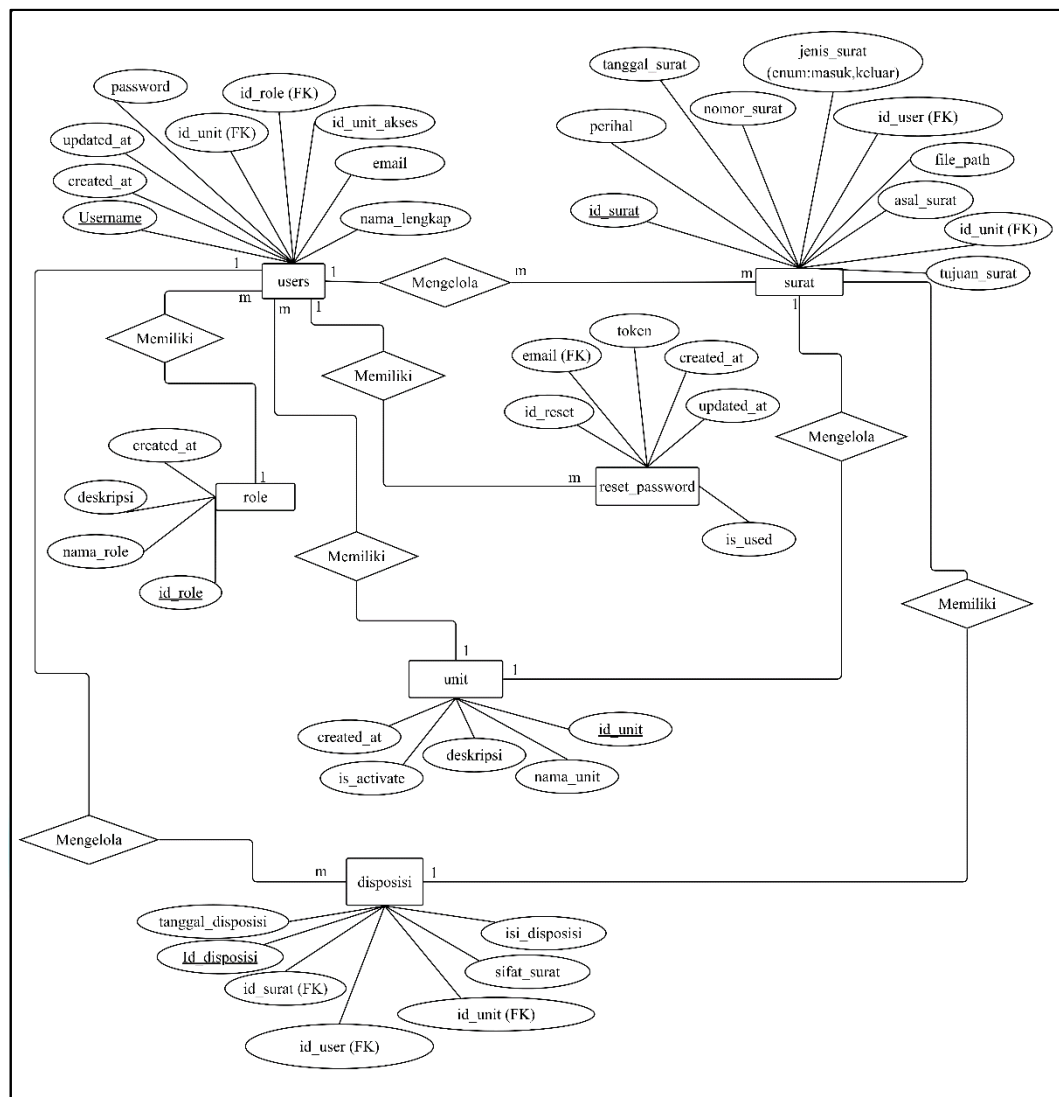
Activity diagram ini menggambarkan proses ketika pimpinan melihat arsip surat keluar yang tersimpan di dalam sistem. Proses dimulai ketika pimpinan membuka halaman *dashboard*, kemudian sistem menampilkan halaman *dashboard* pimpinan. Selanjutnya pimpinan menekan menu arsip surat keluar, sehingga sistem

menampilkan daftar arsip surat keluar yang tersedia di dalam sistem. Pimpinan kemudian dapat melihat daftar surat keluar yang tersimpan.

Untuk melihat isi surat, pimpinan menekan ikon mata sehingga sistem menampilkan *dialog box* berisi surat digital dalam bentuk *PDF*. Melalui tampilan tersebut pimpinan dapat membaca isi surat secara langsung melalui sistem. Apabila pimpinan ingin menyimpan atau mencetak dokumen tersebut, pimpinan dapat memilih opsi mengunduh surat dalam bentuk *PDF*. Sistem kemudian akan menghasilkan *File PDF* yang dapat diunduh. Apabila pimpinan tidak ingin mengunduh *File* tersebut, pimpinan dapat menekan tombol tutup (X) sehingga sistem kembali menampilkan halaman daftar arsip surat keluar.



c. Entity Relational Diagram (ERD)



Sumber : data diolah oleh penulis (2026)

Gambar 4.11 ERD

Entity Relationship Diagram (ERD) pada sistem ini menggambarkan struktur basis data yang digunakan dalam pengembangan aplikasi arsip digital berbasis *OCR* dan *Rule-Based System*. Diagram ini terdiri dari beberapa entitas utama yaitu *User*, *Role*, *Unit*, *Surat*, dan *Disposisi* beserta atribut dan relasinya.

1. Entitas *User* menyimpan data pengguna sistem, seperti *id_user*, *username*, *password*, *email*, *nama_lengkap*, *id_role* (FK), *created_at*, dan *updated_at*. Setiap *user* memiliki satu *role* yang ditentukan melalui relasi dengan entitas *Role*. Relasi ini bersifat *one-to-many*, di mana satu *role* dapat dimiliki oleh

banyak *user*, tetapi satu *user* hanya memiliki satu *role*. Selain itu, entitas *user* juga memiliki relasi dengan entitas unit yang bersifat *one-to-many*, di mana setiap *user* memiliki satu unit dan satu unit dapat berisi beberapa *user*.

2. Entitas *Role* berfungsi untuk menentukan hak akses pengguna dalam sistem, dengan atribut seperti *id_role*, *nama_role*, *deskripsi*, dan *created_at*.
3. Entitas Unit merepresentasikan unit atau bagian dalam organisasi, dengan atribut *id_unit*, *nama_unit*, *deskripsi*, *is_activate*, dan *created_at*. Setiap *user* dapat terhubung dengan satu unit tertentu sesuai struktur organisasi.
4. Entitas Surat merupakan entitas utama dalam sistem arsip digital. Entitas ini menyimpan data surat hasil unggahan dan ekstraksi OCR, dengan atribut seperti *id_surat*, *nomor_surat*, *tanggal_surat*, *perihal*, *jenis_surat* (masuk/keluar), *asal_surat*, *tujuan_surat*, *file_path*, *id_user* (FK), dan *id_unit* (FK). Relasi menunjukkan bahwa satu *user* dapat mengelola banyak surat (*one-to-many*).
5. Entitas Disposisi menyimpan informasi terkait tindak lanjut surat, dengan atribut seperti *id_disposisi*, *tanggal_disposisi*, *isi_disposisi*, *id_surat* (FK), *id_user* (FK), dan *id_unit* (FK), *sifat_surat*. Setiap surat dapat memiliki satu atau lebih data disposisi tergantung alur proses administrasi.
6. Entitas *Reset Password* digunakan untuk menyimpan data permintaan pengaturan ulang kata sandi pengguna dalam sistem. Entitas ini memiliki atribut seperti *id_reset*, *email* (FK), *token*, *created_at*, *updated_at*, *is_used*. Atribut *email* berperan sebagai *foreign key* yang terhubung dengan entitas User, sehingga setiap permintaan reset password hanya dapat dilakukan oleh pengguna yang telah terdaftar dalam sistem. Relasi antara entitas User dan Reset Password bersifat *one-to-many*, di mana satu user dapat memiliki beberapa data reset password. Entitas ini mendukung proses keamanan autentikasi pengguna dalam pengelolaan akses sistem.

4.2.2 Perancangan Database (Fisik)

a. Tabel User

Tabel 4.1 User

Nama Field	Tipe Data	Size	Keterangan
<i>id_user</i>	<i>int</i>	-	<i>Primari key (*)</i>
<i>username</i>	<i>varchar</i>	50	Unique
<i>nama_lengkap</i>	<i>varchar</i>	50	
<i>email</i>	<i>Varchar</i>	50	Unique
<i>password</i>	<i>Varchar</i>	255	
<i>id_role</i>	<i>Int</i>	11	<i>Foreign key</i>
<i>Id_unit</i>	<i>Int</i>	11	<i>Foreign key</i>
<i>created_at</i>	<i>datetime</i>	-	
<i>updated_at</i>	<i>datetime</i>	-	
<i>id_unit_akses</i>	<i>varchar</i>	255	

Sumber : data diolah oleh penulis (2026)

Tabel *user* digunakan untuk menyimpan data pengguna yang dapat mengakses sistem. Data yang tersimpan pada tabel ini meliputi informasi identitas pengguna serta data yang berkaitan dengan proses *Login* ke dalam sistem. kolom *id_user* berfungsi sebagai *primari key* yang menjadi identitas unik bagi setiap pengguna yang terdaftar. Kolom *username* digunakan sebagai nama pengguna yang dipakai saat proses *Login* dan bersifat *unique* agar tidak ada dua pengguna yang memiliki *username* yang sama. Kolom *nama_lengkap* digunakan untuk menyimpan nama lengkap pengguna, sedangkan kolom *email* digunakan untuk menyimpan alamat *email* pengguna yang juga bersifat *unique*. Kolom *password* digunakan untuk menyimpan kata sandi yang digunakan saat proses autentikasi pengguna.

Selanjutnya kolom *id_role* merupakan *foreign key* yang menghubungkan tabel *user* dengan tabel *role* untuk menentukan jenis peran pengguna di dalam sistem. Kolom *id_unit* juga merupakan *foreign key* yang menghubungkan pengguna dengan unit kerja tertentu. Selain itu terdapat kolom *created_at* dan *updated_at* yang digunakan untuk mencatat waktu pembuatan data pengguna serta waktu terakhir data tersebut diperbarui.

b. Tabel *Role*

Tabel 4.2 *Role*

Nama Field	Tipe Data	Size	Keterangan
<i>id_role</i>	int	-	Primari key (*)
<i>nama_role</i>	varchar	25	
<i>deskripsi</i>	varchar	255	
<i>created_at</i>	datetime	-	

Sumber : data diolah oleh penulis (2026)

Tabel *role* digunakan untuk menyimpan data mengenai jenis peran pengguna yang terdapat pada sistem. Setiap pengguna yang terdaftar pada sistem akan memiliki satu *role* yang menentukan hak akses dan fungsi yang dapat dilakukan oleh pengguna tersebut.

Kolom *id_role* merupakan *primari key* yang digunakan sebagai identitas unik untuk setiap *role* yang tersimpan di dalam tabel. Kolom *nama_role* digunakan untuk menyimpan nama peran pengguna, seperti administrator, operator, atau pimpinan. Selanjutnya kolom *deskripsi* digunakan untuk memberikan penjelasan singkat mengenai fungsi atau hak akses dari *role* tersebut.

Selain itu terdapat kolom *created_at* yang digunakan untuk menyimpan informasi waktu kapan data *role* tersebut dibuat di dalam sistem. Dengan adanya tabel *role* ini, sistem dapat mengatur pembagian hak akses pengguna sesuai dengan perannya masing-masing.

c. Tabel Unit

Tabel 4.3 Unit

Nama Field	Tipe Data	Size	Keterangan
id_unit	int	-	Primari key (*)
nama_unit	varchar	80	
deskripsi	varchar	150	
is_activate	tinyint	1	1=aktif, 0=nonaktif
created_at	datetime	-	

Sumber : data diolah oleh penulis (2026)

Tabel unit digunakan untuk menyimpan data unit organisasi yang terdapat dalam institusi. Data unit ini diperlukan untuk mengetahui asal atau tujuan surat serta untuk mengelompokkan pengguna berdasarkan unit kerja masing-masing. Kolom id_unit berfungsi sebagai *primari key* yang menjadi identitas unik dari setiap unit yang tersimpan dalam *database*. Kolom nama_unit digunakan untuk menyimpan nama unit organisasi, seperti fakultas, biro, atau bagian tertentu dalam institusi.

Kolom deskripsi digunakan untuk memberikan penjelasan tambahan mengenai unit tersebut. Selanjutnya terdapat kolom *is_activate* yang digunakan untuk menunjukkan status unit apakah masih aktif atau tidak. Nilai 1 menunjukkan bahwa unit masih aktif, sedangkan nilai 0 menunjukkan bahwa unit sudah tidak aktif. Selain itu terdapat Kolom *created_at* yang digunakan untuk mencatat waktu ketika data unit tersebut dibuat di dalam sistem.

d. Tabel Surat

Tabel 4.4 Surat

Nama Field	Tipe Data	Size	Keterangan
id_surat	int	-	Primari key (*)
nomor_surat	varchar	100	
tanggal_surat	date	-	
perihal	varchar	-	
jenis_surat	enum	-	masuk, keluar
asal_surat	varchar	255	
tujuan_surat	varchar	255	
File_path	varchar	255	
id_user_input	int	-	Foreign key
id_unit	int	-	Foreign key
created_at	timestamp	-	

Sumber : data diolah oleh penulis (2026)

Tabel surat digunakan untuk menyimpan data arsip surat baik surat masuk maupun surat keluar yang diunggah ke dalam sistem. Tabel ini menjadi salah satu tabel utama dalam sistem karena berfungsi sebagai penyimpanan informasi dokumen surat digital. Kolom id_surat merupakan *primari key* yang berfungsi sebagai identitas unik untuk setiap surat yang tersimpan dalam *database*. Kolom nomor_surat digunakan untuk menyimpan nomor surat resmi yang terdapat pada dokumen. Kolom tanggal_surat digunakan untuk menyimpan tanggal surat diterbitkan atau diterima. Selanjutnya kolom perihal digunakan untuk menyimpan informasi mengenai topik atau isi singkat dari surat tersebut.

Kolom jenis_surat menggunakan tipe data *enum* yang berisi dua nilai yaitu masuk dan keluar, yang berfungsi untuk membedakan jenis surat yang tersimpan dalam sistem. Kolom asal_surat digunakan untuk menyimpan informasi mengenai pengirim surat, sedangkan kolom tujuan_surat digunakan untuk menyimpan informasi mengenai pihak yang menjadi tujuan surat. Kolom *File_path* digunakan

untuk menyimpan lokasi penyimpanan *File* dokumen surat di dalam sistem. Selanjutnya kolom *id_user_input* merupakan *foreign key* yang menghubungkan surat dengan pengguna yang melakukan proses penginputan surat tersebut. Selain itu terdapat kolom *id_unit* yang merupakan *foreign key* yang menunjukkan unit yang berkaitan dengan surat tersebut.

e. Tabel Disposisi

Tabel 4.5 Disposisi

Nama Field	Tipe Data	Size	Keterangan
<i>id_disposisi</i>	<i>int</i>	-	<i>Primari key (*)</i>
<i>id_surat</i>	<i>int</i>	-	<i>Foreign key</i>
<i>id_user</i>	<i>int</i>	-	<i>Foreign key</i>
<i>id_unit</i>	<i>int</i>	-	<i>Foreign key</i>
<i>isi_disposisi</i>	<i>text</i>	-	
<i>sifat_surat</i>	<i>enum</i>	-	Biasa, Segera, Rahasia
<i>tanggal_disposisi</i>	<i>Datetime</i>	-	

Sumber : data diolah oleh penulis (2026)

Tabel disposisi digunakan untuk menyimpan data disposisi surat yang dibuat oleh pimpinan terhadap surat masuk yang diterima oleh institusi. Disposisi ini berisi instruksi atau arahan mengenai tindak lanjut dari surat tersebut. Kolom *id_disposisi* merupakan *primari key* yang digunakan sebagai identitas unik untuk setiap data disposisi yang tersimpan dalam sistem. Kolom *id_surat* merupakan *foreign key* yang menghubungkan disposisi dengan surat yang didisposisikan. Kolom *id_user* merupakan *foreign key* yang menunjukkan pengguna yang membuat disposisi, dalam hal ini biasanya adalah pimpinan.

Selanjutnya kolom *id_unit* merupakan *foreign key* yang menunjukkan unit tujuan dari disposisi tersebut. Kolom *isi_disposisi* digunakan untuk menyimpan isi instruksi atau catatan yang diberikan dalam disposisi. Kolom *sifat_surat* menggunakan tipe data *enum* yang menunjukkan tingkat kepentingan surat, yaitu biasa, segera, atau rahasia. Selain itu terdapat kolom *tanggal_disposisi* yang digunakan untuk menyimpan waktu ketika disposisi tersebut dibuat oleh pengguna.

f. Tabel Reset Password

Tabel 4. 6 Reset Password

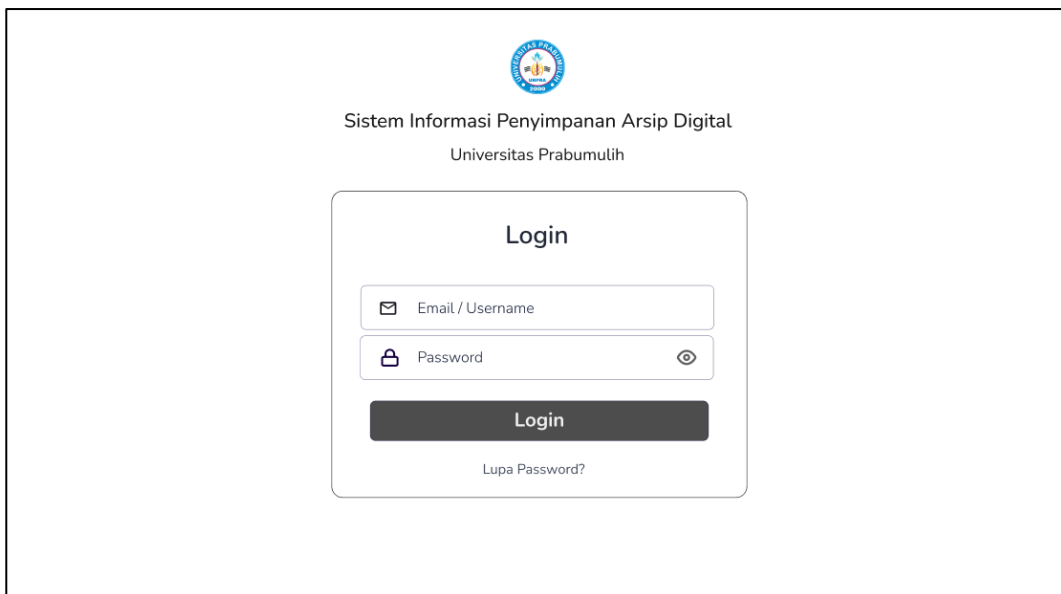
Nama Field	Tipe Data	Size	Keterangan
id_reset	<i>int</i>	-	<i>Primary key*</i>
email	<i>varchar</i>	100	<i>Foreign key</i>
token	<i>varchar</i>	255	
created_at	<i>timestamp</i>	-	
updated_at	<i>timestamp</i>	-	
Is_used	<i>tinyint</i>	1	1 = <i>true</i> 2 = <i>false</i>

Sumber: data diolah oleh penulis (2026)

Tabel *reset_password* digunakan untuk menyimpan data permintaan pengaturan ulang kata sandi pengguna pada sistem. *Field* *id_reset* digunakan sebagai *primary key* yang berfungsi untuk mengidentifikasi setiap data reset password secara unik. *Field* *email* merupakan *foreign key* yang terhubung dengan tabel *users* untuk memastikan bahwa proses reset password hanya dapat dilakukan oleh pengguna yang terdaftar dalam sistem. Selanjutnya, *field* *token* digunakan untuk menyimpan kode verifikasi sementara yang digunakan pada proses validasi reset password. Adapun *field* *created_at* dan *updated_at* digunakan untuk mencatat waktu pembuatan serta pembaruan data pada tabel *reset_password*. Dengan adanya tabel ini, proses pengelolaan reset password dapat dilakukan secara lebih terstruktur dan mendukung aspek keamanan sistem. Selain itu kolom *is_used* digunakan untuk membedakan token yang masih aktif dan belum digunakan dengan token yang sudah digunakan dalam proses perubahan *password*.

4.2.3 Perancangan Antar Muka

a. Halaman *Login*



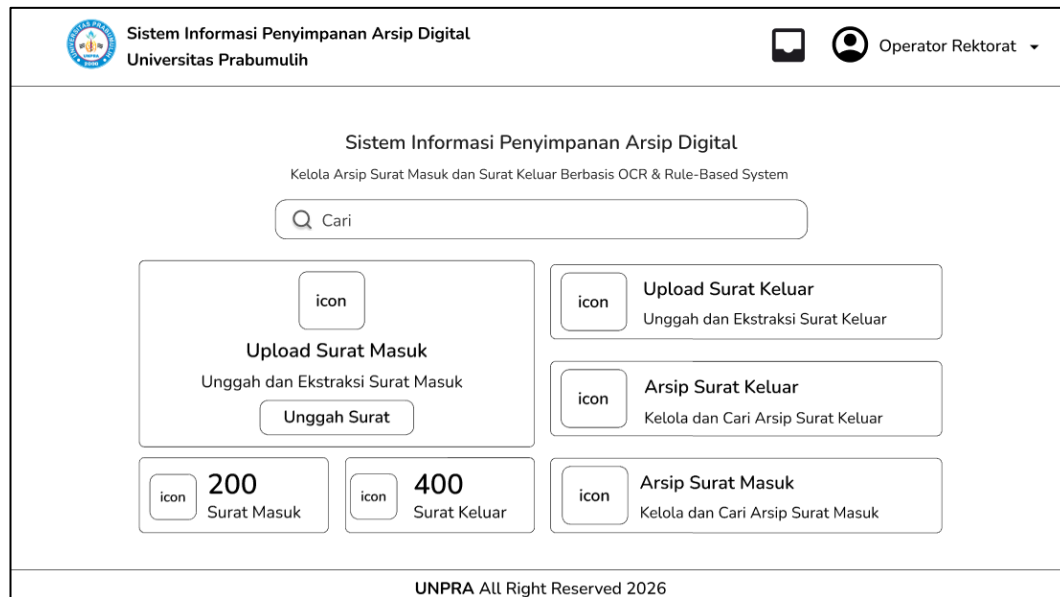
The screenshot shows a web interface for the 'Sistem Informasi Penyimpanan Arsip Digital' at 'Universitas Prabumulih'. At the top center is the university's logo. Below it, the system name and university name are displayed. The main content is a 'Login' form box. Inside this box, there are two input fields: 'Email / Username' with an envelope icon and 'Password' with a lock icon and a toggle eye icon. Below these fields is a dark 'Login' button. At the bottom of the form box is a link that says 'Lupa Password?'. The entire form is centered on a light gray background.

Sumber : data diolah oleh penulis (2026)

Gambar 4.12 Halaman *Login*

Berdasarkan tampilan antarmuka pada gambar di atas, halaman *Login* merupakan halaman awal yang digunakan pengguna untuk masuk ke dalam Sistem Informasi Penyimpanan Arsip Digital. Pada halaman ini pengguna diminta memasukkan *email* atau *username* serta *password* yang telah terdaftar sebagai proses autentikasi sebelum mengakses fitur sistem. Tampilan halaman *Login* terdiri dari kolom *input email/username*, kolom *password*, dan tombol *Login* yang digunakan untuk memproses data yang dimasukkan, sehingga hanya pengguna yang memiliki hak akses yang dapat masuk ke dalam sistem.

b. Halaman *Dashboard Operator*

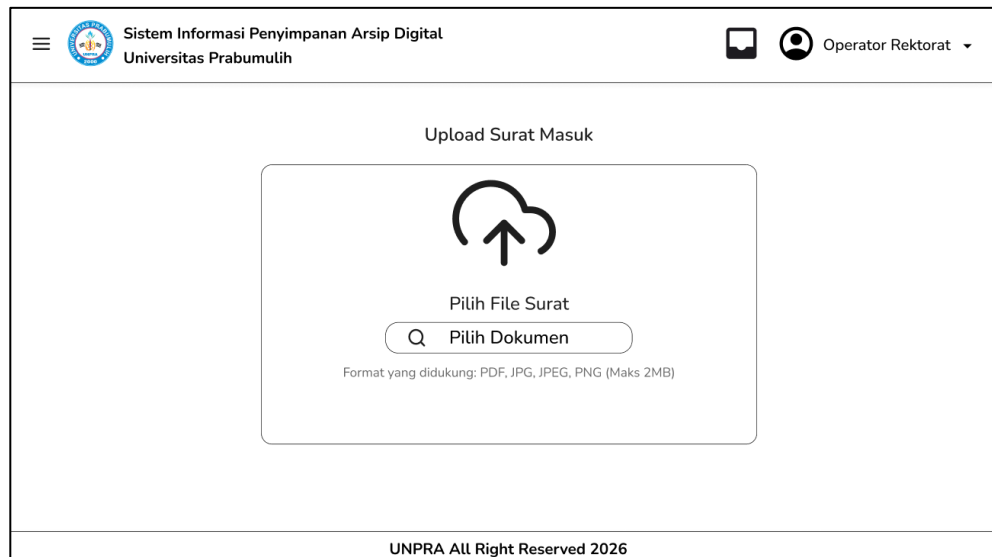


Sumber : data diolah oleh penulis (2026)

Gambar 4.13 Halaman Dashboard Operator

Berdasarkan tampilan antarmuka pada gambar di atas, halaman *dashboard operator* merupakan halaman utama yang ditampilkan setelah pengguna berhasil melakukan *Login* ke dalam sistem. Halaman ini berfungsi sebagai pusat *navigasi* yang memudahkan operator dalam mengakses berbagai fitur yang tersedia. Pada tampilan tersebut terdapat beberapa menu *upload* surat masuk, *upload* surat keluar, arsip surat masuk dan arsip surat keluar, serta pencarian data arsip yang telah tersimpan dalam sistem. Selain itu, *dashboard* juga menampilkan informasi ringkas mengenai data arsip sehingga memudahkan operator dalam mengelola arsip surat masuk dan surat keluar.

c. Halaman *Upload* Surat Masuk Operator



Sumber : data diolah oleh penulis (2026)

Gambar 4.14 Halaman *Upload* Surat Masuk Operator

Dari tampilan antarmuka di atas, halaman *upload* surat masuk merupakan halaman yang digunakan oleh operator untuk mengunggah dokumen surat masuk ke dalam sistem arsip digital. Halaman ini berfungsi sebagai tempat bagi operator untuk memasukkan dokumen surat masuk yang sebelumnya masih berada dalam bentuk *File* digital agar dapat disimpan dan dikelola oleh sistem. Melalui halaman ini operator dapat memulai proses pengarsipan surat dengan cara memilih dokumen yang akan diunggah ke dalam sistem.

Pada halaman tersebut operator dapat mengunggah dokumen surat masuk dengan menekan tombol *Upload File*. Setelah tombol tersebut ditekan, sistem akan menampilkan jendela pemilihan *File* sehingga operator dapat memilih dokumen yang tersimpan pada perangkat yang digunakan. *File* yang dapat diunggah ke dalam sistem adalah *File* dalam bentuk digital seperti *PDF*, *PNG*, *JPEG*, atau *JPG* dengan batas maksimal ukuran dokumen sebesar 2 MB. Pembatasan format dan ukuran *File* ini bertujuan untuk memastikan dokumen yang diunggah dapat diproses dengan baik oleh sistem serta menjaga kinerja penyimpanan data agar tetap efisien.

d. Halaman Hasil Ekstraksi Surat Masuk

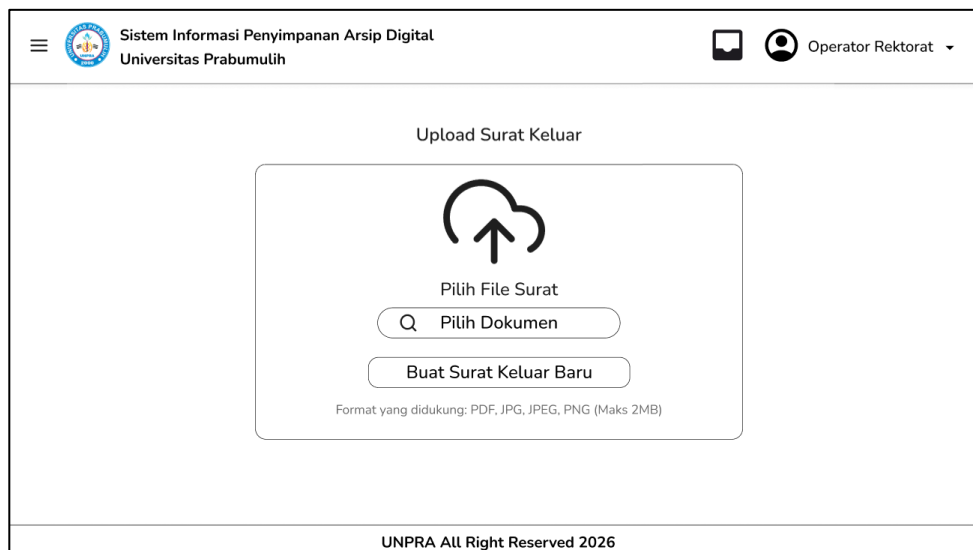
The screenshot shows a web interface for uploading incoming letters. The header includes the university logo, the system name 'Sistem Informasi Penyimpanan Arsip Digital Universitas Prabumulih', and a user profile for 'Operator Rektorat'. The main section is titled 'Upload Surat Masuk'. It features a document preview on the left and a metadata form on the right. The preview shows a PDF document titled 'Document_abc.pdf' with the following text: 'PT Maju Sukses', '27 Februari 2026', 'no : 009/MS/UND/2026', 'perihal : undangan rapat', and 'Kepada Yth Rektor Universitas Prabumulih'. The metadata form, titled 'Keterangan Surat', has four input fields: 'Asal surat', 'Nomor surat', 'Tanggal surat', and 'Perihal'. At the bottom of the form are two buttons: 'Batal' and 'Simpan'. The footer of the page reads 'UNPRA All Right Reserved 2026'.

Sumber : data diolah oleh penulis (2026)

Gambar 4.15 Halaman Hasil Ekstraksi Surat Masuk

Setelah *File* surat masuk berhasil diunggah oleh operator dan proses ekstraksi teks menggunakan teknologi *OCR* selesai dilakukan, sistem akan menampilkan halaman *Upload* Surat Masuk yang memuat hasil ekstraksi dokumen tersebut. Halaman ini terdiri atas dua bagian utama yang ditampilkan secara berdampingan. Bagian kiri menampilkan pratinjau dokumen hasil ekstraksi *OCR* yang memuat sebagian teks surat seperti nama pengirim, tanggal, nomor surat, perihal, dan tujuan surat. Bagian kanan menyediakan formulir metadata surat dengan field asal surat, nomor surat, tanggal surat, dan perihal, yang dapat terisi secara otomatis berdasarkan hasil ekstraksi *OCR* namun tetap dapat diperbaiki secara manual oleh operator. Di bagian bawah formulir terdapat tombol *Batal* untuk membatalkan proses dan tombol *Simpan* untuk menyimpan data yang telah diverifikasi ke dalam basis data sistem arsip digital. Dengan adanya halaman ini, proses pengarsipan dapat dilakukan lebih efisien karena operator tetap memiliki kendali penuh dalam memverifikasi data sebelum surat disimpan.

e. Halaman *Upload* Surat Keluar Operator



Sumber : data diolah oleh penulis (2026)

Gambar 4.16 Halaman *Upload* Surat Keluar Operator

Berdasarkan tampilan antarmuka pada gambar di atas, halaman *Upload* Surat Keluar digunakan oleh pengguna dengan *role* operator untuk melakukan pengunggahan dokumen surat keluar ke dalam sistem arsip digital. Pada halaman ini terdapat area unggah dokumen yang dilengkapi ikon *upload* serta tombol Pilih Dokumen untuk memilih *File* yang akan diunggah ke sistem. Selain itu, sistem juga menyediakan tombol Buat Surat Keluar Baru yang digunakan untuk membuat data surat keluar secara langsung melalui sistem.

Tampilan halaman dirancang sederhana dan mudah dipahami agar pengguna dapat melakukan proses unggah dokumen dengan lebih efisien. Pada bagian bawah area *upload* juga ditampilkan informasi mengenai format *File* yang didukung, yaitu PDF, JPG, JPEG, dan PNG, beserta batas maksimum ukuran *File* sebesar 2 MB sebagai panduan bagi pengguna sebelum melakukan proses unggah. Dengan adanya fitur ini, dokumen surat keluar dapat tersimpan secara digital sehingga memudahkan proses pengarsipan, pengelolaan, dan pencarian dokumen pada sistem arsip digital.

f. Halaman Hasil Ekstraksi Surat Keluar

The screenshot displays the 'Upload Surat Keluar' interface. On the left, a document preview for 'Document_abc.pdf' shows the following text: PT Maju Sukses, 27 Februari 2026, no : 009/MS/UND/2026, perihal : undangan rapat, Kepada Yth, Rektor Universitas Prabumulih. On the right, the 'Keterangan Surat' section contains four input fields: Tujuan, Nomor surat, Tanggal surat, and Perihal. Below these fields are two buttons: Batal and Simpan. The top navigation bar includes the system name 'Sistem Informasi Penyimpanan Arsip Digital Universitas Prabumulih' and the user role 'Operator Rektorat'. The footer states 'UNPRA All Right Reserved 2026'.

Sumber : data diolah oleh penulis (2026)

Gambar 4.17 Halaman Hasil Ekstraksi Surat Keluar

Berdasarkan tampilan antarmuka pada gambar di atas, halaman ini menampilkan hasil proses ekstraksi teks dari dokumen surat yang sebelumnya telah diunggah ke dalam sistem. Setelah sistem berhasil membaca isi dokumen menggunakan teknologi OCR, sebagian informasi yang terdapat di dalam surat akan ditampilkan pada panel sebelah kiri dalam bentuk potongan teks hasil ekstraksi. Informasi tersebut dapat berupa nama instansi, tanggal surat, nomor surat, maupun isi perihal yang terdapat pada dokumen surat.

Pada bagian sebelah kanan terdapat form Keterangan Surat yang digunakan untuk melengkapi data arsip sebelum disimpan ke dalam sistem. Pengguna dapat memasukkan atau menyesuaikan informasi seperti tujuan surat, nomor surat, tanggal surat, dan perihal berdasarkan hasil teks yang telah diekstrak oleh sistem. Selain itu, tersedia beberapa tombol aksi seperti Batal, Lihat File, dan Simpan yang berfungsi untuk membatalkan proses, melihat dokumen asli, atau menyimpan data surat ke dalam *database* arsip digital.

g. Halaman Editor Surat Keluar

Sumber : data diolah oleh penulis(2026)

Gambar 4. 18 Halaman Editor Surat Keluar

Antarmuka halaman Editor Surat Keluar dirancang dengan tampilan dua panel yang saling berkaitan. Panel kiri berfungsi sebagai area pengisian data surat, yang memuat tujuh kolom isian meliputi kota, tanggal surat, nomor surat, lampiran, perihal, penerima, dan isi surat. Panel kanan menampilkan pratinjau dokumen surat secara langsung, sehingga pengguna dapat melihat hasil akhir surat dalam format menyerupai dokumen fisik sebelum dicetak atau disimpan.

Tampilan ini dilengkapi fitur pratinjau surat secara langsung, di mana setiap data yang diisikan pada panel kiri akan langsung tercermin pada tampilan surat di panel kanan secara otomatis. Pada bagian atas halaman terdapat bilah navigasi yang memuat tombol kembali serta dua tombol aksi utama, yaitu Simpan ke Arsip dan Cetak Surat, yang memudahkan pengguna mengakses fungsi inti tanpa perlu berpindah halaman. Keseluruhan desain antarmuka ini mengutamakan kesederhanaan tampilan dan kemudahan penggunaan, sehingga sesuai diterapkan pada sistem informasi manajemen surat dinas berbasis *web* di lingkungan instansi pendidikan.

h. Halaman Daftar Arsip Surat Masuk



Sistem Informasi Penyimpanan Arsip Digital

Universitas Prabumulih



 Operator Rektorat

Daftar Arsip Surat Masuk

Total surat tercatat

20 Surat

Cari



No	Tanggal Surat	Nomor Surat	Asal Surat	Perihal	Status	Aksi
1	17 November 2025	021-022/MKR-SI-KA/UNPRA/2025	Panitia Makrab Prodi SI dan KA	Peminjaman Ruang Kelas	Beri Disposisi	
2	17 November 2025	021-022/MKR-SI-KA/UNPRA/2025	Panitia Makrab Prodi SI dan KA	Peminjaman Ruang Kelas	Lihat Disposisi	
3	17 November 2025	021-022/MKR-SI-KA/UNPRA/2025	Panitia Makrab Prodi SI dan KA	Peminjaman Ruang Kelas	Beri Disposisi	
4	17 November 2025	021-022/MKR-SI-KA/UNPRA/2025	Panitia Makrab Prodi SI dan KA	Peminjaman Ruang Kelas	Lihat Disposisi	

UNPRA All Right Reserved 2026

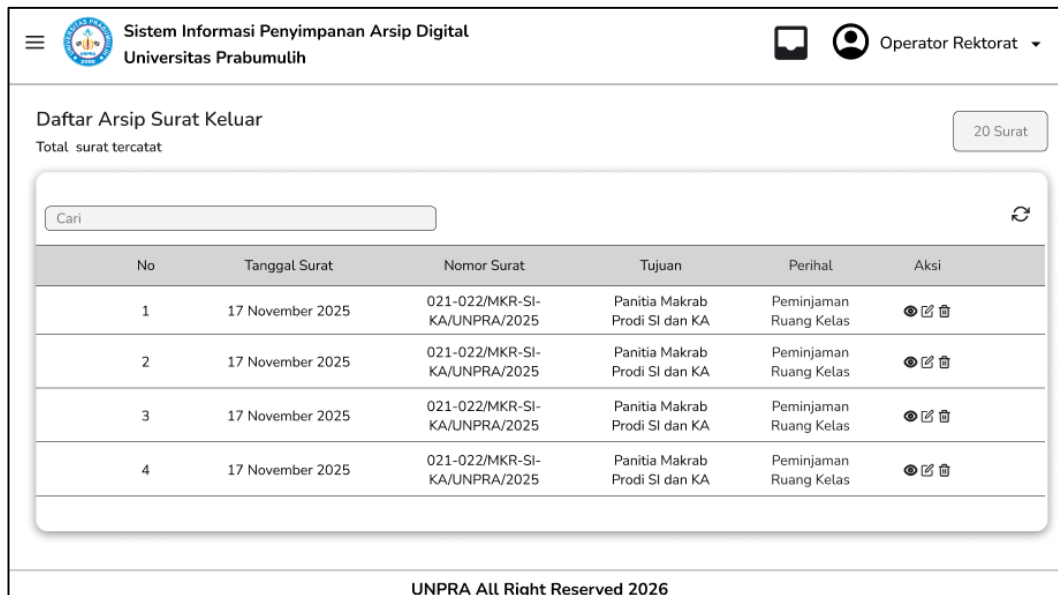
Sumber : data diolah oleh penulis (2026)

Gambar 4.19 Halaman Daftar Arsip Surat Masuk

Berdasarkan tampilan antarmuka pada gambar di atas, halaman Daftar Arsip Surat Masuk berfungsi sebagai halaman untuk menampilkan seluruh data surat masuk yang telah tersimpan di dalam sistem. Informasi surat ditampilkan dalam bentuk tabel agar pengguna dapat melihat data secara lebih terstruktur dan mudah dibaca. Pada tabel tersebut terdapat beberapa kolom informasi penting seperti nomor urut, tanggal surat, nomor surat, asal surat, perihal, serta status disposisi. Selain itu, tersedia juga kolom aksi yang memungkinkan pengguna untuk melakukan tindakan tertentu terhadap data surat yang ditampilkan.

