

**PERANCANGAN MEDIA PEMBELAJARAN BERBASIS
ANDROID PADA MATA PELAJARAN KOMPUTER**

SKRIPSI



**M. AFRI RAHMAN
2022230037**

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

PERANCANGAN MEDIA PEMBELAJARAN BERBASIS *ANDROID* PADA MATA PELAJARAN KOMPUTER

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



M. AFRI RAHMAN
2022230037

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

HALAMAN PENGESAHAN

PERANCANGAN MEDIA PEMBELAJARAN BERBASIS *ANDROID* PADA MATA PELAJARAN KOMPUTER

SKRIPSI

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

Oleh :

M. Afri Rahman
2022230037

Prabumulih, 4 Juni 2026

Pembimbing I



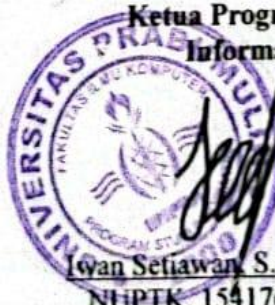
Iwan Setiawan, S.Kom., M.Kom.
NUPTK. 1541769670130303

Pembimbing II



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 “Perancangan Media Pembelajaran Berbasis *Android* Pada Mata Pelajaran Komputer” telah dipertahankan dan disetujui 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:

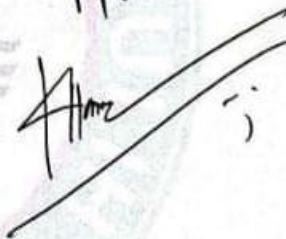
1. Riza Kartina, S.Kom., M.Kom

NUPTK. 0549767668231073

()

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

NUPTK. 7146768669130363

()

Mengetahui,

Dekan Fakultas Ilmu Komputer

Ketua Program Studi

Informatika


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


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

PERNYATAAN INTEGRITAS

Saya yang bertanda tangan di bawah ini

Nama : M. Afri Rahman

NIM : 2022230037

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, 4 Juni 2026

Yang membuat pernyataan,



M. Afri Rahman

NIM. 2022230037

ABSTRAK

Perancangan Media Pembelajaran Berbasis *Android* Pada Mata Pelajaran Komputer

M. Afri Rahman, 2022230037, 2026

Pembelajaran komputer di MTs Ishlahul Ummah merupakan proses penting dalam mendukung pemahaman siswa terhadap teknologi. Metode yang masih bersifat konvensional menyebabkan siswa kesulitan memahami konsep dasar komputer dan menurunkan motivasi belajar. Penelitian ini bertujuan untuk mengembangkan media pembelajaran berbasis *Android* sebagai solusi yang lebih interaktif, mudah diakses, dan sesuai dengan kebutuhan siswa. Media ini memanfaatkan visualisasi dan animasi untuk membantu pemahaman materi komputer, termasuk perangkat keras, perangkat lunak, dan sistem operasi. Hasil penelitian diharapkan dapat meningkatkan kualitas pembelajaran komputer di MTs Ishlahul Ummah secara lebih menarik, efektif, dan kontekstual sesuai perkembangan teknologi digital

Kata Kunci : Pembelajaran komputer, media pembelajaran berbasis *Android*, interaktif

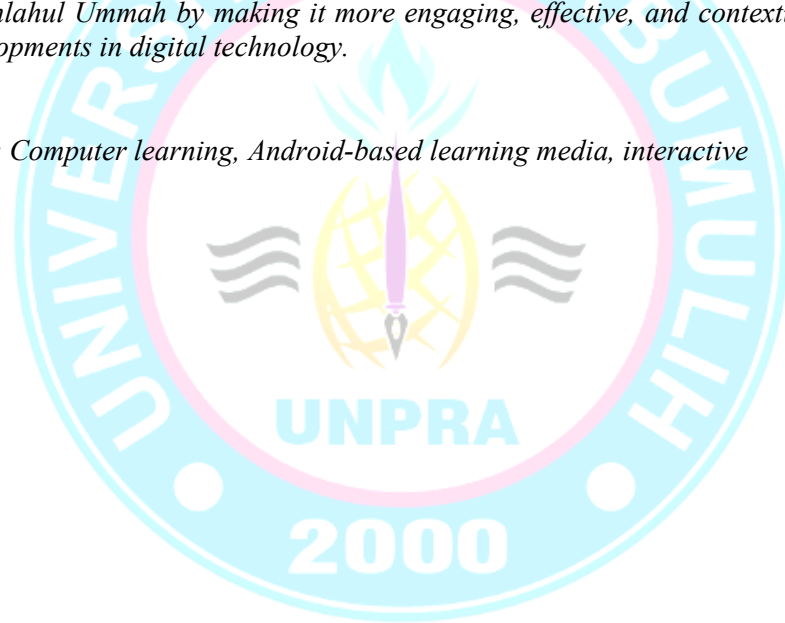
ABSTRACT

Designing Android-Based Learning Media for Computer Science Subjects

M. Afri Rahman, 2022230037, 2026

Computer learning at MTs Ishlahul Ummah is a crucial process in supporting students' understanding of technology. Conventional methods make it difficult for students to grasp basic computer concepts and reduce their motivation to learn. This research aims to develop Android-based learning media as a more interactive, accessible, and tailored solution to students' needs. This media utilizes visualization and animation to aid understanding of computer material, including hardware, software, and operating systems. The research results are expected to improve the quality of computer learning at MTs Ishlahul Ummah by making it more engaging, effective, and contextual, in line with developments in digital technology.

Keywords: *Computer learning, Android-based learning media, interactive*



KATA PENGANTAR

Puji dan syukur penulis ucapkan kehadiran Allah SWT atas limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan Skripsi yang berjudul Perancangan Media Pembelajaran Berbasis *Android* Pada Mata Pelajaran Komputer, 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
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 sekaligus pembimbing akademik.
4. Bapak Iwan Setiawan, S.Kom., M.Kom selaku Ketua Program Studi Informatika sekaligus Pembimbing I.
5. Ibu Riza Kartina, S.Kom., M.Kom sebagai pembimbing II.
6. Teruntuk Papa Hartomi Dan Mama Nuryanti yang tidak bisa menemani secara langsung namun selalu kekal dan abadi di hati penulis
7. Teruntuk Winda Kurniatullah, S.E, M.Mu Amar, S.Tr.Ak, dan M.Mu Amir, S.Pd. yang selalu mendukung dan selalu mengusahakan setiap jalan yang dipilih oleh adik bungsu mereka.
8. Bapak dan Ibu Dosen Program Studi Informatika Fakultas Ilmu Komputer Universitas Prabumulih.
9. Ibu Eva Kurnia Sari, S.Pd.I selaku Kepala Madrasah MTs Ishlahul Ummah yang telah mengizinkan penulis untuk melakukan penelitian
10. Guru dan staf MTs Ishlahul Ummah yang telah membantu penulis dalam melaksanakan penelitian
11. Untuk diri sendiri yang selalu berdiri tegak menghadapi segala masalah yang hadir walau tanpa dampingan dari papa dan mama

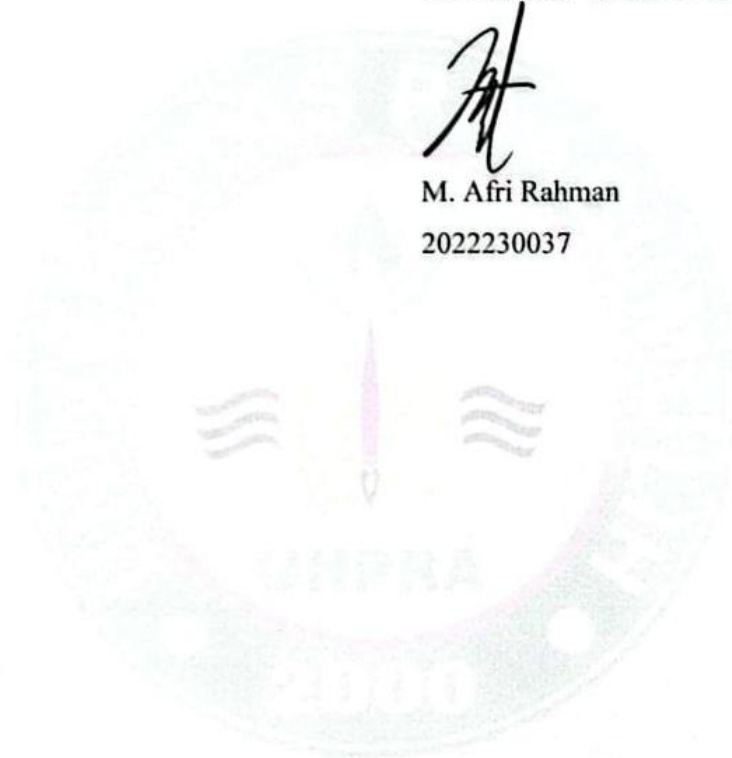
12. Untuk teman-teman angkatan informatika 2022 dan teman teman lain nya terima kasih karna telah menemani penulis dalam membuat dan menyelesaikan skripsi ini
 13. Terakhir untuk buku-bukuku yang selalu menemani dalam diam, terima kasih telah menjadi saksi dan bagian dari perjalanan menyelesaikan skripsi ini.
- Penulisan Skripsi ini masih terdapat kekurangan sehingga membutuhkan kritik dan saran yang bersifat membangun untuk kesempurnaan Skripsi.

Prabumulih, 4 Juni 2026



M. Afri Rahman

2022230037

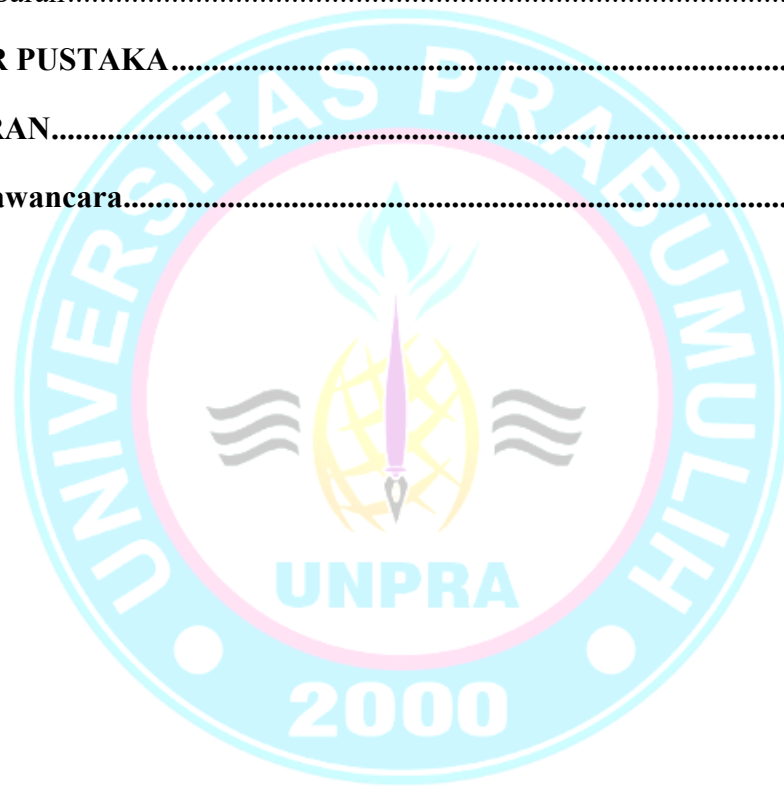


DAFTAR ISI

	Halaman
HALAMAN JUDUL DALAM	i
HALAMAN PENGESAHAN.....	ii
HALAMAN PERSETUJUAN	iii
PERNYATAAN INTEGRITAS	iv
ABSTRAK	v
<i>ABSTRACT</i>	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR TABEL	xii
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN	xvii
DAFTAR SINGKATAN.....	xviii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA	5
2.1 Pengertian Perancangan	5
2.2 Pengertian Media Pembelajaran.....	6
2.3 Pengertian Berbasis	6
2.4 Pengertian <i>Android</i>	7
2.5 Pengertian Pelajaran Komputer.....	8