Melalui halaman ini, pengguna dapat dengan mudah memantau status pengelolaan surat masuk yang ada di dalam sistem. Surat yang belum diberikan disposisi akan ditandai dengan status Belum Disposisi, sedangkan surat yang sudah diproses akan menampilkan tombol Lihat Disposisi untuk melihat informasi disposisi yang telah diberikan. Fitur pencarian yang tersedia pada bagian atas tabel juga membantu pengguna dalam menemukan data surat secara lebih cepat berdasarkan kata kunci tertentu.

i. Halaman Daftar Arsip Surat Keluar



No	Tanggal Surat	Nomor Surat	Tujuan	Perihal	Aksi
1	17 November 2025	021-022/MKR-SI-KA/UNPRA/2025	Panitia Makrab Prodi SI dan KA	Peminjaman Ruang Kelas	  
2	17 November 2025	021-022/MKR-SI-KA/UNPRA/2025	Panitia Makrab Prodi SI dan KA	Peminjaman Ruang Kelas	  
3	17 November 2025	021-022/MKR-SI-KA/UNPRA/2025	Panitia Makrab Prodi SI dan KA	Peminjaman Ruang Kelas	  
4	17 November 2025	021-022/MKR-SI-KA/UNPRA/2025	Panitia Makrab Prodi SI dan KA	Peminjaman Ruang Kelas	  

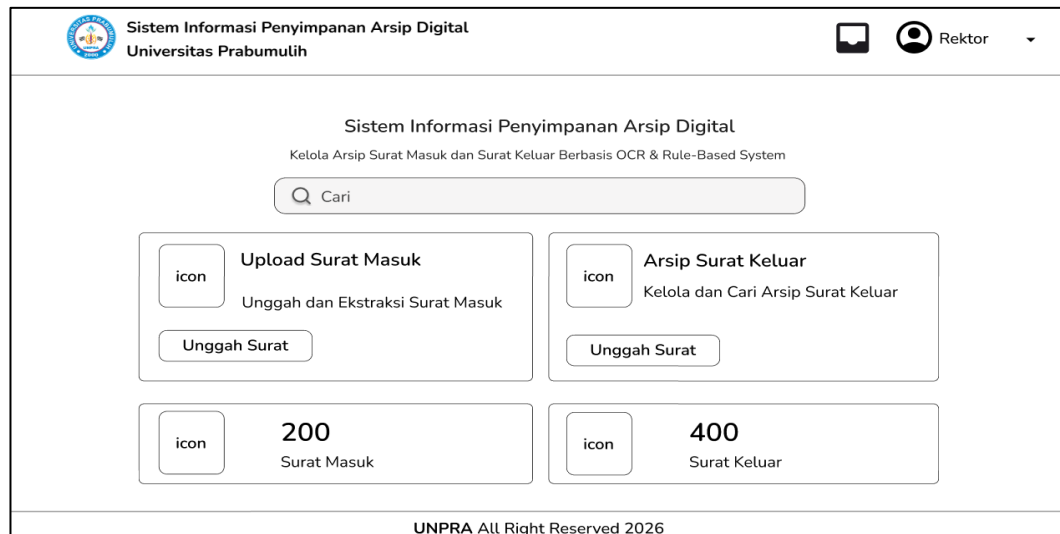
Sumber : data diolah oleh penulis (2026)

Gambar 4.20 Halaman Daftar Arsip Surat Keluar

Berdasarkan tampilan antarmuka pada gambar di atas, halaman Daftar Arsip Surat Keluar digunakan untuk menampilkan data surat keluar yang telah berhasil disimpan ke dalam sistem. Informasi surat disajikan dalam bentuk tabel yang berisi beberapa kolom penting seperti nomor urut, tanggal surat, nomor surat, tujuan surat, serta perihal surat. Penyajian data dalam bentuk tabel bertujuan agar pengguna dapat melihat dan mengelola data arsip secara lebih sistematis dan terstruktur.

Selain menampilkan data surat, halaman ini juga menyediakan kolom aksi yang memungkinkan pengguna untuk melakukan tindakan tertentu terhadap arsip yang tersedia. Pada kolom tersebut terdapat ikon yang dapat digunakan untuk melihat detail dokumen maupun menghapus data arsip apabila diperlukan. Fitur pencarian juga disediakan pada bagian atas tabel untuk membantu pengguna dalam menemukan data surat keluar dengan lebih cepat berdasarkan kata kunci tertentu.

j. Halaman *Dashboard* Pimpinan



Sumber : data diolah oleh penulis (2026)

Gambar 4.21 Halaman *Dashboard* Pimpinan

Halaman *dashboard* pimpinan merupakan halaman utama yang ditampilkan setelah pengguna dengan *role* pimpinan berhasil melakukan *login* ke dalam sistem. Halaman ini berfungsi sebagai tampilan ringkasan informasi yang berkaitan dengan pengelolaan arsip surat di lingkungan universitas. Pada *dashboard* ini terdapat beberapa menu utama yang dapat diakses oleh pimpinan, yaitu menu arsip surat masuk dan arsip surat keluar yang digunakan untuk melihat dan meninjau data surat yang telah tersimpan di dalam sistem.

Selain menyediakan akses ke menu pengelolaan arsip, pada halaman *dashboard* juga ditampilkan diagram atau grafik ringkasan data surat. Grafik ini menampilkan informasi mengenai jumlah surat masuk dan jumlah surat keluar yang tersimpan dalam sistem. Informasi tersebut bertujuan untuk memberikan gambaran umum kepada pimpinan mengenai aktivitas surat menyurat yang terjadi dalam periode tertentu. Dengan adanya tampilan ringkasan ini, pimpinan dapat dengan lebih mudah memantau jumlah dokumen yang masuk maupun keluar tanpa harus membuka data arsip secara satu per satu.

k. Halaman Daftar Arsip Surat Masuk Pimpinan



Sistem Informasi Penyimpanan Arsip Digital

Universitas Prabumulih



 Rektor

Daftar Arsip Surat Masuk

Total surat tercatat

20 Surat

Cari

No	Tanggal Surat	Nomor Surat	Asal Surat	Perihal	Status	Aksi
1	17 November 2025	021-022/MKR-SI-KA/UNPRA/2025	Panitia Makrab Prodi SI dan KA	Peminjaman Ruang Kelas	Beri Disposisi	
2	17 November 2025	021-022/MKR-SI-KA/UNPRA/2025	Panitia Makrab Prodi SI dan KA	Peminjaman Ruang Kelas	Lihat Disposisi	
3	17 November 2025	021-022/MKR-SI-KA/UNPRA/2025	Panitia Makrab Prodi SI dan KA	Peminjaman Ruang Kelas	Beri Disposisi	
4	17 November 2025	021-022/MKR-SI-KA/UNPRA/2025	Panitia Makrab Prodi SI dan KA	Peminjaman Ruang Kelas	Lihat Disposisi	

UNPRA All Right Reserved 2026

Sumber : data diolah oleh penulis (2026)

Gambar 4.22 Halaman Daftar Arsip Surat Masuk Pimpinan

Halaman Daftar Arsip Surat Masuk pada tampilan antarmuka diatas merupakan halaman yang digunakan oleh pengguna dengan *role* pimpinan atau pimpinan untuk melihat seluruh data surat masuk yang telah tersimpan di dalam sistem. Pada halaman ini sistem menampilkan daftar arsip surat dalam bentuk tabel yang berisi beberapa informasi penting seperti nomor surat, tanggal surat, asal surat, perihal, serta status disposisi surat. Penyajian data dalam bentuk tabel bertujuan agar pimpinan dapat dengan mudah melihat dan meninjau informasi surat yang masuk tanpa harus membuka dokumen satu per satu.

Selain menampilkan data arsip surat, halaman ini juga dilengkapi dengan fitur pencarian yang memungkinkan pengguna untuk mencari surat tertentu berdasarkan kata kunci yang dimasukkan. Pada kolom aksi juga tersedia tombol seperti tombol untuk melihat detail surat atau mengelola disposisi surat yang telah dibuat sebelumnya. Dengan adanya halaman ini, pimpinan dapat melakukan pemantauan terhadap surat yang masuk secara lebih terstruktur sehingga proses pengambilan keputusan atau pemberian disposisi dapat dilakukan dengan lebih cepat dan efisien.

1. Tampilan *Dialog Box Form Disposisi*

Disposisi Surat Masuk

Detail Surat Masuk

Nomor Surat	021-022/MKR-SI-KA/UNPRA/2025
Tanggal	17 November 2025
Asal	Panitian Makrab Prodi SI dan KA
Perihal	Peminjaman Ruang Kelas

Form Disposisi

Dari
Rektor

Kepada
Yayasan

Keterangan
catatan atau instruksi disposisi....

Sifat Surat
Segera

Batal
Simpan Disposisi

Sumber : data diolah oleh penulis (2026)

Gambar 4.23 Tampilan Form Disposisi

Halaman Disposisi Surat Masuk merupakan halaman yang digunakan oleh pengguna dengan *role* pimpinan atau pimpinan untuk memberikan instruksi atau arahan terhadap surat masuk yang diterima oleh rektorat. Pada halaman ini sistem menampilkan informasi detail surat pada bagian kiri yang meliputi nomor surat, tanggal surat, asal surat, serta perihal surat. Informasi ini ditampilkan agar pimpinan dapat memahami isi dan konteks surat sebelum memberikan disposisi kepada pihak yang dituju.

Pada bagian kanan halaman terdapat *form* disposisi yang digunakan oleh pimpinan untuk mengisi instruksi atau arahan terhadap surat tersebut. *Form* ini terdiri dari beberapa komponen seperti tujuan disposisi, keterangan atau catatan disposisi, serta sifat surat seperti biasa, segera, atau rahasia. Setelah seluruh informasi disposisi diisi, pimpinan dapat menyimpan disposisi tersebut ke dalam sistem. Data disposisi yang telah disimpan nantinya akan diteruskan kepada unit atau operator yang bersangkutan untuk ditindaklanjuti sesuai dengan instruksi yang diberikan.

m. Tampilan Lihat Disposisi

The screenshot displays two identical side-by-side panels titled "Disposisi Surat Masuk". Each panel is divided into two main sections: "Detail Surat Masuk" on the left and "Instruksi Disposisi" on the right.

Detail Surat Masuk:

Nomor Surat	021-022/MKR-SI-KA/UNPRA/2025
Tanggal	17 November 2025
Asal	Panitian Makrab Prodi SI dan KA
Perihal	Peminjaman Ruang Kelas

Instruksi Disposisi:

Dari	Rektor
Kepada	Yayasan
Keterangan	Untuk dapat diketahui
Kepada	Operator Rektorat
Keterangan	Dibuatkan Balasan
Sifat Surat	Segera
Tanggal Disposisi	18 November 2025

Below the "Instruksi Disposisi" table, there is a button "+ Tambah Instruksi Disposisi". At the bottom of each panel, there are two buttons: "Tutup" and "Cetak Lembar Disposisi".

Sumber : data diolah oleh penulis (2026)

Gambar 4.24 Tampilan Lihat Disposisi

Halaman Instruksi Disposisi Surat Masuk menampilkan hasil disposisi yang telah diberikan oleh pimpinan terhadap suatu surat masuk. Pada tampilan ini sistem menampilkan dua bagian utama, yaitu informasi detail surat pada bagian kiri serta informasi instruksi disposisi pada bagian kanan. Informasi detail surat berfungsi untuk memberikan gambaran mengenai surat yang sedang diproses, seperti nomor surat, tanggal surat, asal surat, serta perihal surat.

Pada bagian kanan halaman ditampilkan instruksi disposisi yang berisi informasi mengenai pihak yang memberikan disposisi, tujuan disposisi, keterangan atau catatan disposisi, sifat surat, serta tanggal disposisi dilakukan. Halaman ini dapat diakses oleh pimpinan maupun operator untuk melihat instruksi yang telah diberikan. Selain itu, sistem juga menyediakan fitur untuk menambahkan instruksi disposisi baru serta mencetak lembar disposisi apabila diperlukan untuk keperluan administrasi atau dokumentasi.

n. Tampilan Inbox Disposisi

 Sistem Informasi Penyimpanan Arsip Digital Universitas Prabumulih   Operator Rektorat						
Inbox Disposisi Unit : Rektorat 2 Masuk 4 Terkirim						
Disposisi Masuk Disposisi Terkirim Cari 						
No	Tanggal Surat	Nomor Surat	Perihal	Ke Unit	Status	Aksi
1	17 November 2025	021-022/MKR-SI-KA/UNPRA/2025	Peminjaman Ruang Kelas	Prodi Informatika	Belum Dibaca	 
2	17 November 2025	021-022/MKR-SI-KA/UNPRA/2025	Peminjaman Ruang Kelas	Rektorat	Sudah Dibaca	 
3	17 November 2025	021-022/MKR-SI-KA/UNPRA/2025	Peminjaman Ruang Kelas	Rektorat	Belum Dibaca	 
4	17 November 2025	021-022/MKR-SI-KA/UNPRA/2025	Peminjaman Ruang Kelas	Prodi Sistem Informasi	Sudah Dibaca	 
UNPRA All Right Reserved 2026						

Sumber : data diolah oleh penulis

Gambar 4. 25 Tampilan Inbox Disposisi

Halaman Inbox Disposisi merupakan tampilan yang digunakan oleh pimpinan maupun operator untuk mengelola surat disposisi pada sistem arsip digital. Pada halaman ini terdapat dua menu utama, yaitu Disposisi Masuk dan Disposisi Terkirim. Menu Disposisi Masuk menampilkan daftar disposisi yang dikirim oleh unit lain kepada unit pengguna saat ini, sehingga pengguna dapat melihat informasi surat yang diterima beserta statusnya. Sementara itu, menu Disposisi Terkirim digunakan untuk menampilkan daftar disposisi yang telah diteruskan atau dikirim oleh unit pengguna kepada unit lain.

Setiap data disposisi yang ditampilkan memuat informasi seperti tanggal surat, nomor surat, perihal, tujuan unit, serta status disposisi apakah sudah dibaca atau belum dibaca. Selain itu, tersedia fitur pencarian untuk mempermudah pengguna dalam menemukan data disposisi tertentu, serta tombol aksi yang dapat digunakan untuk melihat detail maupun melakukan pengelolaan disposisi. Dengan adanya halaman ini, proses distribusi dan pemantauan disposisi surat antar unit dapat dilakukan dengan lebih terstruktur dan efisien.

o. Halaman Daftar Arsip Surat Keluar Pimpinan



Sistem Informasi Penyimpanan Arsip Digital

Universitas Prabumulih

Rektor

Daftar Arsip Surat Keluar

Total surat tercatat

20 Surat

Cari

No	Tanggal Surat	Nomor Surat	Tujuan	Perihal	Aksi
1	17 November 2025	021-022/MKR-SI-KA/UNPRA/2025	Panitia Makrab Prodi SI dan KA	Peminjaman Ruang Kelas	
2	17 November 2025	021-022/MKR-SI-KA/UNPRA/2025	Panitia Makrab Prodi SI dan KA	Peminjaman Ruang Kelas	
3	17 November 2025	021-022/MKR-SI-KA/UNPRA/2025	Panitia Makrab Prodi SI dan KA	Peminjaman Ruang Kelas	
4	17 November 2025	021-022/MKR-SI-KA/UNPRA/2025	Panitia Makrab Prodi SI dan KA	Peminjaman Ruang Kelas	

UNPRA All Right Reserved 2026

UNPRA All Right Reserved 2026

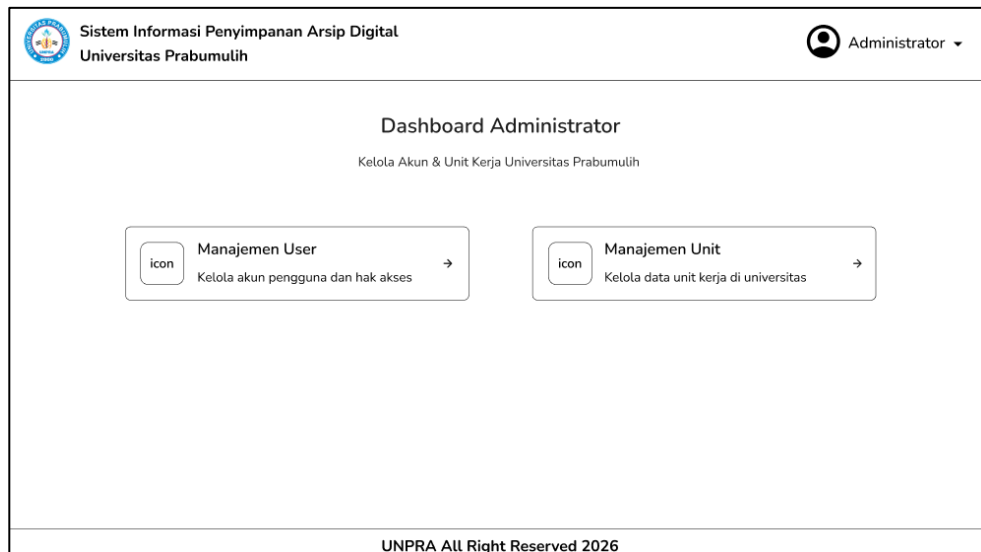
Sumber : data diolah oleh penulis (2026)

Gambar 4.26 Halaman Daftar Arsip Surat Keluar Pimpinan

Halaman Daftar Arsip Surat Keluar merupakan halaman yang digunakan oleh pengguna dengan *role* pimpinan untuk melihat arsip surat keluar yang telah dibuat dan disimpan di dalam sistem. Pada halaman ini sistem menampilkan daftar surat keluar dalam bentuk tabel yang berisi informasi seperti tanggal surat, nomor surat, tujuan surat, serta perihal surat. Informasi tersebut membantu pimpinan untuk mengetahui riwayat surat yang telah dikirim oleh institusi kepada pihak lain.

Melalui halaman ini pimpinan juga dapat melakukan pencarian data surat keluar menggunakan fitur pencarian yang tersedia pada bagian atas tabel. Selain itu terdapat pula beberapa tombol aksi yang memungkinkan pengguna untuk melihat detail surat ataupun melakukan pengelolaan terhadap data surat tersebut. Dengan adanya halaman arsip surat keluar ini, proses pengelolaan dan pemantauan dokumen surat yang telah dikirim menjadi lebih terorganisir dan mudah untuk ditelusuri kembali apabila diperlukan.

p. Halaman *Dashboard Administrator*



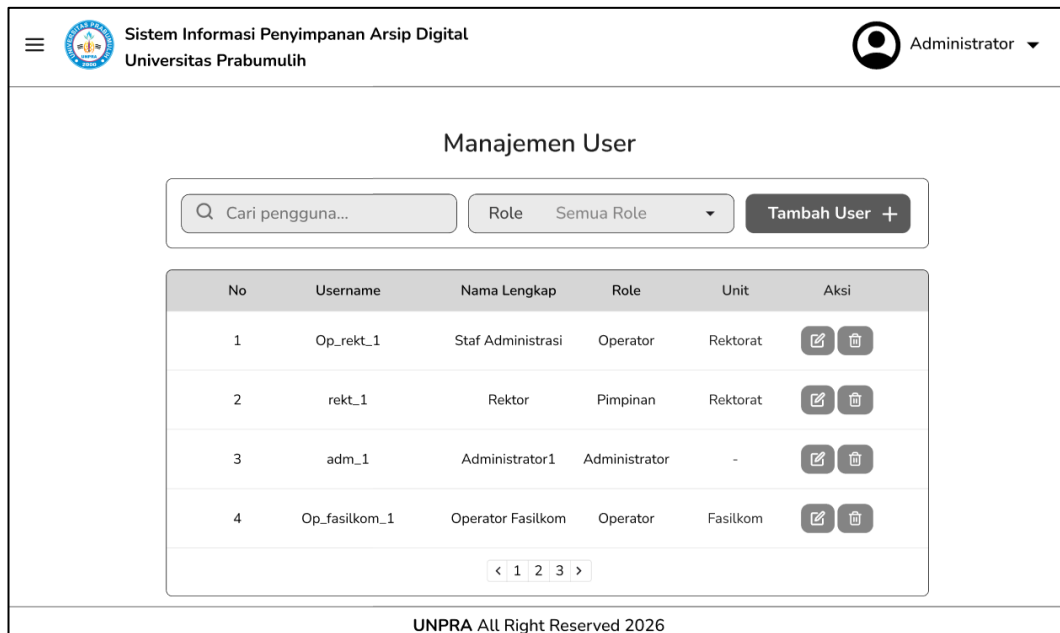
Sumber : data diolah oleh penulis (2026)

Gambar 4.27 Halaman Dashboard Administrator

Halaman Dashboard Administrator merupakan halaman utama yang ditampilkan setelah administrator berhasil masuk ke dalam sistem. Halaman ini berfungsi sebagai pusat navigasi bagi administrator untuk mengakses seluruh fitur pengelolaan sistem, dengan subtitle "Kelola Akun & Unit Kerja Universitas Prabumulih" yang menggambarkan ruang lingkup wewenang administrator dalam sistem.

Halaman ini menampilkan dua menu utama yang disajikan dalam bentuk kartu berdampingan. Kartu pertama adalah Manajemen User yang berfungsi untuk mengelola akun pengguna dan hak akses dalam sistem, sedangkan kartu kedua adalah Manajemen Unit yang digunakan untuk mengelola data unit kerja di lingkungan universitas. Setiap kartu dilengkapi dengan ikon, judul menu, deskripsi singkat, serta tanda panah sebagai penanda bahwa kartu tersebut dapat diklik untuk berpindah ke halaman yang sesuai. Desain halaman ini sengaja dibuat sederhana dan terfokus, sehingga administrator dapat langsung mengakses fitur yang dibutuhkan tanpa melalui navigasi yang rumit.

q. Halaman Manajemen User



The screenshot displays the 'Manajemen User' interface. At the top, the header includes the university logo, the system name 'Sistem Informasi Penyimpanan Arsip Digital Universitas Prabumulih', and the user role 'Administrator'. The main content area features a search bar labeled 'Cari pengguna...', a role filter dropdown set to 'Semua Role', and a 'Tambah User +' button. Below these controls is a table listing four users:

No	Username	Nama Lengkap	Role	Unit	Aksi
1	Op_rekt_1	Staf Administrasi	Operator	Rektorat	[Edit] [Delete]
2	rekt_1	Rektor	Pimpinan	Rektorat	[Edit] [Delete]
3	adm_1	Administrator1	Administrator	-	[Edit] [Delete]
4	Op_fasilkom_1	Operator Fasilkom	Operator	Fasilkom	[Edit] [Delete]

At the bottom of the table, there are pagination controls showing '< 1 2 3 >'. The footer of the page states 'UNPRA All Right Reserved 2026'.

Sumber : data diolah oleh penulis (2026)

Gambar 4.28 Halaman Manajemen User

Halaman *Manajemen User* merupakan halaman yang digunakan oleh pengguna dengan *role* administrator untuk mengelola akun pengguna yang dapat mengakses sistem. Pada halaman ini administrator dapat melihat daftar seluruh pengguna yang terdaftar dalam sistem yang ditampilkan dalam bentuk tabel. Informasi yang ditampilkan meliputi *username*, nama lengkap pengguna, *role* pengguna, serta unit kerja dari masing-masing pengguna.

Selain menampilkan daftar pengguna, halaman ini juga menyediakan beberapa fitur pengelolaan akun seperti menambahkan pengguna baru, mengedit data pengguna, serta menghapus akun pengguna yang sudah tidak digunakan. Administrator juga dapat melakukan pencarian pengguna berdasarkan nama atau *username* serta melakukan penyaringan data berdasarkan *role* pengguna. Dengan adanya halaman ini, pengelolaan hak akses pengguna dalam sistem dapat dilakukan secara lebih terstruktur sehingga keamanan dan pengaturan penggunaan sistem dapat terjaga dengan baik.

BAB V

HASIL DAN PEMBAHASAN

5.1 Spesifikasi Perangkat

Pengembangan aplikasi dalam penelitian ini didukung oleh penggunaan perangkat pada setiap tahapan. Perangkat tersebut digunakan dalam proses perancangan, penyusunan kode, hingga pengujian sistem. Penggunaan perangkat yang sesuai memungkinkan proses pengembangan aplikasi berjalan secara lebih optimal. Perangkat yang digunakan dalam penelitian ini terdiri dari perangkat keras dan perangkat lunak.

5.1.1 Perangkat Keras yang Digunakan

Perangkat keras yang digunakan dalam penelitian ini merupakan perangkat yang secara langsung digunakan dalam proses pengembangan dan pengujian sistem. Perangkat keras tersebut digunakan untuk menjalankan aplikasi, termasuk pada proses pengolahan citra dan ekstraksi teks menggunakan *OCR*. Kinerja perangkat keras yang digunakan berpengaruh terhadap kecepatan pemrosesan data serta kelancaran sistem saat dijalankan. Adapun spesifikasi perangkat keras yang digunakan dalam penelitian ini adalah sebagai berikut.

7. Laptop
8. *Processor* : Intel Core i5-8350
9. RAM : 8 GB
10. *Storage* : 256 GB
11. *Keyboard*
12. *Mouse*

5.1.2 Perangkat Lunak yang Digunakan

Perangkat lunak dalam penelitian ini digunakan sebagai pendukung utama dalam proses pengembangan dan implementasi sistem. Perangkat lunak tersebut berperan dalam menjalankan proses pengkodean, pengolahan data, serta pengujian sistem yang telah dibangun. Penggunaan perangkat lunak yang sesuai memungkinkan sistem dapat berjalan dengan baik, terutama pada fitur yang melibatkan pengolahan citra dan OCR. Adapun perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Sistem Operasi : *Microsoft Windows 11 Pro*
2. Bahasa Pemrograman *Backend* : *Python 3.10*
3. *Framework Backend* : *FastAPI* versi 0.135.2
4. Bahasa Pemrograman *Frontend* : *Dart* versi 3.11.3
5. *Framework Frontend* : *Flutter* versi 3.41
6. *OCR Engine* : *Tesseract OCR* versi 5.5.0.20241111
7. *Database* : *MySQL* versi 8.0.44
8. *Code Editor* : *Visual Studio Code* 1.120.0
9. *Browser* : *Google Chrome* versi 148.0.7778.168

5.2 Implementasi Database

Implementasi *database* pada sistem ini dirancang untuk mendukung pengelolaan data surat dan disposisi secara terstruktur dan terintegrasi. Setiap tabel dalam *database* memiliki peran spesifik yang saling berelasi untuk memastikan alur pengolahan data berjalan dengan baik, mulai dari penyimpanan data pengguna, pengelompokan peran, hingga pengelolaan surat masuk dan disposisi. Perancangan ini bertujuan untuk meningkatkan efisiensi pengolahan data serta meminimalisir redundansi data dalam sistem. Dalam penelitian ini *database* yang digunakan adalah *MySQL*.

5.2.1 Implementasi Tabel *users*

Table: <i>users</i>	
Columns:	
<i>id_user</i>	int AI PK
<i>username</i>	varchar(50)
<i>nama_lengkap</i>	varchar(50)
<i>email</i>	varchar(50)
<i>password</i>	varchar(255)
<i>id_role</i>	int
<i>id_unit</i>	int
<i>created_at</i>	datetime
<i>updated_at</i>	datetime
<i>id_unit_akses</i>	varchar(255)

Sumber : data diolah oleh penulis (2026)

Gambar 5.1 Tabel *Users*

Tabel *users* digunakan untuk menyimpan data pengguna yang memiliki akses ke dalam sistem. Struktur tabel ini terdiri dari beberapa atribut penting seperti *id_user*, *username*, *nama_lengkap*, *email*, *password*, *id_role*, *id_unit*, serta atribut waktu seperti *created_at* dan *updated_at*. Setiap pengguna memiliki identitas unik yang direpresentasikan melalui *id_user* sebagai *primary key*, sehingga memudahkan dalam proses autentikasi dan pengelolaan data.

Selain itu, tabel ini memiliki relasi dengan tabel *role* melalui atribut *id_role* dan tabel *unit* melalui *id_unit*. Relasi ini memungkinkan sistem untuk mengatur hak akses pengguna berdasarkan peran serta unit kerja masing-masing. Dengan adanya struktur ini, sistem dapat membedakan fungsi pengguna seperti administrator, pimpinan, dan operator secara efektif.

5.2.2 Implementasi Tabel *role*

Table: <i>role</i>	
Columns:	
<u>id_role</u>	int AI PK
nama_role	varchar(25)
deskripsi	varchar(255)
created_at	datetime

Sumber: data diolah oleh penulis (2026)

Gambar 5.2 Tabel *Role*

Tabel *role* digunakan untuk mendefinisikan jenis peran atau hak akses yang tersedia dalam sistem. Atribut yang terdapat dalam tabel ini meliputi *id_role*, *nama_role*, *deskripsi*, dan *created_at*. Setiap *role* memiliki identitas unik yang berfungsi sebagai acuan dalam menentukan level akses pengguna terhadap fitur-fitur sistem.

Melalui tabel ini, sistem dapat mengatur pembagian tugas secara terstruktur, seperti administrator yang bertanggung jawab mengelola akun pengguna, pimpinan yang berperan dalam melihat laporan dan disposisi, serta operator yang mengelola data surat. Dengan adanya tabel *role*, sistem menjadi lebih fleksibel dalam pengembangan dan pengaturan hak akses di masa mendatang.

5.2.3 Implementasi Tabel *unit*

Table: <i>unit</i>	
Columns:	
<u>id_unit</u>	int AI PK
nama_unit	varchar(80)
deskripsi	varchar(150)
is_activate	tinyint(1)
created_at	datetime

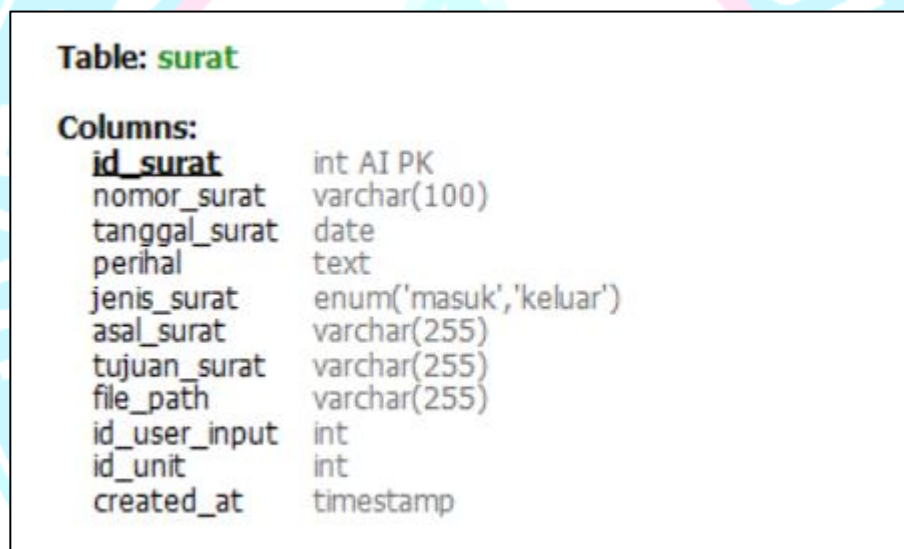
Sumber : data diolah oleh penulis (2026)

Gambar 5.3 Tabel *Unit*

Tabel unit digunakan untuk menyimpan informasi terkait unit kerja atau bagian dalam organisasi. Tabel ini memiliki atribut seperti *id_unit*, *nama_unit*, *deskripsi*, *is_activate*, dan *created_at*. Setiap unit memiliki identitas unik yang memungkinkan sistem mengelompokkan pengguna berdasarkan instansi atau bagian kerja tertentu.

Keberadaan tabel unit sangat penting dalam sistem karena berkaitan langsung dengan distribusi surat dan disposisi. Dengan adanya relasi antara tabel unit dan tabel *users*, sistem dapat mengetahui asal dan tujuan unit dalam proses pengelolaan surat. Hal ini membantu memastikan bahwa alur disposisi berjalan sesuai dengan struktur organisasi.

5.2.4 Implementasi Tabel surat



The diagram shows the structure of the 'surat' table. It lists the columns and their corresponding data types and constraints. The primary key is 'id_surat'.

Table: surat	
Columns:	
id_surat	int AI PK
nomor_surat	varchar(100)
tanggal_surat	date
perihal	text
jenis_surat	enum('masuk','keluar')
asal_surat	varchar(255)
tujuan_surat	varchar(255)
file_path	varchar(255)
id_user_input	int
id_unit	int
created_at	timestamp

Sumber : data diolah oleh penulis (2026)

Gambar 5.4 Tabel Surat

Tabel surat digunakan untuk menyimpan data surat yang dikelola dalam sistem, baik surat masuk maupun surat keluar. Atribut dalam tabel ini meliputi *id_surat*, *nomor_surat*, *tanggal_surat*, *perihal*, *jenis_surat*, *asal_surat*, *tujuan_surat*, *File_path*, *id_user_input*, *id_unit*, dan *created_at*. Setiap surat memiliki identitas unik yang digunakan sebagai *primary key* untuk membedakan data surat.

Tabel ini berperan sebagai inti dari sistem karena menjadi sumber utama dalam proses disposisi. Relasi dengan tabel *users* dan unit memungkinkan sistem

mencatat siapa yang menginput surat serta dari unit mana surat tersebut berasal. Selain itu, penyimpanan *File_path* mempermudah akses terhadap dokumen surat secara digital.

5.2.5 Implementasi Tabel disposisi

Table: disposisi	
Columns:	
<u>id_disposisi</u>	int AI PK
id_surat	int
id_user	int
id_unit	int
isi_disposisi	text
sifat_surat	enum('Biasa', 'Segera', 'Rahasia')
status	enum('baru', 'dibaca', 'ditindaklanjuti')
tanggal_disposisi	datetime

Sumber : data diolah oleh penulis (2026)

Gambar 5.5 Tabel Disposisi

Tabel disposisi digunakan untuk mencatat proses tindak lanjut atau instruksi terhadap suatu surat. Atribut yang terdapat dalam tabel ini antara lain *id_disposisi*, *id_surat*, *id_user*, *id_unit*, *isi_disposisi*, *sifat_surat*, dan *tanggal_disposisi*. Setiap data disposisi memiliki identitas unik yang berfungsi untuk mencatat riwayat tindak lanjut surat.

Melalui tabel ini, sistem dapat merekam alur distribusi dan instruksi yang diberikan oleh pengguna kepada unit terkait. Relasi dengan tabel surat memungkinkan setiap disposisi terhubung langsung dengan surat yang diproses. Dengan demikian, sistem mampu memberikan informasi histori disposisi secara lengkap dan terstruktur.

5.2.6 Implementasi Tabel Reset Password

Table: reset_password	
Columns:	
id_reset	int AI PK
email	varchar(255)
token	varchar(100)
created_at	datetime
updated_at	datetime
is used	tinyint(1)

Sumber : data diolah oleh penulis (2026)

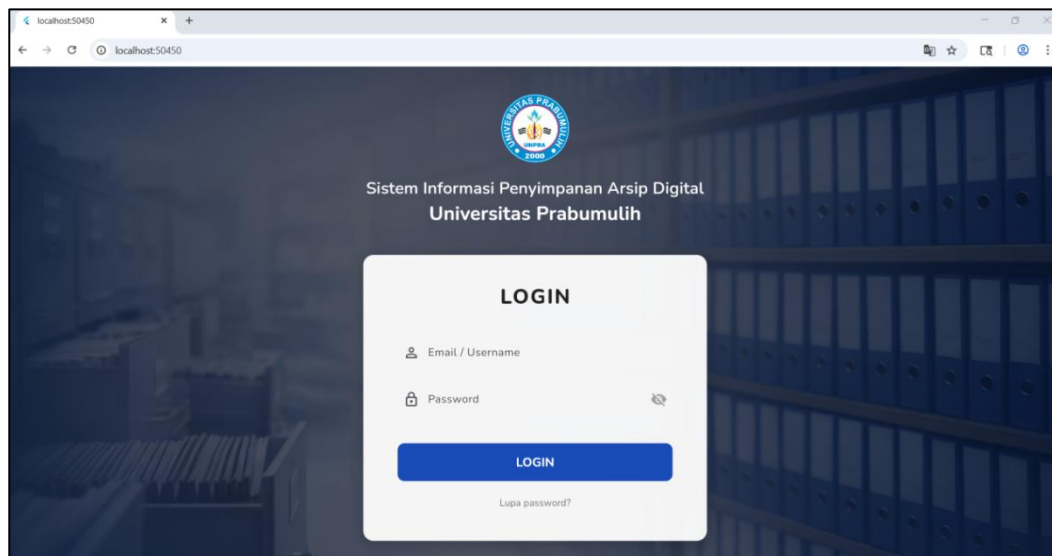
Gambar 5.6 Tabel Reset Password

Tabel `reset_password` digunakan untuk mengelola proses permintaan perubahan kata sandi oleh pengguna. Tabel ini menyimpan atribut seperti `id`, `email`, `token`, dan `created_at`, `updated_at`, dan `is_used` yang berfungsi sebagai identifikasi permintaan reset password.

Keberadaan tabel ini penting untuk menjaga keamanan sistem, terutama dalam proses autentikasi pengguna. Dengan menggunakan token yang bersifat sementara, sistem dapat memastikan bahwa hanya pengguna yang berhak yang dapat melakukan perubahan password. Selain itu, pencatatan waktu pembuatan token membantu dalam membatasi masa berlaku permintaan reset demi meningkatkan keamanan data pengguna.

5.3 Tampilan Antarmuka

5.3.1 Tampilan Halaman *Login*



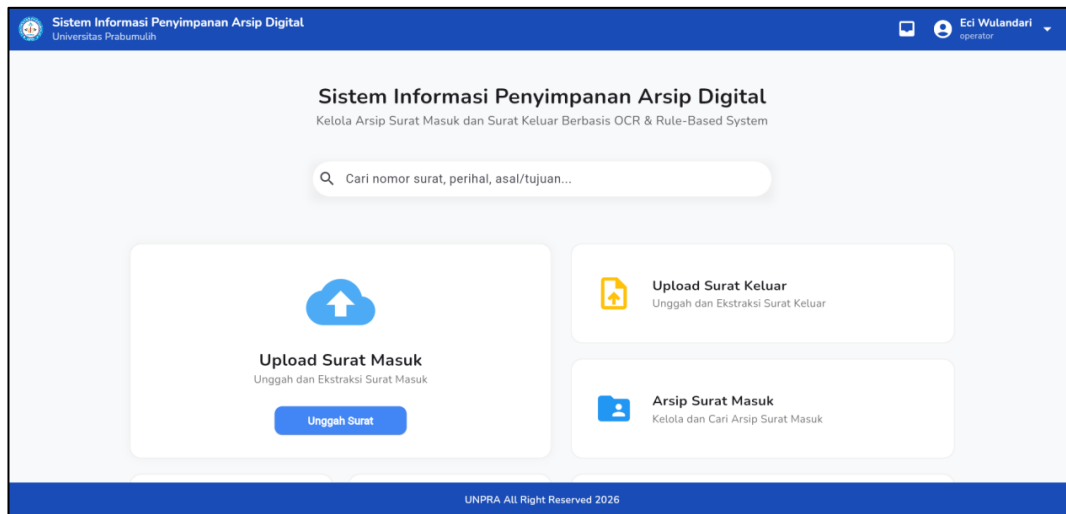
Sumber : data diolah oleh penulis (2026)

Gambar 5.7 Halaman *Login*

Halaman pertama yang ditampilkan kepada pengguna saat mengakses sistem aplikasi arsip digital adalah halaman *Login*. Tampilan antarmuka pada halaman ini dirancang sederhana dan informatif untuk memudahkan pengguna dalam melakukan proses autentikasi. Pada bagian atas halaman terdapat logo Universitas Prabumulih yang berfungsi sebagai identitas institusi, diikuti dengan nama aplikasi yaitu Sistem Informasi Penyimpanan Arsip Digital Universitas Prabumulih sebagai penjelasan fungsi utama sistem yang digunakan.

Di bagian tengah halaman, terdapat sebuah kotak formulir *Login* yang menjadi fokus utama interaksi pengguna. Formulir ini terdiri dari dua field input, yaitu email atau username dan password. Field email atau username digunakan untuk mengidentifikasi pengguna, sedangkan field password berfungsi sebagai kunci keamanan untuk mengakses sistem. Selain itu, tersedia tombol *Login* yang digunakan untuk memproses data yang telah dimasukkan oleh pengguna agar dapat masuk ke dalam sistem sesuai dengan hak akses yang dimiliki.

5.3.2 Tampilan Halaman Dashboard Operator



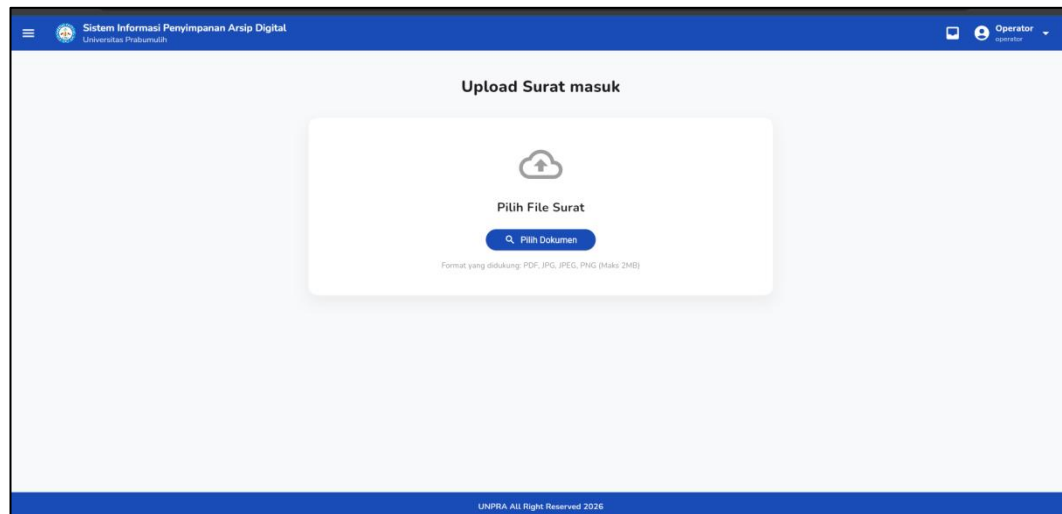
Sumber : data diolah oleh penulis (2026)

Gambar 5.8 Halaman Dashboard

Antarmuka alaman dashboard ini merupakan tampilan utama yang pertama kali dilihat oleh operator setelah berhasil melakukan autentikasi ke dalam sistem. Bagian atas halaman ini memiliki header biru yang memuat identitas institusi serta informasi profil pengguna yang sedang aktif di pojok kanan atas. Fokus utama halaman ini terletak pada judul sistem yang menjelaskan penggunaan teknologi *OCR* dan *Rule-Based System*, serta sebuah *search bar* yang memudahkan pencarian cepat nomor surat atau perihal tertentu di seluruh pangkalan data.

Di bagian bawah bilah pencarian, terdapat dua kartu navigasi utama yang kontras untuk fitur unggah surat masuk dan unggah surat keluar guna memandu alur kerja pengguna. Selain itu, terdapat baris statistik di bagian paling bawah yang menampilkan jumlah total surat masuk dan surat keluar yang telah terdata dalam sistem secara *real-time*. Penggunaan ikon yang intuitif dan tata letak yang bersih bertujuan agar operator dapat beralih antar fitur dengan cepat tanpa memerlukan waktu belajar yang lama.

5.3.3 Tampilan Halaman *Upload* Surat Masuk



Sumber : data diolah oleh penulis (2026)

Gambar 5.9 Halaman *Upload* Surat Masuk

Tampilan fitur *upload* surat masuk dirancang dengan pendekatan minimalis untuk menjaga fokus pengguna pada proses *input* dokumen. Di tengah layar, terdapat kotak pengunggahan yang ditandai dengan ikon awan, disertai keterangan format berkas yang didukung yaitu *PDF*, *JPG*, dan *PNG* dengan batasan ukuran maksimal sebesar 2 MB. Navigasi pada bagian header secara konsisten menunjukkan posisi pengguna saat ini melalui label "*Upload* Surat MASUK" agar tidak terjadi kesalahan input kategori dokumen.

Proses interaksi dimulai saat pengguna menekan tombol "Pilih Dokumen" untuk memilih berkas dari direktori lokal perangkat. Setelah berkas dipilih, sistem akan secara otomatis mempersiapkan dokumen tersebut untuk diproses oleh mesin ekstraksi. Kesederhanaan antarmuka ini secara teknis bertujuan untuk mengurangi beban komputasi pada sisi klien sebelum data dikirimkan ke *backend* untuk dianalisis menggunakan algoritma *OCR*.

5.3.4 Tampilan Halaman Form *Upload* Surat Masuk

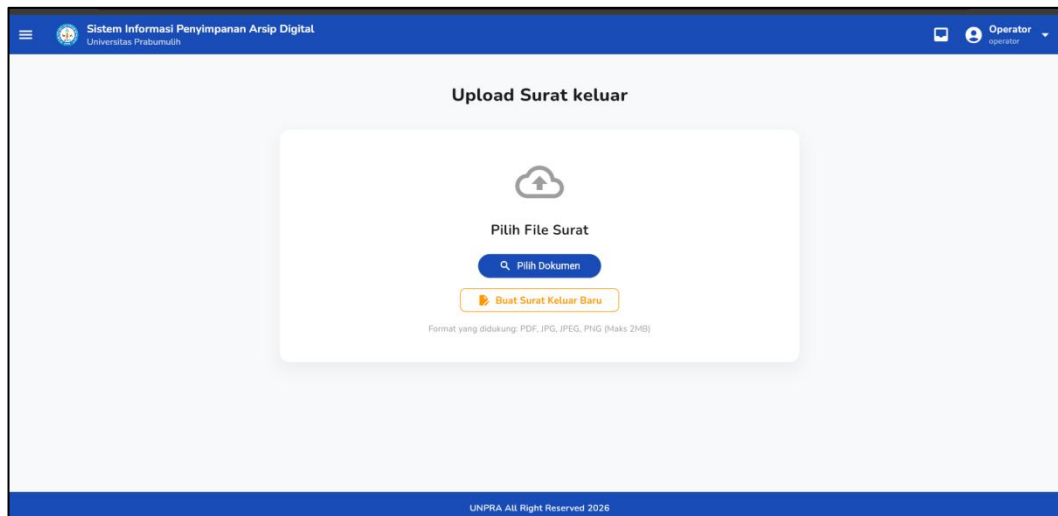
Sumber : data diolah oleh penulis (2026)

Gambar 5.10 Halaman Form *Upload* Surat Masuk

Halaman ini merupakan hasil dari pemrosesan dokumen yang menampilkan perbandingan antara data mentah dan data terstruktur. Di sisi sebelah kiri, terdapat panel "Hasil Pemindaian Mentah (OCR)" yang menampilkan seluruh teks yang berhasil dibaca oleh mesin OCR dari dokumen asli secara apa adanya. Sementara itu, di sisi sebelah kanan terdapat formulir "Metadata Surat" yang berisi kolom-kolom seperti asal surat, nomor surat, tanggal, dan perihal yang telah terisi secara otomatis melalui logika *Rule-Based System*.

Pemisahan antara teks mentah dan formulir metadata bertujuan untuk memudahkan operator dalam melakukan verifikasi atau validasi data jika terdapat karakter yang kurang tepat terbaca oleh sistem. Pengguna memiliki keleluasaan untuk menyunting secara manual kolom metadata tersebut sebelum menekan tombol "Simpan ke Arsip" untuk memasukkannya ke dalam *database*. Adanya tombol "Batal" memberikan opsi bagi pengguna untuk mengulang proses jika dokumen yang diunggah ternyata tidak sesuai.

5.3.5 Tampilan Halaman *Upload* Surat Keluar



Sumber : data diolah oleh penulis (2026)

Gambar 5.11 Halaman *Upload* Surat Keluar

Antarmuka untuk mengunggah surat keluar memiliki kemiripan struktur dengan halaman surat masuk untuk menjaga konsistensi pengalaman pengguna (user experience). Terdapat elemen visual berupa ikon awan di dalam kotak putih besar sebagai instruksi utama bagi pengguna untuk memilih *File* yang akan diekstraksi metadatanya. Perbedaan teknis yang terlihat pada halaman ini adalah batasan ukuran berkas yang lebih kecil, yaitu maksimal 2 MB, yang disesuaikan dengan standar efisiensi penyimpanan dokumen digital internal.

Selain tombol "Pilih Dokumen", halaman ini juga menyediakan tombol alternatif "Buat Surat Keluar Baru" yang berwarna oranye di bagian bawah. Keberadaan tombol ini memungkinkan pengguna untuk langsung beralih ke fitur editor surat jika dokumen fisik belum tersedia dan ingin dibuat secara digital melalui sistem. Header biru tetap dipertahankan untuk menampilkan identitas aplikasi dan informasi pengguna yang sedang masuk guna menjamin keamanan sesi.

5.3.6 Tampilan Halaman Form *Upload* Surat Keluar

Upload Surat keluar

20260226_sdrdekanfakulta...

Hasil Pemindaian Mentah (OCR):

i YA UNPRA Sy a
was a UNIVERSITAS PRABUMULIH
Prabumulih, 26 Februari 2026
Nomor » 067/UNPRA/II/2026
Lampiran 1- Tg
. Perihal - Penetapan Daya Tampung
dan Target Mahasiswa
Kepada Yth.
1. Sdr. Dekan Fakultas
2. Sdr. Ketua Program Studi
i .
- Universitas Prabumulih .
Dengan hormat,
Sehubungan dengan akan
dilaksanakannya Penerimaan Mahasiswa

Keterangan Surat

Tujuan Surat
Sdr. Dekan Fakultas | Sdr. Ketua Program Studi | Universitas Prabun

Nomer Surat
067/UNPRA/II/2026

Tanggal Surat (YYYY-MM-DD)
2026-02-26

Perihal
Penetapan Daya Tampung dan Target Mahasiswa

Batal Simpan ke Arsip

UNPRA All Right Reserved 2026

Sumber : data diolah oleh penulis (2026)

Gambar 5.12 Halaman Form *Upload* Surat Keluar

Setelah dokumen surat keluar berhasil diproses, sistem akan menampilkan pratinjau hasil ekstraksi teks dan pemetaan datanya. Panel kiri menampilkan pratinjau dokumen beserta teks mentah hasil *OCR* yang menunjukkan entitas seperti nama sekolah tujuan dan tanggal surat. Di panel kanan, sistem menyajikan metadata yang telah dipetakan secara otomatis ke dalam field tujuan surat, nomor surat, tanggal surat, dan perihal untuk memastikan akurasi data yang akan diarsipkan.

Antarmuka ini sangat membantu dalam memvalidasi apakah algoritma ekstraksi telah mengenali format surat keluar dengan benar, terutama pada bagian tujuan surat. Pengguna dapat langsung memperbaiki teks di dalam formulir jika terdapat kesalahan pembacaan otomatis tanpa harus mengunggah ulang dokumen. Langkah terakhir dari proses ini adalah menekan tombol "Simpan ke Arsip" yang akan memicu penyimpanan data terstruktur ke dalam tabel surat keluar di *database*.

5.3.7 Tampilan Halaman Editor Surat Keluar

Sumber : data diolah oleh penulis (2026)

Gambar 5.13 Halaman Editor Surat Keluar

Halaman editor surat keluar dirancang dengan sistem panel ganda yang mendukung sinkronisasi data secara langsung atau *real-time preview*. Panel sebelah kiri berfungsi sebagai formulir masukan (*input form*) tempat pengguna mengisi identitas surat seperti kota, tanggal, nomor surat, lampiran, perihal, dan nama penerima. Setiap perubahan teks yang dilakukan pada formulir ini akan secara otomatis memperbarui tampilan visual pada panel sebelah kanan.

Panel sebelah kanan menampilkan pratinjau dokumen dalam format kertas A4 yang sudah terintegrasi dengan kepala surat resmi Universitas Prabumulih. Implementasi tampilan *real-time* ini memudahkan pengguna untuk melihat tata letak dan konten surat sebelum dicetak atau disimpan. Setelah seluruh data dipastikan benar, pengguna dapat mengeklik tombol "Simpan ke Arsip" untuk menyimpan data ke *database* atau tombol "Cetak Surat" untuk mengunduh dokumen dalam format standar resmi.

5.3.8 Tampilan Halaman Arsip Surat Masuk Operator

Sistem Informasi Penyimpanan Arsip Digital
Universitas Prabumulih

Daftar Arsip Surat Masuk
Total 16 surat tercatat

Cari nomor, asal, atau perihal...

No	Tanggal	Nomor Surat	Asal Surat	Perihal	Status	Aksi
1	14 April 2026	B-119/16720/HM.200/2026	BADAN PUSAT STATISTIK KOTA PRABUMULIH	Undangan Forum Konsultasi Publik	Lihat Disposisi	
2	3 Maret 2026	051/P-YPPIII/2026	YAYASAN PENDIDIKAN PRABUMULIH	Undangan Buka Bersama	Belum Disposisi	
3	6 April 2026	002/BGK/UNPRA/IV/2026	BUIJANG GADIS KAMPUS	Undangan Audiensi Kesiapan Pemilihan BGK	Belum Disposisi	
4	21 Januari 2026	008/AK-PRAKERIN/SMK	SMK PGRI 1 PRABUMULIH	Permohonan Pelaksanaan Praktik Kerja Industri Prakerin	Belum Disposisi	
5	1 April 2026	117/FT/UNPRA/IV/2026	Fakultas Teknik	Permohonan SPPD	Belum Disposisi	

UNPRA All Right Reserved 2026

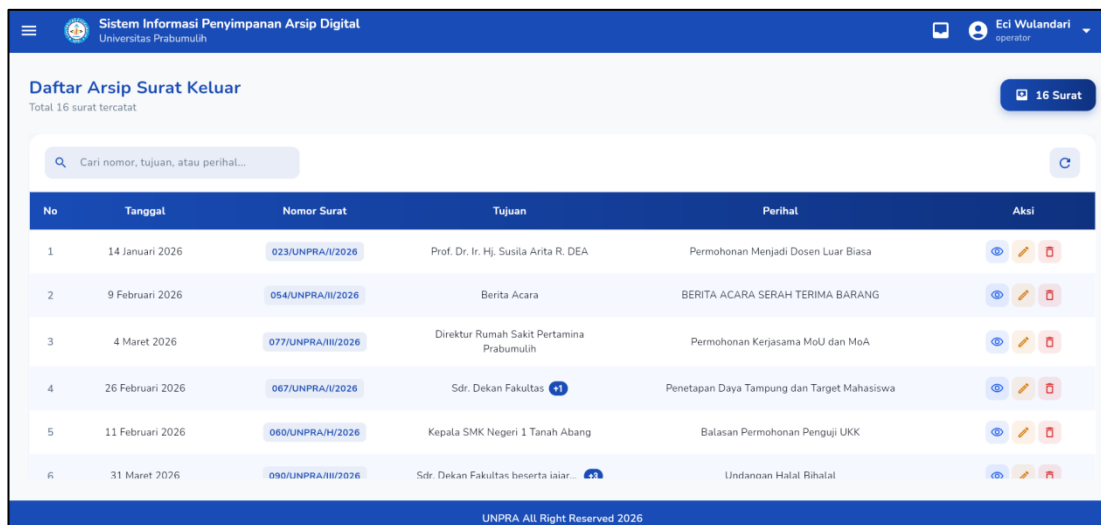
Sumber : data diolah oleh penulis (2026)

Gambar 5.14 Halaman Arsip Surat Masuk Sisi Operator

Antarmuka Daftar Arsip Surat Masuk menyajikan informasi seluruh dokumen yang telah tersimpan dalam bentuk tabel dinamis. Tabel ini memuat metadata penting seperti tanggal, nomor surat, asal surat, dan perihal untuk memudahkan identifikasi dokumen secara cepat. Di atas tabel, tersedia fitur filter pencarian yang memungkinkan operator menyaring data berdasarkan kriteria tertentu guna meningkatkan efisiensi pencarian pada volume data yang besar.

Pada kolom status, terdapat label visual "Belum Disposisi" yang memberikan indikasi mengenai progres tindak lanjut dari surat tersebut dalam alur pengarsipan. Di bagian paling kanan tabel, tersedia kolom aksi yang berisi ikon untuk melihat rincian dokumen, mengubah data, atau menghapus arsip. Total jumlah surat yang tercatat juga ditampilkan pada bagian pojok kanan atas tabel sebagai informasi ringkas mengenai kapasitas pangkalan data saat ini.

5.3.9 Tampilan Halaman Arsip Surat Keluar Operator



No	Tanggal	Nomor Surat	Tujuan	Perihal	Aksi
1	14 Januari 2026	023/UNPRA/I/2026	Prof. Dr. Ir. Hj. Susila Arita R. DEA	Permohonan Menjadi Dosen Luar Biasa	[Icons]
2	9 Februari 2026	054/UNPRA/II/2026	Berita Acara	BERITA ACARA SERAH TERIMA BARANG	[Icons]
3	4 Maret 2026	077/UNPRA/III/2026	Direktur Rumah Sakit Pertamina Prabumulih	Permohonan Kerjasama MoU dan MoA	[Icons]
4	26 Februari 2026	067/UNPRA/I/2026	Sdr. Dekan Fakultas	Penetapan Daya Tampung dan Target Mahasiswa	[Icons]
5	11 Februari 2026	060/UNPRA/H/2026	Kepala SMK Negeri 1 Tanah Abang	Balasan Permohonan Penguji UKK	[Icons]
6	31 Maret 2026	090/UNPRA/III/2026	Sdr. Dekan Fakultas beserta iatar...	Undangan Halal Rihlat	[Icons]

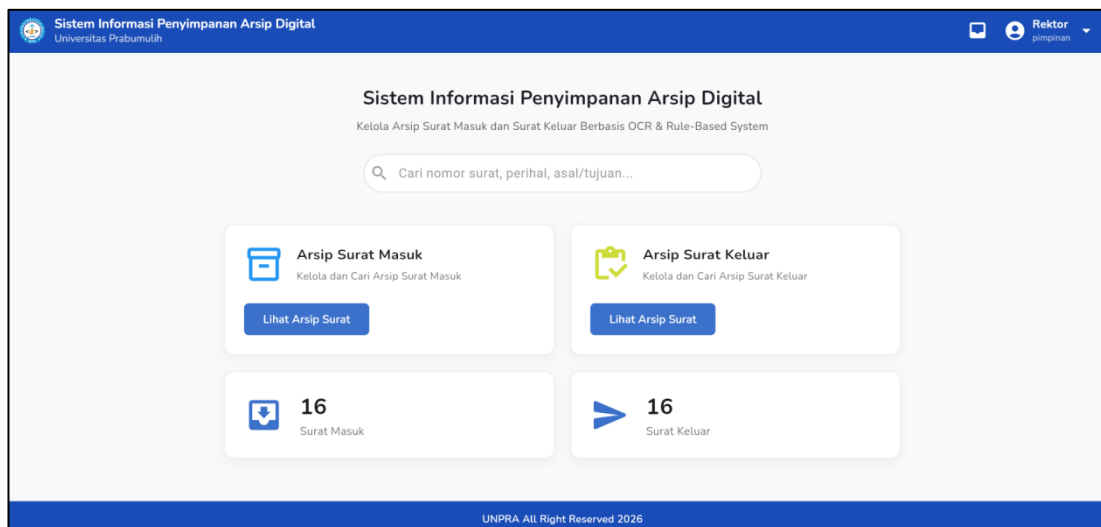
Sumber : data diolah oleh penulis (2026)

Gambar 5.15 Halaman Arsip Surat Keluar Sisi Operator

Antarmuka Daftar Arsip Surat Keluar berfungsi sebagai repositori digital untuk mengelola seluruh dokumen yang tersimpan dalam sistem. Tampilan ini menggunakan format tabel dinamis yang menyajikan metadata seperti tanggal, nomor surat, tujuan, dan perihal secara sistematis. Tersedia pula bilah pencarian responsif pada bagian atas untuk mempermudah penyaringan (*filtering*) data berdasarkan kata kunci. Secara teknis, fitur ini krusial dalam mempercepat proses penemuan kembali dokumen (*document retrieval*) seiring bertambahnya volume arsip.

Pada sisi kanan tabel terdapat kolom aksi yang memudahkan pengguna dalam mengelola data arsip melalui fitur pratinjau, penyuntingan metadata, dan penghapusan arsip. Selain itu, indikator jumlah data pada pojok kanan atas menampilkan informasi total arsip secara *real-time* sehingga memudahkan pengguna dalam memantau volume data secara langsung guna mendukung kebutuhan pelaporan dan audit administratif institusi.

5.3.10 Tampilan Halaman Dashboard Pimpinan



Sumber : data diolah oleh penulis (2026)

Gambar 5.16 halaman Dashboard Pimpinan

Halaman dashboard pimpinan merupakan antarmuka khusus yang dirancang untuk pengguna dengan hak akses sebagai pimpinan unit Rektorat. Secara visual, halaman ini memiliki tata letak yang serupa dengan dashboard operator, namun berfokus pada fungsi pemantauan (monitoring) dan disposisi surat. Di bagian tengah, sistem menyajikan ringkasan statistik berupa jumlah surat masuk dan surat keluar yang memerlukan tindak lanjut, serta bilah pencarian global yang memungkinkan pimpinan mencari arsip secara spesifik berdasarkan nomor atau perihal surat.

Elemen utama pada halaman ini adalah dua kartu navigasi besar untuk mengakses arsip surat masuk dan surat keluar secara langsung melalui tombol "Lihat Arsip Surat". Di bagian bawah, terdapat indikator numerik yang menunjukkan total volume surat yang telah masuk ke dalam sistem. Penggunaan skema warna biru dan putih tetap dipertahankan untuk menjaga konsistensi identitas visual sistem, sementara struktur navigasi yang ringkas bertujuan untuk mempercepat proses pengambilan keputusan pimpinan terhadap dokumen-dokumen yang masuk.

5.3.11 Tampilan Halaman Arsip Surat Masuk Pimpinan

Daftar Arsip Surat Masuk
Total 16 surat tercatat

Cari nomor, asal, atau perihal...

No	Tanggal	Nomor Surat	Asal Surat	Perihal	Status	Aksi
1	14 April 2026	B-119/16720/HM.200/2026	BADAN PUSAT STATISTIK KOTA PRABUMULIH	Undangan Forum Konsultasi Publik	Lihat Disposisi	
2	3 Maret 2026	051/P-YPPIII/2026	YAYASAN PENDIDIKAN PRABUMULIH	Undangan Buka Bersama	Beri Disposisi	
3	6 April 2026	002/BGK/UNPRA/V/2026	BUIJANG GADIS KAMPUS	Undangan Audiensi Kesiapan Pemilihan BGK	Beri Disposisi	
4	21 Januari 2026	008/AK-PRAKERIN/SMK	SMK PGRI 1 PRABUMULIH	Permohonan Pelaksanaan Praktik Kerja Industri Prakerin	Beri Disposisi	
5	1 April 2026	117/FT/UNPRA/IV/2026	Fakultas Teknik	Permohonan SPPD	Beri Disposisi	

UNPRA All Right Reserved 2026

Sumber : data diolah oleh penulis (2026)

Gambar 5.17 Halaman Arsip Surat Masuk Sisi Pimpinan

Halaman daftar arsip untuk pimpinan menyajikan informasi tabel surat masuk yang telah terintegrasi dengan fungsi manajemen alur kerja. Tabel ini memuat data kronologis seperti tanggal, nomor surat, asal surat, dan perihal, yang memudahkan pimpinan dalam memetakan prioritas dokumen. Perbedaan teknis pada antarmuka pimpinan dibandingkan operator terletak pada kolom status dan aksi, di mana terdapat tombol berwarna oranye bertuliskan "Beri Disposisi" sebagai fitur utama untuk meneruskan instruksi kepada unit terkait.

Indikator numerik di pojok kanan atas menunjukkan jumlah total surat yang tersedia di dalam pangkalan data pimpinan secara *real-time*. Fitur pencarian dan penyegaran (*refresh*) data juga disediakan untuk memastikan informasi yang ditampilkan selalu akurat. Dengan adanya tombol aksi "Beri Disposisi" pada setiap baris data, sistem memungkinkan pimpinan untuk merespons setiap surat masuk dengan cepat tanpa harus berpindah ke menu lain, sehingga efisiensi pengarsipan digital dapat tercapai.

5.3.12 Tampilan Form Disposisi

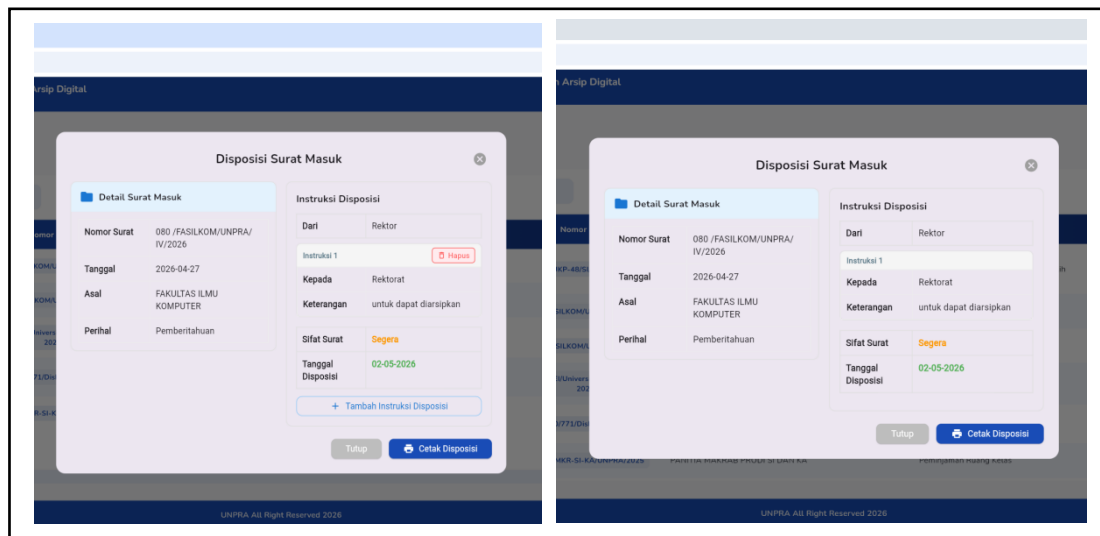
Sumber : data diolah oleh penulis (2026)

Gambar 5.18 Form Disposisi

Antarmuka formulir disposisi muncul dalam bentuk jendela *pop-up* atau modal saat pimpinan menekan tombol "Berikan Disposisi". Halaman ini dibagi menjadi dua panel fungsional, panel kiri menampilkan "Detail Surat Masuk" sebagai referensi data agar pimpinan dapat melihat kembali informasi surat yang akan didisposisikan. Panel kanan merupakan "Form Disposisi" yang berisi kolom input terstruktur untuk memasukkan data tujuan unit, keterangan instruksi, serta pemilihan sifat surat melalui menu *dropdown*.

Secara teknis, pemilihan sifat surat seperti "Biasa", "Segera" atau "Penting" berfungsi untuk memberikan label prioritas pada *database* yang nantinya akan memengaruhi urutan tampilan pada akun penerima disposisi. Di bagian bawah formulir, terdapat tombol "Batal" dan "Simpan Disposisi" sebagai kendali interaksi pengguna. Penempatan elemen formulir yang rapi di dalam kotak dialog ini bertujuan untuk meminimalisir distraksi dan memastikan pimpinan dapat fokus pada pemberian instruksi yang tepat sesuai dengan perihal surat yang dibaca.

5.3.13 Tampilan Lihat Disposisi



Sumber : data diolah oleh penulis (2026)

Gambar 5.19 Tampilan Lihat Disposisi

Antarmuka detail instruksi disposisi ini tidak hanya dapat diakses oleh pimpinan, melainkan juga tersedia bagi level akses operator untuk menjamin transparansi alur administrasi di lingkungan Universitas Prabumulih. Secara visual, desain yang disajikan kepada operator dirancang konsisten dengan tampilan pimpinan guna menjaga keseragaman pengalaman pengguna (*user experience*). Hal ini memungkinkan operator untuk memantau sejauh mana surat masuk telah ditindaklanjuti dan melihat instruksi apa saja yang telah diberikan oleh pimpinan terhadap dokumen tersebut secara *real-time*.

Meskipun memiliki struktur visual yang serupa, terdapat perbedaan fungsional pada sisi operator, yaitu tidak tersedianya tombol untuk menambah instruksi disposisi baru. Pembatasan ini merupakan penerapan hak akses (*access control*), di mana kewenangan pemberian instruksi sepenuhnya berada pada pimpinan, sedangkan operator hanya bertugas memverifikasi dan mencatat data disposisi. Fitur disposisi saat ini berfungsi sebagai sistem pencatatan digital yang mendokumentasikan tujuan penerusan instruksi secara terpusat, sehingga memudahkan pelacakan kronologi disposisi sebagai bagian dari arsip digital institusi.

5.3.14 Tampilan Halaman Inbox Disposisi

No	Tanggal	Nomor Surat	Perihal	Ke Unit	Sifat	Status	Aksi
1	23-05-2026 23:15	800/771/Dishub I/2025	Permohonan Pelaksanaan Sosialisasi	Prodi Informatika	Segera	Belum Dibaca	Detail Hapus
2	16-05-2026 09:38	B-119/16720/HM.200/2026	Undangan Forum Konsultasi Publik	Rektorat	Segera	Sudah Dibaca	Detail Hapus
3	16-05-2026 09:31	080/FASILKOM/UNPRA/IV/2026	Pemberitahuan	Rektorat	Segera	Sudah Dibaca	Detail Hapus
4	15-05-2026 10:57	B-119/16720/HM.200/2026	Undangan Forum Konsultasi Publik	Prodi Informatika	Segera	Sudah Dibaca	Detail Hapus

Halaman 1 dari 1 • Menampilkan 4 dari 4 data

UNPRA All Right Reserved 2026

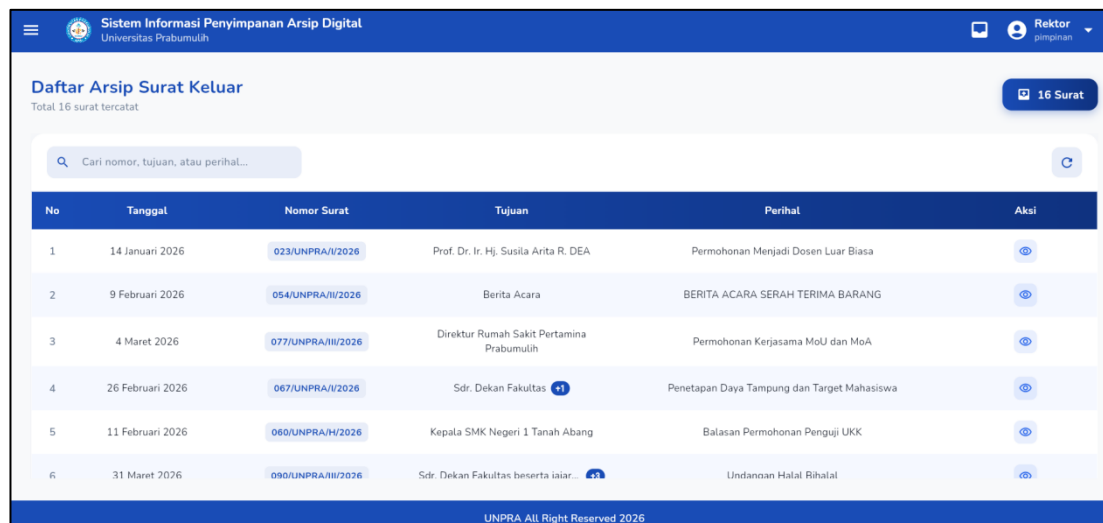
Sumber : data diolah oleh penulis (2026)

Gambar 5.20 Halaman Inbox Disposisi

Halaman Inbox Disposisi merupakan salah satu halaman utama pada aplikasi Sistem Informasi Penyimpanan Arsip Digital yang digunakan oleh pimpinan maupun operator untuk mengelola disposisi surat antar unit. Pada tampilan halaman ini terdapat dua tab utama, yaitu Disposisi Masuk dan Disposisi Terkirim. Tab Disposisi Masuk menampilkan daftar surat disposisi yang diterima dari unit lain ke unit pengguna saat ini, sedangkan tab Disposisi Terkirim menampilkan daftar disposisi yang telah diteruskan atau dikirim oleh unit pengguna ke unit lain.

Pada halaman ini pengguna dapat melihat informasi disposisi secara lengkap, seperti nomor surat, tanggal surat, perihal surat, tujuan unit, serta status disposisi apakah sudah dibaca atau belum dibaca. Selain itu, tersedia fitur pencarian untuk memudahkan pengguna dalam menemukan data disposisi tertentu dengan lebih cepat. Sistem juga menyediakan tombol aksi pada setiap data disposisi untuk melihat detail surat maupun melakukan pengelolaan disposisi lebih lanjut. Dengan tampilan yang terstruktur dan sederhana, halaman inbox disposisi membantu proses distribusi, pemantauan, dan pengelolaan surat disposisi menjadi lebih efektif dan terorganisir.