2.6 Pengertian Media Pembelajaran Berbasis <i>Android</i>	9
2.7 Penelitian Terdahulu	10
BAB III METODE PENELITIAN	16
3.1 Objek dan Jadwal Penelitian	16
3.2 Metode Penelitian	20
3.3 Sumber Data	20
3.3.1 Data Primer	21
3.3.2 Data Sekunder	21
3.4 Teknik Pengumpulan Data	21
3.5 Metode Pengembangan Sistem	23
3.6 Alat Bantu Perancangan	25
3.6.1 Alat Bantu Utama	26
3.6.2 Alat Bantu Pemodelan Visual	27
3.6.3 Alat Bantu Desain	32
3.6.4 Alat Bantu Pengembangan dan <i>Debugging</i>	32
3.7 Pengujian Sistem	33
BAB IV ANALISIS DAN PERANCANGAN	34
4.1 Analisa Sistem	34
4.1.1 Analisa Sistem Berjalan	32
4.1.2 Analisa Kebutuhan	32
4.2 Perancangan Sistem yang Diusulkan	39
4.2.1 Perancangan Sistem (Logis)	32
4.2.2 Perancangan Data Base (Fisik)	58
4.2.3 Perancangan Antar Muka	60
BAB V HASIL DAN PEMBAHASAN	80
5.1 Spesifikasi Perangkat	80

5.1.1 Perangkat Keras Yang Digunakan	80
5.1.2 Perangkat Lunak Yang Digunakan	81
5.2 Implementasi Database	81
5.3 Tampilan Antarmuka.....	88
5.4 Hasil Pengujian Sistem.....	108
BAB VI KESIMPULAN DAN SARAN	212
6.1 Kesimpulan.....	212
6.2 Saran.....	213
DAFTAR PUSTAKA.....	215
LAMPIRAN.....	218
Draft Wawancara.....	223



DAFTAR TABEL

	Halaman
Tabel 2. 1 Penelitian Terdahulu	10
Tabel 3. 1 <i>Use Case Diagram</i>	27
Tabel 3. 2 <i>Activity Diagram</i>	29
Tabel 3. 3 Class Diagram	31
Tabel 4. 1 <i>User</i>	58
Tabel 4. 2 Materi	58
Tabel 4. 3 File Latihan Soal	59
Tabel 4. 4 Latihan Soal	59
Tabel 4. 5 Hasil Latihan Soal	59
Tabel 5. 1 Identifikasi <i>Statement</i> Fungsi <i>Login</i>	112
Tabel 5. 2 <i>Test Case</i> Fungsi <i>Login</i>	118
Tabel 5. 3 Identifikasi <i>Statement</i> Fungsi <i>Logout</i>	122
Tabel 5. 4 <i>Test Case</i> Fungsi <i>Logout</i>	127
Tabel 5. 5 Identifikasi <i>Statement</i> Fungsi Simpan Materi	130
Tabel 5. 6 <i>Test Case</i> Fungsi Simpan Materi	136
Tabel 5. 7 Identifikasi <i>Statement</i> Fungsi Hapus Materi	139
Tabel 5. 8 <i>Test Case</i> Fungsi Hapus Materi	143
Tabel 5. 9 Identifikasi <i>Statement</i> Fungsi Tambah Latihan	145
Tabel 5. 10 <i>Test Case</i> Fungsi Tambah Latihan	151
Tabel 5. 11 Identifikasi <i>Statement</i> Fungsi Hapus Latihan	152
Tabel 5. 12 <i>Test Case</i> Fungsi Hapus Latihan	156
Tabel 5. 13 Identifikasi <i>Statement</i> Fungsi Tambah Soal	158
Tabel 5. 14 <i>Test Case</i> Fungsi Tambah Soal	165
Tabel 5. 15 Identifikasi <i>Statement</i> Fungsi Hapus Soal	167
Tabel 5. 16 <i>Test Case</i> Fungsi Hapus Soal	171
Tabel 5. 17 Identifikasi <i>Statement</i> Fungsi <i>Next</i>	172
Tabel 5. 18 <i>Test Case</i> Fungsi <i>Next</i>	176
Tabel 5. 19 Identifikasi <i>Statement</i> Fungsi Hitung Nilai	177
Tabel 5. 20 <i>Test Case</i> Fungsi Hitung Nilai	180
Tabel 5. 21 Identifikasi <i>Statement</i> Fungsi Hitung Nilai	183
Tabel 5. 22 <i>Test Case</i> Fungsi Hitung Nilai	187
Tabel 5. 23 Identifikasi <i>Statement</i> Fungsi Tambah Akun <i>User</i>	190

Tabel 5. 24 <i>Test Case</i> Fungsi Tambah Akun <i>User</i>	196
Tabel 5. 25 Identifikasi <i>Statement</i> Fungsi Reset <i>Password</i>	200
Tabel 5. 26 <i>Test Case</i> Fungsi Reset <i>Password</i>	205
Tabel 5. 27 Identifikasi <i>Statement</i> Fungsi Hapus Akun <i>User</i>	207
Tabel 5. 28 <i>Test Case</i> Fungsi Hapus Akun <i>User</i>	211



DAFTAR GAMBAR

	Halaman
Gambar 2. 1 Kerangka Berpikir Penelitian	14
Gambar 3. 1 Struktur Organisasi.....	18
Gambar 3. 2 Metode Pengembangan Sistem <i>Design Thinking</i>	24
Gambar 4. 1 Rancangan <i>Use Case Diagram</i>	40
Gambar 4. 2 <i>Login Activity</i>	41
Gambar 4. 3 <i>Upload Materi Activity</i>	42
Gambar 4. 4 Mengedit Materi <i>Activity</i>	43
Gambar 4. 5 Menghapus Materi <i>Activity</i>	44
Gambar 4. 6 Menambahkan Latihan Soal <i>Activity</i>	45
Gambar 4. 7 Edit Latihan Soal <i>Activity</i>	46
Gambar 4. 8 Hapus Latihan Soal <i>Activity</i>	47
Gambar 4. 9 Melihat Materi <i>Activity</i>	48
Gambar 4. 10 Mengerjakan Latihan Soal <i>Activity</i>	49
Gambar 4. 11 Melihat dan Mengunduh Hasil Latihan Soal <i>Activity</i>	50
Gambar 4. 12 Melihat Hasil Latihan Soal <i>Activity</i>	51
Gambar 4. 13 Menambahkan Akun Siswa	52
Gambar 4. 14 Reset <i>Password</i>	53
Gambar 4. 15 Menghapus Akun Siswa	54
Gambar 4. 16 <i>Logout Umum User</i>	55
Gambar 4. 17 <i>Class Diagram</i> Login Umum	56
Gambar 4. 18 <i>Class Diagram</i> Kelola User	56
Gambar 4. 19 <i>Class Diagram</i> Materi	57
Gambar 4. 20 <i>Class Diagram</i> Latihan Soal.....	57
Gambar 4. 21 <i>Class Diagram</i> Hasil Latihan Soal	58
Gambar 4. 22 Halaman <i>Login</i>	60
Gambar 4. 23 Halaman Menu Utama.....	61
Gambar 4. 24 Halaman Materi Guru.....	62
Gambar 4. 25 Halaman Menambahkan Materi	63
Gambar 4. 26 Halaman Edit Materi	64
Gambar 4. 27 Halaman Hapus Materi.....	65
Gambar 4. 28 Halaman Materi Siswa	66
Gambar 4. 29 Halaman File Latihan Soal	67

Gambar 4. 30 Halaman Menambahkan File Latihan Soal.....	68
Gambar 4. 31 Halaman Latihan Soal Guru	69
Gambar 4. 32 Halaman Menambahkan Latihan Soal	70
Gambar 4. 33 Halaman Edit Latihan Soal.....	71
Gambar 4. 34 Halaman Hapus Latihan Soal	72
Gambar 4. 35 Halaman Latihan Soal Siswa.....	73
Gambar 4. 36 Halaman Hasil Latihan Soal Guru	74
Gambar 4. 37 Halaman Hasil Latihan Soal Siswa.....	75
Gambar 4. 38 Halaman Guru Kelola <i>User</i>	76
Gambar 4. 39 Halaman Tambah Akun Siswa	77
Gambar 4. 40 Halaman Reset <i>Password</i> Akun Siswa	78
Gambar 4. 41 Halaman Hapus Akun Siswa	79
Gambar 5. 1 Tabel <i>Users</i>	82
Gambar 5. 2 Table Materi <i>Hardware</i>	83
Gambar 5. 3 Tabel Materi <i>Software</i>	84
Gambar 5. 4 Table File Latihan Soal	85
Gambar 5. 5 Tabel Soal	86
Gambar 5. 6 Tabel Hasil Latihan	87
Gambar 5. 7 Halaman <i>Login</i>	88
Gambar 5. 8 Halaman Menu Utama.....	89
Gambar 5. 9 Halaman Materi.....	90
Gambar 5. 10 Menambahkan Materi.....	91
Gambar 5. 11 Edit Materi	92
Gambar 5. 12 Hapus Materi.....	93
Gambar 5. 13 Halaman Materi Siswa	94
Gambar 5. 14 File Latihan Soal	95
Gambar 5. 15 Menambahkan File Latihan Soal	96
Gambar 5. 16 Halaman Latihan Soal Guru	97
Gambar 5. 17 Menambahkan Soal	98
Gambar 5. 18 <i>Edit</i> Latihan Soal.....	99
Gambar 5. 19 Hapus Latihan Soal	100
Gambar 5. 20 Halaman Soal Siswa.....	101
Gambar 5. 21 Halaman Hasil Latihan Guru.....	102
Gambar 5. 22 Halaman Hasil Latihan Siswa.....	103
Gambar 5. 23 Halaman Guru Kelola <i>User</i>	104