5.3.15 Tampilan Halaman Arsip Surat Keluar Pimpinan



No	Tanggal	Nomor Surat	Tujuan	Perihal	Aksi
1	14 Januari 2026	023/UNPRA/I/2026	Prof. Dr. Ir. Hj. Susila Arita R. DEA	Permohonan Menjadi Dosen Luar Biasa	
2	9 Februari 2026	054/UNPRA/II/2026	Berita Acara	BERITA ACARA SERAH TERIMA BARANG	
3	4 Maret 2026	077/UNPRA/III/2026	Direktur Rumah Sakit Pertamina Prabumulih	Permohonan Kerjasama MoU dan MoA	
4	26 Februari 2026	067/UNPRA/I/2026	Sdr. Dekan Fakultas	Penetapan Daya Tampung dan Target Mahasiswa	
5	11 Februari 2026	060/UNPRA/H/2026	Kepala SMK Negeri 1 Tanah Abang	Balasan Permohonan Penguji UKK	
6	31 Maret 2026	090/UNPRA/III/2026	Sdr. Dekan Fakultas beserta iatir.	Undangan Halal Rihlat	

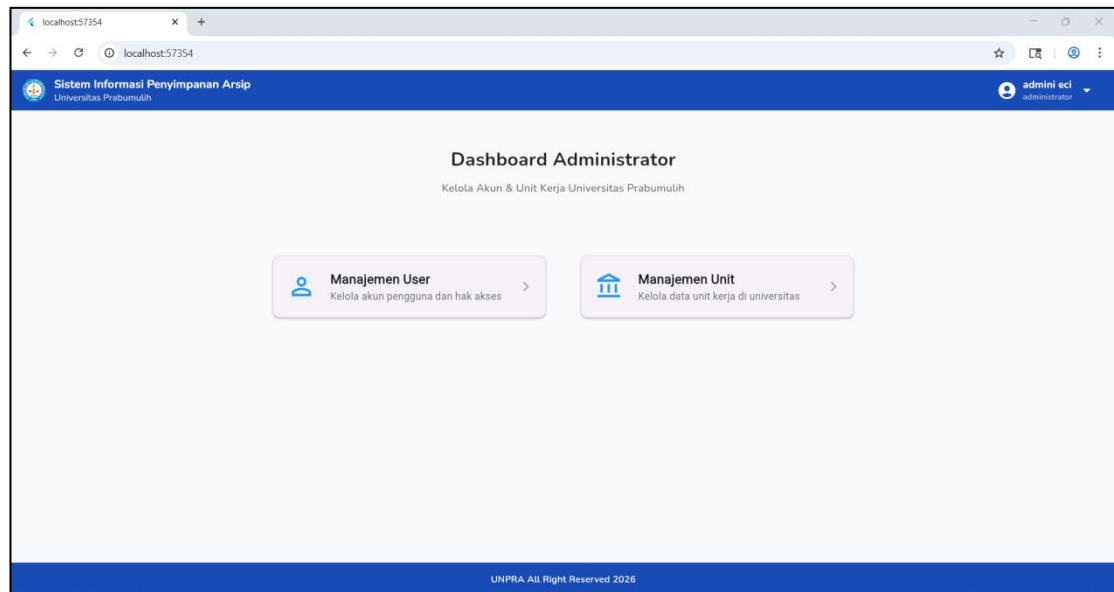
Sumber : data diolah oleh penulis (2026)

Gambar 5.21 Halaman Daftar Arsip Surat Keluar Sisi Pimpinan

Antarmuka Daftar Arsip Surat Keluar untuk level akses pimpinan dirancang dengan fokus utama pada fungsi monitoring dan verifikasi dokumen yang telah diunggah oleh operator. Halaman ini menyajikan informasi melalui komponen tabel dinamis yang terstruktur, mencakup kolom tanggal pengarsipan, nomor surat, instansi atau individu tujuan, serta perihal surat. Pada bagian atas tabel, tersedia bilah pencarian responsif yang berfungsi untuk memfasilitasi pimpinan dalam menelusuri arsip spesifik secara efisien di tengah volume pangkalan data yang besar.

Perbedaan teknis yang paling signifikan antara tampilan pimpinan dan operator pada fitur ini terletak pada pembatasan wewenang (*access control*) yang diterapkan pada kolom aksi di sisi kanan tabel. Pada antarmuka ini, pimpinan hanya diberikan satu opsi interaksi berupa ikon pratinjau untuk melihat detail dan pratinjau visual dokumen surat keluar. Peniadaan fitur penyuntingan (*edit*) maupun penghapusan (*delete*) pada level pimpinan merupakan implementasi langsung dari prinsip integritas data, yang bertujuan untuk memastikan bahwa arsip digital yang telah didistribusikan tidak dapat diubah secara sepihak, sekaligus menjaga keaslian riwayat dokumen institusi.

5.3.16 Tampilan Halaman Dashboard Administrator

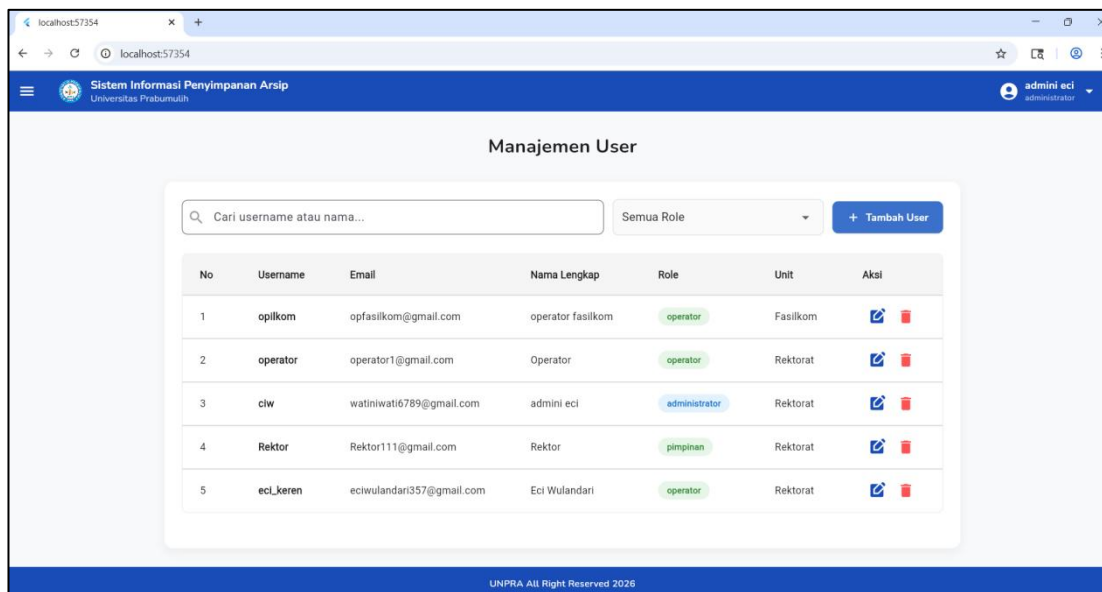


Sumber : data diolah oleh penulis (2026)

Gambar 5.22 Halaman Dashboard Administrator

Halaman Dashboard Administrator merupakan pusat kendali manajemen data master yang dirancang dengan tata letak minimalis dan fungsional. Pada bagian header, sistem tetap mempertahankan konsistensi identitas Universitas Prabumulih di sisi kiri dan informasi akun administrator yang sedang aktif di sisi kanan. Antarmuka ini memfokuskan interaksi pada dua kartu navigasi utama yang menjadi akses cepat menuju fitur Manajemen User dan Manajemen Unit. Penggunaan ikon yang intuitif pada setiap kartu bertujuan untuk mempermudah administrator dalam mengelola entitas pengguna dan struktur organisasi di dalam sistem. Skema warna biru dan putih yang dominan tetap dipertahankan guna menjaga estetika profesional, sementara desain yang bersih bertujuan untuk mengoptimalkan efisiensi administrator dalam menjalankan fungsi kontrol administratif tanpa adanya elemen visual yang tidak relevan.

5.3.17 Tampilan Halaman Manajemen User



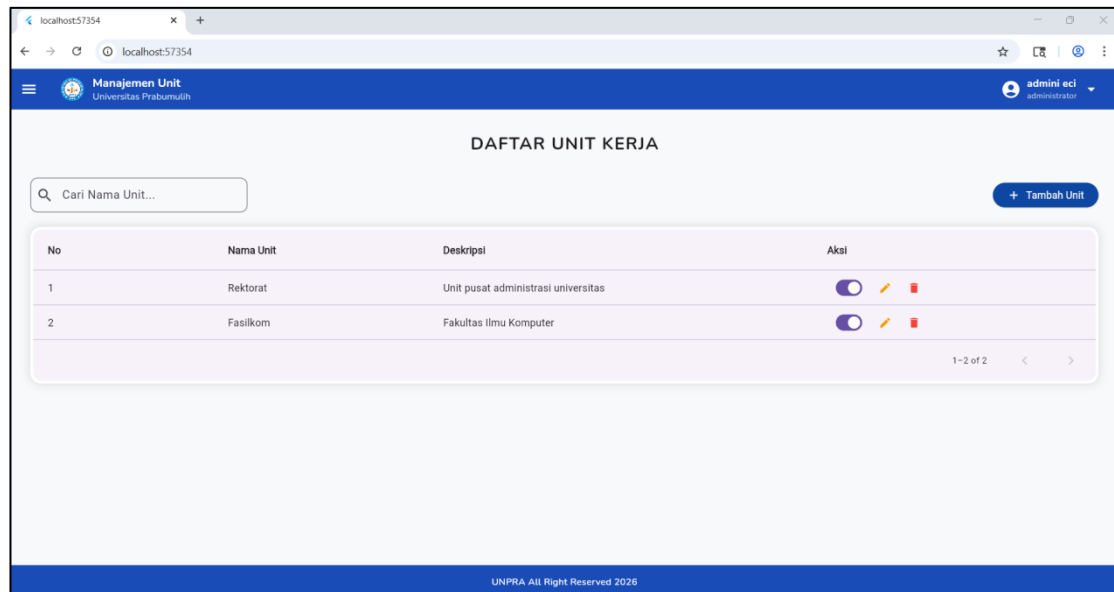
Sumber : data diolah oleh penulis (2026)

Gambar 5.23 Halaman Manajemen User

Halaman manajemen user merupakan antarmuka eksklusif bagi administrator untuk mengelola hak akses dan profil seluruh pengguna di dalam sistem. Data disajikan dalam bentuk tabel yang memuat informasi identitas pengguna secara lengkap, mulai dari *username*, *email*, nama lengkap, peran (*role*), hingga unit kerja masing-masing. Di atas tabel, tersedia fitur pencarian berdasarkan nama dan filter peran untuk mempermudah administrator dalam melakukan audit atau pengelolaan akun dalam jumlah besar.

Tombol "Tambah User" yang berwarna biru di pojok kanan atas berfungsi sebagai pintu masuk untuk mendaftarkan akun baru ke dalam sistem. Pada setiap baris data, tersedia kolom aksi yang berisi ikon untuk menyunting profil atau menghapus akun jika sudah tidak aktif. Secara teknis, halaman ini merupakan dasar dari sistem keamanan *Role-Based Access Control (RBAC)*, yang memastikan bahwa setiap pengguna (seperti operator, pimpinan, atau admin) hanya dapat mengakses fitur yang sesuai dengan tugas dan tanggung jawabnya di Universitas Prabumulih.

5.3.18 Tampilan Halaman Manajemen Unit



Sumber : data diolah oleh penulis (2026)

Gambar 5.24 halaman Manajemen unit

Halaman manajemen unit kerja digunakan oleh administrator untuk mendefinisikan struktur organisasi di lingkungan universitas yang akan terintegrasi dengan sistem pengarsipan. Tabel pada halaman ini menampilkan daftar nama unit kerja beserta deskripsi singkat mengenai fungsinya, seperti Rektorat sebagai pusat administrasi atau FASILKOM sebagai fakultas. Struktur ini sangat krusial karena data unit kerja akan digunakan sebagai referensi otomatis pada fitur ekstraksi OCR dan formulir disposisi.

Fitur teknis yang menonjol pada halaman ini adalah penggunaan tombol switch atau toggle pada kolom aksi yang memungkinkan administrator mengaktifkan atau menonaktifkan unit kerja secara instan tanpa menghapus data permanen. Selain itu, terdapat tombol "Tambah Unit" untuk memperbarui struktur organisasi jika terjadi penambahan fakultas atau lembaga baru. Dengan antarmuka yang bersih dan fungsional ini, pengelolaan pangkalan data entitas organisasi dapat dilakukan secara terpusat dan konsisten.

5.4 Hasil Pengujian Sistem

Pengujian sistem pada penelitian ini dilakukan menggunakan metode *white box testing* untuk menganalisis alur logika program pada sisi backend aplikasi arsip digital berbasis *OCR* dan *Rule-Based System*. Pengujian dilakukan dengan menganalisis *source code* pada beberapa proses utama sistem, seperti proses *Login*, *Preprocessing* dokumen, ekstraksi teks *OCR*, penerapan *Rule-Based System*, hingga proses penyimpanan arsip dan disposisi. Analisis dimulai dari identifikasi *source code* yang digunakan pada setiap fungsi, kemudian dilakukan identifikasi *statement* untuk menentukan bagian-bagian logika program yang mempengaruhi alur eksekusi sistem. *Statement* tersebut selanjutnya dijadikan dasar dalam pembentukan *flowchart* dan *Flowgraph* untuk menggambarkan aliran proses program secara terstruktur.

Setelah *Flowgraph* terbentuk, dilakukan perhitungan *Cyclomatic Complexity* untuk mengetahui jumlah jalur independen minimum yang harus diuji pada setiap fungsi program. Nilai kompleksitas tersebut digunakan sebagai dasar dalam menentukan *independent path* yang merepresentasikan jalur eksekusi berbeda pada sistem. Berdasarkan jalur independen tersebut, kemudian disusun *test case* pengujian untuk memastikan seluruh alur logika program dapat berjalan sesuai dengan kondisi yang diharapkan. Tahap akhir pengujian dilakukan menggunakan *statement coverage* untuk mengevaluasi apakah seluruh *statement* pada *source code* telah berhasil dieksekusi selama proses pengujian berlangsung.

5.4.1 Pengujian Fungsi *Login*

a. Potongan *Source code* Fungsi *Login*

Fungsi *Login* ini merupakan salah satu dari beberapa fungsi utama yang akan diuji dalam penelitian sistem arsip digital ini untuk memastikan keamanan akses pengguna. Komponen di dalamnya mencakup penangkapan input kredensial, *query* data pengguna dengan relasi tabel *role* dan unit, serta verifikasi kata sandi untuk penerbitan token akses. Berikut adalah potongan *source code* fungsi *Login* yang digunakan dalam tahap pengujian:

```
def Login(data: dict, db: Session = Depends(get_db)):
    Login_id = data.get("email")
    password = data.get("password")
    print(f"\n--- DEBUG LOGIN ---")
    print(f"Mencoba Login dengan ID: {Login_id}")

    query = text("""
        SELECT
            u.id_user, u.username, u.nama_lengkap, u.email, u.password,
            u.id_role, u.id_unit, u.id_unit_akses,
            r.nama_role as role_name,
            n.nama_unit as unit_name
        FROM users u
        JOIN role r ON u.id_role = r.id_role
        JOIN unit n ON u.id_unit = n.id_unit
        WHERE u.email = :Login_param OR u.username = :Login_param
    """)

    result = db.execute(query, {"Login_param": Login_id})
    user = result.fetchone()

    if user:
        user_dict = dict(user._mapping)
```

```

print(f"User ditemukan: {user_dict['username']}
(Role: {user_dict['role_name']})")

if check_password(password, user_dict['password']):
    print("Status: LOGIN BERHASIL.")
    token = generate_token(user_dict)
    return {
        "success": True,
        "token": token,
        "user": {
            "id_user": user_dict["id_user"],
            "nama": user_dict["nama_lengkap"],
            "username": user_dict["username"],
            "email": user_dict["email"],
            "role": user_dict["role_name"],
            "id_role": user_dict["id_role"],
            "unit": user_dict["unit_name"],
            "id_unit": user_dict["id_unit"],
            "id_unit_akses": user_dict["id_unit_akses"]
        }
    }
else:
    print("Status: PASSWORD SALAH!")
    raise HTTPException(
        status_code=401,
        detail="Password salah"
    )
else:
    print("Status: USER TIDAK DITEMUKAN!")
    raise HTTPException(
        status_code=404,
        detail="Akun tidak ditemukan"
    )

```

)

b. Identifikasi *Statement* Fungsi *Login*Tabel 5. 1 Identifikasi *Statement* Fungsi *Login*

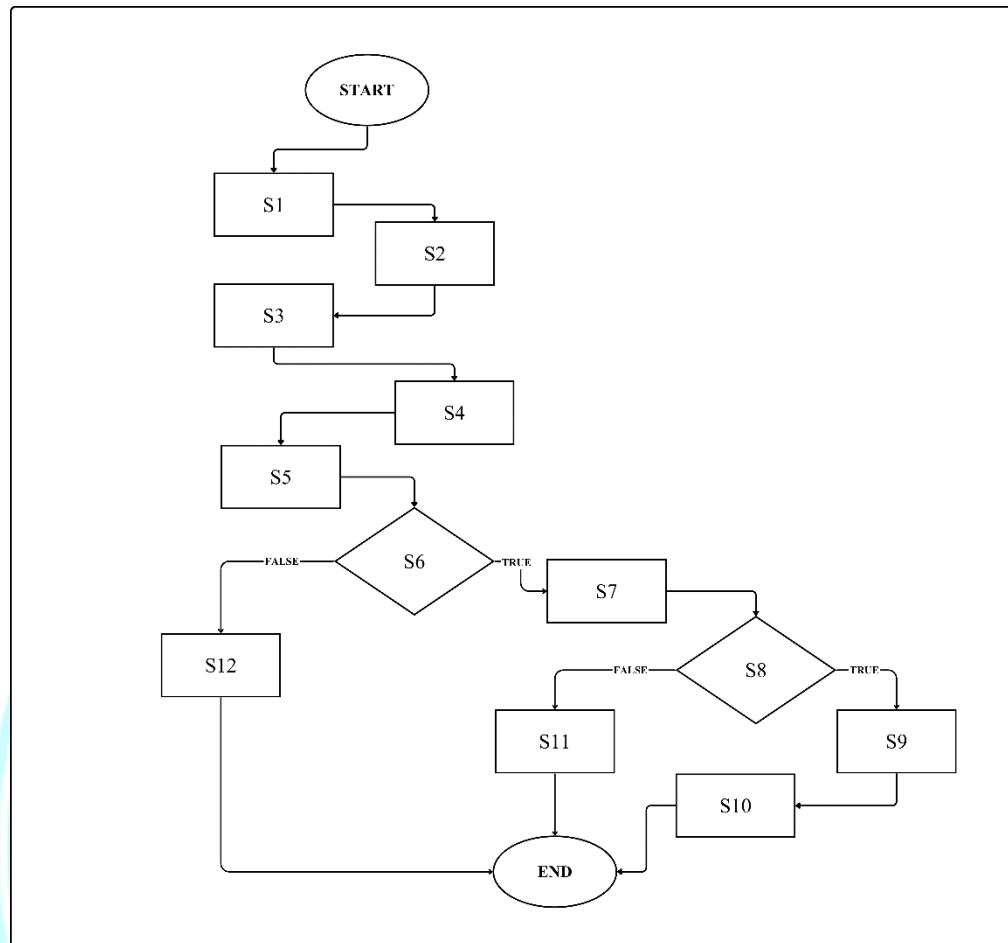
<i>Statement</i>	<i>Source code</i>	<i>Keterangan</i>
S1	<code>Login_id = data.get("email")</code>	Mengambil input email atau username
S2	<code>password = data.get("password")</code>	Mengambil input password
S3	<pre> query = text(""" SELECT u.id_user, u.username, u.nama_lengkap, u.email, u.password, u.id_role, u.id_unit, u.id_unit_akses, r.nama_role as role_name, n.nama_unit as unit_name FROM users u JOIN role r ON u.id_role = r.id_role JOIN unit n ON u.id_unit = n.id_unit WHERE u.email = :Login_param OR u.username = :Login_param""") </pre>	Membuat query pencarian user
S4	<code>result = db.execute(query, {"Login_param": Login_id})</code>	Menjalankan query ke database
S5	<code>user = result.fetchone()</code>	Mengambil hasil query user
S6	<code>if user:</code>	Memeriksa apakah user ditemukan
S7	<code>user_dict = dict(user._mapping)</code>	Mengubah hasil query menjadi dictionary
S8	<code>if check_password(password, user_dict['password']):</code>	Memeriksa kecocokan password
S9	<code>token = generate_token(user_dict)</code>	Membuat token autentikasi

<i>Statement</i>	<i>Source code</i>	<i>Keterangan</i>
S10	<code>return {"success": True, ...}</code>	Mengembalikan data <i>Login</i> berhasil
S11	<code>raise HTTPException(status_code=401, detail="Password salah")</code>	Mengembalikan pesan password salah
S12	<code>raise HTTPException(status_code=404, detail="Akun tidak ditemukan")</code>	Mengembalikan pesan akun tidak ditemukan

Sumber : data diolah oleh penulis(2026)



c. *Flowchart Fungsi Login*



Sumber : data diolah oleh penulis (2026)

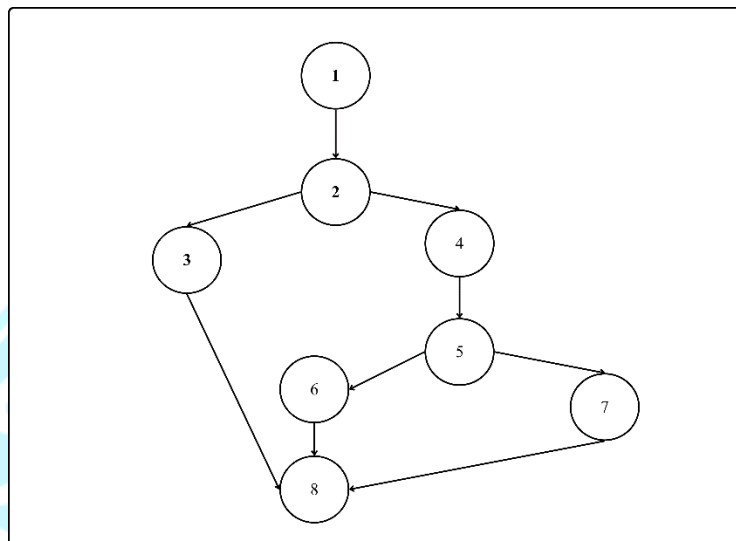
Berdasarkan diagram alir pada Gambar 5. *Flowchart Fungsi Login*, proses *Login* dimulai ketika pengguna memasukkan *Login_id* berupa email atau username serta password melalui antarmuka *Flutter*. Setelah itu, sistem menyiapkan dan mengeksekusi kueri basis data dengan melakukan join pada tabel *users*, *role*, dan *unit* untuk mencari data akun pengguna.

Selanjutnya, sistem melakukan pengecekan apakah akun ditemukan di dalam basis data. Jika akun tidak ditemukan, maka sistem akan menampilkan pesan bahwa akun tidak tersedia dan proses *Login* berakhir. Namun apabila akun ditemukan, sistem melanjutkan proses dengan mengambil data profil pengguna dan melakukan verifikasi kecocokan password menggunakan fungsi *check_password*.

Jika hasil verifikasi menunjukkan password tidak sesuai, maka sistem menampilkan pesan kesalahan password. Sebaliknya, apabila password benar,

sistem akan menghasilkan token akses (generate token) dan mengembalikan data lengkap pengguna ke aplikasi sebagai tanda bahwa proses *Login* berhasil dilakukan.

d. *Flowgraph Fungsi Login*



Sumber : data diolah oleh penulis (2026)

Gambar 5.25 *Flowgraph Fungsi Login*

Flowgraph pada fungsi *Login* ini merepresentasikan alur eksekusi program berdasarkan hasil pengelompokan *statement* yang telah diidentifikasi sebelumnya. *Flowgraph* yang dibentuk terdiri dari 8 *node* dan 9 *edge*, dengan penambahan *node* 8 sebagai terminal *node* (*end process*) yang berfungsi untuk menandai berakhirnya seluruh alur eksekusi program. *Node* pertama (*node* 1) terdiri dari S1 hingga S5 yang mencakup proses pengambilan input email atau username, password, serta penyusunan query untuk pencarian data pengguna pada *database*. Seluruh proses tersebut masih berada pada alur linear tanpa adanya percabangan sehingga digabungkan ke dalam satu *node*.

Selanjutnya, *Node* 2 merepresentasikan *statement* S6 yang merupakan titik keputusan untuk memeriksa keberadaan user di dalam *database*. Apabila kondisi tidak terpenuhi, alur akan berakhir pada *Node* 3 (S12). Apabila kondisi terpenuhi, alur dilanjutkan ke *Node* 4 yang berisi S7, kemudian menuju *Node* 5 sebagai pengecekan kecocokan password (S8). Berdasarkan hasil evaluasi tersebut, alur

program akan berakhir pada tiga kemungkinan terminal, yaitu *Node* 3 (S12), *Node* 6 (S11), dan *Node* 7 (S9–S10), yang kemudian semuanya bermuara pada *Node* 8 sebagai *node* akhir (*end process*).

e. *Cyclomatic Complexity* Fungsi *Login*

Berdasarkan *Flowgraph* yang telah dibentuk, terdapat 2 *predicate node* (P), yaitu pada *Node* 2 yang melakukan pengecekan keberadaan user dalam *database*, dan *Node* 5 yang melakukan pengecekan kecocokan password. Oleh karena itu, nilai *Cyclomatic Complexity* dapat dihitung menggunakan rumus berikut :

$$V(G) = P + 1$$

$$V(G) = 2 + 1$$

$$V(G) = 3$$

Dari perhitungan tersebut nilai 3 dimana nilai *Cyclomatic Complexity* tersebut menunjukkan bahwa terdapat tiga jalur independen dalam proses eksekusi fungsi *Login*, yang merepresentasikan seluruh kemungkinan alur logika berdasarkan kondisi yang terjadi pada program. Nilai ini kemudian digunakan sebagai dasar untuk menentukan jumlah *independent path* yang akan diuji pada tahap pengujian berikutnya, sehingga setiap jalur eksekusi dapat diverifikasi untuk memastikan seluruh kondisi pada fungsi *Login* telah ditangani dengan benar.

f. *Independen path Fungsi Login*

Berdasarkan hasil perhitungan *Cyclomatic Complexity* yang diperoleh dari *Flowgraph* fungsi *Login*, nilai tersebut digunakan sebagai dasar untuk menentukan jumlah jalur independen yang harus diuji dalam proses *white box testing*. Setiap jalur independen merepresentasikan kemungkinan alur eksekusi program yang berbeda berdasarkan kondisi percabangan, khususnya pada pengecekan keberadaan user di *database* dan kesesuaian password. Dengan demikian, setiap kemungkinan skenario pada fungsi *Login* dapat dianalisis dan diuji secara menyeluruh untuk memastikan seluruh logika program berjalan dengan benar. Sehingga didapatkan 3 jalur independen (*independent path*) berikut ini:

1. Path 1:

$1 \rightarrow 2 \rightarrow 3 \rightarrow 8$

(User tidak ditemukan di *database* / akun tidak ditemukan)

2. Path 2:

$1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 8$

(User ditemukan, namun password tidak sesuai)

3. Path 3:

$1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 7 \rightarrow 8$

(User ditemukan dan password sesuai, *Login* berhasil)

g. *Test case Fungsi Login*

Pengujian dilakukan berdasarkan jalur independen (*independent path*) yang telah diperoleh dari hasil analisis *Flowgraph* fungsi *Login*. Setiap jalur independen direpresentasikan ke dalam beberapa *test case* dengan kombinasi input username dan password yang berbeda untuk menguji seluruh kemungkinan alur eksekusi program. Jalur eksekusi pada masing-masing *test case* diperoleh dari *node-node* yang dilalui pada *Flowgraph* sehingga setiap proses pengujian dapat memverifikasi fungsi *Login* secara menyeluruh.

Tabel 5. 2 *Test case* Fungsi *Login*

<i>Test case</i>	Username	Password	Jalur Eksekusi	Output
TC1	Username tidak terdaftar	Bebas	1 → 2 → 3 → 8	Akun tidak ditemukan
TC2	Username valid	Password salah	1 → 2 → 4 → 5 → 6 → 8	Password salah
TC3	Username valid	Password benar	1 → 2 → 4 → 5 → 7 → 8	Login berhasil

Sumber : data diolah oleh penulis (2026)

h. *Statement Coverage* Fungsi *Login*

Statement Coverage digunakan untuk mengukur sejauh mana seluruh *statement* pada fungsi *Login* telah dieksekusi melalui *test case* yang telah dirancang. Pengujian ini bertujuan untuk memastikan bahwa setiap *statement* pada program telah dijalankan minimal satu kali sehingga seluruh logika program dapat diverifikasi dengan baik. *Statement* yang Tereksekusi:

1. TC1:

S1, S2, S3, S4, S5, S6, S12

2. TC2:

S1, S2, S3, S4, S5, S6, S7, S8, S11

3. TC3:

S1, S2, S3, S4, S5, S6, S7, S8, S9, S10

Berdasarkan hasil pengujian yang telah dilakukan, seluruh *statement* pada fungsi *Login* yaitu S1 hingga S12 berhasil dieksekusi melalui kombinasi *test case* TC1, TC2, dan TC3. Dengan demikian diperoleh nilai *statement coverage* sebesar:

$$12/12 \times 100\% = 100\%$$

Nilai tersebut menunjukkan bahwa seluruh *statement* pada fungsi *Login* telah diuji dan seluruh kemungkinan alur eksekusi program berhasil dicakup dalam proses pengujian *white box*.

5.4.2 Pengujian Fungsi Upload

a. Potongan Source code Fungsi Upload

Endpoint ini berfungsi sebagai penghubung utama antara aplikasi frontend dan proses pengolahan dokumen di sisi server. Proses yang dilakukan meliputi penerimaan *File* dokumen dari pengguna, validasi tipe dan ukuran *File*, proses decoding gambar, hingga pembuatan *job OCR* yang diproses secara asynchronous menggunakan worker thread. Berikut adalah potongan *source code* fungsi *upload* yang digunakan dalam tahap pengujian:

```

async def OCR_endpoint(
    File: UploadFile = File(...),
    coords: Optional[str] = Form(None),
    width: Optional[float] = Form(None),
    height: Optional[float] = Form(None),
    jenis_surat: Optional[str] = Form("masuk"),
):
    try:
        content = await File.read()
        Filename = File.Filename.lower()

        ext = os.path.splitext(Filename)[1]
        if ext not in _ALLOWED_EXT:
            raise HTTPException(400, f"Tipe File tidak didukung. Gunakan: {_ALLOWED_EXT_STR}.")

        if len(content) > _MAX_FILE_SIZE:
            raise HTTPException(400, f"Ukuran File terlalu besar. Maksimal {_MAX_FILE_SIZE // (1024 * 1024)}MB.")

    except ValueError as e:
        raise HTTPException(400, str(e))

```

```

job_id      = str(uuid.uuid4())
_jobs[job_id] = {"status": "queued", "result": None, "created_at": time.time()}
def _worker():
    try:
        result = process_document(
            Image, coords, width, height, jenis_surat,
            on_status=lambda s: _jobs[job_id].update({"status": s}) )
        _jobs[job_id]["status"] = "done"
        _jobs[job_id]["result"] = result
    except ValueError as e:
        _jobs[job_id]["status"]      = "error"
        _jobs[job_id]["error_type"] = "unprocessable"
        _jobs[job_id]["message"]    = str(e)
    except Exception as e:
        _jobs[job_id]["status"]      = "error"
        _jobs[job_id]["error_type"] = "server_error"
        _jobs[job_id]["message"]    = str(e)
    Thread(target=_worker, daemon=True).start()
    _cleanup_old_jobs()

    print(f" [JOB {job_id}] Diterima & antri diproses.")
    return {"success": True, "job_id": job_id}

except HTTPException:
    raise

except Exception as e:
    print(f"ERROR OCR ENDPOINT: {str(e)}")
    raise HTTPException(500, f"Terjadi kesalahan: {str(e)}")

```

b. Identifikasi *Statement* Fungsi *Upload*

Berdasarkan *source code* Fungsi *Upload* tersebut, dilakukan identifikasi *statement* untuk mempermudah proses analisis alur program dalam metode *white box testing*.

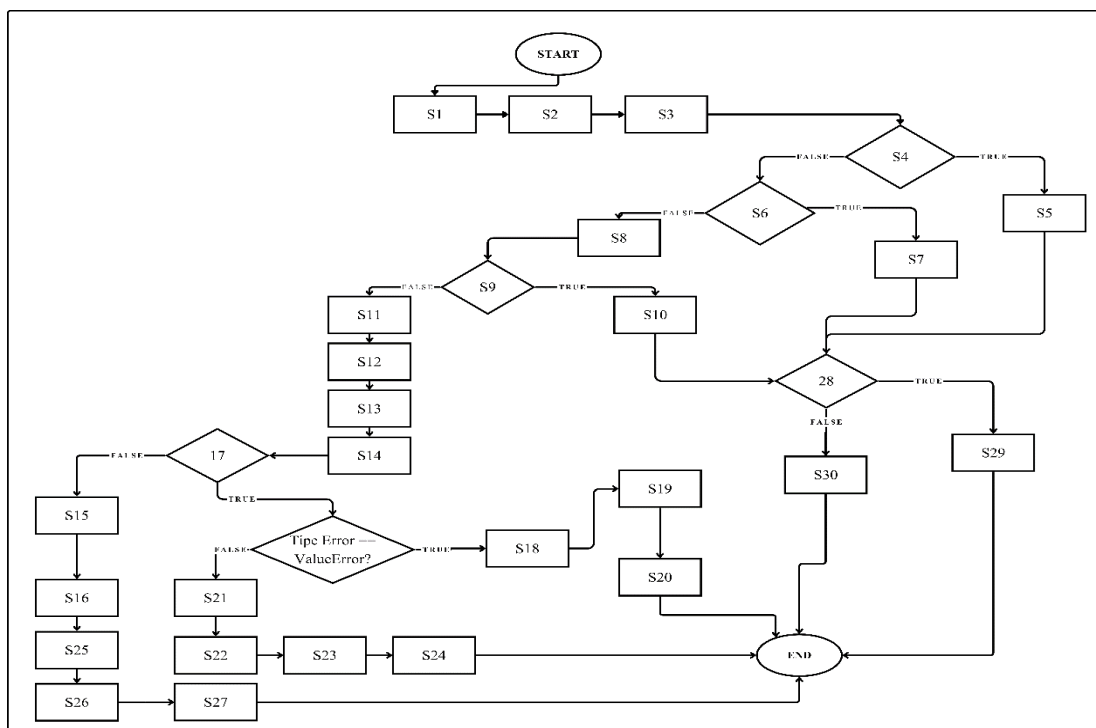
Tabel 5. 3 Identifikasi *Statement* Fungsi *Upload*

<i>Statement</i>	<i>Source code</i>	<i>Keterangan</i>
S1	<code>content = await File.read()</code>	Membaca isi <i>File</i> yang diunggah
S2	<code>Filename = File.Filename.lower()</code>	Mengambil nama <i>File</i> dan mengubah menjadi huruf kecil
S3	<code>ext = os.path.splitext(Filename)[1]</code>	Mengambil ekstensi <i>File</i>
S4	<code>if ext not in _ALLOWED_EXT:</code>	Memeriksa apakah ekstensi <i>File</i> didukung
S5	<code>raise HTTPException(400, ...)</code>	Menampilkan <i>error</i> tipe <i>File</i> tidak didukung
S6	<code>if len(content) > _MAX_FILE_SIZE:</code>	Memeriksa ukuran <i>File</i>
S7	<code>raise HTTPException(400, ...)</code>	Menampilkan <i>error</i> ukuran <i>File</i> terlalu besar
S8	<code>Image = decode_bytes_to_Image(content, Filename)</code>	Mengubah <i>File</i> menjadi <i>Image</i> untuk proses OCR
S9	<code>except ValueError as e:</code>	Menangkap <i>error</i> saat proses decode <i>Image</i>
S10	<code>raise HTTPException(400, str(e))</code>	Menampilkan <i>error</i> hasil decode <i>Image</i>
S11	<code>job_id = str(uuid.uuid4())</code>	Membuat ID job OCR
S12	<code>_jobs[job_id] = {"status": "queued", ...}</code>	Menyimpan status awal job OCR
S13	<code>def _worker():</code>	Mendefinisikan worker untuk proses OCR
S14	<code>result = process_document(...)</code>	Memproses dokumen OCR
S15	<code>_jobs[job_id]["status"] = "done"</code>	Mengubah status job menjadi selesai
S16	<code>_jobs[job_id]["result"] = result</code>	Menyimpan hasil OCR
S17	<code>except ValueError as e:</code>	Menangkap <i>error</i> saat proses OCR

Statement	Source code	Keterangan
S18	<code>_jobs[job_id]["status"] = "error"</code>	Mengubah status job menjadi <i>error</i>
S19	<code>_jobs[job_id]["error_type"] = "unprocessable"</code>	Menentukan jenis <i>error</i> unprocessable
S20	<code>_jobs[job_id]["message"] = str(e)</code>	Menyimpan pesan <i>error</i> OCR
S21	<code>except Exception as e:</code>	Menangkap server <i>error</i> pada worker
S22	<code>_jobs[job_id]["status"] = "error"</code>	Mengubah status job menjadi <i>error</i>
S23	<code>_jobs[job_id]["error_type"] = "server_error"</code>	Menentukan jenis server <i>error</i>
S24	<code>_jobs[job_id]["message"] = str(e)</code>	Menyimpan pesan server <i>error</i>
S25	<code>Thread(target=_worker, daemon=True).start()</code>	Menjalankan proses OCR menggunakan thread
S26	<code>_cleanup_old_jobs()</code>	Membersihkan job lama
S27	<code>return {"success": True, "job_id": job_id}</code>	Mengembalikan response berhasil
S28	<code>except HTTPException:</code>	Menangkap <i>HTTPException</i>
S29	<code>raise</code>	Melempar kembali <i>HTTPException</i>
S30	<code>raise HTTPException(400, ...)</code>	Menampilkan <i>error</i> server internal

Sumber : data diolah oleh penulis (2026)

c. Flowchart Fungsi Upload



Sumber : data diolah oleh penulis (2026)

Gambar 5.26 Flowchart Fungsi Upload

Flowchart fungsi upload menggambarkan proses sistem dimulai dari pembacaan isi File yang diunggah pengguna, kemudian sistem mengambil nama File, mengubahnya menjadi huruf kecil, serta mengekstrak ekstensi File. Setelah itu dilakukan validasi terhadap jenis File dan ukuran File melalui simpul keputusan. Jika ekstensi File tidak didukung atau ukuran File melebihi batas maksimum, sistem akan langsung menghasilkan `HTTPException` sebagai bentuk penanganan kesalahan.

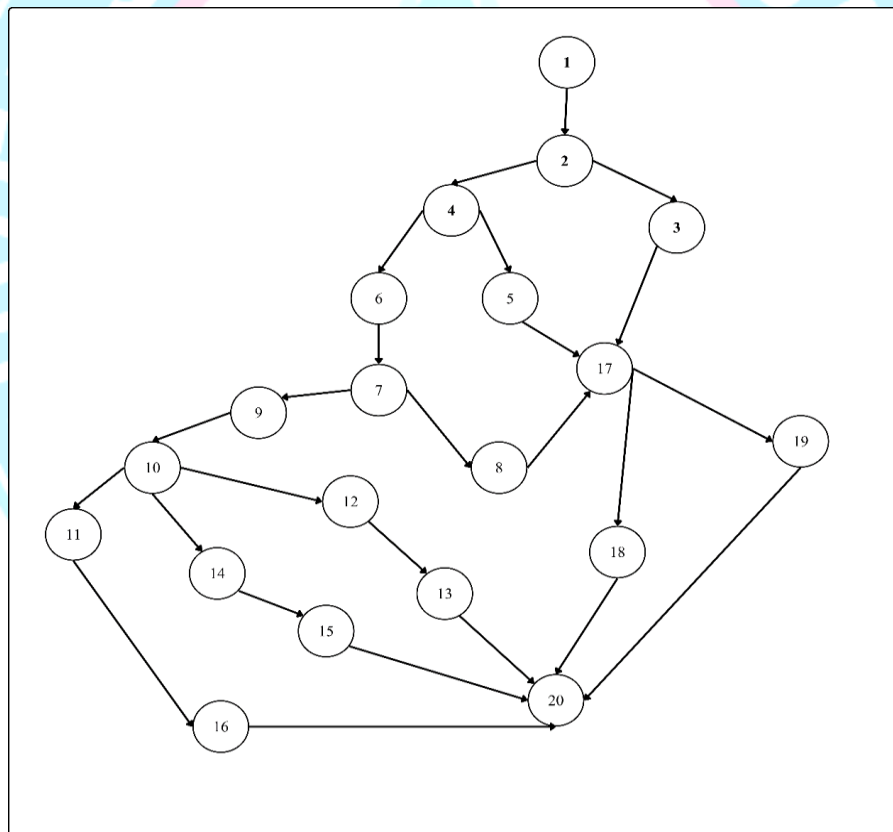
Apabila File lolos tahap validasi, sistem melanjutkan proses konversi data bytes menjadi objek gambar. Tahap ini juga dilengkapi mekanisme penanganan error untuk mendeteksi kemungkinan kerusakan atau ketidaksesuaian format gambar. Jika proses decode berhasil, sistem akan membuat ID job OCR menggunakan UUID, menyimpan status awal job sebagai `queued`, serta mendefinisikan fungsi worker asinkronus untuk menjalankan proses OCR.

Pada tahap pemrosesan OCR, sistem memantau kemungkinan terjadinya runtime error. Jika proses berjalan normal, status job diperbarui menjadi `done` dan

hasil ekstraksi teks disimpan. Sebaliknya, jika terjadi kegagalan, sistem akan mengelompokkan jenis *error* menjadi *unprocessable* atau *server_error*, kemudian mencatat status dan pesan kegagalannya ke dalam sistem.

Setelah worker selesai didefinisikan, sistem menjalankan *background thread* untuk memproses *OCR* secara paralel, dilanjutkan dengan pembersihan data job yang telah kedaluwarsa. Selanjutnya sistem mengembalikan response sukses berupa data JSON yang berisi ID job kepada pengguna. Selain itu, seluruh *error* utama juga ditangani melalui mekanisme *global exception handling* untuk memastikan setiap kesalahan dapat diteruskan sebagai response yang sesuai maupun sebagai indikasi internal server *error*.

d. *Flowgraph Fungsi Upload*



Sumber : data diolah oleh penulis (2026)

Gambar 5.27 *Flowgraph Fungsi Upload*

Flowgraph pada fungsi *upload* merepresentasikan alur eksekusi program berdasarkan hasil pengelompokan *statement* yang telah diidentifikasi sebelumnya.

Flowgraph ini terdiri dari 20 *node* dan 25 *edge*, dengan *Node* 20 sebagai terminal *node* (*end process*) yang menandai berakhirnya proses eksekusi program. Beberapa *statement* linear yang tidak memiliki percabangan digabungkan ke dalam satu *node* agar *Flowgraph* lebih ringkas tanpa mengubah logika program.

Proses dimulai dari *Node* 1 yang mencakup *statement* S1, S2 dan S3, yaitu proses penerimaan *File upload*, pembacaan isi *File*, dan identifikasi ekstensi *File*. Alur kemudian menuju *Node* 2 (S4) sebagai *predicate node* untuk memvalidasi ekstensi *File*. Jika ekstensi tidak valid, alur diarahkan ke *Node* 3 (S5) untuk menghasilkan *HTTPException* tipe *File*, lalu diteruskan ke *Node* 17 sebagai pusat penanganan *exception* global sebelum berakhir di *Node* 20. Namun jika valid, alur dilanjutkan ke *Node* 4 (S6) untuk memvalidasi ukuran *File*. Jika ukuran melebihi batas maksimum, sistem menuju *Node* 5 (S7) untuk menghasilkan *HTTPException* ukuran *File*, sedangkan jika valid proses dilanjutkan ke *Node* 6 (S8) untuk *decode File* menjadi objek *Image*.

Setelah proses *decode* dilakukan, *Node* 7 (S9) berfungsi sebagai *predicate node* untuk menentukan keberhasilan *decode Image*. Jika gagal, alur menuju *Node* 8 (S10) untuk menghasilkan *HTTPException* *decode File* dan diteruskan ke mekanisme *exception* global. Jika berhasil, alur dilanjutkan ke *Node* 9 yang mencakup *statement* S11–S13, yaitu pembuatan *job_id*, penyimpanan status awal *OCR*, dan pendefinisian *worker thread*.

Selanjutnya proses *OCR* dijalankan pada *Node* 10 (S14) dan dipantau oleh *Node* 12 (S17) untuk mendeteksi kesalahan selama proses berlangsung. Jika *OCR* berhasil, alur menuju *Node* 11 (S15–S16) untuk memperbarui status *job* menjadi *done* dan menyimpan hasil ekstraksi teks sebelum berakhir di *Node* 20. Sebaliknya, jika terjadi *error*, alur menuju *Node* 14 untuk memeriksa jenis kesalahan. *Error* bertipe *ValueError* akan diarahkan ke *Node* 13 (S18–S20) untuk mencatat *unprocessable error*, sedangkan *error* sistem lainnya akan diarahkan ke *Node* 15 (S22–S24) untuk mencatat *server_error* sebelum berakhir pada terminal *node*.

Secara paralel, *main thread* dari *Node* 9 akan melanjutkan alur menuju *Node* 16 (S25–S27) untuk menjalankan *worker thread*, membersihkan data *job* kedaluwarsa, dan mengembalikan respons sukses berupa *job_id* kepada pengguna sebelum berakhir di *Node* 20. Sementara itu, seluruh *HTTPException* dari *Node* 3,

Node 5, dan *Node 8* akan diproses pada *Node 17* (S28). Jika *error* termasuk *HTTPException*, alur menuju *Node 18* (S29) untuk melempar kembali *error* tersebut, sedangkan *error* lain akan dialihkan ke *Node 19* (S30) sebagai internal server *error* sebelum seluruh alur berakhir di *Node 20*.

e. *Cyclomatic Complexity Fungsi Upload*

Berdasarkan *Flowgraph* yang telah dibentuk, terdapat 6 *predicate node* (P), yaitu pada *Node 2* yang melakukan validasi ekstensi *File*, *Node 4* yang melakukan validasi ukuran *File*, *Node 7* yang melakukan pengecekan keberhasilan proses *decode Image*, *Node 12* yang melakukan pengecekan interupsi kesalahan proses *OCR*, *Node 14* yang melakukan klasifikasi tipe *error* pada worker thread, dan *Node 17* yang melakukan validasi tipe pengecualian pada penanganan *exception* global. Oleh karena itu, nilai *Cyclomatic Complexity* dapat dihitung menggunakan rumus berikut:

$$V(G) = P+1$$

$$V(G) = 6+1$$

$$V(G) = 7$$

Dari perhitungan tersebut diperoleh nilai *Cyclomatic Complexity* sebesar 7. Nilai tersebut menunjukkan bahwa terdapat tujuh jalur independen (*independent path*) pada proses eksekusi fungsi *upload*. Setiap jalur independen merepresentasikan kemungkinan alur logika yang berbeda berdasarkan kondisi percabangan yang terjadi selama proses *upload* dan pengolahan dokumen *OCR*. Nilai *Cyclomatic Complexity* ini kemudian digunakan sebagai dasar dalam menentukan jumlah jalur independen yang akan diuji pada tahap *white box testing*, sehingga seluruh kemungkinan alur eksekusi program dapat diverifikasi untuk memastikan fungsi *upload* berjalan dengan benar sesuai kondisi yang diharapkan.

f. *Independen path Fungsi Upload*

Berdasarkan hasil perhitungan *Cyclomatic Complexity* yang diperoleh dari *Flowgraph* fungsi *upload*, nilai tersebut digunakan sebagai dasar untuk menentukan jumlah jalur independen yang harus diuji dalam proses *white box testing*. Setiap jalur independen merepresentasikan kemungkinan alur eksekusi program yang

berbeda berdasarkan kondisi percabangan, khususnya pada validasi dokumen, inisialisasi *background thread*, dan penanganan *exception* global. Dengan demikian, setiap kemungkinan skenario pada fungsi *upload* dapat dianalisis dan diuji secara menyeluruh untuk memastikan seluruh logika program berjalan dengan benar. Sehingga didapatkan 7 jalur independen (*independent path*) berikut ini:

1. Path 1 (Validasi Ekstensi File Gagal):

1 → 2 → 3 → 17 → 18 → 20

(File yang diunggah memiliki format atau ekstensi yang tidak didukung oleh sistem sehingga proses dihentikan dan sistem mengembalikan *HTTPException*.)

2. Path 2 (Validasi Ukuran File Gagal):

1 → 2 → 4 → 4 → 17 → 18 → 20

(File memiliki format yang valid, namun ukuran File melebihi batas maksimum yang ditentukan oleh sistem sehingga proses *upload* dibatalkan.)

3. Path 3 (Proses Decode Image Gagal):

1 → 2 → 4 → 6 → 7 → 8 → 17 → 18 → 20

(File berhasil melewati validasi format dan ukuran, namun proses konversi data gambar gagal karena File rusak, korup, atau tidak dapat didekode.)

4. Path 4 (Skenario Utama Berhasil):

1 → 2 → 4 → 6 → 7 → 9 → 10 → 11 → 16 → 20

(Seluruh proses validasi berhasil dilalui, sistem mengembalikan respons sukses berupa *job_id*, kemudian *background thread* menjalankan proses *OCR* hingga status job berhasil diselesaikan.)

5. Path 5 (Worker Thread Mengalami *ValueError*):

1 → 2 → 4 → 6 → 7 → 9 → 10 → 12 → 13 → 20

(Seluruh validasi awal berhasil dilalui, namun proses *OCR* pada *background thread* mengalami *ValueError* akibat dokumen tidak dapat diproses secara valid oleh sistem.)

6. Path 6 (Worker Thread Mengalami *General Exception*):

1 → 2 → 4 → 6 → 7 → 9 → 10 → 14 → 15 → 20

(Sistem berhasil melewati seluruh proses validasi awal, namun terjadi gangguan atau kesalahan sistem internal pada proses *OCR* sehingga memicu *general exception/server error*.)

7. Path 7 (Penanganan Global Non-*HTTPException*):

1 → 2 → 4 → 6 → 7 → 8 → 17 → 19 → 20

(Terjadi kesalahan internal yang tidak termasuk kategori *HTTPException* pada alur utama sistem sehingga *exception* dialihkan ke mekanisme penanganan global internal server.)

g. *Test case Fungsi Upload*

Pengujian dilakukan berdasarkan jalur independen (*independent path*) yang telah diperoleh dari hasil analisis *Flowgraph* fungsi *upload*. Setiap jalur independen direpresentasikan ke dalam beberapa *test case* dengan kombinasi kondisi *File* input dan proses *background* thread yang berbeda untuk menguji seluruh kemungkinan alur eksekusi program. Jalur eksekusi pada masing-masing *test case* diperoleh dari *node-node* yang dilalui pada *Flowgraph* sehingga setiap proses pengujian dapat memverifikasi fungsi *upload* secara menyeluruh.

Tabel 4. 7 *Test case* Fungsi *Upload*

<i>Test case</i>	Kondisi <i>File</i> Input	Status Proses Worker Thread	Jalur Eksekusi	Output / Respon Sistem
TC1	<i>File</i> teks (.txt)	Tidak dieksekusi	1 → 2 → 3 → 17 → 18 → 20	<i>HTTPException</i> (400): Tipe <i>File</i> tidak didukung
TC2	Gambar JPG berukuran 50MB	Tidak dieksekusi	1 → 2 → 4 → 5 → 17 → 18 → 20	<i>HTTPException</i> (400): Ukuran <i>File</i> terlalu besar
TC3	<i>File</i> gambar .jpg yang korup/rusak	Tidak dieksekusi	1 → 2 → 4 → 6 → 7 → 8 → 17 → 18 → 20	<i>HTTPException</i> (400): Gagal melakukan decode <i>File</i> gambar
TC4	Gambar valid (PNG/JPG, ukuran sesuai)	Sukses tanpa interupsi	1 → 2 → 4 → 6 → 7 → 9 → 10 → 11 → 16 → 20	JSON <i>job_id</i> berhasil dikembalikan dan status <i>job</i> berubah menjadi “done”
TC5	Gambar valid (PNG/JPG, ukuran sesuai)	Interupsi <i>ValueError</i>	1 → 2 → 4 → 6 → 7 → 9 → 10 → 12 → 13 → 20	Status <i>job</i> berubah menjadi “error” dengan tipe “unprocessable”
TC6	Gambar valid (PNG/JPG, ukuran sesuai)	Interupsi <i>Exception</i> Sistem	1 → 2 → 4 → 6 → 7 → 9 → 10 → 14 → 15 → 20	Status <i>job</i> berubah menjadi “error” dengan tipe “server_error”

<i>Test case</i>	<i>Kondisi File Input</i>	<i>Status Proses Worker Thread</i>	<i>Jalur Eksekusi</i>	<i>Output / Respon Sistem</i>
TC7	Memicu <i>error</i> internal sistem utama	Tidak dieksekusi	1 → 2 → 4 → 6 → 7 → 8 → 17 → 19 → 20	<i>HTTPException (500): Internal Server Error</i>

Sumber : data diolah oleh penulis (2026)

h. *Statement Coverage* Fungsi *Upload*

Statement Coverage digunakan untuk mengukur sejauh mana seluruh *statement* pada fungsi *upload* telah dieksekusi melalui *test case* yang telah dirancang. Pengujian ini bertujuan untuk memastikan bahwa setiap *statement* pada program telah dijalankan minimal satu kali sehingga seluruh logika program dapat diverifikasi dengan baik.

Statement yang Tereksekusi pada Masing-Masing *Test case*:

1. TC1: S1, S2, S3, S4, S5, S28, S29
2. TC2: S1, S2, S3, S4, S6, S7, S28, S29
3. TC3: S1, S2, S3, S4, S6, S8, S9, S10, S28, S30
4. TC4: S1, S2, S3, S4, S6, S8, S9, S11, S12, S13, S14, S15, S16, S17, S25, S26, S27
5. TC5: S1, S2, S3, S4, S6, S8, S11, S12, S13, S14, S17, S18, S19, S20
6. TC6: S1, S2, S3, S4, S6, S8, S11, S12, S13, S14, S17, S21, S22, S23, S24
7. TC7: S1, S2, S3, S4, S6, S8, S9, S10, S28, S29

Berdasarkan hasil pengujian yang telah dilakukan, seluruh *statement* pada fungsi *upload* yaitu S1 hingga S30 berhasil dieksekusi melalui kombinasi *test case* TC1, TC2, TC3, TC4, TC5, TC6, dan TC7. Dengan demikian diperoleh nilai *statement coverage* sebesar:

$$(30 / 30) \times 100\% = 100\%$$

Nilai tersebut menunjukkan bahwa seluruh *statement* pada fungsi *upload* telah diuji dan seluruh kemungkinan alur eksekusi program berhasil dicakup dalam proses pengujian *white box*.

5.4.3 Pengujian *Preprocessing* Dokumen

- a. *Source code* Fungsi *Preprocessing* Dokumen
def enhance_for_pdf(img):

```

h, w = img.shape[:2]
if w < 1500:
    scale = 1500 / w
    img = cv2.resize(img, None, fx=scale, fy=scale, interpolation =
        cv2.INTER_LANCZOS4)
    gaussian = cv2.GaussianBlur(img, (0, 0), 2)
    img_color = cv2.addWeighted(img, 1.3, gaussian, -0.3, 0)
    img_OCR = cv2.cvtColor(img_color, cv2.COLOR_BGR2GRAY)
    img_OCR = cv2.normalize(img_OCR, None, 0, 255,
cv2.NORM_MINMAX)
    return img_color, img_OCR
def prepare_Image_for_OCR(Image, manual_points=None):
    if manual_points is not None and len(manual_points) == 4:
        print("[PREPROCESSING] Mode manual crop")
        pts = np.array(manual_points, dtype="float32")
        warped = four_point_transform(Image, pts)
    else:
        print("[PREPROCESSING] Mode auto detect")
        corners = detect_document_corners(Image)
        if corners is None:
            print("[PREPROCESSING] Auto detect gagal, pakai full Image")
            warped = Image
        else:
            print("[PREPROCESSING] Auto detect sukses")
            warped = four_point_transform(Image, corners)
    img_color, img_OCR = enhance_for_pdf(warped)
    return img_color, img_OCR

```

b. Identifikasi *Statement Fungsi Preprocessing* Dokumen

Berdasarkan *source code* fungsi *Preprocessing* dokumen tersebut, dilakukan identifikasi *statement* untuk mempermudah proses analisis alur program dalam metode *white box testing*

Tabel 5. 4 *Statement Fungsi Preprocessing* Dokumen

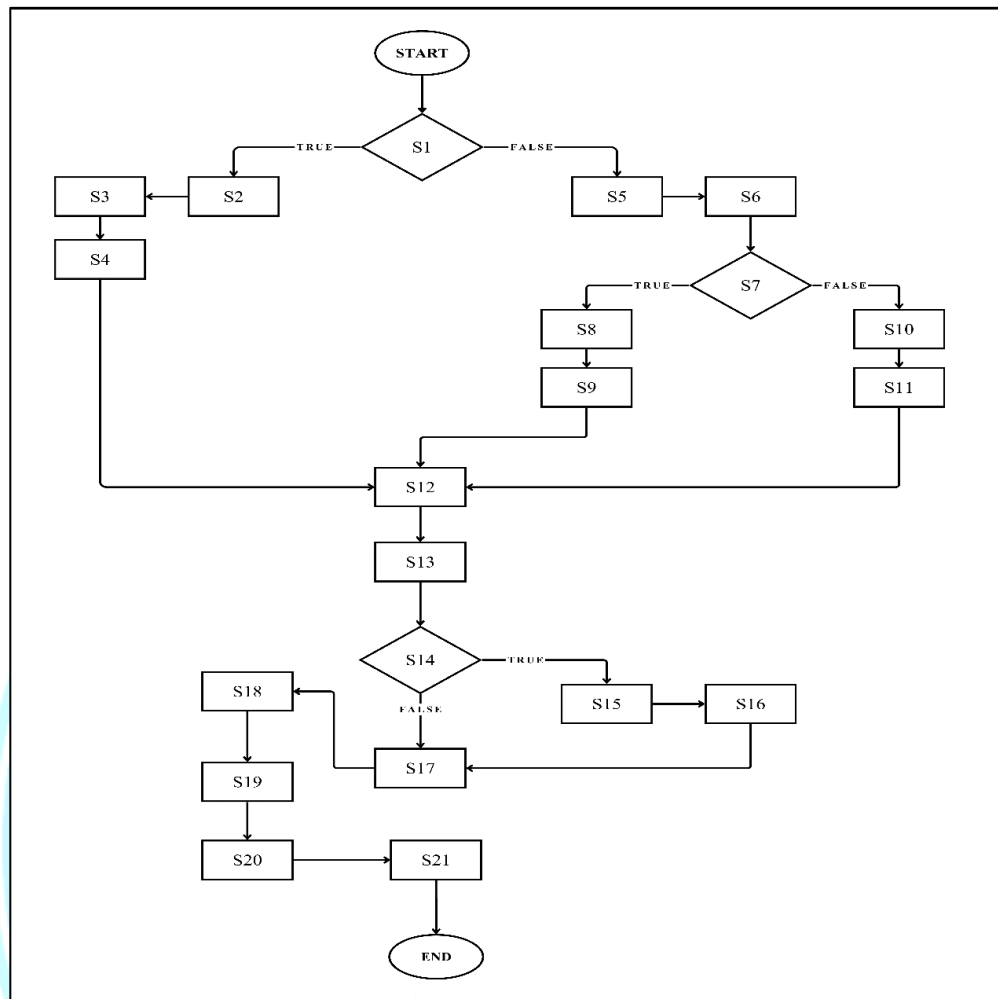
<i>Statement</i>	<i>Source code</i>	<i>Keterangan</i>
S1	if manual_points is not None and len(manual_points) == 4:	Memeriksa kelengkapan dan jumlah koordinat manual sebanyak 4 titik. (Percabangan)
S2	print(" [PREPROCESSING] Mode manual crop")	Menampilkan log inisialisasi mode manual crop.
S3	pts = np.array(manual_points, dtype="float32")	Mengonversi koordinat manual menjadi array NumPy.
S4	warped = four_point_transform(Image, pts)	Melakukan transformasi perspektif berdasarkan koordinat manual.
S5	print(" [PREPROCESSING] Mode auto detect")	Menampilkan log inisialisasi mode auto detect.
S6	corners = detect_document_corners(Image)	Mendeteksi koordinat sudut dokumen secara otomatis.
S7	if corners is None:	Memeriksa apakah deteksi sudut dokumen gagal. (Percabangan)
S8	print(" [PREPROCESSING] Auto detect gagal...")	Menampilkan log kegagalan deteksi otomatis.
S9	warped = Image	Menggunakan gambar asli tanpa proses crop.
S10	print(" [PREPROCESSING] Auto detect sukses")	Menampilkan log keberhasilan deteksi otomatis.
S11	warped = four_point_transform(Image, corners)	Melakukan transformasi perspektif berdasarkan koordinat otomatis.
S12	img_color, img_OCR = enhance_for_pdf(warped)	Memanggil fungsi <i>enhance_for_pdf</i> untuk meningkatkan kualitas citra.
S13	h, w = img.shape[:2]	Mengambil ukuran dimensi gambar.
S14	if w < 1500:	Memeriksa apakah lebar gambar kurang dari 1500 piksel. (Percabangan)

<i>Statement</i>	<i>Source code</i>	<i>Keterangan</i>
S15	<code>scale = 1500 / w</code>	Menghitung rasio skala perbesaran gambar.
S16	<code>img = cv2.resize(img, None, fx=scale, fy=scale, interpolation=cv2.INTER_LANCZOS4)</code>	Memperbesar resolusi gambar menggunakan interpolasi Lanczos.
S17	<code>gaussian = cv2.GaussianBlur(img, (0, 0), 2)</code>	Mengurangi <i>noise</i> menggunakan Gaussian Blur.
S18	<code>img_color = cv2.addWeighted(img, 1.3, gaussian, -0.3, 0)</code>	Meningkatkan ketajaman teks pada gambar.
S19	<code>img_OCR = cv2.cvtColor(img_color, cv2.COLOR_BGR2GRAY)</code>	Mengubah gambar menjadi grayscale.
S20	<code>img_OCR = cv2.normalize(img_OCR, None, 0, 255, cv2.NORM_MINMAX)</code>	Menormalisasi kontras gambar OCR.
S21	<code>return img_color, img_OCR</code>	Mengembalikan hasil <i>Preprocessing</i> dokumen.

Sumber : data diolah oleh penulis (2026)



c. *Flowchart Fungsi Preprocessing Dokumen*



Sumber : data diolah oleh penulis (2026)

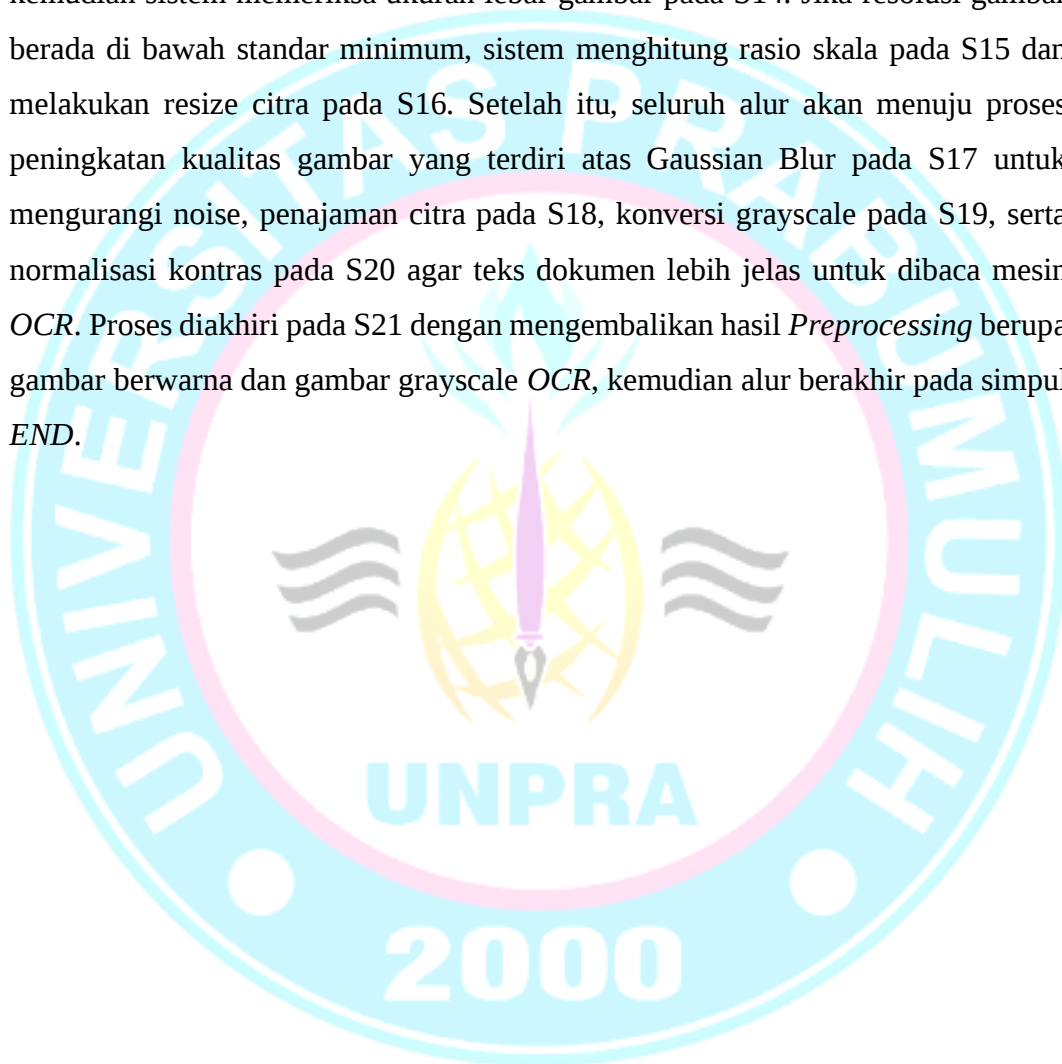
Gambar 5.28 *Flowchart Fungsi Preprocessing Dokumen*

Flowchart pada fungsi *Preprocessing* dokumen ini menggambarkan alur pengolahan citra sebelum proses *OCR* dilakukan. Proses dimulai dari pengecekan keberadaan koordinat manual pada S1. Apabila koordinat manual tersedia dan valid, sistem akan menjalankan mode manual crop melalui S2 dan S3, kemudian melakukan transformasi perspektif dokumen menggunakan fungsi four point transform pada S4. Sebaliknya, apabila koordinat manual tidak tersedia, sistem akan menjalankan mode auto detect melalui S5 dan S6 untuk mendeteksi sudut dokumen secara otomatis menggunakan metode contour detection OpenCV.

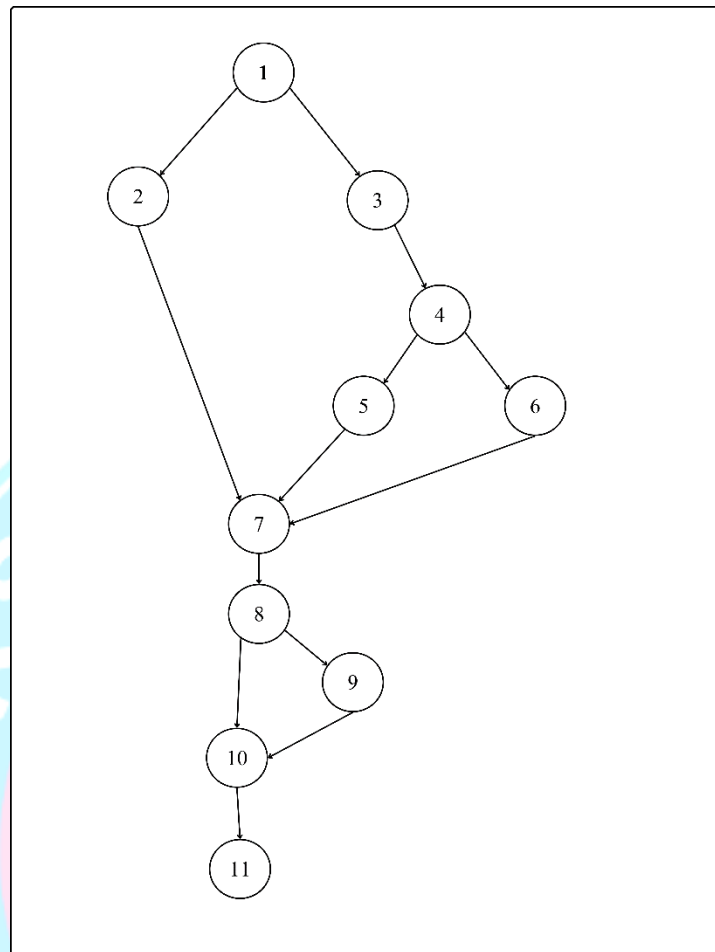
Selanjutnya, hasil deteksi otomatis diperiksa pada S7. Jika deteksi sudut dokumen gagal, sistem menampilkan log kegagalan pada S8 dan menggunakan

gambar asli tanpa pemotongan pada S9. Namun apabila deteksi berhasil, sistem akan menampilkan log keberhasilan pada S10 dan melakukan transformasi perspektif otomatis pada S11. Seluruh jalur tersebut kemudian bermuara pada proses enhancement citra di S12 untuk meningkatkan kualitas gambar dokumen sebelum OCR dilakukan.

Tahap berikutnya dimulai dari pengambilan dimensi gambar pada S13, kemudian sistem memeriksa ukuran lebar gambar pada S14. Jika resolusi gambar berada di bawah standar minimum, sistem menghitung rasio skala pada S15 dan melakukan resize citra pada S16. Setelah itu, seluruh alur akan menuju proses peningkatan kualitas gambar yang terdiri atas Gaussian Blur pada S17 untuk mengurangi noise, penajaman citra pada S18, konversi grayscale pada S19, serta normalisasi kontras pada S20 agar teks dokumen lebih jelas untuk dibaca mesin OCR. Proses diakhiri pada S21 dengan mengembalikan hasil *Preprocessing* berupa gambar berwarna dan gambar grayscale OCR, kemudian alur berakhir pada simpul *END*.



d. *Flowgraph Fungsi Preprocessing Dokumen*



Sumber : data diolah oleh penulis (2026)

Gambar 5.29 *Flowgraph Fungsi Preprocessing Dokumen*

Flowgraph pada fungsi *Preprocessing* dokumen ini merepresentasikan alur eksekusi program berdasarkan hasil pengelompokan *statement* yang telah diidentifikasi sebelumnya. *Flowgraph* yang dibentuk terdiri dari 11 *node* dan 12 *edge*, dengan penambahan *End Node* sebagai titik terminal untuk menandai berakhirnya seluruh alur eksekusi program. *Node* pertama (*Node 1*) merupakan titik keputusan yang memeriksa ketersediaan koordinat manual (*S1*). Apabila koordinat tersedia, alur berlanjut ke *Node 2* (*S2–S4*) untuk melakukan pemotongan dokumen secara manual. Sebaliknya, apabila tidak tersedia, alur diarahkan ke *Node 3* (*S5–S6*) untuk memulai proses deteksi sudut dokumen secara otomatis.

Selanjutnya, *Node 4* merepresentasikan *statement S7* yang menjadi titik keputusan untuk memverifikasi keberhasilan algoritma deteksi otomatis. Apabila

deteksi gagal, alur akan menuju *Node* 5 (S8–S9) dengan menggunakan gambar utuh (*full Image*), sedangkan apabila deteksi berhasil, alur dilanjutkan ke *Node* 6 (S10–S11) untuk melakukan transformasi perspektif secara otomatis. Ketiga jalur tersebut kemudian konvergen menuju *Node* 7 (S12–S13) untuk memanggil fungsi *enhance_for_pdf* serta mengekstraksi dimensi citra.

Pada tahap berikutnya, alur menuju *Node* 8 (S14) sebagai titik keputusan untuk validasi resolusi lebar gambar. Jika lebar gambar kurang dari 1500 piksel, alur berlanjut ke *Node* 9 (S15–S16) untuk melakukan perbesaran resolusi. Namun, jika resolusi telah memenuhi syarat, alur akan melompati proses tersebut dan langsung menuju *Node* 10 (S17) untuk penerapan Gaussian Blur. Akhirnya, seluruh proses peningkatan kualitas citra diselesaikan pada *Node* 11 (S18–S21), yang sekaligus menjadi terminal *node* (*end process*) dari fungsi *Preprocessing* ini.

e. *Cyclomatic Complexity* Fungsi *Preprocessing* Dokumen

Berdasarkan *Flowgraph* yang telah dibentuk, terdapat 3 *predicate node* (\$P\$), yaitu pada *Node* 1 yang melakukan validasi pemilihan mode pemotongan dokumen, *Node* 4 yang melakukan validasi hasil deteksi sudut dokumen otomatis, dan *Node* 8 yang melakukan validasi resolusi lebar gambar. Oleh karena itu, nilai *Cyclomatic Complexity* dapat dihitung menggunakan rumus berikut :

$$V(G) = P + 1$$

$$V(G) = 3 + 1$$

$$V(G) = 4$$

Dari perhitungan tersebut, diperoleh nilai *Cyclomatic Complexity* sebesar 4. Nilai tersebut menunjukkan bahwa terdapat empat jalur independen (*independent path*) pada proses eksekusi fungsi *Preprocessing*. Setiap jalur independen merepresentasikan kemungkinan alur logika yang berbeda berdasarkan kondisi percabangan yang terjadi selama proses pemotongan dan peningkatan kualitas citra. Nilai *Cyclomatic Complexity* ini kemudian digunakan sebagai dasar dalam menentukan jumlah jalur independen yang akan diuji pada tahap *white-box testing*, sehingga seluruh kemungkinan alur eksekusi program dapat diverifikasi untuk memastikan fungsi *Preprocessing* berjalan dengan benar sesuai kondisi yang diharapkan.

f. *Independen path* Fungsi *Preprocessing* Dokumen

Berdasarkan hasil perhitungan *Cyclomatic Complexity* yang diperoleh dari *Flowgraph* fungsi *Preprocessing*, nilai tersebut digunakan sebagai dasar untuk menentukan jumlah jalur independen yang harus diuji dalam proses white-box *testing*. Dengan demikian, didapatkan 4 jalur independen (*independent path*) berikut ini:

1. Path 1 (Mode Manual - Resolusi Cukup):

1 → 2 → 7 → 8 → 10 → 11

(Pengguna memilih mode manual dengan menyediakan 4 titik koordinat, dan resolusi gambar dokumen sudah memenuhi standar ≥ 1500 px, sehingga proses resize tidak diperlukan.)

2. Path 2 (Mode Manual - Resolusi Perlu *Resize*):

1 → 2 → 7 → 8 → 9 → 10 → 11

(Pengguna memilih mode manual, namun resolusi gambar dokumen di bawah 1500 px sehingga sistem melakukan proses perbesaran resolusi menggunakan interpolasi Lanczos.)

3. Path 3 (Mode Auto - Deteksi Gagal - Resolusi Cukup):

1 → 3 → 4 → 5 → 7 → 8 → 10 → 11

(Sistem mencoba deteksi sudut otomatis namun gagal, sehingga menggunakan full *Image* sebagai alternatif. Resolusi gambar asli sudah mencukupi standar sehingga tidak diperlukan resize.)

4. Path 4 (Mode Auto - Deteksi Sukses - Resolusi Perlu *Resize*):

1 → 3 → 4 → 6 → 7 → 8 → 9 → 10 → 11

(Deteksi sudut otomatis berhasil menemukan koordinat dokumen, namun resolusi dokumen rendah, sehingga sistem melakukan pemotongan otomatis dilanjutkan dengan perbesaran resolusi sebelum diproses ke tahap enhancement.)

g. *Test case Fungsi Preprocessing Dokumen*Tabel 5. 4 *Test case Fungsi Preprocessing Dokumen*

<i>Test case</i>	<i>Jalur Eksekusi</i>	<i>Kondisi Input</i>	<i>Hasil yang Diharapkan (Expected Result)</i>
TC-01	1 → 2 → 7 → 8 → 10 → 11	Manual, $W \geq 1500\text{px}$	Gambar dipotong manual, tanpa <i>resize</i> , kualitas meningkat.
TC-02	1 → 2 → 7 → 8 → 9 → 10 → 11	Manual, $W < 1500\text{px}$	Gambar dipotong manual, dilakukan <i>resize</i> (interpolasi), kualitas meningkat.
TC-03	1 → 3 → 4 → 5 → 7 → 8 → 10 → 11	Auto Gagal, $W \geq 1500\text{px}$	Deteksi gagal, memakai <i>full Image</i> , tanpa <i>resize</i> , kualitas meningkat.
TC-04	1 → 3 → 4 → 6 → 7 → 8 → 9 → 10 → 11	Auto Sukses, $W < 1500\text{px}$	Deteksi sukses, dipotong otomatis, dilakukan <i>resize</i> , kualitas meningkat.