Gambar 5. 24 Halaman Tambah Akun Siswa	105
Gambar 5. 25 <i>Reset Password</i> Akun Siswa	106
Gambar 5. 26 Halaman Hapus Akun Siswa	107
Gambar 5. 27 <i>Flowchart</i> Fungsi <i>Login</i>	113
Gambar 5. 28 <i>Flowgraph</i> Fungsi <i>Login</i>	115
Gambar 5. 29 <i>Flowchart</i> Fungsi <i>Logout</i>	123
Gambar 5. 30 <i>Flowgraph</i> Fungsi <i>Logout</i>	124
Gambar 5. 31 <i>Flowchart</i> Fungsi <i>Simpan Materi</i>	130
Gambar 5. 32 <i>Flowgraph</i> Fungsi <i>Simpan Materi</i>	132
Gambar 5. 33 <i>Flowchart</i> Fungsi <i>Hapus Materi</i>	140
Gambar 5. 34 <i>Flowgraph</i> Fungsi <i>Hapus Materi</i>	141
Gambar 5. 35 <i>Flowchart</i> Fungsi <i>Tambah Latihan</i>	146
Gambar 5. 36 <i>Flowgraph</i> Fungsi <i>Tambah Latihan</i>	148
Gambar 5. 37 <i>Flowchart</i> Fungsi <i>Hapus Latihan</i>	153
Gambar 5. 38 <i>Flowgraph</i> <i>Hapus Latihan</i>	154
Gambar 5. 39 <i>Flowchart</i> Fungsi <i>Tambah Soal</i>	160
Gambar 5. 40 <i>Flowgraph</i> Fungsi <i>Tambah Soal</i>	162
Gambar 5. 41 <i>Flowchart</i> Fungsi <i>Hapus Soal</i>	168
Gambar 5. 42 <i>Flowgraph</i> Fungsi <i>Hapus Soal</i>	169
Gambar 5. 43 <i>Flowchart</i> Fungsi <i>Next</i>	173
Gambar 5. 44 <i>Flowgraph</i> Fungsi <i>Next</i>	174
Gambar 5. 45 <i>Flowchart</i> Fungsi <i>Hitung Nilai</i>	178
Gambar 5. 46 <i>Flowgraph</i> Fungsi <i>Hitung Nilai</i>	179
Gambar 5. 47 <i>Flowchart</i> Fungsi <i>Generate PDF</i>	184
Gambar 5. 48 <i>Flowgraph</i> Fungsi <i>Generate PDF</i>	185
Gambar 5. 49 <i>Flowchart</i> Fungsi <i>Tambah Akun User</i>	192
Gambar 5. 50 <i>Flowgraph</i> Fungsi <i>Tambah Akun User</i>	194
Gambar 5. 51 <i>Flowchart</i> Fungsi <i>Reset Password</i>	201
Gambar 5. 52 <i>Flowgraph</i> Fungsi <i>Reset Password</i>	202
Gambar 5. 53 <i>Flowchart</i> Fungsi <i>Hapus Akun User</i>	208
Gambar 5. 54 <i>Flowgraph</i> Fungsi <i>Hapus Akun User</i>	209

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Surat Permohonan Penelitian.....	218
Lampiran 2 Surat Balasan Permohonan Penelitian	219
Lampiran 3 Dokumentasi Wawancara & Observasi.....	220



DAFTAR SINGKATAN

IDE	: <i>Integrated Development Environment</i>
UI	: <i>User Interface</i>
UX	: <i>User Experience</i>
MTs	: Madrasah Tsanawiyah
API	: <i>Application Programming Interface</i>
ADDIE	: <i>Analysis, Design, Development, Implementation, Evaluation</i>
DDD-E	: <i>Decide, Design, Develop, Evaluate</i>
R&D	: <i>Research and Development</i>
SAC	: <i>Smart Apps Creator</i>
APK	: <i>Android Package Kit</i>
SMK	: Sekolah Menengah Kejuruan
SMKN	: Sekolah Menengah Kejuruan Negeri
UML	: <i>Unified Modeling Language</i>



BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam proses pembelajaran di era digital saat ini, penggunaan media yang inovatif sangat dibutuhkan agar siswa lebih mudah memahami materi. Pembelajaran komputer di MTs Ishlahul Ummah masih menggunakan metode konvensional berupa ceramah dan penjelasan langsung dari guru dengan bantuan papan tulis dan buku teks. Pendekatan seperti ini membuat siswa kesulitan memahami konsep dasar komputer, terutama mengenai perangkat keras, perangkat lunak, serta fungsi sistem operasi dan latihan soal sesuai dengan materi yang di ajarkan. Kurangnya media pembelajaran visual dan interaktif juga menyebabkan suasana belajar menjadi monoton, sehingga motivasi dan partisipasi siswa dalam pembelajaran menurun.

Kondisi tersebut menunjukkan perlunya inovasi pembelajaran yang dapat membantu siswa memahami materi dengan cara yang lebih menarik dan mudah dipahami. Salah satu alternatif yang potensial adalah pemanfaatan media pembelajaran berbasis *Android*, karena perangkat ini mudah diakses, bersifat terbuka (*open source*), dan sudah akrab digunakan oleh siswa. *Android* juga memungkinkan pengembangan aplikasi pembelajaran interaktif yang mendukung pembelajaran di mana saja dan kapan saja, sesuai dengan karakteristik generasi digital saat ini.

Media pembelajaran berbasis *Android* dapat meningkatkan minat belajar dan pemahaman siswa terhadap materi komputer melalui visualisasi dan animasi interaktif (Haryadi et al., 2024). Media semacam ini memberikan pengalaman belajar yang lebih menyenangkan sekaligus mengatasi keterbatasan fasilitas komputer di sekolah. Selain itu, hasil penelitian tersebut juga menegaskan pentingnya penerapan teknologi dalam meningkatkan efektivitas dan kemandirian belajar siswa.

Media pembelajaran berbasis digital merupakan konsekuensi logis dari perkembangan teknologi informasi dan komunikasi yang mendorong terciptanya

pembelajaran yang fleksibel, interaktif, dan berorientasi pada kebutuhan peserta didik (Afriyadi et al., 2023). Media digital memungkinkan penyajian materi melalui kombinasi teks, gambar, suara, dan video secara terpadu, sehingga meningkatkan motivasi dan keterlibatan siswa dalam proses belajar. Dengan demikian, media digital tidak hanya menjadi alat bantu, tetapi juga komponen penting dalam mewujudkan pembelajaran yang relevan dengan tuntutan pada saat ini.

Dari latar belakang tersebut dapat disimpulkan bahwa, integrasi media pembelajaran berbasis *Android* menjadi langkah strategis untuk meningkatkan kualitas pembelajaran komputer di MTs Ishlahul Ummah. Melalui penerapan media ini, diharapkan siswa dapat memahami konsep dasar komputer dengan lebih baik, termotivasi untuk belajar secara mandiri, serta memperoleh pengalaman belajar yang menarik dan kontekstual. Selain itu, penggunaan media berbasis *Android* juga diharapkan dapat membantu guru menyampaikan materi secara lebih inovatif dan efisien, sehingga mendukung transformasi pendidikan yang adaptif terhadap perkembangan teknologi di era digital.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana merancang media pembelajaran komputer berbasis *Android* yang dapat mendukung proses belajar mengajar di MTs Ishlahul Ummah?
2. Bagaimana mengembangkan media pembelajaran berbasis *Android* yang mudah digunakan serta mampu meningkatkan motivasi dan pemahaman siswa dalam mempelajari materi komputer?
3. Bagaimana merancang media pembelajaran berbasis *Android* yang mudah diakses melalui berbagai perangkat *mobile* siswa?

1.3 Batasan Masalah

Agar penelitian ini lebih fokus dan terarah, maka dibatasi pada ruang lingkup sebagai berikut:

1. Media pembelajaran yang dikembangkan difokuskan pada mata pelajaran komputer di MTs Ishlahul Ummah, dengan konten yang

mencakup materi dasar komputer serta latihan soal yang disesuaikan dengan materi yang diajarkan.

2. Media pembelajaran dirancang berbasis *Android* agar dapat diakses menggunakan perangkat *mobile* siswa, seperti *smartphone* dan tablet.
3. Akses terhadap aplikasi media pembelajaran dibatasi hanya untuk siswa dan guru yang bersangkutan, guna menjaga keamanan data serta memastikan penggunaan aplikasi sesuai dengan tujuan pembelajaran di lingkungan MTs Ishlahul Ummah.

1.4 Tujuan Penelitian

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah merancang media pembelajaran komputer berbasis *Android* yang sesuai dengan kebutuhan proses belajar mengajar di MTs Ishlahul Ummah. Media pembelajaran yang dikembangkan diharapkan mudah digunakan dan bersifat interaktif sehingga mampu meningkatkan motivasi belajar serta pemahaman siswa terhadap materi dasar komputer. Selain itu, penelitian ini juga bertujuan menghasilkan media pembelajaran berbasis *Android* yang responsif dan dapat diakses secara efektif melalui berbagai perangkat *mobile* yang digunakan oleh siswa.

1.5 Manfaat Penelitian

Penelitian ini diharapkan memberikan manfaat sebagai berikut:

1. Memberikan kemudahan bagi siswa dalam memahami materi komputer melalui media pembelajaran berbasis *Android* yang interaktif dan dapat diakses kapan saja.
2. Membantu guru dalam menyampaikan materi pembelajaran komputer secara lebih variatif, efektif, dan menarik.
3. Mendukung pemanfaatan teknologi informasi di lingkungan sekolah untuk meningkatkan kualitas pembelajaran serta menumbuhkan kemandirian belajar siswa.

1.6 Sistematika Penulisan

Agar pembahasan dalam penelitian ini lebih terstruktur, skripsi ini disusun dengan sistematika sebagai berikut:

1. BAB I PENDAHULUAN

Bab ini berisi latar belakang penelitian, rumusan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

2. BAB II TINJAUAN PUSTAKA

Bab ini berisi teori-teori yang mendukung penelitian, hasil penelitian terdahulu yang relevan, serta kerangka berpikir yang menjadi dasar dalam perancangan media pembelajaran berbasis *Android* pada mata pelajaran komputer.

3. BAB III METODE PENELITIAN

Bab ini menjelaskan metode penelitian yang digunakan, meliputi jenis penelitian, lokasi dan waktu penelitian, metode pengumpulan data, analisis kebutuhan sistem, perancangan sistem, serta tahapan pengembangan sistem

4. BAB IV ANALISA DAN PERANCANGAN

Bab ini membahas hasil analisis kebutuhan sistem yang meliputi analisis masalah, analisis pengguna, serta analisis kebutuhan perangkat lunak dan perangkat keras. Selain itu, bab ini juga menjelaskan perancangan sistem, meliputi perancangan alur sistem, perancangan basis data, perancangan antarmuka (UI), serta desain media pembelajaran berbasis *Android* pada mata pelajaran komputer.

5. BAB V HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil implementasi dari perancangan sistem yang telah dibuat, meliputi tampilan dan fungsi media pembelajaran berbasis *Android*. Selain itu, bab ini juga membahas hasil pengujian sistem, evaluasi kinerja aplikasi, serta pembahasan terhadap hasil penelitian yang diperoleh.

6. BAB VI KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang diambil berdasarkan hasil penelitian dan pembahasan yang telah dilakukan. Selain itu, bab ini juga menyajikan saran-saran yang diharapkan dapat menjadi bahan pertimbangan untuk pengembangan sistem dan penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Pengertian Perancangan

Perancangan merupakan suatu proses sistematis yang dilakukan untuk merancang, mengembangkan, dan menyusun suatu produk atau sistem agar sesuai dengan tujuan yang telah ditetapkan (Resa Ayu et al., 2022). Tahapan ini meliputi analisis kebutuhan, perencanaan struktur, serta pembuatan rancangan antarmuka dan alur kerja aplikasi agar sesuai dengan tujuan pembelajaran dan karakteristik pengguna. Melalui proses perancangan yang terarah, media pembelajaran yang dihasilkan akan lebih mudah dioperasikan, menarik, serta relevan dengan kebutuhan belajar siswa di era digital.

Perancangan berfungsi sebagai jembatan antara tahap analisis kebutuhan dan implementasi sistem (Fatqurhohman et al., 2023). Tahapan ini berperan penting dalam memastikan bahwa produk yang dikembangkan benar-benar memenuhi kebutuhan pengguna serta dapat diterapkan secara optimal. Dalam bidang rekayasa perangkat lunak maupun pengembangan media pembelajaran, perancangan tidak hanya menitikberatkan pada aspek teknis, tetapi juga memperhatikan faktor ergonomi, estetika, dan *User experience* agar hasil rancangan tidak hanya fungsional, tetapi juga nyaman digunakan.