Sumber : data diolah oleh penulis (2026)

Tabel di atas menyajikan rancangan pengujian white-box yang dilakukan terhadap fungsi *Preprocessing* untuk memverifikasi kebenaran logika alur program berdasarkan empat jalur independen yang telah diidentifikasi sebelumnya. Setiap *test case* dirancang dengan skenario input yang spesifik untuk mencakup seluruh percabangan logika, mulai dari pemilihan mode pemotongan dokumen (manual atau otomatis), penanganan kegagalan deteksi sudut dokumen, hingga validasi resolusi gambar yang menentukan perlu atau tidaknya proses *resize*. Melalui pengujian ini, diharapkan setiap jalur eksekusi dapat dipastikan berjalan sesuai dengan hasil yang diharapkan (*expected result*), di mana setiap proses transformasi citra baik melalui pemotongan koordinat maupun perbaikan resolusi mampu menghasilkan kualitas citra akhir yang optimal dan siap untuk diproses ke tahap pengenalan karakter (OCR).

h. *Statement Coverage Fungsi Preprocessing Dokumen*

Statement Coverage digunakan untuk mengukur sejauh mana seluruh *statement* pada fungsi *Preprocessing* telah dieksekusi melalui *test case* yang telah dirancang. Pengujian ini bertujuan untuk memastikan bahwa setiap *statement* pada program telah dijalankan minimal satu kali sehingga seluruh logika program dapat

diverifikasi dengan baik. *Statement* yang Tereksekusi pada masing-masing *Test case*:

1. TC-01 (Path 1): S1, S2, S3, S4, S12, S13, S14, S17, S18, S19, S20, S21
2. TC-02 (Path 2): S1, S2, S3, S4, S12, S13, S14, S15, S16, S17, S18, S19, S20, S21
3. TC-03 (Path 3): S1, S5, S6, S7, S8, S9, S12, S13, S14, S17, S18, S19, S20, S21
4. TC-04 (Path 4): S1, S5, S6, S7, S10, S11, S12, S13, S14, S15, S16, S17, S18, S19, S20, S21

Berdasarkan hasil pengujian yang telah dilakukan, seluruh *statement* pada fungsi *Preprocessing* yaitu S1 hingga S21 berhasil dieksekusi melalui kombinasi *test case* TC-01, TC-02, TC-03, dan TC-04. Dengan demikian diperoleh nilai *statement coverage* sebesar:

$$(21 / 21) \times 100\% = 100\%$$

Nilai tersebut menunjukkan bahwa seluruh *statement* pada fungsi *Preprocessing* telah diuji dan seluruh kemungkinan alur eksekusi program berhasil dicakup dalam proses pengujian *white-box*.

5.4.4 Pengujian Fungsi Ekstraksi Teks dengan OCR

a. *Source code* Fungsi Ekstraksi Teks dengan OCR

Source code tersebut berfungsi untuk melakukan *Preprocessing* gambar dan ekstraksi teks menggunakan OCR Tesseract. Proses *Preprocessing* dilakukan dengan mengubah gambar menjadi grayscale dan menerapkan adaptive threshold agar teks lebih mudah dikenali, kemudian sistem mengekstrak teks dari gambar menggunakan konfigurasi OCR khusus untuk meningkatkan akurasi hasil pembacaan.

```
def preprocess_for_tesseract(Image):
    if len(Image.shape) == 3:
        Image = cv2.cvtColor(Image, cv2.COLOR_BGR2GRAY)
    Image = cv2.adaptiveThreshold(
        Image, 255,
```

```

cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
cv2.THRESH_BINARY,
21, 10)

return Image

def extract_text(Image):
    Image = preprocess_for_tesseract(Image)
    pil_Image = Image.fromarray(Image)

    custom_config = r'--oem 3 --psm 6 -l ind+eng --dpi 300 -c
    tessedit_char_blacklist=~`!@#%&*_+=+[]{}|;:"<>'

    text = pytesseract.Image_to_string(pil_Image, config=custom_config)
    return text

```

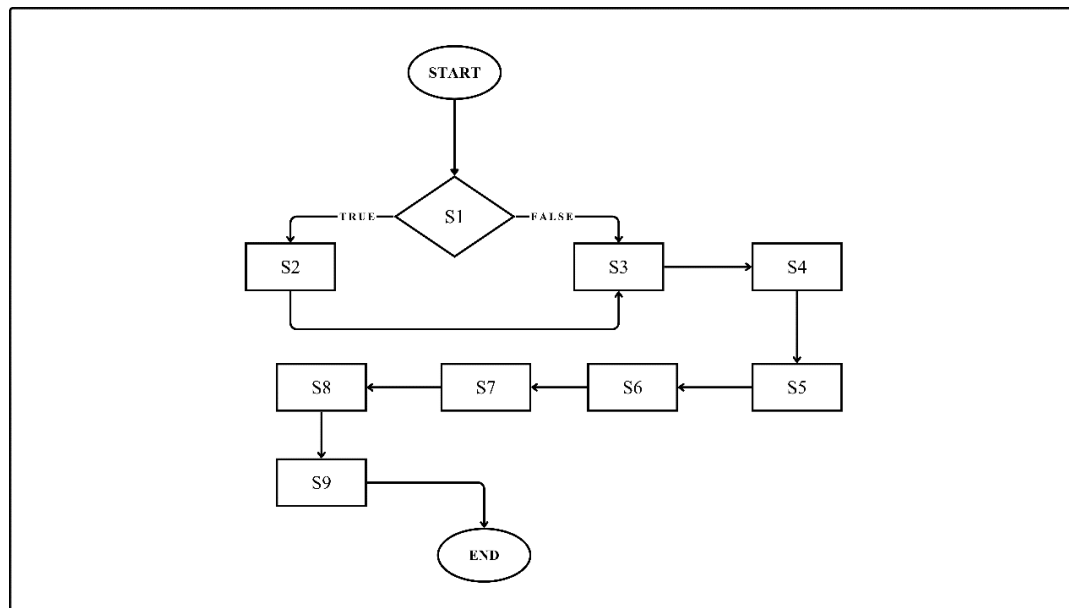
b. Identifikasi *Statement* Fungsi Ekstraksi Teks dengan OCR

Tabel 5. 6 Identifikasi *Statement* Fungsi Ekstraksi Teks dengan OCR

<i>Statement</i>	<i>Source code</i>	<i>Keterangan</i>
S1	<code>if len(Image.shape) == 3:</code>	Pengecekan dimensi kanal citra (RGB/BGR).
S2	<code>Image = cv2.cvtColor(Image, cv2.COLOR_BGR2GRAY)</code>	Konversi citra ke <i>grayscale</i> .
S3	<code>Image = cv2.adaptiveThreshold(...)</code>	Penerapan <i>adaptive thresholding</i> .
S4	<code>return Image</code>	Mengembalikan citra hasil prapemrosesan.
S5	<code>Image = preprocess_for_tesseract(Image)</code>	Pemanggilan fungsi prapemrosesan.
S6	<code>pil_Image = Image.fromarray(Image)</code>	Konversi format OpenCV ke PIL <i>Image</i> .
S7	<code>custom_config = r'--oem 3 --psm 6 -l ind+eng --dpi 300 -c tessedit_char_blacklist=~`!@#%&*_+=+[]{} ;:"<>'</code>	Definisi konfigurasi Tesseract.
S8	<code>text = pytesseract.Image_to_string(pil_Image, config=custom_config)</code>	Eksekusi mesin OCR Tesseract.
S9	<code>return text</code>	Mengembalikan hasil teks OCR.

Sumber : data diolah oleh penulis (2026)

c. *Flowchart* Fungsi Ekstraksi Teks dengan OCR



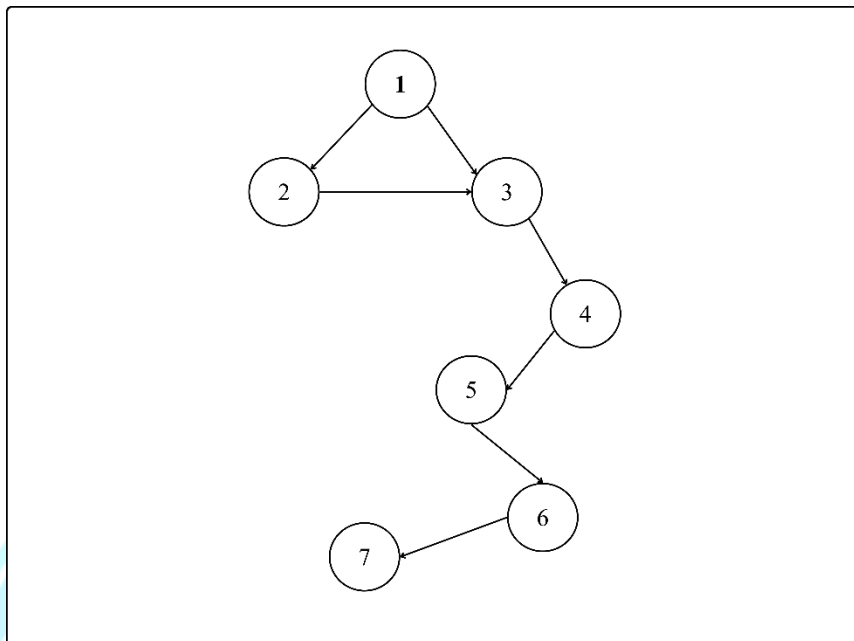
Sumber : data diolah oleh penulis (2026)

Gambar 5.30 *Flowchart* Fungsi Ekstraksi Teks dengan OCR

Flowchart pada fungsi Ekstraksi Teks dengan OCR merepresentasikan alur prosedural sistem dalam melakukan ekstraksi teks dari citra dokumen. Proses diawali dari titik *START* yang langsung diarahkan menuju titik keputusan (*decision node*) pada S1 untuk memvalidasi dimensi kanal citra. Jika hasil validasi bernilai *true*, alur diarahkan menuju S2 untuk melakukan konversi warna menjadi grayscale sebelum melanjutkan ke proses berikutnya. Sebaliknya, jika hasil validasi bernilai *false*, alur akan langsung melewati proses konversi tersebut menuju S3 untuk tahapan *adaptive thresholding*.

Setelah proses prapemrosesan selesai, alur berlanjut secara linear melalui S4 (*return Image*) hingga dipanggil oleh fungsi utama pada S5. Tahapan berikutnya mencakup konversi format citra ke *Pillow Image* (S6), penentuan konfigurasi Tesseract (S7), serta eksekusi mesin OCR untuk membaca teks (S8). Rangkaian proses ini diakhiri dengan pengembalian nilai teks hasil ekstraksi pada S9, yang kemudian membawa sistem menuju titik akhir (*END*). Alur yang terstruktur ini memastikan bahwa setiap tahapan transformasi citra telah divalidasi dan diproses sesuai dengan kebutuhan format input yang didukung oleh mesin OCR.

d. *Flowgraph* Fungsi Ekstraksi Teks dengan OCR



Sumber : data diolah oleh penulis (2026)

Gambar 5.31 *Flowgraph* Fungsi Ekstraksi Teks dengan OCR

Flowgraph ini merepresentasikan alur logika fungsi ekstraksi OCR dengan total 7 node dan 7 edge. Node 1 bertindak sebagai *predicate node* yang mengatur dua jalur eksekusi independen. Jalur pertama (Node 1, Node 2, Node 3) menguji efektivitas sistem dalam menangani input berwarna, sedangkan jalur kedua (Node 1, Node 3) menguji efisiensi sistem pada input yang sudah dalam format grayscale. Setelah melewati Node 3, alur eksekusi menjadi linear (*sequential*) hingga mencapai Node 7 sebagai terminal node. Struktur ini memberikan gambaran yang jelas bagi penguji bahwa seluruh baris kode (S1 hingga S9) tercakup dalam alur pengujian yang sistematis dan tidak terdapat dead code dalam sistem.

e. *Cyclomatic Complexity* Fungsi Ekstraksi Teks dengan OCR

Berdasarkan *Flowgraph* yang telah dibentuk, terdapat 1 *predicate node* (P), yaitu pada Node 1 yang melakukan pengecekan dimensi kanal citra (*Image channel*) untuk menentukan apakah diperlukan proses konversi warna atau tidak. Oleh karena itu, nilai *Cyclomatic Complexity* dapat dihitung menggunakan dua rumus berikut:

$$V(G) = P + 1$$

$$V(G) = 1 + 1$$

$$V(G) = 2$$

Dari perhitungan tersebut, diperoleh nilai 2, di mana nilai *Cyclomatic Complexity* tersebut menunjukkan bahwa terdapat dua jalur independen dalam proses eksekusi fungsi ekstraksi OCR, yang merepresentasikan seluruh kemungkinan alur logika berdasarkan kondisi yang terjadi pada program. Nilai ini kemudian digunakan sebagai dasar untuk menentukan jumlah *independent path* yang akan diuji pada tahap pengujian berikutnya, sehingga setiap jalur eksekusi dapat diverifikasi untuk memastikan seluruh kondisi pada fungsi ekstraksi OCR telah ditangani dengan benar.

f. *Independent path* Fungsi Ekstraksi Teks dengan OCR

Berdasarkan hasil perhitungan *Cyclomatic Complexity* yang diperoleh dari *Flowgraph* fungsi ekstraksi OCR, nilai tersebut digunakan sebagai dasar untuk menentukan jumlah jalur independen yang harus diuji dalam proses *white box testing*. Dengan demikian, didapatkan 2 jalur independen (*independent path*) berikut ini:

1. Path 1 (Input Berwarna):

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7$$

(Sistem mendeteksi citra input memiliki 3 kanal warna (input BGR), sehingga sistem melakukan proses konversi warna ke grayscale di *Node* 2 sebelum melanjutkan ke tahap *adaptive thresholding* di *Node* 3 hingga proses OCR selesai.)

2. Path 2 (Input *Grayscale*):

$$1 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7$$

(Sistem mendeteksi citra input sudah dalam format satu kanal (grayscale), sehingga proses konversi warna di *Node* 2 dilewati dan sistem langsung melakukan *adaptive thresholding* di *Node* 3 sebelum melanjutkan proses hingga akhir.)

g. *Test case* Fungsi Ekstraksi Teks dengan OCRTabel 5. 7 *Test case* Fungsi Ekstraksi Teks dengan OCR

<i>Test case</i>	Skenario Pengujian	Input Data	Jalur Eksekusi	Hasil yang Diharapkan
TC-01	Pengujian dengan citra berwarna (BGR)	Gambar format .jpg (3-channel)	1 → 2 → 3 → 4 → 5 → 6 → 7	Teks berhasil diekstraksi dengan benar.
TC-02	Pengujian dengan citra <i>grayscale</i>	Gambar format .jpg (1-channel)	1 → 3 → 4 → 5 → 6 → 7	Teks berhasil diekstraksi dengan benar.

Sumber : data diolah oleh penulis (2026)

Tabel di atas menyajikan dua *test case* utama yang dirancang untuk memverifikasi seluruh alur logika pada fungsi `extract_text`. TC-01 bertujuan untuk menguji kebenaran alur pada kondisi citra berwarna (3 kanal), di mana sistem harus melalui tahap konversi warna terlebih dahulu sebelum melakukan prapemrosesan (*thresholding*). Sebaliknya, TC-02 menguji efisiensi alur pada kondisi citra yang sudah dalam format satu kanal (*grayscale*). Dengan menjalankan kedua skenario ini, seluruh *statement* (S1-S9) dipastikan telah dieksekusi setidaknya satu kali, sehingga pengujian *Statement Coverage* mencapai nilai 100%

h. *Statement Coverage* Fungsi Ekstraksi Teks dengan OCR

Statement Coverage digunakan untuk mengukur sejauh mana seluruh *statement* pada fungsi ekstraksi OCR telah dieksekusi melalui *test case* yang telah dirancang. Pengujian ini bertujuan untuk memastikan bahwa setiap *statement* pada program telah dijalankan minimal satu kali sehingga seluruh logika program dapat diverifikasi dengan baik. *Statement* yang Tereksekusi :

1. TC1:
S1, S2, S3, S4, S5, S6, S7, S8, S9
2. TC2:
S1, S3, S4, S5, S6, S7, S8, S9

Berdasarkan hasil pengujian yang telah dilakukan, seluruh *statement* pada fungsi ekstraksi OCR yaitu S1 hingga S9 berhasil dieksekusi melalui kombinasi *test case* TC1 dan TC2. Dengan demikian diperoleh nilai *statement coverage* sebesar :

$$(9 / 9) \times 100\% = 100\%$$

Nilai tersebut menunjukkan bahwa seluruh *statement* pada fungsi ekstraksi OCR telah diuji dan seluruh kemungkinan alur eksekusi program berhasil dicakup dalam proses pengujian white-box.

5.4.5 Pengujian Fungsi *Rule-Based System*

a. *Source code* Fungsi *Rule-Based System*

Source code fungsi *Rule-Based System* ini berfungsi untuk mengekstrak metadata surat secara otomatis dari hasil teks OCR menggunakan pendekatan *Rule-Based System*. Sistem melakukan identifikasi informasi penting seperti asal surat, tujuan surat, nomor surat, tanggal surat, dan perihal berdasarkan pola kata kunci, format penulisan, serta aturan tertentu sesuai jenis surat masuk atau surat keluar.

```
def extract_metadata(raw_text: str, jenis_surat: Optional[str] = None) -> Dict[str, str]:
```

```
    lines = normalize_text(raw_text)
```

```
    data = {"asal_surat": "",
```

```
           "tujuan_surat": "",
```

```
           "nomor_surat": "",
```

```
           "tanggal_surat": "",
```

```
           "perihal": "",}
```

```
    if not lines:
```

```
        return data
```

```
    if jenis_surat == "masuk":
```

```
        for line in lines[:10]:
```

```
            upper = line.upper()
```

```
            if any(x in upper for x in [
```

```
                "FAKULTAS", "PROGRAM", "UNIVERSITAS",
```

```
                "DINAS", "PEMERINTAH", "SEKOLAH", "LEMBAGA"]):
```

```
                data["asal_surat"] = line
```

```
                break
```

```
    if not data["asal_surat"]:
```

```
        data["asal_surat"] = lines[0]
```

```

if jenis_surat == "keluar":

    for i, line in enumerate(lines):
        if re.search(r'\b(Kepada|Yth\.|?)|Yang\s+Terhormat|To)\b', line,
re.IGNORECASE):

            isi = re.sub(
                r'\b(Kepada\s*(Yth\.|?)|Yth\.|Yang\s+Terhormat|To)\s*[:\~]?s*',
                "", line, flags=re.IGNORECASE).strip()

            if len(isi) < 3 and i + 1 < len(lines):
                isi = lines[i + 1]
                if re.search(r'\b(Yth\.|?)\b', isi, re.IGNORECASE):
                    isi = re.sub(
                        r'\b(Yth\.|?)\s*[:\~]?s*', "",
                        isi, flags=re.IGNORECASE
                    ).strip()
                    if len(isi) < 3 and i + 2 < len(lines):
                        isi = lines[i + 2]
                    data["tujuan_surat"] = isi.strip(" :-")
                    tujuan_list = [ ]
                    if data["tujuan_surat"]:
                        tujuan_list.append(data["tujuan_surat"])

            start = i + 1
            for j in range(start, min(start + 10, len(lines))):
                baris = lines[j].strip()

                if re.search(
                    r'\b(Nomor|Perihal|Hal|Lampiran|Dari|Tanggal|Sifat)\b',
                    baris, re.IGNORECASE):
                    break

```

```

if len(baris) == 0 and len(tujuan_list) > 1:
    break

match_nomor = re.match(r'^\d+[\.\,)]s*(.+)', baris)
if match_nomor:
    kandidat = match_nomor.group(1).strip(" :-")
    if kandidat and kandidat not in tujuan_list:
        tujuan_list.append(kandidat)
    continue

match_dash = re.match(r'^[-\,)]s*(.+)', baris)
if match_dash:
    kandidat = match_dash.group(1).strip(" :-")
    if kandidat and kandidat not in tujuan_list:
        tujuan_list.append(kandidat)
    continue

if len(tujuan_list) > 1:
    data["tujuan_surat"] = " | ".join(tujuan_list)
    break

if not data["tujuan_surat"]:
    for i, line in enumerate(lines):
        if re.search(r'\bKepada\b', line, re.IGNORECASE):
            for j in range(i + 1, min(i + 5, len(lines))):
                candidate = lines[j].strip()
                upper = candidate.upper()
                if any(x in upper for x in [
                    "FAKULTAS", "PROGRAM", "UNIVERSITAS",
                    "DINAS", "PEMERINTAH", "BAPAK", "IBU",
                    "DIREKTUR", "REKTOR", "KEPALA", "KETUA"]):
                    data["tujuan_surat"] = candidate

```

```

        break
    break
pola_keyword = r'(:Nomor|No|Nornor|Nomar|N0|NO)'
for line in lines:
    line_clean = line.strip()
    if any(x in line_clean.lower() for x in ["telp", "fax", "kode pos"]):
        continue

    match = re.search(
        rf'{pola_keyword}\s*[:]?s*([A-Z0-9./-]{5,})',
        line_clean, re.IGNORECASE)
    if match:
        kandidat = match.group(1).strip()
        if any(sym in kandidat for sym in ["/", "-", "."]):
            data["nomor_surat"] = kandidat.strip(" :-")
            break

    if not data["nomor_surat"]:
        for line in lines:
            match = re.search(
                r'([A-Z0-9]{1,10}[./-][A-Z0-9./-]{5,})',
                line, re.IGNORECASE)
            if match:
                kandidat = match.group(1).strip()
                if any(sym in kandidat for sym in ["/", "-", "."]):
                    data["nomor_surat"] = kandidat.strip(" :-")
                    break

bulan_map = {"januari": "01", "februari": "02", "maret": "03",
             "april": "04", "mei": "05", "juni": "06",
             "juli": "07", "agustus": "08", "september": "09",
             "oktober": "10", "november": "11", "desember": "12"}

```

```

for line in lines:
    for bulan in bulan_map:
        if bulan in line.lower():
            match = re.search(
                r'(\d{1,2})\s+' + bulan + r'\s+(\d{4})',
                line, re.IGNORECASE)
            if match:
                hari = match.group(1).zfill(2)
                tahun = match.group(2)
                data["tanggal_surat"] = f"{tahun}-{bulan_map[bulan]}-{hari}"
                break
for i, line in enumerate(lines):
    if re.search(r'(Perihal|Hal)', line, re.IGNORECASE):
        isi = re.sub(
            r'(Perihal|Hal)\s*[:\-\]?\s*', "",
            line, flags=re.IGNORECASE)
        if len(isi) < 5 and i + 1 < len(lines):
            isi += " " + lines[i + 1]
        data["perihal"] = isi.strip()
        break
return data

```

b. Identifikasi *Statement Fungsi Rule-Based System*

Dari *source code* fungsi *Rule-Based System* tersebut, diidentifikasi sebanyak 25 *statement* utama yang merepresentasikan seluruh titik percabangan, validasi, dan jalur kritikal pada proses ekstraksi metadata surat. *Statement* yang dipilih difokuskan pada bagian logika utama seperti proses *filtering*, pencocokan regex, *fallback extraction*, validasi data, serta pengembalian hasil ekstraksi. Dengan melakukan pengujian terhadap 25 *statement* utama tersebut, maka baris-baris program yang bersifat sekuensial dan berada di antara jalur percabangan akan ikut tereksekusi secara otomatis, sehingga keseluruhan alur logika fungsi tetap dapat

direpresentasikan dengan baik tanpa harus menguji seluruh baris kode secara terpisah.

Tabel 5. 8 Identifikasi *Statement* Fungsi *Rule-Based System*

<i>Statement</i>	<i>Source code</i>	<i>Keterangan</i>
S1	lines = normalize_text(raw_text)	Eksekusi normalisasi teks input menghasilkan output, bukan percabangan
S2	if not lines: return data	Percabangan cek apakah teks kosong, menentukan alur selanjutnya
S3	if jenis_surat == "masuk":	Percabangan filter berdasarkan jenis surat
S4	if any(x in upper for x in [...]):	Percabangan evaluasi kecocokan keyword instansi/kop surat
S5	if not data["asal_surat"]:	Percabangan cek apakah asal_surat kosong (pemicu <i>fallback</i>)
S6	data["asal_surat"] = lines[0]	Eksekusi assignment nilai <i>fallback</i> ke variabel data
S7	if jenis_surat == "keluar":	Percabangan filter berdasarkan jenis surat
S8	if re.search(r'\b(Kepada Yth\. Yang\ s+Terhormat To)\b', line, re.IGNORECASE)	Percabangan cek keberadaan kata tujuan surat
S9	if len(isi) < 3 and i + 1 < len(lines):	Percabangan <i>fallback</i> jika isi terlalu pendek, ambil baris +1
S10	if re.search(r'\b(Yth\. ?)\b', isi, re.IGNORECASE):	Percabangan cek apakah baris <i>fallback</i> +1 masih mengandung "Yth"
S11	if len(isi) < 3 and i + 2 < len(lines):	Percabangan <i>fallback</i> kedua jika isi masih pendek, ambil baris +2
S12	if re.search(r'\b(Nomor Perihal Hal)\b', isi, ...)	Percabangan filter agar isi tujuan bukan bagian metadata surat
S13	if len(baris) == 0 and len(tujuan_list) > 1:	Percabangan <i>stop condition loop</i> saat baris kosong setelah multi-tujuan
S14	if match_nomor:	Percabangan cek kecocokan format penerima bernomor (1., 2.)
S15	if match_dash:	Percabangan cek kecocokan format penerima tanda hubung (-)
S16	if len(tujuan_list) > 1:	Percabangan cek apakah ada lebih dari satu tujuan untuk digabung
S17	if not data["tujuan_surat"]:	Percabangan pemicu <i>fallback</i> global jika tujuan masih kosong
S18	if re.search(r'\bKepada\b', line, ...)	Percabangan pencarian agresif kata "Kepada" pada <i>fallback</i>
S19	if any(x in upper for x in ["ykh", "rektor", ...]):	Percabangan evaluasi keyword jabatan/instansi pada <i>fallback</i> tujuan

<i>Statement</i>	<i>Source code</i>	<i>Keterangan</i>
S20	if any(x in line_clean.lower() for x in ["telp", "fax", ...]):	Percabangan filter noise baris kontak agar tidak masuk pencarian nomor
S21	if match: (<i>pola nomor utama</i>)	Percabangan cek kecocokan regex nomor surat format utama
S22	if not data["nomor_surat"]:	Percabangan pemicu <i>fallback</i> pencarian nomor format generik
S23	if match: (<i>pola nomor generik</i>)	Percabangan cek kecocokan regex nomor surat format generik
S24	if bulan in line.lower():	Percabangan cek kecocokan nama bulan untuk ekstraksi tanggal
S25	if re.search(r'(Perihal Hal)', line, ...)	Percabangan cek keberadaan kata "Perihal" atau "Hal"
S26	return data	Eksekusi pengembalian hasil akhir bukan percabangan

Sumber : data diolah oleh penulis (2026)



data kosong atau tidak. Jika hasil validasi bernilai *true*, sistem langsung mengembalikan data kosong melalui S26 dan diarahkan menuju titik akhir (*END*). Sebaliknya, apabila hasil validasi bernilai *false*, alur proses dilanjutkan menuju tahap identifikasi jenis surat pada S3.

Pada S3, sistem melakukan percabangan berdasarkan jenis surat yang diproses. Jika jenis surat merupakan surat masuk, alur diarahkan menuju S4 untuk melakukan pencarian keyword instansi atau kop surat seperti “FAKULTAS”, “UNIVERSITAS”, “DINAS”, dan keyword lainnya. Jika keyword ditemukan, sistem menetapkan asal surat berdasarkan baris yang sesuai. Namun apabila asal surat belum ditemukan, proses dilanjutkan menuju S5 dan S6 sebagai mekanisme *fallback* dengan menggunakan baris pertama dokumen sebagai nilai asal surat. Setelah proses ekstraksi asal surat selesai, alur dilanjutkan menuju proses ekstraksi metadata surat keluar pada S7.

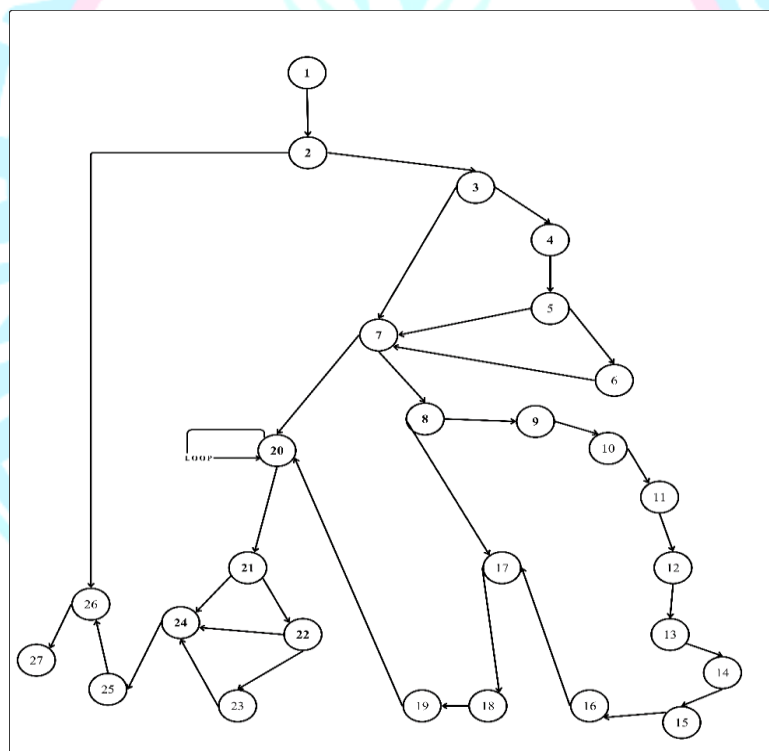
Pada S7, sistem memvalidasi apakah dokumen termasuk surat keluar. Jika validasi bernilai *true*, proses dilanjutkan menuju S8 untuk mendeteksi keyword tujuan surat seperti “Kepada”, “Yth”, dan “Yang Terhormat”. Setelah keyword ditemukan, sistem melakukan beberapa tahapan validasi *fallback* pada S9, S10, dan S11 apabila isi tujuan terlalu pendek atau masih mengandung kata formalitas surat. Selanjutnya, sistem melakukan filtering metadata surat pada S12 dan validasi kondisi penghentian iterasi pada S13 sebelum memproses format daftar tujuan surat pada S14 dan S15. Jika ditemukan lebih dari satu tujuan surat, sistem akan menggabungkan seluruh tujuan surat pada S16 sebelum melanjutkan proses berikutnya. Apabila tujuan surat belum berhasil ditemukan, sistem menjalankan mekanisme *fallback* tambahan melalui S17, S18, dan S19 untuk mencari tujuan surat berdasarkan keyword jabatan atau instansi tertentu.

Setelah proses ekstraksi tujuan surat selesai, alur berlanjut menuju proses ekstraksi nomor surat pada S20 hingga S23. Pada tahap ini, sistem terlebih dahulu melakukan filtering terhadap noise seperti “telp”, “fax”, dan “kode pos” sebelum memvalidasi pola nomor surat utama. Jika nomor surat utama tidak ditemukan, sistem akan menjalankan proses *fallback* menggunakan pola nomor generik. Selanjutnya, sistem melakukan proses ekstraksi tanggal surat pada S24 dengan mendeteksi nama bulan pada dokumen, kemudian dilanjutkan dengan proses

ekstraksi perihal surat pada S25 menggunakan keyword “Perihal” atau “Hal”. Seluruh rangkaian proses diakhiri dengan pengembalian data metadata pada S26 yang kemudian mengarahkan sistem menuju titik akhir (*END*).

Dalam penyusunannya, *flowchart* ini difokuskan pada blok-blok utama yang merepresentasikan alur proses sistem secara keseluruhan namun tetap sejalan dengan *source code* program. Beberapa validasi internal dan proses yang masih berada dalam satu konteks logika direpresentasikan dalam bentuk alur yang lebih ringkas untuk mempertahankan keterbacaan diagram tanpa menghilangkan representasi proses utama sistem. Dengan demikian, *flowchart* tetap mampu menggambarkan percabangan, proses iterasi, mekanisme *fallback*, serta tahapan ekstraksi metadata sesuai dengan implementasi program yang sebenarnya.

d. *Flowgraph* Fungsi Rule-Based System



Sumber : data diolah oleh penulis (2026)

Gambar 5.33 *Flowgraph* Fungsi Rule-Based System

Flowgraph ini merepresentasikan alur logika fungsi `extract_metadata()` dalam proses ekstraksi metadata surat berbasis OCR dengan total 27 *node* dan 34

edge. Alur dimulai dari proses normalisasi teks pada *Node 1* hingga validasi awal pada *Node 2* untuk memeriksa apakah hasil *OCR* kosong atau tidak. Jika data kosong, proses langsung diarahkan menuju *Node 26* dan berakhir pada terminal *node (END)*. Namun apabila data tersedia, alur dilanjutkan menuju proses identifikasi jenis surat pada *Node 3* yang memisahkan proses ekstraksi surat masuk dan surat keluar.

Pada jalur surat masuk, *Node 4* dan *Node 5* digunakan untuk melakukan evaluasi keyword instansi atau kop surat. Jika keyword ditemukan, sistem akan menetapkan nilai asal surat melalui *Node 6* sebelum melanjutkan proses berikutnya. Sedangkan pada jalur surat keluar, *Node 7* hingga *Node 19* merepresentasikan proses pencarian tujuan surat, validasi isi tujuan, *fallback* pengambilan baris berikutnya, serta penggabungan multi-tujuan surat. Beberapa proses validasi internal seperti pengecekan format tujuan bernomor, tanda hubung, dan penggabungan daftar tujuan tetap direpresentasikan dalam *Flowgraph* melalui *Node 14*, *Node 15*, dan *Node 16* agar tetap sinkron dengan *source code* dan *statement program*.

Selanjutnya, *Node 20* hingga *Node 24* menggambarkan proses ekstraksi nomor surat dan tanggal surat. Pada *Node 20* terdapat self-loop yang merepresentasikan proses iterasi filtering noise seperti “telp”, “fax”, dan “kode pos” sebelum sistem melanjutkan validasi pola nomor surat utama maupun *fallback* nomor generik. Setelah itu, *Node 24* dan *Node 25* merepresentasikan proses pencarian metadata tanggal dan perihal surat sebelum seluruh proses diarahkan menuju *Node 26* dan diakhiri pada terminal *node*.