Berdasarkan kedua pendapat tersebut, dapat disimpulkan bahwa perancangan merupakan tahapan penting yang menentukan keberhasilan suatu sistem atau media pembelajaran. Perancangan tidak hanya berfokus pada penyusunan struktur dan alur kerja, tetapi juga memperhatikan interaksi pengguna serta kemudahan akses. Dalam konteks pengembangan media pembelajaran, perancangan berperan dalam menentukan penyajian materi, pola interaksi, dan tampilan aplikasi agar mampu menciptakan pengalaman belajar yang efektif, menarik, dan bermakna bagi peserta didik.

2.2 Pengertian Media Pembelajaran

Media pembelajaran merupakan segala bentuk alat, bahan, atau sarana yang digunakan untuk menyalurkan pesan dan informasi dalam proses belajar mengajar guna membantu peserta didik memahami materi secara lebih efektif (Tugiman et al., 2024). Media ini dapat berupa teks, gambar, audio, video, grafik, animasi, maupun interaktivitas yang diakses melalui aplikasi atau platform berbasis teknologi. Pemanfaatan media tersebut membuat kegiatan pembelajaran lebih menarik, tidak monoton, serta mampu menciptakan suasana belajar yang menyenangkan dan bermakna. Selain itu, media berperan sebagai jembatan antara guru dan peserta didik dalam menyampaikan pesan pembelajaran secara lebih jelas, konkret, dan mudah dipahami.

Media pembelajaran memberikan keleluasaan bagi guru untuk berinovasi dalam menyusun materi sesuai karakteristik peserta didik dan tujuan pembelajaran (Tugiman et al., 2024). Sementara itu, siswa memperoleh kesempatan belajar secara mandiri tanpa dibatasi ruang dan waktu. Melalui penerapan *digital learning*, peserta didik dapat berinteraksi langsung dengan materi, mengeksplorasi pelajaran secara lebih mendalam, serta membangun pemahaman yang kontekstual. Penggunaan media yang interaktif, seperti *interactive media*, terbukti mampu meningkatkan motivasi dan hasil belajar karena menggabungkan unsur teks, gambar, audio, dan video secara terpadu.

Berdasarkan pendapat tersebut, dapat disimpulkan bahwa media pembelajaran memiliki peran strategis dalam meningkatkan kualitas proses belajar mengajar di era teknologi. Media tidak hanya menjadi alat bantu penyampaian informasi, tetapi juga sarana pembentuk pengalaman belajar yang aktif, kreatif, dan mandiri. Melalui pemanfaatan *e-learning* dan *interactive media*, proses pembelajaran dapat berlangsung lebih fleksibel, komunikatif, serta menumbuhkan kemampuan berpikir kritis dan pemecahan masalah yang dibutuhkan dalam menghadapi tantangan pendidikan modern.

2.3 Pengertian Berbasis

Istilah berbasis dalam konteks pendidikan menunjukkan bahwa suatu sistem, metode, atau kegiatan pembelajaran berlandaskan pada komponen tertentu

yang menjadi dasar pelaksanaannya. Istilah ini digunakan untuk menegaskan bahwa kegiatan pembelajaran memiliki titik tumpu atau fondasi yang dijadikan acuan utama dalam pengembangan dan pelaksanaannya. Dengan demikian, ketika dikatakan berbasis digital atau berbasis *Android*, artinya seluruh proses dan instrumen pembelajaran dibangun serta dijalankan menggunakan teknologi digital atau sistem *Android* sebagai pijakan utama dalam mendukung tercapainya tujuan pendidikan (Wahyuni et al., 2022). Konsep ini juga menandakan adanya perubahan paradigma dari pembelajaran konvensional menuju pembelajaran modern yang lebih adaptif terhadap perkembangan teknologi informasi.

Istilah berbasis *Android* mengacu pada kegiatan atau aplikasi yang dikembangkan dengan menggunakan sistem operasi *Android* sebagai fondasi teknologinya (Maharani et al., 2023). *Android* dijelaskan sebagai sistem operasi berbasis *Linux* yang dirancang khusus untuk perangkat bergerak (*mobile device*), dengan karakteristik *open source* yang memungkinkan pengguna untuk mengembangkan, memodifikasi, serta menyesuaikan aplikasi sesuai kebutuhan. Dengan demikian, penggunaan istilah berbasis tidak hanya menunjukkan ketergantungan teknis terhadap suatu platform, tetapi juga menggambarkan adanya hubungan konseptual dan fungsional antara teknologi yang digunakan dengan tujuan pembelajaran yang ingin dicapai.

Berdasarkan kedua pendapat tersebut, dapat disimpulkan bahwa istilah berbasis mencerminkan fondasi atau landasan utama dari suatu kegiatan, sistem, maupun media pembelajaran. Dalam ranah pendidikan modern, istilah ini sering digunakan untuk menegaskan bahwa kegiatan pembelajaran dirancang dan dikembangkan dengan mengandalkan teknologi tertentu sebagai sarana utama untuk mencapai efektivitas, fleksibilitas, serta inovasi dalam proses belajar mengajar.

2.4 Pengertian *Android*

Android merupakan sistem operasi berbasis *Linux* yang bersifat *open source* dan dikembangkan oleh Google untuk perangkat bergerak seperti smartphone dan tablet (Ady Prabowo et al., 2021). Sistem ini dirancang dengan arsitektur terbuka yang memberikan kebebasan kepada para pengembang untuk

membuat, memodifikasi, serta mengembangkan aplikasi sesuai kebutuhan pengguna. *Android* juga menyediakan berbagai pustaka pendukung dan *Application Programming Interface* (API) yang mempermudah proses pengembangan aplikasi interaktif di berbagai bidang, termasuk bidang pendidikan. Dengan dukungan tersebut, *Android* menjadi salah satu platform yang paling populer dan banyak digunakan dalam pengembangan media pembelajaran modern yang inovatif dan menarik bagi peserta didik.

Salah satu keunggulan *Android* terletak pada kemampuannya menjalankan aplikasi dengan tampilan antarmuka yang sederhana dan mudah digunakan oleh berbagai kalangan (Wijaya et al., 2024). Sistem ini juga mampu menyesuaikan tampilan dengan berbagai ukuran layar perangkat, sehingga memberikan pengalaman pengguna yang optimal. Dalam konteks pendidikan, *Android* memberi peluang besar untuk mengembangkan media pembelajaran inovatif karena mendukung berbagai elemen multimedia seperti video, gambar, animasi, dan suara yang dapat meningkatkan daya tarik serta efektivitas proses belajar mengajar.

Berdasarkan kedua pendapat tersebut, dapat disimpulkan bahwa *Android* merupakan platform yang ideal untuk pengembangan media pembelajaran digital karena bersifat terbuka, fleksibel, dan mudah diakses. Dengan sifatnya yang portabel, media pembelajaran berbasis *Android* memungkinkan siswa belajar di mana saja dan kapan saja tanpa batasan ruang maupun waktu. Selain itu, dukungan fitur multimedia menjadikan *Android* sebagai sarana efektif dan efisien untuk menciptakan pembelajaran yang interaktif, menarik, serta relevan dengan kebutuhan pendidikan di era digital.

2.5 Pengertian Pelajaran Komputer

Komputer merupakan seperangkat alat elektronik yang digunakan untuk melakukan *input* data, kemudian data tersebut diproses menggunakan instruksi tertentu sesuai dengan program yang dijalankan, dan menghasilkan *output* berupa informasi yang telah diolah sehingga menjadi lebih bermakna bagi penggunanya (Ahmad et al., 2023). Berdasarkan pengertian tersebut, pelajaran komputer dapat diartikan sebagai mata pelajaran yang bertujuan membekali peserta didik dengan kemampuan memahami cara kerja komputer, mengoperasikan perangkat keras

(*hardware*) dan perangkat lunak (*software*), serta menerapkan teknologi informasi dalam kehidupan sehari-hari. Pelajaran komputer juga membantu siswa memahami konsep dasar logika dan algoritma yang menjadi fondasi dalam pengolahan data.

Pelajaran komputer memiliki peran penting dalam meningkatkan literasi digital siswa dan menumbuhkan keterampilan berpikir tingkat tinggi (Ahmad et al., 2023). Di era digital saat ini, pelajaran komputer tidak hanya menekankan pada aspek teknis penggunaan perangkat, tetapi juga berperan dalam mengembangkan kemampuan berpikir kritis, kreatif, kolaboratif, dan komunikatif. Dalam prosesnya, guru berperan sebagai fasilitator yang membantu siswa memahami penerapan teknologi secara kontekstual, sehingga siswa mampu menggunakan komputer tidak hanya sebagai alat bantu belajar, tetapi juga sebagai sarana inovasi dan pemecahan masalah dalam kehidupan nyata.

Berdasarkan pendapat tersebut, dapat disimpulkan bahwa pelajaran komputer merupakan mata pelajaran yang berfungsi membentuk literasi digital dan kesiapan siswa menghadapi perkembangan teknologi informasi yang pesat. Melalui kegiatan pelajaran komputer, siswa dapat belajar dengan cara yang lebih interaktif dan aplikatif sehingga mampu memahami prinsip dasar sistem komputer dan jaringan. Dengan demikian, pelajaran komputer tidak hanya meningkatkan kompetensi teknis siswa, tetapi juga menumbuhkan kemampuan berpikir logis, kreatif, dan adaptif dalam menghadapi tantangan di era digital.

2.6 Pengertian Media Pembelajaran Berbasis *Android*

Media pembelajaran berbasis *Android* merupakan sarana yang mengintegrasikan teknologi *mobile* dengan proses belajar mengajar untuk membantu peserta didik memahami materi secara lebih menarik dan interaktif (Wahyuni et al., 2022). Media ini dikembangkan dalam bentuk aplikasi yang dapat dijalankan pada perangkat *smartphone* atau *tablet* berbasis *Android*, berisi materi pembelajaran yang disusun secara sistematis sesuai tujuan pendidikan. Melalui penerapan *mobile learning*, siswa dapat belajar secara mandiri dengan memanfaatkan fitur seperti animasi, video pembelajaran, *interactive quiz*, dan simulasi yang dirancang agar sesuai dengan karakteristik peserta didik. Dengan

demikian, media berbasis *Android* berperan penting dalam menciptakan proses pembelajaran yang inovatif dan adaptif terhadap perkembangan teknologi.

Media pembelajaran berbasis *Android* memberikan kemudahan bagi guru dalam menyampaikan materi ajar (Wahyuni et al., 2022). Guru dapat mengemas konten secara menarik, memperbarui materi sesuai kurikulum, serta menyesuaikan penyajiannya dengan kebutuhan dan gaya belajar siswa. Keunggulan lain media ini adalah fleksibilitasnya, karena memungkinkan peserta didik belajar kapan pun dan di mana pun tanpa dibatasi oleh ruang dan waktu.

Berdasarkan pandangan tersebut, dapat disimpulkan bahwa media pembelajaran berbasis *Android* memiliki peran strategis dalam meningkatkan kualitas pembelajaran melalui pemanfaatan teknologi digital yang interaktif. Media ini berfungsi tidak hanya sebagai alat bantu visual, tetapi juga sebagai sarana yang mampu meningkatkan motivasi belajar dan menciptakan suasana belajar yang menyenangkan, kontekstual, serta sesuai dengan tuntutan pendidikan di era digital. Selain itu, keberadaan media ini juga mendorong terciptanya proses pembelajaran yang lebih efisien dan relevan dengan kebiasaan belajar generasi modern yang akrab dengan perangkat digital.

2.7 Penelitian Terdahulu

Penelitian terdahulu merupakan salah satu rujukan penting dalam penyusunan penelitian ini karena memberikan gambaran mengenai perkembangan kajian pada topik yang sejenis. Dengan menelaah berbagai studi sebelumnya, peneliti dapat memahami pendekatan, metode, serta temuan yang telah diperoleh oleh peneliti lain. Hal ini membantu menghindari terjadinya duplikasi penelitian serta membuka peluang untuk mengidentifikasi celah atau aspek yang belum banyak dikaji. Beberapa penelitian yang relevan dan dijadikan referensi dalam penelitian ini disajikan pada tabel 2.1 berikut

Tabel 2. 1 Penelitian Terdahulu

No	Judul Artikel	Penulis (Tahun)	Rangkuman
1	Pengembangan Media Pembelajaran Interaktif Berbasis <i>Android</i> pada	Salman Alfarizi , Aditya Prapanca (2021)	Penelitian ini mengembangkan media pembelajaran interaktif berbasis <i>Android</i> bernama “WE

	Mata Pelajaran 3D di SMKN 2 Surabaya		PLAY 3D” untuk mendukung pembelajaran mata pelajaran 3D di SMKN 2 Surabaya. Media dikembangkan menggunakan model ADDIE dan berisi materi dasar, Hasil uji validitas menunjukkan kategori sangat valid
2	Pengembangan Media Pembelajaran Berbasis <i>Android</i> – Pembelajaran Inovatif Untuk Meningkatkan Kompetensi Siswa Smk	Said Thaha Ghafara, Wakhinuddin Simatupang, Ambiyar, Mukhlidi Muskhir, Dedy Irfan (2022)	Penelitian ini bertujuan untuk mengembangkan dan menghasilkan media pembelajaran berbasis <i>Android</i> (mobile learning) dalam bentuk aplikasi yang bisa diakses menggunakan smartphone untuk siswa kelas 10 Sekolah Menengah Kejuruan.
3	Perancangan Media Pembelajaran Berbasis <i>Android</i> pada Mata Pelajaran Simulasi dan Komunikasi Digital	Risda Wati, Yuliawati Yunus, Popi Radyuli (2023)	Penelitian ini mengembangkan media pembelajaran berbasis <i>Android</i> untuk mata pelajaran Simulasi dan Komunikasi Digital menggunakan model ADDIE. Produk dirancang dengan Adobe Flash CS5 dan memuat fitur materi, video, evaluasi, serta petunjuk penggunaan.
4	Pengembangan Media Pembelajaran Berbasis <i>Android</i> Materi Sistem Perawatan Baterai Pada Kelas XI SMK N 11 Malang	Ichwan Ma'ruf Adityawarman, Syarif Suhartadi, Mokhammad Samsul Huda (2023)	Penelitian ini mengembangkan media pembelajaran berbasis <i>Android</i> pada materi Sistem Perawatan Baterai untuk siswa kelas XI SMK. Pengembangan dilakukan menggunakan model R&D Borg & Gall melalui sepuluh tahap, mulai dari identifikasi masalah hingga produksi massal.
5	Pengembangan Media Pembelajaran Berbasis <i>Android</i> pada Mata Pelajaran Sistem	I Made Arya Riananda Putra, Made Windu Antara Kesiman, I Gede Mahendra Darmawiguna (2023)	Penelitian ini mengembangkan media pembelajaran berbasis <i>Android</i> pada mata pelajaran Sistem Komputer mengenai struktur dan fungsi komputer untuk

	Komputer di Kelas X SMKN 1 Manggis		siswa kelas X. Pengembangan dilakukan menggunakan model ADDIE.
6	Pengembangan Media Pembelajaran Informatika Berbasis <i>Android</i> Menggunakan Google Sites dan Appsgeyser	Mita Sendia Ulfa, Riska Wulandari, Hendra Kurniawan (2023)	Penelitian ini menghasilkan media pembelajaran informatika berbasis <i>Android</i> yang dibuat melalui Google Sites dan Appsgeyser dengan model pengembangan DDD-E. Aplikasi menyediakan menu materi, video, tugas, serta profil, dan dikemas dalam bentuk file .apk agar dapat digunakan kapan saja.
7	Pengembangan Aplikasi Media Pembelajaran Interaktif Berbasis <i>Android</i> Tanpa Koding di Sekolah Menengah Kejuruan	Santi Ariani, Achmad Noor Fatirul, Atiqoh (2023)	Penelitian ini mengembangkan aplikasi media pembelajaran interaktif berbasis <i>Android</i> tanpa koding dengan menggunakan Smart Apps Creator (SAC). Aplikasi ditujukan untuk membantu pembelajaran materi Peluang pada siswa SMK. Hasil validasi menunjukkan bahwa produk memperoleh nilai sangat layak dari ahli materi, media, dan desain.
8.	Pengembangan Media Pembelajaran Berbasis <i>Android</i> dan HTML Berbantu Lectora Inspire Pada Materi Jurnal Umum di SMK Muhammadiyah 04 Medan	Chairina Rizkianda, Andri Zainal, Haryadi, Tuti Sriwedari, Ramdhansyah (2023)	Hasil penelitian ini menunjukkan bahwa media pembelajaran interaktif berbasis <i>Android</i> dan HTML yang telah dikembangkan dengan model ADDIE layak, praktis dan efektif untuk diterapkan ke dalam pembelajaran
9.	Media Pembelajaran Pengenalan Hardware dan Software Berbasis <i>Android</i> pada SMK	Octadino Haryadi, Yola Permata Bunda, Erni Rouza (2024)	Penelitian mengembangkan aplikasi <i>Android</i> berbasis Adobe Animate menggunakan metode MDLC. Aplikasi berisi materi

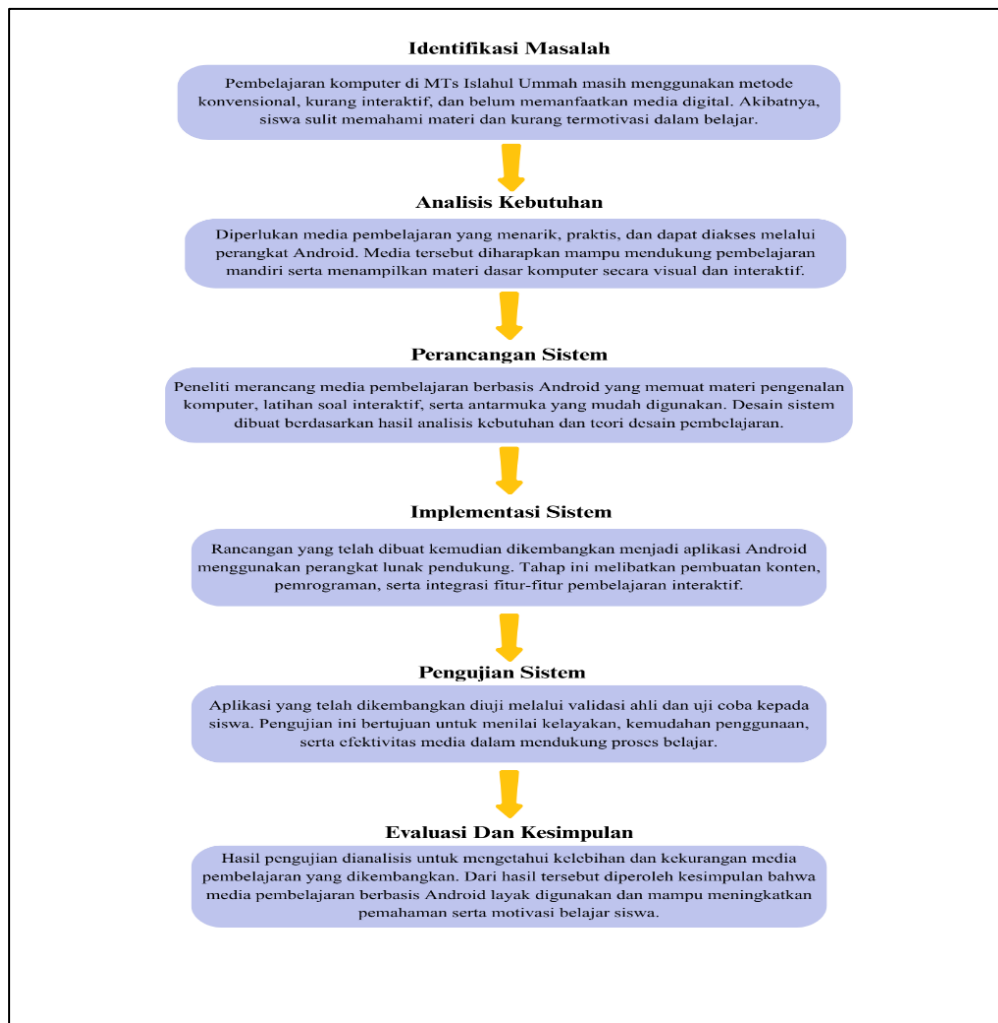
		hardware, software, dan kuis interaktif.
10.	Pengembangan Media Pembelajaran Interaktif Berbasis <i>Android</i> Pada Mata Pelajaran Bahasa Indonesia Kelas V SD Negeri 1 Beji Kabupaten Tulungagung	Penelitian ini mengembangkan media pembelajaran interaktif berbasis <i>Android</i> untuk materi teks petunjuk di kelas V dengan model pengembangan ADDIE. Aplikasi berisi materi, latihan, kuis, serta fitur interaktif untuk membantu pemahaman siswa.

Sumber: Penulis 2026

Berdasarkan berbagai penelitian terdahulu ini, dapat disimpulkan bahwa media pembelajaran berbasis *Android* terbukti efektif dalam meningkatkan hasil belajar, minat, dan motivasi siswa. Keunggulan utamanya terletak pada kemudahan akses, tampilan yang menarik, serta kemampuannya dalam mendukung pembelajaran mandiri. Oleh karena itu, penulis melakukan penelitian berjudul “*Perancangan Media Pembelajaran Berbasis Android pada Mata Pelajaran Komputer*” di MTs Ishlahul Ummah dengan tujuan untuk mengembangkan media pembelajaran yang sesuai dengan karakteristik dan kebutuhan siswa tingkat MTs, sehingga dapat membantu mereka memahami konsep dasar komputer secara lebih interaktif, menarik, dan efisien.

2.8 Kerangka Berpikir Penelitian

Kerangka berpikir penelitian merupakan suatu alur berpikir yang logis dan sistematis yang menggambarkan hubungan antara teori, konsep, serta hasil penelitian terdahulu dengan variabel yang diteliti. Kerangka ini berfungsi sebagai dasar dalam menjelaskan arah penelitian dan memberikan gambaran mengenai bagaimana teori-teori yang relevan saling berhubungan untuk menjawab rumusan masalah. Melalui kerangka berpikir yang tersusun dengan baik, peneliti dapat menunjukkan keterkaitan antara landasan teoritis dan pelaksanaan penelitian secara jelas, runtut, dan terarah.



Sumber : Penulis 2026

Gambar 2. 1 Kerangka Berpikir Penelitian

Dari Gambar 2.1 tentang kerangka berpikir penelitian, dapat dijelaskan dan disimpulkan sebagai berikut:

1. Identifikasi Masalah

Pembelajaran komputer di MTs Ishlahul Ummah masih menggunakan metode konvensional, kurang interaktif, dan belum memanfaatkan media digital. Akibatnya, siswa sulit memahami materi dan kurang termotivasi dalam belajar.

2. Analisis Kebutuhan

Diperlukan media pembelajaran yang menarik, praktis, dan dapat diakses melalui perangkat *Android*. Media tersebut diharapkan mampu

mendukung pembelajaran mandiri serta menampilkan materi dasar komputer secara visual dan interaktif.