Dalam pembentukan *Flowgraph* ini, beberapa percabangan internal dan validasi bertingkat disederhanakan menjadi alur linear untuk mempertahankan keterbacaan *Flowgraph* tanpa menghilangkan representasi logika utama program. Penyederhanaan dilakukan terutama pada rangkaian validasi internal yang masih berada dalam satu konteks proses ekstraksi tujuan surat, seperti validasi *fallback* isi tujuan, pengecekan format penerima, dan penggabungan daftar tujuan surat. Meskipun demikian, seluruh proses utama, percabangan penting, loop, serta mekanisme *fallback* tetap direpresentasikan sehingga *Flowgraph* tetap sesuai dengan *source code*, *statement program*, dan kebutuhan analisis *white box testing*.

e. *Cyclomatic Complexity Fungsi Rule-Based System*

Berdasarkan *flowgraph* yang telah dibentuk, terdapat 8 *predicate node* (P), yaitu pada *Node* 2 yang melakukan pengecekan apakah hasil normalisasi teks kosong, *Node* 3 yang melakukan validasi jenis surat masuk, *Node* 5 yang memvalidasi keberadaan data asal surat, *Node* 7 yang melakukan validasi jenis surat keluar, *Node* 8 yang melakukan pengecekan keyword tujuan surat, *Node* 20 yang melakukan filtering noise pada proses pencarian nomor surat, *Node* 21 yang melakukan validasi pola nomor surat utama, serta *Node* 22 yang melakukan pengecekan *fallback* nomor surat generik. Oleh karena itu, nilai *Cyclomatic Complexity* dapat dihitung menggunakan rumus berikut:

$$V(G) = P + 1$$

$$V(G) = 8 + 1$$

$$V(G) = 9$$

Berdasarkan hasil perhitungan tersebut diperoleh nilai *Cyclomatic Complexity* sebesar 9. Nilai tersebut menunjukkan bahwa terdapat sembilan jalur independen (*independent path*) dalam proses eksekusi fungsi *extract_metadata()*, yang merepresentasikan berbagai kemungkinan alur logika berdasarkan kondisi dan validasi yang terjadi pada program. Nilai *Cyclomatic Complexity* ini kemudian digunakan sebagai dasar dalam penentuan *independent path* pada tahap *white box testing*, sehingga setiap jalur eksekusi utama dapat diuji dan diverifikasi untuk memastikan seluruh proses ekstraksi metadata surat berjalan sesuai dengan logika program yang telah dirancang.

f. *Independen path Fungsi Rule-Based System*

Berdasarkan hasil perhitungan *Cyclomatic Complexity* yang diperoleh dari *flowgraph* fungsi *Rule-Based System*, nilai tersebut digunakan sebagai dasar untuk menentukan jumlah jalur independen (*independent path*) yang harus diuji pada proses *white box testing*. Setiap jalur independen merepresentasikan kemungkinan alur eksekusi program yang berbeda berdasarkan kondisi percabangan pada proses ekstraksi metadata surat, seperti validasi data kosong, identifikasi jenis surat, proses *fallback* metadata, pencarian nomor surat, serta proses ekstraksi informasi surat lainnya. Dengan demikian, seluruh kemungkinan alur utama pada fungsi

`extract_metadata()` dapat dianalisis dan diuji secara sistematis untuk memastikan logika program berjalan sesuai dengan implementasi yang dirancang. Berdasarkan *flowgraph* yang telah dibentuk, diperoleh 9 jalur independen (*independent path*) sebagai berikut:

1. Path 1:

$1 \rightarrow 2 \rightarrow 26 \rightarrow \text{END}$

(Hasil normalisasi teks kosong sehingga sistem langsung mengembalikan data kosong)

2. Path 2:

$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 20 \rightarrow 21 \rightarrow 24 \rightarrow 25 \rightarrow 26 \rightarrow \text{END}$

(Proses ekstraksi surat masuk berhasil menemukan *fallback* asal surat dan nomor surat utama)

3. Path 3:

$1 \rightarrow 2 \rightarrow 3 \rightarrow 7 \rightarrow 20 \rightarrow 21 \rightarrow 22 \rightarrow 23 \rightarrow 24 \rightarrow 25 \rightarrow 26 \rightarrow \text{END}$

(Proses surat bukan surat masuk dan nomor surat ditemukan melalui *fallback* nomor generik)

4. Path 4:

$1 \rightarrow 2 \rightarrow 3 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 15 \rightarrow 16 \rightarrow 17 \rightarrow 18 \rightarrow 19 \rightarrow 20 \rightarrow 21 \rightarrow 24 \rightarrow 25 \rightarrow 26 \rightarrow \text{END}$

(Proses surat keluar berhasil menemukan tujuan surat serta nomor surat utama)

5. Path 5:

$1 \rightarrow 2 \rightarrow 3 \rightarrow 7 \rightarrow 8 \rightarrow 17 \rightarrow 18 \rightarrow 19 \rightarrow 20 \rightarrow 21 \rightarrow 24 \rightarrow 25 \rightarrow 26 \rightarrow \text{END}$

(Proses *fallback* tujuan surat dijalankan karena keyword tujuan surat tidak ditemukan pada proses awal)

6. Path 6:

$1 \rightarrow 2 \rightarrow 3 \rightarrow 7 \rightarrow 20 \rightarrow 20 \rightarrow 21 \rightarrow 24 \rightarrow 25 \rightarrow 26 \rightarrow \text{END}$

(Proses filtering noise nomor surat menghasilkan iterasi ulang sebelum nomor surat utama ditemukan)

7. Path 7:

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 7 \rightarrow 20 \rightarrow 21 \rightarrow 22 \rightarrow 24 \rightarrow 25 \rightarrow 26 \rightarrow \text{END}$$

(Nomor surat utama tidak ditemukan dan *fallback* nomor generik juga tidak menghasilkan data valid)

8. Path 8:

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 7 \rightarrow 20 \rightarrow 21 \rightarrow 24 \rightarrow 25 \rightarrow 26 \rightarrow \text{END}$$

(Proses surat masuk berhasil menemukan asal surat tanpa menggunakan *fallback*)

9. Path 9:

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 15 \rightarrow 16 \rightarrow 17 \rightarrow 18 \rightarrow 19 \rightarrow 20 \rightarrow 21 \rightarrow 22 \rightarrow 23 \rightarrow 24 \rightarrow 25 \rightarrow 26 \rightarrow \text{END}$$

(Proses surat keluar berhasil menemukan tujuan surat dan nomor surat diperoleh melalui *fallback* nomor generik)

g. *Test case Fungsi Rule-Based System*

Pengujian dilakukan berdasarkan jalur independen (*independent path*) yang telah diperoleh dari hasil analisis *flowgraph* fungsi *Rule-Based System*. Setiap jalur independen direpresentasikan ke dalam beberapa *test case* dengan kombinasi kondisi input dokumen OCR yang berbeda untuk menguji seluruh kemungkinan alur eksekusi program. Jalur eksekusi pada masing-masing *test case* diperoleh dari *node-node* yang dilalui pada *flowgraph* sehingga setiap proses pengujian dapat memverifikasi proses ekstraksi metadata surat secara menyeluruh, mulai dari validasi data kosong, ekstraksi asal surat, tujuan surat, nomor surat, tanggal surat, hingga proses ekstraksi perihal surat.

Tabel 5. 9 *Test case Fungsi Rule-Based System*

<i>Test case</i>	Skenario Pengujian	Jalur Eksekusi	Output
TC1	Hasil OCR kosong	$1 \rightarrow 2 \rightarrow 26 \rightarrow \text{END}$	Sistem mengembalikan data metadata kosong
TC2	Surat masuk dengan keyword instansi dan nomor surat valid	$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 7 \rightarrow 20 \rightarrow 21 \rightarrow 24 \rightarrow 25 \rightarrow 26 \rightarrow \text{END}$	Metadata surat berhasil diekstraksi
TC3	Surat masuk tanpa keyword instansi sehingga menggunakan <i>fallback</i> asal surat	$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 20 \rightarrow 21 \rightarrow 24 \rightarrow 25 \rightarrow 26 \rightarrow \text{END}$	Asal surat berhasil diambil dari baris pertama dokumen

TC4	Surat keluar dengan tujuan surat berhasil ditemukan	1 → 2 → 3 → 7 → 8 → 9 → 10 → 11 → 12 → 13 → 14 → 15 → 16 → 17 → 18 → 19 → 20 → 21 → 24 → 25 → 26 → END	Tujuan surat berhasil diekstraksi
TC5	Surat keluar menggunakan <i>fallback</i> pencarian tujuan surat	1 → 2 → 3 → 7 → 8 → 17 → 18 → 19 → 20 → 21 → 24 → 25 → 26 → END	Tujuan surat berhasil diperoleh melalui <i>fallback</i>
TC6	Nomor surat utama tidak ditemukan sehingga menggunakan <i>fallback</i> nomor generik	1 → 2 → 3 → 7 → 20 → 21 → 22 → 23 → 24 → 25 → 26 → END	Nomor surat berhasil diperoleh dari pola generik
TC7	Terdapat noise seperti “telp” atau “fax” pada dokumen	1 → 2 → 3 → 7 → 20 → 20 → 21 → 24 → 25 → 26 → END	Noise berhasil difilter sebelum ekstraksi nomor surat
TC8	Nomor surat utama gagal dan <i>fallback</i> nomor generik juga gagal	1 → 2 → 3 → 7 → 20 → 21 → 22 → 24 → 25 → 26 → END	Sistem tidak berhasil menemukan nomor surat
TC9	Surat keluar dengan multi tujuan surat	1 → 2 → 3 → 7 → 8 → 9 → 10 → 11 → 12 → 13 → 14 → 15 → 16 → 17 → 18 → 19 → 20 → 21 → 22 → 23 → 24 → 25 → 26 → END	Sistem berhasil mengekstraksi lebih dari satu tujuan surat

Sumber : data diolah oleh penulis (2026)

h. *Statement Coverage Fungsi Rule-Based System*

Statement Coverage digunakan untuk mengukur sejauh mana seluruh *statement* pada fungsi `extract_metadata()` telah dieksekusi melalui *test case* yang telah dirancang. Pengujian ini bertujuan untuk memastikan bahwa setiap *statement* pada program telah dijalankan minimal satu kali sehingga seluruh logika program dapat diverifikasi dengan baik. *Statement* yang Tereksekusi :

1. TC1
S1, S2, S26
2. TC2
S1, S2, S3, S4, S5, S7, S20, S21, S24, S25, S26
3. TC3
S1, S2, S3, S4, S5, S6, S7, S20, S21, S24, S25, S26
4. TC4
S1, S2, S3, S7, S8, S9, S10, S11, S12, S13, S14, S15, S16, S17, S18,
S19, S20, S21, S24, S25, S26

5. TC5

S1, S2, S3, S7, S8, S17, S18, S19, S20, S21, S24, S25, S26

6. TC6

S1, S2, S3, S7, S20, S21, S22, S23, S24, S25, S26

7. TC7

S1, S2, S3, S7, S20, S21, S24, S25, S26

8. TC8

S1, S2, S3, S7, S20, S21, S22, S24, S25, S26

9. TC9

S1, S2, S3, S7, S8, S9, S10, S11, S12, S13, S14, S15, S16, S17, S18,
S19, S20, S21, S22

Berdasarkan hasil pengujian yang telah dilakukan, seluruh *statement* pada fungsi *Rule-Based System* yaitu S1 hingga S26 berhasil dieksekusi melalui kombinasi *test case* TC1 hingga TC9. Dengan demikian diperoleh nilai *statement coverage* sebesar:

$$(26/26) \times 100\% = 100\%$$

Nilai tersebut menunjukkan bahwa seluruh *statement* pada fungsi *Rule-Based System* telah diuji dan seluruh kemungkinan alur eksekusi utama program berhasil dicakup dalam proses pengujian *white box testing*.

5.4.6 Pengujian Fungsi Simpan Arsip

a. Source code Fungsi Simpan Arsip

```
@router.post("/api/simpan-arsip")
async def simpan_arsip(
    File: UploadFile = File(...),
    nomor_surat: str = Form(...),
    tanggal_surat: str = Form(...),
    perihal: str = Form(...),
    jenis_surat: str = Form(...),
    asal_tujuan: str = Form(...),
    id_user_input: int = Form(...),
    id_unit: int = Form(...),
```

```

db: Session = Depends(get_db),):
try:
    pdf_bytes = await File.read()
    ext = os.path.splitext(File.filename.lower())[1]
    if ext not in _ALLOWED_EXT:
        raise HTTPException(400, f"Tipe File tidak didukung. Gunakan:
{_ALLOWED_EXT_STR}.")
    if len(pdf_bytes) > _MAX_FILE_SIZE:
        raise HTTPException(400, "Ukuran File terlalu besar. Maksimal
3MB.")
    cek_duplikat = db.execute(
        text("SELECT nomor_surat, tanggal_surat, perihal FROM surat
WHERE nomor_surat = :nomor AND jenis_surat = :jenis LIMIT 1"),
        {"nomor": nomor_surat.strip(), "jenis": jenis_surat}
    ).fetchone()

    if cek_duplikat:
        return { "success" : False, "duplikat" : True,
            "message" : f"Nomor surat '{nomor_surat}' sudah tercatat di
arsip {jenis_surat}.",
            "data_lama": {
                "nomor_surat" : cek_duplikat.nomor_surat,
                "tanggal_surat": str(cek_duplikat.tanggal_surat),
                "perihal" : cek_duplikat.perihal,} }

    col_count = "asal_surat" if jenis_surat == "masuk" else
"tujuan_surat"
    result_count = db.execute(text(f"SELECT COUNT(*) FROM surat
WHERE {col_count} = :val"), {"val": asal_tujuan}).scalar()
    prefix = "m" if jenis_surat == "masuk" else "k"
    urutan = (result_count or 0) + 1

```

```

try:
    tgl_obj = datetime.strptime(tanggal_surat, "%Y-%m-%d")
except Exception:
    tgl_obj = datetime.now()

safe_asal_tujuan = re.sub(r'[^a-zA-Z0-9]', "", asal_tujuan.lower())
Filename =
f"{tgl_obj.strftime('%Y%m%d')}_{safe_asal_tujuan}_{prefix}{urutan}.p
df"

folder_path = "uploads/arsip"
os.makedirs(folder_path, exist_ok=True)
File_path_fisik = os.path.join(folder_path, Filename)

with open(File_path_fisik, "wb") as f:
    f.write(pdf_bytes)

db_path = File_path_fisik.replace("\\", "/")
db.execute(
    text(f"""
        INSERT INTO surat (
            nomor_surat, tanggal_surat, perihal, jenis_surat,
            {col_count}, File_path, id_user_input, id_unit, created_at
        ) VALUES (:no, :tgl, :hal, :jenis, :asal, :path, :uid, :unit, NOW()) """),
    {"no": nomor_surat, "tgl": tgl_obj.date(), "hal": perihal, "jenis":
jenis_surat,
    "asal": asal_tujuan, "path": db_path, "uid": id_user_input, "unit":
id_unit})
    db.commit()

    return {"success": True, "message": "Arsip Universitas Prabumulih
    Berhasil Disimpan!", "Filename": Filename}
except HTTPException:

```

```

raise
except Exception as e:
    db.rollback()
    print(f"ERROR SIMPAN ARSIP: {str(e)}")
return {"success": False, "message": f"Gagal masuk Database: {str(e)}"}

```

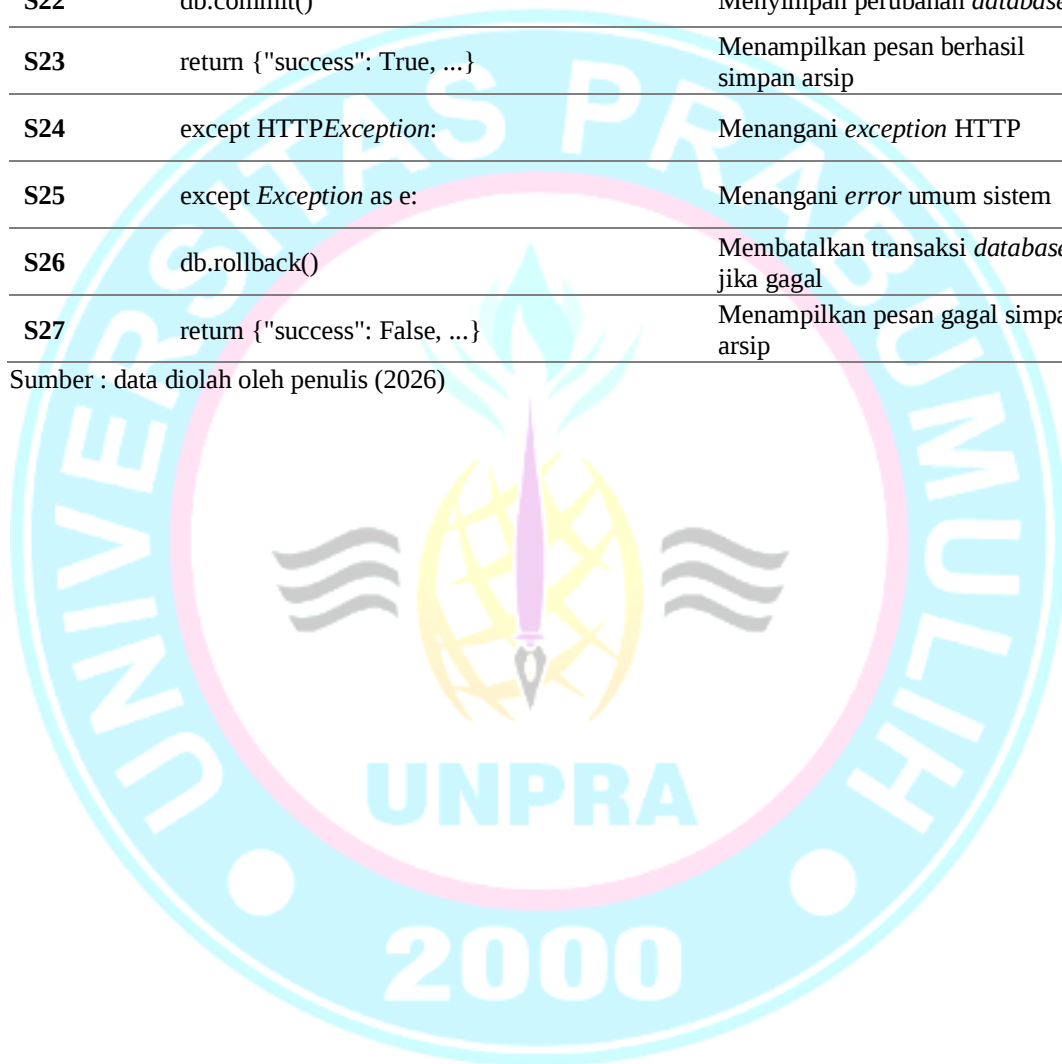
b. Identifikasi *Statement* Fungsi Simpan Arsip

Tabel 4. 10 Fungsi Simpan Arsip

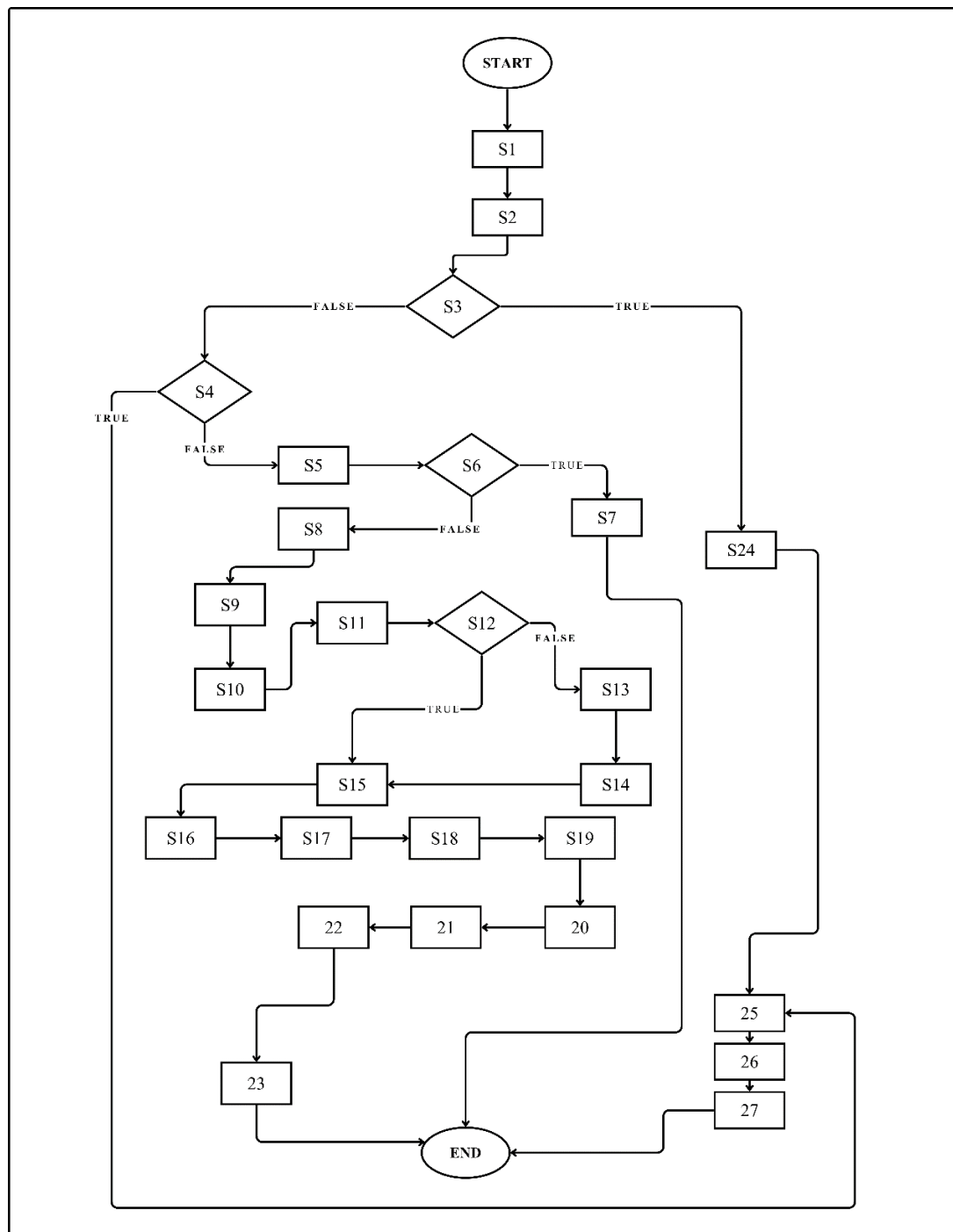
<i>Statement</i>	<i>Source code</i>	<i>Keterangan</i>
S1	pdf_bytes = await <i>File</i> .read()	Membaca <i>File upload</i> dari user
S2	ext = os.path.splitext(<i>File</i> .Filename.lower())[1]	Mengambil ekstensi <i>File</i>
S3	if ext not in _ALLOWED_EXT	Validasi tipe <i>File</i>
S4	if len(pdf_bytes) > _MAX_FILE_SIZE	Validasi ukuran <i>File</i>
S5	cek_duplikat = db.execute(...).fetchone()	Mengecek data surat duplikat di <i>database</i>
S6	if cek_duplikat	Percabangan jika data surat sudah ada
S7	return { "success": False, ... }	Mengembalikan pesan data duplikat
S8	col_count = "asal_surat" if jenis_surat == "masuk" else "tujuan_surat"	Menentukan kolom berdasarkan jenis surat
S9	result_count = db.execute(...).scalar()	Menghitung jumlah surat
S10	prefix = "m" if jenis_surat == "masuk" else "k"	Menentukan prefix nama <i>File</i>
S11	urutan = (result_count or 0) + 1	Menentukan nomor urut arsip
S12	tgl_obj = datetime.strptime(tanggal_surat, "%Y-%m-%d")	Konversi tanggal surat
S13	except <i>Exception</i> :	Penanganan <i>error</i> format tanggal
S14	tgl_obj = datetime.now()	Menggunakan tanggal saat ini jika format salah
S15	safe_asal_tujuan = re.sub(...)	Membersihkan karakter pada nama asal/tujuan
S16	Filename = f"{tgl_obj.strftime("%Y%m%d")}_{safe_asal_tujuan}_{prefix}{urutan}.pdf"	Membuat nama <i>File</i> arsip
S17	os.makedirs(folder_path, exist_ok=True)	Membuat folder penyimpanan jika belum ada

<i>Statement</i>	<i>Source code</i>	<i>Keterangan</i>
S18	with open(<i>File_path_fisik</i> , "wb") as f:	Membuka <i>File</i> untuk penyimpanan
S19	f.write(pdf_bytes)	Menyimpan <i>File</i> PDF ke folder
S20	db_path = <i>File_path_fisik</i> .replace("\\", "/")	Mengubah format path <i>File</i>
S21	db.execute(text(f"INSERT INTO surat ...	Menyimpan data arsip ke <i>database</i>
S22	db.commit()	Menyimpan perubahan <i>database</i>
S23	return {"success": True, ...}	Menampilkan pesan berhasil simpan arsip
S24	except HTTPException:	Menangani <i>exception</i> HTTP
S25	except Exception as e:	Menangani <i>error</i> umum sistem
S26	db.rollback()	Membatalkan transaksi <i>database</i> jika gagal
S27	return {"success": False, ...}	Menampilkan pesan gagal simpan arsip

Sumber : data diolah oleh penulis (2026)



c. *Flowchart Fungsi Simpan Arsip*



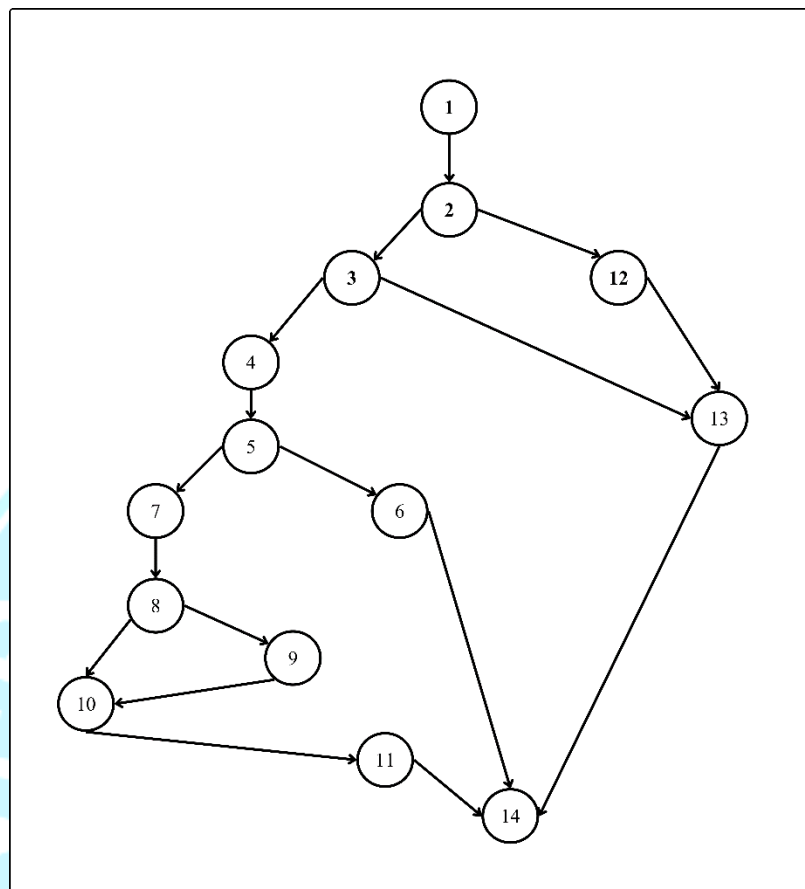
Sumber : data diolah oleh penulis (2026)

Gambar 5.34 Flowchart Fungsi Simpan Arsip

Flowchart pada fungsi simpan arsip ini merepresentasikan alur kontrol sistem dalam melakukan validasi dan penyimpanan berkas dokumen ke dalam basis

data. Proses diawali dari titik *START* saat menerima berkas unggahan, yang dilanjutkan secara linear menuju tahap pembacaan data *bytes* dokumen (S1) dan pengambilan ekstensi berkas (S2). Alur kemudian diarahkan menuju titik keputusan (*decision node*) pertama pada S3 untuk memvalidasi tipe berkas. Jika ekstensi tidak didukung (*true*), sistem akan memicu *HTTP Exception* (S24). Jika valid (*false*), alur berlanjut ke S4 untuk memeriksa ukuran dokumen dengan batas maksimal 3MB. Setelah lolos dari kedua validasi berkas tersebut, sistem melakukan query ke *database* pada S5 untuk mengecek duplikasi nomor surat berdasarkan aturan jenis surat yang sama. Jika data surat terdeteksi duplikat pada titik percabangan S6 (*true*), alur akan dihentikan dan langsung mengembalikan pesan penolakan berupa status gagal (S7).

Apabila data surat dinyatakan unik (*false* pada S6), alur memasuki tahapan penentuan parameter penyimpanan dokumen yang dinamis. Sistem mengevaluasi jenis surat pada S8 untuk menetapkan kolom target *database* (*asal_surat* atau *tujuan_surat*) dan menghitung jumlah arsip serupa pada S9 guna menentukan nomor urut serta prefix berkas (S10 dan S11). Selanjutnya, sistem melakukan penanganan *error* (*try-except block*) pada proses konversi *string* tanggal surat menjadi objek *datetime* (S12); jika terjadi kesalahan format (S13), sistem secara otomatis beralih menggunakan tanggal saat ini sebagai pengaman (*fallback* pada S14). Berkas fisik *PDF* kemudian diberi nama terstandarisasi setelah melalui pembersihan karakter noise pada S15 dan S16, lalu disimpan ke dalam direktori lokal server (S17 hingga S19). Alur diakhiri dengan proses normalisasi path (S20) dan eksekusi instruksi *SQL INSERT* ke dalam *database* (S21). Jika seluruh rangkaian penyimpanan data berhasil, transaksi akan di-*commit* pada S22 dan mengembalikan pesan sukses (S23) menuju titik akhir (*END*). Namun, apabila terjadi kegagalan sistem di tengah proses (*exception* pada S25), alur otomatis diarahkan menuju fungsi *rollback* pada S26 untuk membatalkan perubahan *database* dan mengembalikan pesan kegagalan (S27) sebelum mencapai titik *END*.

d. *Flowgraph* Fungsi Simpan Arsip

Sumber : data diolah oleh penulis (2026)

Gambar 5.35 *Flowgraph* Fungsi Simpan Arsip

Flowgraph pada fungsi simpan arsip merupakan representasi alur program dalam bentuk *node* dan *edge* untuk mempermudah analisis kompleksitas struktur. Proses ini diawali pada *node* 1 yang merepresentasikan pembacaan *File upload* sekaligus pengambilan ekstensi dokumen. Selanjutnya, validasi tipe *File* dilakukan pada *node* 2 sebagai *decision node*, di mana kondisi tidak valid akan diarahkan ke *node* 12 untuk penanganan *HTTP exception*, sedangkan kondisi valid dilanjutkan ke *node* 3 yang memvalidasi ukuran *File*. Pada *node* 3 juga terdapat percabangan, di mana *File* yang melebihi batas ukuran akan menuju *node* 13 sebagai *global exception handler*, sedangkan *File* yang valid diteruskan ke *node* 4 untuk pengecekan duplikasi data di *database*.

Tahapan berikutnya berada pada *node* 5 yang mengevaluasi hasil pengecekan duplikasi, di mana kondisi duplikat akan mengarah ke *node* 6 sebagai terminal return, sedangkan kondisi unik akan melanjutkan proses ke *node* 7 yang mencakup penentuan metadata arsip seperti kolom, prefix, dan urutan. Selanjutnya, proses parsing tanggal dilakukan pada *node* 8 sebagai decision *node*, di mana jika terjadi *error* maka akan dialihkan ke *node* 9 untuk *fallback* menggunakan tanggal saat ini, sedangkan jika berhasil proses langsung dilanjutkan. Setelah itu, alur berjalan secara linear pada *node* 10 untuk proses sanitasi nama *File*, pembuatan nama *File*, dan pembuatan folder penyimpanan, kemudian dilanjutkan ke *node* 11 yang mencakup proses penyimpanan *File* ke server, pemformatan path *database*, insert data ke *database*, commit transaksi, hingga pengembalian respon sukses. Seluruh alur diakhiri pada *node* 14, dengan mekanisme *exception* global pada *node* 12 dan 13 yang menangani HTTP *exception*, rollback *database*, serta pengembalian pesan gagal apabila terjadi kesalahan sistem.

e. *Cyclomatic Complexity* Fungsi Simpan Arsip

Berdasarkan *flowgraph* yang telah dibentuk pada fungsi simpan arsip, terdapat 4 *predicate node* (P), yaitu pada S3 (validasi tipe *File*), S4 (validasi ukuran *File*), S6 (validasi data duplikat), dan S12 (validasi format tanggal). Oleh karena itu, nilai *Cyclomatic Complexity* dapat dihitung menggunakan rumus berikut :

$$V(G) = P+1$$

$$V(G) = 4+1$$

$$V(G) = 5$$

Berdasarkan hasil perhitungan tersebut diperoleh nilai *Cyclomatic Complexity* sebesar 5. Nilai tersebut menunjukkan bahwa terdapat lima jalur independen (*independent path*) dalam proses eksekusi fungsi *simpan_arsip()*, yang merepresentasikan berbagai kemungkinan alur logika berdasarkan kondisi validasi *File*, pengecekan duplikasi data, serta validasi format tanggal. Nilai *Cyclomatic Complexity* ini digunakan sebagai dasar dalam penentuan *independent path* pada tahap *white box testing* sehingga seluruh alur eksekusi utama dapat diuji secara menyeluruh.

f. *Independent path* Fungsi Simpan Arsip

Berdasarkan hasil Cyclomatic Complexity, diperoleh 5 jalur independen (*independent path*) sebagai berikut:

1. Path 1:

$S1 \rightarrow S2 \rightarrow S3 \rightarrow S24 \rightarrow \text{END}$

(File dengan ekstensi tidak valid)

2. Path 2:

$S1 \rightarrow S2 \rightarrow S3 \rightarrow S4 \rightarrow S25 \rightarrow S26 \rightarrow S27 \rightarrow \text{END}$

(File melebihi ukuran maksimum)

3. Path 3:

$S1 \rightarrow S2 \rightarrow S3 \rightarrow S4 \rightarrow S5 \rightarrow S6 \rightarrow S7 \rightarrow \text{END}$

(Data arsip sudah pernah tersimpan / duplikat)

4. Path 4:

$S1 \rightarrow S2 \rightarrow S3 \rightarrow S4 \rightarrow S5 \rightarrow S6 \rightarrow S8 \rightarrow S9 \rightarrow S10 \rightarrow S11 \rightarrow S12$
 $\rightarrow S15 \rightarrow S16 \rightarrow S17 \rightarrow S18 \rightarrow S19 \rightarrow S20 \rightarrow S21 \rightarrow S22 \rightarrow S23 \rightarrow$
 END

(Proses simpan arsip berhasil dengan format tanggal valid)

5. Path 5:

$S1 \rightarrow S2 \rightarrow S3 \rightarrow S4 \rightarrow S5 \rightarrow S6 \rightarrow S8 \rightarrow S9 \rightarrow S10 \rightarrow S11 \rightarrow S12$
 $\rightarrow S13 \rightarrow S14 \rightarrow S15 \rightarrow S16 \rightarrow S17 \rightarrow S18 \rightarrow S19 \rightarrow S20 \rightarrow S21 \rightarrow$
 $S22 \rightarrow S23 \rightarrow \text{END}$

(Proses simpan arsip berhasil dengan *fallback* tanggal sistem)

g. *Test case* Fungsi Simpan Arsip

Pengujian dilakukan berdasarkan jalur independen yang diperoleh dari *flowgraph* fungsi simpan arsip. Setiap *test case* merepresentasikan variasi kondisi input *File* dan data arsip untuk memastikan seluruh alur eksekusi sistem berjalan dengan benar.

Tabel 4. 11 *Test case* Fungsi Simpan Arsip

Test case	Skenario Pengujian	Jalur Eksekusi	Output
TC1	<i>File</i> dengan ekstensi tidak valid	S1 → S2 → S3 → S24 → END	<i>File</i> ditolak (tipe tidak didukung)
TC2	<i>File</i> melebihi batas ukuran	S1 → S2 → S3 → S4 → S25 → S26 → S27 → END	<i>File</i> ditolak (ukuran terlalu besar)
TC3	Data arsip sudah pernah disimpan	S1 → S2 → S3 → S4 → S5 → S6 → S7 → END	Sistem menolak data duplikat
TC4	Simpan arsip berhasil (tanggal valid)	S1 → S2 → S3 → S4 → S5 → S6 → S8 → S9 → S10 → S11 → S12 → S15 → S16 → S17 → S18 → S19 → S20 → S21 → S22 → S23 → END	Arsip berhasil disimpan
TC5	Simpan arsip berhasil (<i>fallback</i> tanggal)	S1 → S2 → S3 → S4 → S5 → S6 → S8 → S9 → S10 → S11 → S12 → S13 → S14 → S15 → S16 → S17 → S18 → S19 → S20 → S21 → S22 → S23 → END	Arsip berhasil disimpan

Sumber : data diolah oleh penulis (2026)

h. *Statement Coverage* Fungsi Simpan Arsip

Statement Coverage digunakan untuk mengukur sejauh mana seluruh *statement* pada fungsi simpan arsip telah dieksekusi melalui *test case* yang dirancang. *Statement* yang tereksekusi:

1. TC1
S1, S2, S3, S24
2. TC2
S1, S2, S3, S4, S25, S26, S27
3. TC3
S1, S2, S3, S4, S5, S6, S7
4. TC4
S1, S2, S3, S4, S5, S6, S8, S9, S10, S11, S12, S15, S16, S17, S18, S19, S20, S21, S22, S23
5. TC5
S1, S2, S3, S4, S5, S6, S8, S9, S10, S11, S12, S13, S14, S15, S16, S17, S18, S19, S20, S21, S22, S23

Berdasarkan hasil pengujian, seluruh *statement* pada fungsi *simpan_arsip()* yaitu S1 hingga S27 telah dieksekusi melalui kombinasi *test case* TC1 hingga TC5. Dengan demikian diperoleh *Statement Coverage* sebesar :

$$(27/27) \times 100\% = 100\%$$

Nilai tersebut menunjukkan bahwa seluruh *statement* pada fungsi *Rule-Based System* telah diuji dan seluruh kemungkinan alur eksekusi utama program berhasil dicakup dalam proses pengujian *white box testing*.

Berdasarkan hasil pengujian *white box* yang telah dilakukan pada beberapa fungsi utama sistem, dapat disimpulkan bahwa seluruh alur logika program pada aplikasi arsip digital berbasis *OCR* dan *Rule-Based System* telah berjalan sesuai dengan rancangan sistem. Pengujian dilakukan melalui tahapan identifikasi *statement*, pembentukan *flowchart* dan *flowgraph*, perhitungan *Cyclomatic Complexity*, penentuan *independent path*, penyusunan *test case*, hingga pengukuran *statement coverage* pada setiap fungsi yang diuji. Hasil pengujian menunjukkan bahwa seluruh jalur independen berhasil dieksekusi dan seluruh *statement* program telah tercakup dalam proses pengujian dengan nilai *statement coverage* mencapai 100% pada setiap fungsi yang diuji. Dengan demikian, sistem mampu menangani berbagai kondisi proses, baik pada kondisi normal maupun kondisi *error handling*, sehingga dapat dinyatakan bahwa implementasi logika program pada sistem telah berjalan dengan baik, stabil, dan sesuai dengan tujuan pengembangan aplikasi arsip digital yang mendukung proses pengelolaan dokumen secara terstruktur dan terintegrasi.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil penelitian, analisa perancangan, implementasi, serta pengujian terstruktur yang telah dilakukan pada bab-bab sebelumnya mengenai pengembangan Aplikasi Arsip Digital berbasis *Optical Character Recognition (OCR)* dan *Rule-Based System* di Universitas Prabumulih, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Penelitian ini berhasil mengembangkan sistem arsip digital terintegrasi berbasis *Optical Character Recognition (OCR)* dan *Rule-Based System* yang mampu mengotomatisasi proses digitalisasi, ekstraksi metadata, klasifikasi, serta pengelolaan arsip surat masuk dan surat keluar di Universitas Prabumulih.
2. Sistem yang dikembangkan mampu mendukung transformasi pengelolaan arsip dari metode manual berbasis dokumen fisik menjadi sistem digital yang lebih efisien, terstruktur, dan mudah diakses. Hasil pengujian menunjukkan bahwa seluruh fungsi utama sistem berjalan sesuai rancangan dengan nilai statement coverage sebesar 100%, sehingga sistem dapat digunakan untuk mendukung proses pengarsipan di lingkungan Universitas Prabumulih.