3. Perancangan Sistem

Peneliti merancang media pembelajaran berbasis *Android* yang memuat materi pengenalan komputer, latihan soal interaktif, serta antarmuka yang mudah digunakan. Desain sistem dibuat berdasarkan hasil analisis kebutuhan dan teori desain pembelajaran

4. Implementasi Sistem

Rancangan yang telah dibuat kemudian dikembangkan menjadi aplikasi *Android* menggunakan perangkat lunak pendukung. Tahap ini melibatkan pembuatan konten, pemrograman, serta integrasi fitur-fitur pembelajaran interaktif.

5. Pengujian Sistem

Aplikasi yang telah dikembangkan diuji melalui validasi ahli dan uji coba kepada siswa. Pengujian ini bertujuan untuk menilai kelayakan, kemudahan penggunaan, serta efektivitas media dalam mendukung proses belajar.

6. Evaluasi dan Kesimpulan

Hasil pengujian dianalisis untuk mengetahui kelebihan dan kekurangan media pembelajaran yang dikembangkan. Dari hasil tersebut diperoleh kesimpulan bahwa media pembelajaran berbasis *Android* layak digunakan dan mampu meningkatkan pemahaman serta motivasi belajar siswa.

Dengan demikian, kerangka berpikir penelitian ini menjadi pedoman bagi peneliti dalam merancang dan mengembangkan media pembelajaran komputer berbasis *Android* di MTs Ishlahul Ummah. Melalui pendekatan yang sistematis mulai dari identifikasi masalah hingga evaluasi, diharapkan media ini dapat menciptakan pengalaman belajar yang fleksibel, interaktif, dan relevan dengan perkembangan teknologi pendidikan masa kini.

BAB III

METODE PENELITIAN

3.1 Objek dan Jadwal Penelitian

Objek penelitian ini adalah MTs Ishlahul Ummah, sebuah madrasah tsanawiyah (sekolah Islam setingkat SMP) swasta yang berlokasi di Dusun 5, Desa Beringin, Kecamatan Lubai, Kabupaten Muara Enim, tepatnya di Jalan Raya Prabumulih–Baturaja. Penelitian ini berfokus pada siswa MTs Ishlahul Ummah beserta kegiatan pembelajaran komputer yang mereka jalani. Fokus penelitian diarahkan pada perancangan media pembelajaran berbasis *Android* untuk mendukung proses belajar komputer di sekolah tersebut. Pemilihan objek ini didasarkan pada kebutuhan siswa akan media pembelajaran yang interaktif, mudah diakses, dan sesuai dengan perkembangan teknologi, guna meningkatkan pemahaman materi serta motivasi belajar mereka.

3.1.1 Sejarah

Madrasah Tsanawiyah Ishlahul Ummah (MTs Ishlahul Ummah) adalah sekolah Islam swasta yang berlokasi di Dusun 5, Desa Beringin, Kecamatan Lubai, Kabupaten Muara Enim, Sumatera Selatan, Indonesia. Madrasah ini didirikan pada tahun 2007 di bawah naungan Yayasan Ishlahul Ummah, dengan tujuan utama untuk memberikan pendidikan yang berlandaskan pada nilai-nilai Islam serta membekali peserta didik dengan pengetahuan umum yang relevan dengan perkembangan zaman.

Sejak berdirinya, MTs Ishlahul Ummah telah menunjukkan perkembangan yang signifikan dalam berbagai aspek, baik dari sisi akademik, sarana dan prasarana, maupun kualitas tenaga pendidik. Madrasah ini memiliki komitmen kuat untuk menciptakan lingkungan belajar yang religius, disiplin, dan berorientasi pada pembentukan karakter peserta didik. Dalam rangka menjaga dan meningkatkan mutu pendidikan, MTs Ishlahul Ummah secara aktif melakukan inovasi

pembelajaran dan menjalin kerja sama dengan berbagai pihak dalam bidang pendidikan dan teknologi.

Sebagai bukti dari upaya peningkatan mutu tersebut, pada tahun 2024 MTs Ishlahul Ummah berhasil memperoleh akreditasi A berdasarkan Surat Keputusan Badan Akreditasi Nasional Pendidikan Dasar dan Menengah (BAN-PDM) Nomor: 187/BAN-PDM/SK/2025. Pencapaian ini menunjukkan bahwa madrasah tersebut telah memenuhi standar nasional pendidikan dalam hal manajemen sekolah, proses pembelajaran, serta hasil lulusan yang berkualitas.

3.1.2 Visi, Misi, dan Tujuan MTs Ishlahul Ummah

Visi, misi, dan tujuan Madrasah Tsanawiyah Ishlahul Ummah merupakan dasar utama dalam pelaksanaan seluruh kegiatan pendidikan, pembinaan karakter, serta pengembangan kompetensi peserta didik. Ketiganya menjadi arah dan pedoman bagi seluruh warga madrasah dalam mewujudkan tujuan institusional serta mendukung tercapainya mutu pendidikan Islam yang unggul, religius, dan berdaya saing. Oleh karena itu, penting untuk memahami visi, misi, dan tujuan MTs Ishlahul Ummah sebagai landasan dalam pelaksanaan penelitian ini.

a. Visi

“Mewujudkan madrasah yang religius, berakhlak mulia, berilmu, mandiri, nasionalis, dan peduli lingkungan.”

b. Misi

1. Menjalankan nilai-nilai agama dan berakhlak mulia dalam segala aspek kehidupan.
2. Mewujudkan siswa/siswi yang berilmu pengetahuan dan beramal saleh.
3. Mewujudkan siswa/siswi yang berprestasi.
4. Menumbuhkan kesadaran berbangsa dan bernegara.
5. Menjadikan siswa/siswi yang disiplin sesuai dengan aturan yang berlaku.
6. Mewujudkan ekosistem madrasah yang peduli lingkungan dan menjaga kebersihan.

c. Tujuan

1. Menghasilkan peserta didik yang beriman, bertakwa, dan berakhlak mulia.
2. Meningkatkan kualitas pembelajaran agar peserta didik memiliki pengetahuan, keterampilan, dan sikap yang seimbang.
3. Menumbuhkan semangat belajar dan motivasi berprestasi di bidang akademik maupun nonakademik.
4. Membangun karakter peserta didik yang mandiri, disiplin, dan bertanggung jawab.
5. Menyiapkan lulusan yang mampu bersaing di era global tanpa meninggalkan nilai-nilai keislaman.

3.1.3 Struktur Organisasi

Struktur organisasi merupakan hal penting dalam lembaga pendidikan karena menggambarkan pembagian tugas, wewenang, dan tanggung jawab setiap unsur di dalamnya. Adapun struktur organisasi MTs Ishlahul Ummah adalah sebagai berikut:



Sumber : Penulis (2026)

Gambar 3. 1 Struktur Organisasi

Berdasarkan Gambar 3.1 Struktur Organisasi MTs Ishlahul Ummah, dapat dijelaskan bahwa struktur ini menggambarkan susunan dan hubungan kerja antar unsur yang berperan dalam penyelenggaraan kegiatan pendidikan di madrasah. Setiap bagian memiliki tanggung jawab yang saling berkaitan untuk mencapai tujuan pendidikan yang efektif, efisien, serta sejalan dengan visi dan misi MTs Ishlahul Ummah.

1. Yayasan Ishlahul Ummah, merupakan lembaga tertinggi yang menaungi dan mengawasi seluruh kegiatan operasional di MTs Ishlahul Ummah. Yayasan bertanggung jawab dalam menentukan arah kebijakan, pengembangan lembaga, serta memastikan seluruh kegiatan madrasah berjalan sesuai visi dan misi yang telah ditetapkan.
2. Kepala Madrasah, bertugas memimpin dan mengoordinasikan seluruh kegiatan pendidikan, administrasi, dan pembinaan tenaga pendidik serta peserta didik. Kepala madrasah juga menjadi penghubung antara pihak yayasan dan seluruh unsur madrasah.
3. Wakil Kepala Madrasah Bidang Kurikulum, membantu kepala madrasah dalam perencanaan, pelaksanaan, dan evaluasi kegiatan akademik agar sesuai dengan standar kurikulum yang berlaku.
4. Wakil Kepala Madrasah Bidang Kesiswaan, bertanggung jawab atas pembinaan disiplin, kegiatan ekstrakurikuler, serta pengembangan minat dan bakat peserta didik.
5. Guru, melaksanakan proses pembelajaran, bimbingan, dan penilaian terhadap peserta didik serta berperan dalam pengembangan karakter dan prestasi siswa.
6. Tenaga Pendidik, membantu kegiatan belajar mengajar dan mendukung pelaksanaan tugas-tugas akademik maupun nonakademik di madrasah.
7. Tata Usaha, melaksanakan administrasi umum, pengarsipan, dan kepegawaian untuk mendukung kelancaran kegiatan madrasah.
8. Perpustakaan, menyediakan sumber belajar dan literasi guna menunjang kegiatan akademik di lingkungan madrasah.

3.1.4 Jadwal Penelitian

Pelaksanaan penelitian ini direncanakan selama kurang lebih enam bulan, dimulai pada tanggal 03 November 2025 dan berakhir pada 04 Mei 2026. Dalam kurun waktu tersebut, kegiatan penelitian akan dilaksanakan secara bertahap, meliputi penyusunan rencana penelitian, pengumpulan dan pengolahan data, analisis kebutuhan sistem, perancangan serta pengembangan sistem, hingga tahap evaluasi dan penyusunan laporan akhir. Dengan pembagian waktu yang terencana, diharapkan seluruh tahapan penelitian dapat berjalan efektif dan menghasilkan hasil penelitian yang optimal.

3.2 Metode Penelitian

Metode penelitian kualitatif adalah pendekatan ilmiah yang digunakan untuk memahami dan menafsirkan makna suatu peristiwa sosial berdasarkan pandangan partisipan. Penelitian ini menekankan pada proses, konteks, dan makna yang muncul secara alami di lapangan tanpa manipulasi dari peneliti. penelitian kualitatif berupaya menggambarkan fenomena secara mendalam dengan memanfaatkan peneliti sebagai instrumen utama dalam pengumpulan dan analisis data (Abdussamad, 2021).

Sejalan dengan fokus penelitian yang menekankan pengembangan media pembelajaran, pendekatan deskriptif kualitatif digunakan karena mampu memberikan gambaran menyeluruh mengenai proses perancangan dan hasil implementasi media pembelajaran berbasis *Android* secara kontekstual. Pendekatan ini memungkinkan peneliti untuk memperoleh pemahaman mendalam terhadap kebutuhan pengguna, efektivitas media, serta faktor-faktor yang memengaruhi keberhasilan penerapan dalam lingkungan pembelajaran.

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 pertama melalui kegiatan observasi, wawancara, dan dokumentasi di lapangan. Data ini bersifat orisinal karena dikumpulkan sendiri oleh peneliti untuk menjawab rumusan masalah penelitian (Abdussamad, 2021). Dalam penelitian ini, data primer diperoleh melalui wawancara dan pengamatan langsung terhadap kegiatan pembelajaran komputer di MTs Ishlahul Ummah. Data tersebut digunakan untuk mengetahui kondisi pembelajaran, kebutuhan siswa, serta kendala yang dihadapi dalam proses belajar sebelum diterapkannya media pembelajaran berbasis *Android*.

3.3.2 Data Sekunder

Data sekunder adalah data yang telah dikumpulkan dan dipublikasikan oleh pihak lain sebelum digunakan oleh peneliti (Abdussamad, 2021). Data ini dapat berupa buku, jurnal, artikel ilmiah, laporan penelitian terdahulu, serta dokumen resmi yang berkaitan dengan topik penelitian. Dalam penelitian ini, data sekunder digunakan sebagai landasan teoritis untuk memperkuat konsep pengembangan media pembelajaran, penerapan teknologi berbasis *Android*, serta mendukung analisis hasil dari data primer.

3.4 Teknik Pengumpulan Data

Pengumpulan data merupakan tahapan penting dalam pelaksanaan penelitian ini karena kualitas data yang diperoleh sangat berpengaruh terhadap ketepatan dan keberhasilan perancangan sistem yang dikembangkan. Untuk memperoleh data yang akurat dan relevan, penelitian ini menggunakan beberapa metode pengumpulan data, yaitu observasi, wawancara, dokumentasi, dan studi literatur.

3.4.1 Observasi

Observasi dilakukan dengan cara mengamati secara langsung proses pembelajaran komputer di MTs Ishlahul Ummah. Kegiatan ini bertujuan untuk mengetahui kondisi nyata pembelajaran yang berlangsung, meliputi metode

pengajaran yang digunakan guru, tingkat keterlibatan siswa, serta fasilitas pendukung yang tersedia. Melalui observasi ini, peneliti dapat mengidentifikasi berbagai kendala yang muncul dalam proses pembelajaran, seperti keterbatasan media belajar, kurangnya variasi metode penyampaian materi, serta rendahnya minat siswa dalam mengikuti pelajaran komputer. Hasil observasi tersebut menjadi dasar penting dalam merancang media pembelajaran berbasis *Android* yang diharapkan mampu meningkatkan efektivitas dan motivasi belajar peserta didik.

3.4.2 Wawancara

Wawancara dilakukan dengan guru mata pelajaran komputer sebagai informan utama. Kegiatan ini bertujuan untuk memperoleh informasi mendalam mengenai proses pembelajaran yang berlangsung, kendala yang dihadapi dalam penyampaian materi, serta kebutuhan media pembelajaran yang sesuai dengan karakteristik peserta didik. Melalui wawancara ini, peneliti memperoleh data yang berguna untuk memahami bagaimana media pembelajaran berbasis *Android* dapat membantu guru dalam menjelaskan materi komputer secara lebih interaktif, menarik, dan mudah dipahami oleh siswa. Hasil wawancara juga menjadi dasar dalam menentukan fitur dan rancangan media yang sesuai dengan kebutuhan pembelajaran di MTs Ishlahul Ummah.

3.4.3 Dokumentasi

Dokumentasi dilakukan dengan mengumpulkan berbagai dokumen dan data pendukung yang berkaitan dengan proses pembelajaran komputer di MTs Ishlahul Ummah. Dokumen yang dikumpulkan meliputi materi ajar, hasil evaluasi belajar siswa, serta arsip kegiatan pembelajaran yang relevan. Data tersebut dianalisis untuk memahami struktur materi, kompetensi yang ingin dicapai, serta kebutuhan siswa terhadap media pembelajaran yang lebih menarik dan interaktif. Selain itu, teknik dokumentasi ini juga membantu peneliti dalam menyesuaikan rancangan konten dan tampilan media pembelajaran berbasis *Android* agar selaras dengan kurikulum dan karakteristik peserta didik pada mata pelajaran komputer.

3.4.4 Studi Literatur

Studi literatur dilakukan dengan menelaah berbagai sumber akademik maupun non-akademik, seperti jurnal ilmiah, buku, prosiding, dan penelitian terdahulu yang berkaitan dengan media pembelajaran berbasis *Android*, teknologi pendidikan, serta pembelajaran komputer. Kegiatan ini bertujuan untuk memperoleh landasan teori yang kuat mengenai konsep media pembelajaran digital, strategi pembelajaran interaktif, serta pendekatan yang efektif dalam meningkatkan motivasi belajar siswa. Melalui studi literatur ini, peneliti juga dapat memahami perkembangan teknologi terkini dan menentukan metode pengembangan media yang sesuai dengan kebutuhan peserta didik pada mata pelajaran komputer di MTs Ishlahul Ummah. Dengan demikian, penelitian ini memiliki dasar ilmiah yang relevan dan mendukung penerapan media pembelajaran yang inovatif di lingkungan madrasah.

3.5 Metode Pengembangan Sistem

Dalam penelitian ini digunakan metode pengembangan Design Thinking, yaitu pendekatan pemecahan masalah yang berpusat pada manusia dan berfokus pada penciptaan solusi yang inovatif, relevan, serta sesuai dengan kebutuhan pengguna. Metode ini dipilih karena dinilai paling sesuai untuk penelitian yang mengembangkan media pembelajaran berbasis *Android* pada mata pelajaran komputer, di mana keterlibatan pengguna (guru dan siswa) sangat penting dalam proses perancangan dan evaluasi media.

Dalam buku *Design Thinking dan Inovasi*, Design Thinking merupakan proses kreatif dan analitis yang melibatkan pemahaman mendalam terhadap pengguna, eksplorasi ide-ide solutif, pembuatan purwarupa (*prototype*), serta pengujian iteratif untuk menghasilkan solusi yang efektif dan berorientasi pada kebutuhan manusia (Lantana & Fauziah, 2025). Pendekatan ini menekankan empati, kolaborasi lintas disiplin, serta iterasi berkelanjutan untuk mencapai hasil yang optimal.

Adapun alasan peneliti menggunakan metode *Design Thinking* adalah sebagai berikut:

1. Berpusat pada pengguna (*User-centered*).

Proses pengembangan menempatkan guru dan siswa sebagai fokus utama dalam merancang media pembelajaran yang sesuai dengan kebutuhan dan karakteristik mereka.

2. Fleksibel dan iteratif.

Metode ini memungkinkan perubahan dan penyempurnaan rancangan berdasarkan umpan balik pengguna pada setiap tahap pengembangan.

3. Mendorong kreativitas dan inovasi.

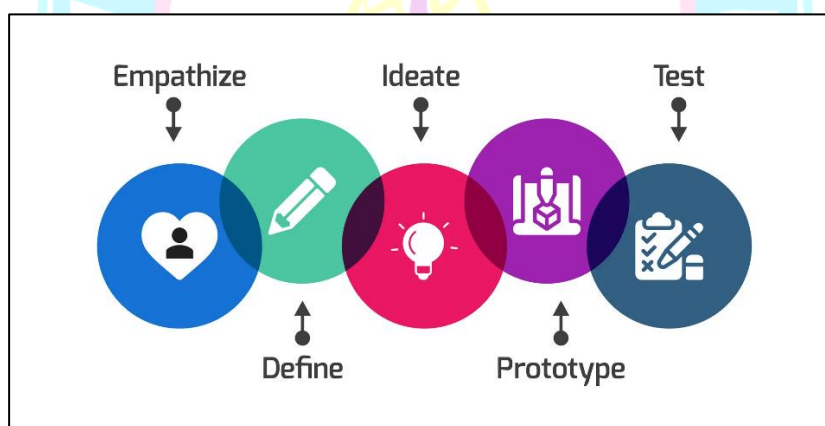
Design Thinking memfasilitasi eksplorasi ide-ide baru yang dapat menghasilkan media pembelajaran yang menarik, interaktif, dan efektif.

4. Kolaboratif dan partisipatif.

Prosesnya melibatkan interaksi langsung antara peneliti dan pengguna, sehingga solusi yang dihasilkan lebih kontekstual.

5. Mengurangi risiko kegagalan.

Karena setiap tahap diuji melalui *prototype* dan umpan balik, kesalahan dapat diperbaiki lebih awal sebelum implementasi penuh.



Sumber: Easyretro.io

Gambar 3. 2 Metode Pengembangan Sistem Design Thinking

Berdasarkan tahapan yang dijelaskan oleh (Lantana & Fauziah, 2025), metode *Design Thinking* terdiri dari lima tahap utama, yaitu:

1. *Empathize* (Empati),

Empathize adalah kegiatan Mengidentifikasi kebutuhan, pengalaman, dan permasalahan pengguna melalui observasi dan wawancara. Penulis

sudah memahami kebutuhan dan kendala pada kegiatan pembelajaran di MTs Ishlahul Ummah

2. *Define* (Menentukan Masalah)

Define bertujuan Merumuskan inti permasalahan yang ada di objek. Penulis sudah menganalisis masalah utama, yaitu perlunya media pembelajaran berbasis *Android* pada mata pelajaran komputer

3. *Ideate* (Pengembangan Ide)

Ideate bertujuan mengembangkan berbagai ide dan alternatif solusi kreatif yang dapat diterapkan dalam media pembelajaran. Dalam hal ini penulis sudah menghasilkan ide solusi dari rumusan masalah, lalu memilih konsep terbaik berupa aplikasi media pembelajaran berbasis *Android*

4. *Prototype* (Pembuatan Prototipe)

Prototype Bertujuan mewujudkan ide terpilih dalam bentuk desain awal media pembelajaran berbasis *Android*. Dalam hal ini penulis akan mewujudkan rancangan awal sistem (*mockup*) dan versi aplikasi sederhana menggunakan *Flutter*.

5. *Test* (Uji Coba)

Test ialah tahapan pengujian terhadap pengguna untuk memperoleh umpan balik dan melakukan penyempurnaan produk. Prototipe akan diuji oleh pengguna untuk mendapatkan masukan dan melakukan perbaikan agar sistem lebih optimal dan sesuai kebutuhan.

Melalui kelima tahap tersebut, penelitian ini berupaya menghasilkan media pembelajaran berbasis *Android* yang tidak hanya fungsional dan menarik, tetapi juga mampu meningkatkan minat belajar serta pemahaman siswa pada mata pelajaran komputer di MTs Ishlahul Ummah.

3.6 Alat Bantu Perancangan

Dalam proses pengembangan sistem, penggunaan alat bantu perancangan memiliki peran penting untuk memudahkan pengembang dalam merancang, mengimplementasikan, serta menguji aplikasi secara optimal. Pada penelitian ini, digunakan beberapa jenis alat bantu seperti *development tools*, perangkat

pemodelan sistem, perancangan basis data, desain antarmuka pengguna (UI/UX), serta alat pengujian dan analisis. Pemilihan alat bantu tersebut bertujuan untuk menunjang kelancaran proses pembuatan media pembelajaran berbasis *Android* pada mata pelajaran komputer, sehingga hasil yang diperoleh lebih terstruktur, efisien, dan mudah dipahami.

3.6.1 Alat Bantu Utama

Visual Studio Code merupakan Integrated Development Environment (IDE) yang digunakan dalam proses pengembangan media pembelajaran. IDE ini menyediakan berbagai fitur seperti syntax highlighting, auto-completion, dukungan ekstensi, serta integrasi dengan version control system seperti Git. Fitur-fitur tersebut memudahkan pengembang dalam menulis, menguji, dan mengelola kode program secara efisien dan terstruktur. Pemilihan *Visual Studio Code* didasarkan pada keunggulannya yang ringan, fleksibel, serta mendukung berbagai bahasa pemrograman dan *framework modern*, sehingga sangat sesuai untuk pengembangan media pembelajaran berbasis *Android* pada mata pelajaran komputer.