6.2 Saran

Meskipun aplikasi pengarsipan digital ini telah memenuhi target spesifikasi perancangan utama dan lulus pengujian logika struktur internal, rancangan ini masih memiliki ruang pengembangan untuk meningkatkan performa dan pengalaman pengguna di masa mendatang. Oleh karena itu, beberapa saran yang dapat diajukan untuk pengembangan selanjutnya adalah sebagai berikut:

1. Penambahan Fitur *Filter* dan Pengurutan Surat: Untuk pengembangan antarmuka dan fungsionalitas manajemen dokumen, disarankan untuk menambahkan menu filter surat khusus yang memungkinkan pengguna untuk mengatur dan menyaring surat dalam urutan tertentu, seperti pengurutan kronologis berdasarkan tanggal surat, filter berdasarkan kategori unit asal/tujuan, status disposisi, maupun klasifikasi nomor surat guna mempercepat proses pencarian dokumen dalam volume besar.
2. Peningkatan Akurasi *Preprocessing* Citra OCR: Diperlukan penambahan modul *image Preprocessing* (seperti otomatisasi binarisasi, *deskewing*, dan pembersihan *noise*) sebelum dokumen teks dialirkan ke mesin OCR *Tesseract* guna meningkatkan kualitas pembacaan karakter pada dokumen fisik atau hasil pemindaian (*scanning*) yang resolusinya kurang optimal.
3. Ekspansi Pengujian Sistem: Mengingat penelitian ini berfokus pada pengujian struktural (*White box testing*), untuk tahap selanjutnya sangat disarankan melakukan pengujian fungsionalitas secara menyeluruh menggunakan metode *Black Box Testing* serta pengujian kepuasan pengguna akhir (*User Acceptance Testing/UAT*) untuk mengukur aspek usability sistem secara nyata dari sudut pandang operator dan pimpinan.
4. Integrasi Fitur Keamanan *Enkripsi*: Untuk menjamin keamanan dokumen arsip yang bersifat rahasia (*confidential*) di lingkungan rektorat, pengembangan berikutnya dapat menerapkan metode kriptografi atau enkripsi berkas pada dokumen *PDF* yang disimpan di dalam direktori penyimpanan server.

DAFTAR PUSTAKA

- Akhsa, A. T. P. D., Agus, M., Rosmiati, & Ulum, A. M. B. (2024). *Designing E-Office Services And Digital Archiving Using Web-Based OCR Methods. Journal of Information Technology and Computer Science (INTECOMS)*, 7(1).
- ALazzawi, A., Yas, Q. M., & Rahmatullah, B. (2023). A Comprehensive Review of *Software development Life Cycle* methodologies: Pros, Cons, and Future Directions. *Iraqi Journal for Computer Science and Mathematics*, 4(4), 173–190. <https://doi.org/10.52866/ijcsm.2023.04.04.014>
- Andriyanto, S. (2022). *Pemodelan Perangkat Lunak Behavior Diagram* (M. Ayu, Ed.). Politeknik Manufaktur Negeri Bangka Belitung.
- Ardiansah, I., & Permana, R. H. (2023). *Memulai Python Belajar Python dari Nol* (N. Septaryana, Ed.). CV. Cendekia Press. <https://play.google.com/books/reader?id=h0nfEAAQBAJ&pg=GBS.PA1&hl=id>
- Evi, T., Gai, A. M., Fauza, M., & Hatchi, I. (2025). *Buku Referensi Metode Penelitian Kualitatif dan Kuantitatif Teknik dan Strategi Dalam Mengolah Data*.
- Fad'li, G. A., Marsofiyati, & Suherdi. (2023). Implementasi Arsip Digital Untuk Penyimpanan Dokumen Digital. *Jurnal Manuhara : Pusat Penelitian Ilmu Manajemen Dan Bisnis*, 1(4), 01–10. <https://doi.org/10.61132/manuhara.v1i4.115>
- Fauzi, A., & Yunial, A. H. (2025). *Testing dan QA Perangkat Lunak: Memahami Konsep, Metode, Teknik, dan Pendekatan dalam Pengujian*. Eureka Media Aksara.
- Fradana, R. A., & Marlina, L. (2025). *Perancangan Aplikasi Database Sistem Pengelolaan Arsip Digital Berbasis Web Mobile Menggunakan Algoritma Levensthein (Studi Kasus : SMA PLUS TARUNA AKTERLIS MEDAN)*.

- Gosal, F. A., & Rompas, P. T. D. (2023). *Penerapan Teknologi Optical Character Recognition Pada Pengarsipan Dokumen (Studi Kasus: PT Pertamina Geothermal Energy Area Lahendong)*.
- Harahap, F. L., Andrianto, R., Panggabean, P., Hanafia, P., Panusunan, P., & Garang, M. (2023). Pengembangan Aplikasi Kalkulator Multifungsi Menggunakan *Flutter Framework*. *Jurnal Penelitian Teknologi Informasi Dan Sains*, 1(3), 76–84. <https://doi.org/10.54066/jptis.v1i3.815>
- Iskandar, M. S., Munazilin, A., & Susanto, A. (2025). Implementasi Aplikasi Manajemen Arsip Surat Berbasis *Optical Character Recognition* Pada Badan Pusat Statistik Banyuwangi. *Jurnal Teknologi Dan Manajemen Industri Terapan (JTMIT)*, 4(3), 622–631.
- Kolawole, I., & Fakokunde, A. (2024). Improving *Software development* with Continuous Integration and Deployment for Agile DevOps in Engineering Practices. *International Journal of Computer Applications Technology and Research*. <https://doi.org/10.7753/ijcatr1401.1002>
- Kostopoulos, G., Davrazos, G., & Kotsiantis, S. (2024). Explainable *Artificial intelligence*-Based Decision Support Systems: A Recent Review. In *Electronics (Switzerland)* (Vol. 13, Number 14). Multidisciplinary Digital Publishing Institute (MDPI). <https://doi.org/10.3390/electronics13142842>
- Lantana, D. A., & Fauziah. (2025). *Design Thinking dan Inovasi*. Lembaga Penerbitan Universitas Nasional (LPU UNAS).
- Lestanti, C. Q. R. O., & Pribadi, J. D. (2025). *Digital Archives Management System for Archives Security at the Balai Pemberdayaan Industri Persepatuan Indonesia (BPIPI)*. <https://ejournal.mandalanursa.org/index.php/Bisnis>
- Nasution, M. A. A. H., Siswanto, & Suryana, E. (2023). Rancangan Media Pembelajaran Berupa Aplikasi Augmented Reality Berbasis *Android*. In *Sawah Lebar Kota Bengkulu* (Vol. 19, Number 2).
- Nuraeni, R., Santoso, A. P. A., & Rezi. (2025). Perlindungan Konsumen Terhadap Produk Overclaim Skincare Di Indonesia. In *JURNAL INOVASI HUKUM* (Vol. 6, Number 1). <https://ejournals.com/ojs/index.php/jih>
- Nurdiansyah, A., Valentino, R. A., & Yussuf, S. A. (2025). Preservasi Digital sebagai Upaya Akuntabilitas Tata Kelola Arsip di Lembaga Kearsipan di

- Indonesia. *Khazanah: Jurnal Pengembangan Kearsipan*, 18(1), 186.
<https://doi.org/10.22146/khazanah.106468>
- Panjaitan, R. M., Rahayudi, B., & Ratnawati, D. E. (2024). *Optical Character Recognition Mobile App for Address Matching in Integrated Social Welfare Data Verification Process*. In *Journal of Information Technology and Computer Science* (Vol. 9, Number 1). www.jitecs.ub.ac.id
- Pramono, T., & Soekamto, Y. S. (2024). Information Extraction Berbasis Rule untuk Laporan Keuangan Perusahaan. *Journal of Information System, Graphics, Hospitality and Technology*, 6(1), 17–22.
<https://doi.org/10.37823/insight.v6i1.316>
- Pranata, M. R. (2024). Arsip Digital Untuk Efisiensi Pengelolaan Dan Percepatan Pelayanan. *SIGNIFICANT: Journal Of Research And Multidisciplinary*, 3(01), 162–172. <https://doi.org/10.62668/significant.v3i01.891>
- Purbasari, Y., Suhartini, & Sakti, E. M. S. (2023). *Perancangan Aplikasi Arsip Elektronik Fakultas Ilmu Komputer Universitas Prabumulih*.
<https://journals.upi-yai.ac.id/index.php/ikraith-informatika/issue/archive>
- Putri, N. D., Oktofa, M. A., Rahmadhani, A. A., & Nurbaiti. (2023). *Pentingnya Peranan Perangkat Keras Dalam Sistem Informasi Manajemen* (Vol. 2, Number 1).
- Rahman, S., Sembiring, A., Siregar, D., Khair, H., Prahmana, I. G., Puspadini, R., & Zen, M. (2023). *Python : Dasar Dan Pemrograman Berorientasi Objek*. Tahta Media Group.
- Risparyanto, A. (2021). Pengelolaan Arsip Perpustakaan. *Buletin Perpustakaan Universitas Islam Indonesia*, 4(2), 161–172.
- Runtu, R. R., Maramis, G. D. P., & Hasibuan, A. (2025). Implementasi Algoritma Rule-Based Classification pada Aplikasi Arsip Web Menggunakan Metode RAD dan Evaluasi ISO/IEC 25010. *Jurnal Minfo Polgan*, 14(2), 2792–2798.
<https://doi.org/10.33395/jmp.v14i2.15505>
- Sari, I. P., Batubara, I. H., Al-Khowarizmi, & Hariani, P. P. (2022). Perancangan Sistem Informasi Pengelolaan Arsip Digital Berbasis Web untuk Mengatur Sistem Kearsipan di SMK Tri Karya. *Wahana Jurnal Pengabdian Kepada Masyarakat*.

- Sari, L., & Siregar, guna Y. K. S. (2021). Perancangan Aplikasi Pendataan Data Kepegawaian Negeri Sipil Pada Dinas Komunikasi Dan Informatika Kota Metro. *Jurnal Mahasiswa Ilmu Komputer*.
- Setiawan, I., Ariansyah, & Purbasari, Y. (2020). *Buku Ajar Kuliah Metode Penelitian Kualitatif*.
- Setiawan, P. R., Ramadhan, R. A., Labellapansa, D. A., Koresponden, P., Panji, :, & Setiawan, R. (2022). Jurnal Pengabdian Masyarakat dan Penerapan Ilmu Pengetahuan Pelatihan Pemrograman Flutter. *Jurnal Pengabdian Masyarakat Dan Penerapan Ilmu Pengetahuan*, 03, No. 01.
- Setiorini, & Aulianto, D. R. (2024). Digital Archive Management in the Organization Section of the Regional Secretariat of Tuban Regency. In *Jurnal Ilmu Informasi Perpustakaan dan Kearsipan* (Vol. 13, Number 01). <http://ejournal.unp.ac.id/index.php/iipk/index>
- Silaen, I. J. J., Sari, J. E. O., & Steven, J. (2022). *Literature Review Faktor-Faktor Yang Mempengaruhi Implementasi Si: Hardware, Software, Dan Database*. 1(1). <https://doi.org/10.38035/jim.v1i1>
- Silalahi, F. D. (2022). *Manajemen Database MySQL (Structured Query Language)*.
- Suharni, Susilowati, E., & Pakusadewa, F. (2023). Perancangan Website Rumah Makan Ninik Sebagai Media Proposi Menggunakan Unified Modelling Language. *Jurnal Rekayasa Informasi*.
- Sulkowski, Lukasz. (2023). *Managing the digital university: paradigms, leadership, and organization*. Routledge.
- Ulwani, R. F., Mailin, M., & Zainun, Z. (2021). Strategi Komunikasi Al-Ittihadiyah Sumatera Utara dalam Mengembangkan Dakwah di Sumatera Utara. *JURNAL LENSEA MUTIARA KOMUNIKASI*, 4(1), 10–20. <https://doi.org/10.51544/jlmk.v5i1.1496>
- Universitas Cakrawala. (2023, December 19). *Mengenai MySQL, Sistem Manajemen Database yang Menjadi Pilihan Utama*. Cakrawala.Ac.Id. <https://www.cakrawala.ac.id/berita/mysql-adalah>
- Yunita, H. D., Winarko, T., & Winariyani. (2022). Penerapan Aplikasi Website Dalam Pengolahan Data Posyandu Pada Posyandu Bina Sejahtera. In *Jurnal Cendikia* (Vol. 22, Number 1).

Zulvikri, M., Amani, A., Negeri Bandung Alamat, P., Gegerkalong Hilir, J., Parongpong, K., Bandung Barat, K., Barat, J., & Penulis, K. (2024). Transformasi Digital: Menggali Potensi Teknologi Terkini Dalam Pengelolaan Dokumen Dan Informasi Kantor Pada Perusahaan PT. Victory Prima Abadi. *Jurnal Transformasi Bisnis Digital (JUTRABIDI)*, 1(3), 1–12. <https://doi.org/10.61132/jutrabidi.v1i2.107>



LAMPIRAN

Lampiran 1 Surat Permohonan Penelitian



The image shows a formal letter on the official letterhead of the Faculty of Computer Science at Universitas Prabumulih. The letter is dated October 24, 2025, and is addressed to the Rector of the university. It is a request for research and data collection, specifically for a thesis project. The letter includes the student's name (Eci Wulandari), ID number (2022230032), and the title of the thesis (Development of a Digital Archive and Dynamic Assessment Application Based on OCR and Rule-Based System). The letter is signed by the Head of the Informatics Study Program, Iwan Setiawan, S.Kom., M.Kom., with a stamp and NIDN number.


FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH

Prabumulih, 24 Oktober 2025

Nomor : 018 /FASILKOM/UNPRA/Inf-S1/X/2025
Lampiran : -
Perihal : Permohonan Penelitian dan Pengambilan Data

Kepada Yth,
Rektor Universitas Prabumulih
Di
Tempat

Dalam rangka pelaksanaan skripsi mahasiswa Program Studi Informatika Fakultas Ilmu Komputer Universitas Prabumulih, maka kami harapkan bantuan dari Ibu kiranya mengizinkan mahasiswa:

Nama : Eci Wulandari
Nim : 2022230032
Judul Skripsi : Pengembangan Aplikasi Pengajuan Dinas dan Arsip Digital Terpadu Berbasis OCR dan Rule-Based System

Untuk melaksanakan penelitian dan pengambilan data ditempat Ibu Pimpin guna menyelesaikan skripsi. Atas bantuan dan kerjasamanya kami ucapkan terima kasih.

Ketua Program Studi,
Informatika

Iwan Setiawan, S.Kom., M.Kom.
NIDN.0209129101

Jalan Patra No.50 RT.01 RW.03 Kelurahan Sukaraja Kecamatan Prabumulih Selatan
Kota Prabumulih Sumatera Selatan Indonesia Telpun 0713-3224417 Fax 0713-322418
24 Okt 2025 14.28

Lampiran 2 Surat Balasan Permohonan Penelitian


UNIVERSITAS PRABUMULIH

Prabumulih, 28 Oktober 2025

Nomor : 343/UNPRA/X/2025
Lampiran : -
Perihal : **Balasan Permohonan Penelitian dan Pengambilan Data**

Kepada Yth.
Ketua Program Studi Informatika
Fakultas Ilmu Komputer - Universitas Prabumulih
Di -
Tempat

Dengan hormat,
Menindaklanjuti surat dari Ketua Program Studi Informatika Fakultas Ilmu Komputer Universitas Prabumulih pada tanggal 24 Oktober 2025 nomor: 018/FASILKOM/UNPRA/Inf-S1/X/2025 perihal: Permohonan Penelitian dan Pengambilan Data atas nama mahasiswa:

Nama : Eci Wulandari
NIM : 2022230032

Pada dasarnya kami tidak keberatan dan silahkan saja melakukan penelitian untuk mendapatkan data pendukung yang dibutuhkan dalam penyusunan Skripsi tersebut dengan mengikuti peraturan yang ada di lingkungan Universitas Prabumulih.

Demikianlah, atas perhatian dan kerjasamanya diucapkan terima kasih.


Dr. H. Pratiwi, S.Si., M.Si.

Jalan Patra No.50 RT.01 RW.03 Kelurahan Sukaraja Kecamatan Prabumulih Selatan
Kota Prabumulih Sumatera Selatan Indonesia Telpn 0713-3224417 Fax 0713-322418
e-mail : mail@unpra.ac.id

Lampiran 3 Dokumentasi Wawancara & Observasi



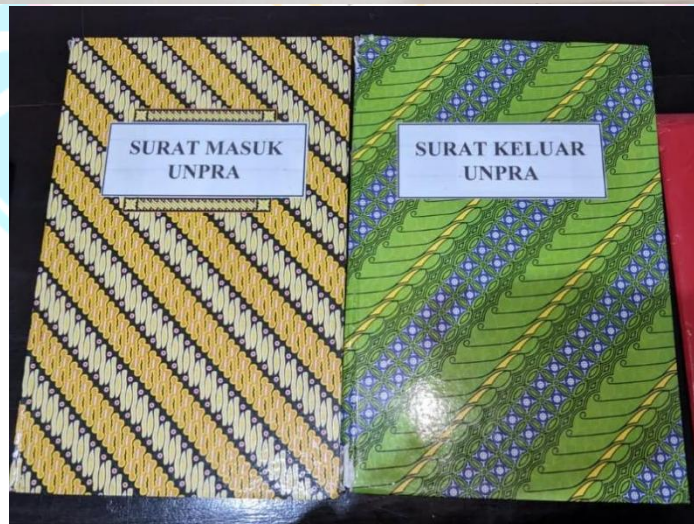


Surat dari DISKOPIL
No. Surat 100.12/1262/Diskopil/2014
Tgl. Surat 03.10.2015

LEMBAR DISPOSISI
Diterima tgl. 23.10.2015
No. Agenda
Sifat ☐ Biasa ☒ Segera ☐ Rahasia

Perihal Undangan

Dari	Kepada	Keterangan	Paraf
Rektor	Adm Rektorat	4 dapat dihadiri	04
	Kesekretariatan	u/ menginformasikan	04
	Unit		



Lampiran 4 Daftar Pertanyaan Wawancara

Draft Wawancara

Nama Narasumber : Ranti Yuliningsih, A.Md.
Jabatan : Staf Administrasi Rektorat
Tempat Wawancara : Kantor Rektorat Universitas Prabumulih
Tanggal Wawancara : 30 Oktober 2025
Pewawancara : Eci Wulandari

No	Pertanyaan	Jawaban
1	Dapatkah Ibu menjelaskan secara singkat mengenai Universitas Prabumulih?	Universitas Prabumulih merupakan perguruan tinggi yang berperan dalam pengembangan pendidikan tinggi di Kota Prabumulih. Universitas ini bertujuan mendukung peningkatan kualitas sumber daya manusia melalui kegiatan pendidikan, penelitian, dan pengabdian. Informasi sejarah resmi dapat dilihat melalui dokumen dan situs universitas.
2	Apa terdapat visi dan misi Universitas Prabumulih menurut dokumen resmi universitas?	Untuk visi dan misi kampus, saya menyarankan untuk melihat langsung melalui dokumen resmi atau situs Universitas Prabumulih, karena informasi tersebut sudah dijelaskan lebih lengkap dan terperinci di sana.
3	Bagaimana struktur umum organisasi di Rektorat Universitas Prabumulih?	Untuk struktur organisasi, secara umum tetap mengikuti ketentuan universitas, namun informasi detailnya juga dapat dilihat pada situs atau dokumen struktur organisasi yang sudah ditetapkan.

4	Boleh dijelaskan mengenai posisi dan tanggung jawab Ibu sebagai staf administrasi di Rektorat?	Saya bertugas menangani surat-menyurat dan pengelolaan arsip. Kegiatan saya meliputi pencatatan surat yang diterima maupun dikirim, penyusunan lembar disposisi, serta penataan arsip agar mudah ditemukan kembali.
5	Bagaimana proses surat masuk dan keluar di Rektorat Universitas Prabumulih?	Proses surat masuk dicatat terlebih dahulu dalam buku agenda, kemudian dilampirkan lembar disposisi dan diteruskan kepada pimpinan untuk diarahkan ke bagian terkait. Surat keluar juga dicatat dalam buku agenda surat keluar sebelum didistribusikan ke pihak tujuan.
6	Siapa yang bertanggung jawab dalam pengelolaan arsip dan dokumen di rektorat?	Pengelolaan arsip dilakukan oleh staf administrasi yang mengatur pencatatan, penyimpanan map arsip, dan penataan dokumen sesuai kategori penyimpanan.
7	Apakah terdapat SOP pengelolaan arsip secara tertulis?	Belum terdapat SOP pengelolaan arsip yang tertulis secara resmi.
8	Bagaimana sistem penyimpanan arsip yang diterapkan saat ini?	Arsip disimpan dalam bentuk fisik dan sebagian telah didigitalisasi secara sederhana.
9	Berapa lama masa simpan arsip?	Masa simpan arsip minimal selama 5 tahun.
10	Apa yang dilakukan terhadap arsip setelah masa simpan berakhir?	Arsip disimpan di gudang atau dimusnahkan setelah masa simpan berakhir.
11	Apakah diperbolehkan melakukan pendokumentasian proses pengelolaan arsip untuk keperluan penelitian?	Diperbolehkan, selama hanya untuk kepentingan penelitian dan tidak mencakup dokumen yang bersifat rahasia atau berisi informasi internal tertentu. Jika ada dokumen yang tidak dapat ditampilkan, hal tersebut akan diberitahukan.
12	Apakah saat ini sudah ada sistem komputerisasi dalam pengelolaan arsip?	Saat ini penyimpanan dilakukan dalam bentuk arsip fisik, dicatat melalui buku agenda, dan disimpan dalam map serta rak penyimpanan. Belum ada sistem berbasis aplikasi atau digital yang digunakan.

13	Apakah pernah terjadi kendala dalam proses pengarsipan?	Kendala dapat terjadi terutama ketika jumlah arsip meningkat, karena proses pencarian memerlukan waktu lebih lama. Ketelitian sangat dibutuhkan agar penempatan arsip tetap sesuai kategori.
14	Bagian mana dari proses arsip yang dianggap paling penting atau memerlukan ketelitian lebih tinggi?	Pencatatan detail seperti nomor surat, tanggal, asal, dan tujuan surat sangat penting. Ketepatan informasi ini mempengaruhi kemudahan pencarian dan penelusuran arsip di kemudian hari.
15	Apakah pencarian arsip dapat memakan waktu dalam situasi tertentu?	Ya, terutama jika arsip yang dicari adalah arsip lama atau jumlah dokumennya banyak. Karena penyimpanan berbasis fisik, proses pengecekan perlu dilakukan secara teliti.
16	Apakah kondisi seperti ini berpotensi menimbulkan hambatan di masa mendatang?	Berpotensi, terutama jika volume administrasi meningkat atau struktur kerja berubah. Ketergantungan pada satu orang pengelola juga dapat menjadi tantangan tersendiri.
17	Jika /suatu saat diterapkan sistem arsip digital, manfaat apa yang paling terasa bagi bagian administrasi?	Sistem arsip digital dapat mempercepat pencatatan, memudahkan pencarian dokumen, mengurangi risiko kesalahan pendataan, serta membantu menjaga kerapian penyimpanan data.

18 Bagaimana pendapat Ibu jika peneliti mengusulkan pengembangan aplikasi arsip digital untuk mendukung pengelolaan administrasi?

Usulan tersebut sangat baik dan dapat membantu pekerjaan menjadi lebih efektif. Selama sistemnya mudah dioperasikan dan sesuai kebutuhan, kami menyambut baik pengembangan tersebut.

19 Apakah Ibu bersedia jika nantinya dilakukan uji coba aplikasi yang dikembangkan peneliti?

Ya, selama proses uji coba dilakukan dengan penjelasan terlebih dahulu dan tidak mengganggu aktivitas kerja inti.

Narasumber



Kantor Administrasi Rektorat

Rantiyulningsih, A.Md.

Prabumulih, 30 Oktober 2025

Mahasiswa

Eci Wulandari

Lampiran 5 Berita Acara Demonstrasi Aplikasi

BERITA ACARA **DEMONSTRASI APLIKASI DAN PENGUMPULAN DATA** **PENELITIAN SKRIPSI**

Pada hari dan tanggal tersebut di bawah ini, telah dilaksanakan kegiatan demonstrasi aplikasi dan pengumpulan data pendukung dalam rangka penelitian skripsi dengan rincian sebagai berikut:

Hari / Tanggal : Kamis, 30 April 2026
Waktu : Menyesuaikan jam kerja
Tempat : Ruang Rektorat Universitas Prabumulih
Narasumber : Ranti Yuliningsih, A.Md.
Jabatan : Staf Administrasi Rektorat UNPRA
Peneliti : Eci Wulandari (NIM. 2022230032)
Kegiatan : **Demonstrasi Aplikasi & Pengumpulan Data Pendukung**

Adapun agenda kegiatan yang dilaksanakan adalah sebagai berikut:

No.	Agenda	Keterangan
1.	Pengumpulan data pendukung	Peneliti meminta contoh dokumen fisik surat masuk dan surat keluar yang biasa diterima dan dikirim oleh Unit Rektorat Universitas Prabumulih sebagai data pengembangan sistem.
2.	Demonstrasi aplikasi	Peneliti mendemonstrasikan aplikasi Sistem Informasi Penyimpanan Arsip Digital yang telah dikembangkan kepada narasumber, mencakup fitur upload surat masuk, proses ekstraksi OCR otomatis, tampilan hasil ekstraksi metadata, pengelolaan arsip, dan disposisi surat.
3.	Sesi umpan balik	Narasumber memberikan tanggapan dan masukan terhadap aplikasi yang didemonstrasikan. Masukan utama yang diberikan berkaitan dengan fitur pembuatan surat keluar,

		narasumber menyampaikan bahwa apabila pembuatan surat keluar masih menggunakan aplikasi pengolah kata pihak ketiga (seperti Microsoft Word), maka ketergantungan terhadap aplikasi pihak ketiga masih belum dapat dieliminasi sepenuhnya.
4.	Tindak lanjut	Berdasarkan masukan tersebut, peneliti melakukan penambahan fitur editor pembuatan surat keluar langsung di dalam aplikasi pada tahap pengembangan selanjutnya, sehingga seluruh proses pembuatan, pengunggahan, dan pengarsipan surat keluar dapat dilakukan dalam satu platform yang terintegrasi.

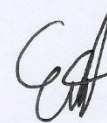
Demikian Berita Acara ini dibuat dengan sebenar-benarnya sebagai dokumentasi kegiatan penelitian. Berita Acara ini ditandatangani oleh pihak-pihak yang terlibat sebagai bukti bahwa kegiatan tersebut telah benar-benar dilaksanakan.

Prabumulih, 30 April 2026

Narasumber / Informan,
Staf Administrasi Rektorat

Peneliti,


Ranti Yuliningsih, A.Md.



Eci Wulandari
NIM. 2022230032



**FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH**

**KESEDIAAN SEBAGAI PEMBIMBING SKRIPSI
PROGRAM STUDI INFORMATIKA**

Yang bertanda tangan di bawah ini:

Nama : Iwan Setiawan, S.Kom., M.Kom.
NUPTK : 1541769670130303
Program Studi : Informatika
Fakultas : Ilmu Komputer

Bersedia menjadi pembimbing I Skripsi, dan

Nama : Riza Kartina, S.Kom., M.Kom.
NUPTK : 0549767668231073
Program Studi : Informatika
Fakultas : Ilmu Komputer

Bersedia menjadi pembimbing II Skripsi

dari mahasiswa Program Studi Informatika Fakultas Ilmu Komputer Universitas Prabumulih
di bawah ini:

Nama : Eci Wulandari
NIM : 2022230032
Judul Skripsi : Pengembangan Aplikasi Arsip Digital Berbasis OCR dan
Rule-Based System

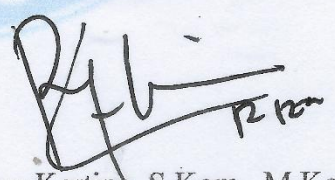
Demikian surat kesediaan membimbing ini kami buat, atas perhatiannya kami ucapkan terima kasih.

Prabumulih, 2 Juni 2026

Dosen Pembimbing I

Dosen Pembimbing II


(Iwan Setiawan, S.Kom., M.Kom.)
NUPTK.1541769670130303


(Riza Kartina, S.Kom., M.Kom.)
NUPTK.0549767668231073



FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH

KARTU BIMBINGAN SKRIPSI

NAMA : Eci Wulandari
NIM : 2022230032
PROGRAM STUDI : Informatika
NAMA PEMBIMBING I : Iwan Setiawan, S.Kom., M.Kom.
JUDUL : Pengembangan Aplikasi Arsip Digital Berbasis *OCR* Dan
Rule-Based System

NO	TGL KONSULTASI	TOPIK POKOK YANG DIBICARAKAN	TANDA TANGAN PEMB. I	TGL MENG-HADAP KEMBALI
1	28-1-2026	BAB I a. Tujuan		
		BAB II a. Pengertian flutter b. Judul penelitian Terdahulu		
		BAB III a. Struktur Organisasi unpa b. observasi		
2	29-1-2026	BAB I ACC		
		BAB II ACC		
		BAB III ACC		

Mulai Bimbingan : 28-1-2026
Batas Akhir Bimbingan :

Mengetahui Kaprodi Informatika



IWAN SETIAWAN, S.Kom.M.Kom

PERHATIAN !
TIDAK BOLEH HILANG
SETIAP BIMBINGAN HARUS DIBAWA



FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH

KARTU BIMBINGAN SKRIPSI

NAMA : Eci Wulandari
NIM : 2022230032
PROGRAM STUDI : Informatika
NAMA PEMBIMBING I : Iwan Setiawan, S.Kom., M.Kom.
JUDUL : Pengembangan Aplikasi Arsip Digital Berbasis OCR Dan
Rule-Based System

NO	TGL. KONSULTASI	TOPIK POKOK YANG DIBICARAKAN	TANDA TANGAN PEMB. I	TGL MENG-HADAP KEMBALI
4.	2/3/2026	BAB IV a. penjabaran tabel & gambar b. Antar muka		
5	12/3/2026	BAB IV ACC		
6.	13/5/2026	BAB V a. halaman ✓ b. processor ✓ c. windows 11 pro ✓ d. flutter 3.4.5 ✓, OCR engine ✓ e. chrome ✓, urutan tabel ✓ f. Samakan Tabel database ✓ g. Bab IV buat desain UI/UX ✓ h. upload surat masuk ✓	 i. ket surat masuk /keluar ✓ j. urutan surat ✓ k. di Spesifikasi ✓ 	

Mulai Bimbingan :

Batas Akhir Bimbingan :

PERHATIAN !
TIDAK BOLEH HILANG
SETIAP BIMBINGAN HARUS DIBAWA

Mengetahui Kaprodi Informatika

IWAN SETIAWAN, S.Kom.M.Kom



FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH

KARTU BIMBINGAN SKRIPSI

NAMA : Eci Wulandari
NIM : 2022230032
PROGRAM STUDI : Informatika
NAMA PEMBIMBING I : Iwan Setiawan, S.Kom., M.Kom.
JUDUL : Pengembangan Aplikasi Arsip Digital Berbasis OCR Dan
Rule-Based System

NO	TGL. KONSULTASI	TOPIK POKOK YANG DIBICARAKAN	TANDA TANGAN PEMB.	TGL MENG-HADAP KEMBALI
7.	21/5/2022	BAB V a. pilih file surat b. file filter malukan surat c. Pengy! an		
8.	22/5/2022	BAB V ACC BAB VI ACC	 	
9.	26/5/2022	Kelengkapan a. Lembar Kerja b. pengantar & powerpoint		
10	29/5/2022	Kelengkapan SIAP UJI		

Mulai Bimbingan :

Batas Akhir Bimbingan :

PERHATIAN !
TIDAK BOLEH HILANG
SETIAP BIMBINGAN HARUS DIBAWA

Mengetahui Kaprodi Informatika



IWAN SETIAWAN, S.Kom.M.Kom



FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH

KARTU BIMBINGAN SKRIPSI

NAMA : Eci Wulandari
NIM : 2022230032
PROGRAM STUDI : Informatika
NAMA PEMBIMBING II : Riza Kartina, S.Kom., M.Kom
JUDUL : Pengembangan Aplikasi Arsip Digital Berbasis OCR Dan
Rule-Based System

NO	TGL KONSULTASI	TOPIK POKOK YANG DIBICARAKAN	TANDA TANGAN PEMB. II	TGL MENG-HADAP KEMBALI
1)	21/1/2026	Bab. I Revisi - Fokus siapa pengguna aplikasi - Rumusan, Batasan, Tujuan disesuaikan kebutuhan.	Riz	23/1/2026.
2)	23/1/2026	Bab I Acc	Riz	26/1/2026.
3)	26/1/2026	Bab II Acc	Riz	
4)	27/1/2026	Bab III Revisi - Sesuaikan template - Sistematika penulisan	R	28/1/2026
5)	28/1/2026	Bab III Acc	R	

Mulai Bimbingan : 21/1/2026
Batas Akhir Bimbingan :

Mengetahui Kaprodi Informatika



IWAN SETIAWAN, S.Kom.M.Kom

PERHATIAN !
TIDAK BOLEH HILANG
SETIAP BIMBINGAN HARUS DIBAWA



FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH

KARTU BIMBINGAN SKRIPSI

NAMA : Eci Wulandari
NIM : 2022230032
PROGRAM STUDI : Informatika
NAMA PEMBIMBING II : Riza Kartina, S.Kom., M.Kom
JUDUL : Pengembangan Aplikasi Arsip Digital Berbasis OCR Dan
Rule-Based System

NO	TGL KONSULTASI	TOPIK POKOK YANG DIBICARAKAN	TANDA TANGAN PEMB. II	TGL MENG-HADAP KEMBALI
6	4/2/2026	Bab. IV Revisi analisis sistem berjalan	Rt.	
7.	6/2/2026	Bab II analisis sistem berjalan (ACC) Rt		
8.	6/2/2026	Bab. IV Revisi - Activity Diagram - Usecase	Rt	
9.	13/2/2026	Bab IV - Uscase ACC - Activity Diagram revisi	Rt	
10.	25/2/2026	Bab IV - Activity diagram ACC - ERD Revisi - Tabel database Revisi	Rt	

Mulai Bimbingan :

Batas Akhir Bimbingan :

Mengetahui Kaprodi Informatika



IWAN SETIAWAN, S.Kom.M.Kom

PERHATIAN !
TIDAK BOLEH HILANG
SETIAP BIMBINGAN HARUS DIBAWA



FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH

KARTU BIMBINGAN SKRIPSI

NAMA : Eci Wulandari
NIM : 2022230032
PROGRAM STUDI : Informatika
NAMA PEMBIMBING II : Riza Kartina, S.Kom., M.Kom
JUDUL : Pengembangan Aplikasi Arsip Digital Berbasis *OCR* Dan
Rule-Based System

NO	TGL. KONSULTASI	TOPIK POKOK YANG DIBICARAKAN	TANDA TANGAN PEMB. II	TGL MENG-HADAP KEMBALI
11.	26/2/2024	ERD & Tabel Relasi ACC	Rt.	
12.	2/3/2026	Revisi		
		Perancangan antar muka - sesuai dg standar.	Rt.	
13.	11/3/2026	Bab IV ACC	Rt.	
14.	1/4/2026	Bab V Revisi	Rt.	
15.	29/4/2026	Bab V Aplikasi ACC	Rt.	
16.	6/5/2026	Bab V Desain antar Muka ^{Revisi}	Rt.	12
17.	11/5/2026	Revisi - Pengujian sistem	Rt.	
18.	25/5/2026	Bab V ACC		
19.	26/5/2026	Bab VI ACC	Rt.	
20.	26/5/2026	ACC Sign off	Rt.	

Mulai Bimbingan :

Batas Akhir Bimbingan :

Mengetahui Kaprodi Informatika

PERHATIAN !
TIDAK BOLEH HILANG
SETIAP BIMBINGAN HARUS DIBAWA



IWAN SETIAWAN, S.Kom.M.Kom



**FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH**

PENILAIAN REVIEW SKRIPSI

Nama : Eci Wulandari
NIM : 2022230032
Judul Skripsi : Pengembangan Aplikasi Arsip Digital Berbasis OCR dan
Rule-Based System

Hari/Tanggal : KAMIS / 11-6-2026

Saran dan masukan :

IKUTI ARAHAN PENGUJI DENGAN BERKORDINASI
PADA PEMBIMPIG DAHULU

ACC SIAP JILID

17/6/2026

Rekomendasi*:

- a. Dilanjutkan tanpa perbaikan
- ☒ b. Dilanjutkan dengan perbaikan
- c. Ditolak

Prabumulih, 11 Juni 2026
Pembimbing I

(Iwan Setiawan, S.Kom., M.Kom)
NUPTK. 1541769670130303

*pilih salah satu



**FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH**

PENILAIAN REVIEW SKRIPSI

Nama : Eci Wulandari
NIM : 2022230032
Judul Skripsi : Pengembangan Aplikasi Arsip Digital Berbasis OCR dan
Rule-Based System
Hari/Tanggal :
Saran dan masukan :

Ikuti arahan penguji

ACC Sap Jilid skripsi

Rekomendasi*:

- a. Dilanjutkan tanpa perbaikan
- b. Dilanjutkan dengan perbaikan
- c. Ditolak

Prabumulih, 17/6/..... Juni 2026
Pembimbing II

Riza

(Riza Kartina, S.Kom., M.Kom.)
NUPTK.0549767668231073

**pilih salah satu*



**FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH**

PENILAIAN REVIEW SKRIPSI

Nama : Eci Wulandari
NIM : 2022230032
Judul Skripsi : Pengembangan Aplikasi Arsip Digital Berbasis OCR dan
Rule-Based System
Hari/Tanggal : Kamis, 11 - 06 - 2026
Saran dan masukan :

- Sistematisa Penulisan / Peckman
- Program - Activity
- Use Case
- Berikan Review Uji Sistem

Ace Rizki
GAP 15/6/2026

Rekomendasi*:

- a. Dilanjutkan tanpa perbaikan
- ☒ b. Dilanjutkan dengan perbaikan
- c. Ditolak

Prabumulih, Juni 2026
Dosen Penguji

(Khana Wijaya, S.Kom., M.Kom)
NUP TK. 7146768669130363

*pilih salah satu