Dalam penelitian ini, Flutter digunakan sebagai framework utama untuk mengembangkan aplikasi. Flutter merupakan *framework open-source* yang dikembangkan oleh *Google* untuk membangun aplikasi *mobile*, *web*, dan *desktop* menggunakan bahasa pemrograman Dart. Framework ini mengandalkan konsep *widget* sebagai komponen utama dalam penyusunan antarmuka, sehingga mampu menghasilkan tampilan yang konsisten di berbagai platform (Gunawan, 2024). Dengan pendekatan *single codebase*, Flutter memungkinkan pengembang membuat aplikasi yang responsif dan memiliki performa tinggi pada berbagai platform, khususnya *Android*. Selain itu, fitur hot reload yang dimilikinya membantu mempercepat proses pengujian dan perbaikan, sehingga pengembangan media pembelajaran berbasis *Android* pada mata pelajaran komputer dapat dilakukan secara lebih efisien dan efektif.

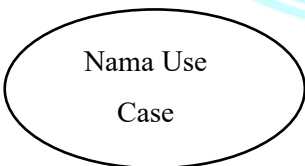
3.6.2 Alat Bantu Pemodelan Visual

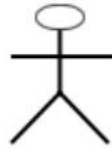
Bahasa pemodelan visual (*Unified Modeling Language*) Adalah sebuah bahasa berbasis diagram/grafik yang digunakan untuk memvisualisasi, membangun, dan mendokumentasikan sistem perangkat lunak berorientasi objek (Saputra et al., 2023). Melalui pemodelan visual, pengembang dapat memahami hubungan antar komponen, aliran proses pembelajaran, serta interaksi antara pengguna dan sistem dengan lebih jelas dan terstruktur. Selain itu, diagram visual juga berperan sebagai sarana komunikasi antara pengembang, pembimbing, dan pihak terkait lainnya agar rancangan aplikasi dapat dipahami secara seragam sebelum proses pengkodean dimulai.

a. *Use Case Diagram*

Use case diagram merupakan tools untuk mengembangkan sistem dengan menjelaskan hubungan antara actor yang terlibat dalam sistem. *Use case* diagram adalah deskripsi fungsi dari sebuah perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *User* (pengguna) sebuah sistem Dalam sebuah pembicaraan tentang *use case*, pengguna biasanya disebut dengan actor (Wayahdi & Ruziq, 2023). Berikut Adalah simbol-simbol *Use case* Diagram dapat dilihat pada tabel 3.1

Tabel 3. 1 Use Case Diagram

Simbol	Deskripsi
	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit dan aktor.

Aktor/Actor

Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi.

Asosiasi/association _____	Komunikasi antar aktor dan <i>Use case</i> yang berpartisipasi.
Ekstensi/extend <<extend>> _____→	Relasi <i>Use case</i> tambahan ke sebuah <i>Use case</i> dimana <i>Use case</i> yang ditambah dapat berdiri sendiri walau tanpa <i>Use case</i> tambahan.
Generalisasi/generalization _____→	Hubungan generalisasi dan spesialisasi antara dua buah <i>Use case</i> yang mana fungsi yang satu lebih umum dariyang lainnya.
Include/Use case <<include>> _____→	Relasi <i>Use case</i> tambahan ke sebuah <i>Use case</i> dimana <i>Use case</i> yang ditambahkan memerlukan <i>Use case</i> ini untuk menjalankan fungsinya.
Uses _____→	

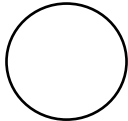
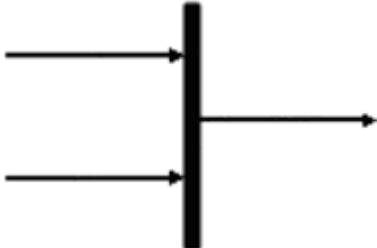
Sumber : Penulis, 2025

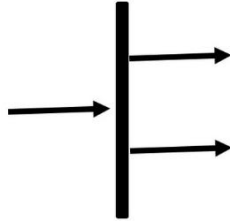
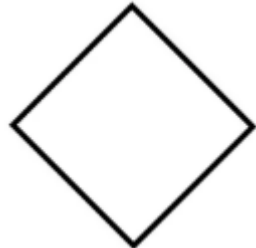
b. *Activity Diagram*

Activity diagram digunakan untuk menggambarkan alur fungsional dalam suatu sistem informasi. Diagram ini menjelaskan secara menyeluruh bagaimana suatu workflow dimulai, bagaimana proses tersebut berjalan, aktivitas apa saja yang terjadi selama proses berlangsung, serta bagaimana urutan aktivitas hingga proses berakhir (Andriyanto, 2022). Fokus utama activity diagram adalah pada aktivitas dan proses yang terjadi di dalam sistem, bukan pada tindakan yang dilakukan oleh aktor. Oleh karena itu, diagram aktivitas lebih menekankan pada alur proses dan

jalur aktivitas secara umum dari tingkat atas. Activity diagram biasanya disusun berdasarkan satu atau beberapa use case yang telah dirancang sebelumnya.

Tabel 3. 2 Activity Diagram

Simbol	Deskripsi
 Activity	Menunjukkan aktivitas atau proses yang sedang dilakukan dalam sistem
 Control Flow	Menunjukkan alur atau urutan eksekusi dalam proses
 Object Flow	Menunjukkan aliran objek dari satu aktivitas ke aktivitas lainnya
 Start Point	Menandai awal dimulai atau dibentuk suatu objek
 End Point	Menyatakan suatu objek di bentuk atau di akhiri
 Join/Penggabungan	Menggabungkan kembali beberapa alur aktivitas menjadi satu

Fork	Memecah satu alur aktivitas menjadi beberapa alur parallel
	
Decision	Menunjukkan penggambaran suatu keputusan yang harus diambil pada kondisi tertentu
	

Sumber : Penulis, 2026

c. *Class Diagram*

Class Diagram merupakan diagram yang menggambarkan hubungan antar kelas serta memberikan penjelasan secara rinci mengenai setiap kelas yang terdapat dalam model perancangan suatu sistem. Diagram ini juga menunjukkan aturan-aturan dan tanggung jawab setiap entitas yang berperan dalam menentukan perilaku system (Ramdany et al., 2024).

Secara umum, class diagram dapat dipahami sebagai representasi visual dari struktur sistem perangkat lunak yang dibangun, yang menunjukkan jenis-jenis kelas beserta keterkaitan di antaranya. Diagram ini berfungsi untuk memodelkan struktur data dan hubungan logis antar kelas yang menjadi dasar dalam pengelolaan data pada sistem yang akan dikembangkan. Selain itu, class diagram tersusun atas sekumpulan kelas yang saling berelasi dan membentuk satu kesatuan sistem. Setiap kelas digambarkan dalam bentuk persegi panjang yang terbagi menjadi tiga bagian utama, yaitu bagian atas berisi nama kelas, bagian tengah berisi atribut atau data yang dimiliki kelas, dan bagian bawah berisi metode atau operasi yang dapat dilakukan oleh kelas tersebut

Tabel 3. 3 Class Diagram

Simbol	Deskripsi
 Class	Himpunan dari objek berbagi atribut serta operasi yang sama
Association	Sebuah hubungan yang menunjukkan adanya interaksi antar class.
Aggregation	Mengindikasikan keseluruhan bagian relationship dan biasanya disebut relasi.
 Composition	Sebuah class yang tidak bisa berdiri sendiri dan harus merupakan bagian dari class yang lain, maka class tersebut memiliki relasi Composition terhadap class tempat dia bergantung tersebut.
 Generalization	Sebuah hubungan antar class yang bersifat dari khusus ke umum.
 Dependency	Sebuah class menggunakan class yang lain. Hal ini disebut dependency

Sumber : Penulis, 2026

3.6.3 Alat Bantu Desain

Dalam proses pengembangan aplikasi, tahap desain memiliki peranan penting untuk mengubah kebutuhan pengguna menjadi rancangan visual dan fungsional dari sistem yang akan dibangun. Tahap ini meliputi pembuatan tata letak tampilan, penentuan elemen antarmuka, pemilihan warna, ikon, serta penyusunan alur interaksi antara pengguna dengan aplikasi. Proses desain dilakukan sebelum tahap implementasi agar pengembang memiliki panduan yang jelas dalam membangun tampilan dan fitur aplikasi sesuai hasil analisis kebutuhan.

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

3.6.4 Alat Bantu Pengembangan dan *Debugging*

Dalam proses pengembangan aplikasi, tahap *debugging* memiliki peran penting untuk memastikan bahwa sistem berfungsi sesuai rancangan dan bebas dari kesalahan (*error*). Kegiatan ini dilakukan untuk menemukan, memisahkan, dan memperbaiki kesalahan yang ada pada suatu aplikasi. Proses ini mencakup pemahaman terhadap kode sumber aplikasi, pemeriksaan catatan log, serta pemanfaatan perangkat bantu *debugging* (Wijaya et al., 2021).

Pada penelitian ini, proses *debugging* dilakukan menggunakan Visual Studio Code dengan bantuan fitur *debugger* yang terintegrasi dengan Flutter. Aplikasi dijalankan baik melalui emulator *Android* maupun langsung di perangkat *smartphone* guna menguji kestabilan dan fungsionalitasnya secara nyata. Dengan cara ini, pengembang dapat memantau jalannya program secara *real-time*, mengenali bagian kode yang bermasalah, serta melakukan perbaikan secara cepat

melalui fitur *hot reload* yang disediakan oleh Flutter, sehingga proses pengujian menjadi lebih efisien.

3.7 Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa media pembelajaran berbasis *Android* pada mata pelajaran komputer telah berfungsi sesuai dengan rancangan yang telah dibuat. Metode pengujian yang digunakan dalam penelitian ini adalah White Box Testing. Pengujian ini berfokus pada struktur internal program, seperti alur logika, percabangan, serta jalur eksekusi kode, untuk memastikan bahwa setiap bagian dari program bekerja dengan benar sesuai fungsinya (Wintana et al., 2022). Pengujian white box pada penelitian ini dilakukan menggunakan tools bawaan framework Flutter yaitu flutter test yang digunakan untuk menguji fungsi dan logika program berdasarkan *source code* aplikasi yang telah dikembangkan.

Metode White Box dipilih karena aplikasi yang dikembangkan memiliki logika program yang cukup kompleks. Melalui pengujian ini, peneliti dapat memastikan bahwa setiap fungsi utama, seperti proses login pengguna, akses materi pembelajaran, hingga penyimpanan hasil evaluasi, berjalan sesuai dengan yang diharapkan tanpa kesalahan logika dalam kode program. Selain itu, penggunaan flutter test membantu proses pengujian menjadi lebih terstruktur dan memudahkan peneliti dalam mendeteksi kesalahan pada setiap modul aplikasi.

Proses pengujian dilakukan dengan menjalankan setiap modul aplikasi, menelusuri jalur eksekusi, serta memeriksa hasil keluaran untuk memastikan kesesuaian dengan input yang diberikan. Hasil dari pengujian digunakan untuk menemukan dan memperbaiki kesalahan (*error*) yang mungkin muncul sebelum aplikasi diimplementasikan secara menyeluruh. Dengan demikian, White Box Testing memberikan jaminan bahwa aplikasi media pembelajaran yang dikembangkan memiliki tingkat keandalan, stabilitas, dan akurasi yang baik dalam mendukung proses pembelajaran komputer di lingkungan sekolah.

BAB IV

ANALISIS DAN PERANCANGAN

4.1 Analisa Sistem

4.1.1 Analisa dan Evaluasi Sistem Berjalan

a. Analisa Sistem Berjalan

Proses pembelajaran mata pelajaran komputer di MTs Ishlahul Ummah saat ini masih didominasi oleh metode konvensional, yaitu penyampaian materi melalui penjelasan lisan guru dan penggunaan buku cetak sebagai sumber utama pembelajaran. Media pembelajaran berbasis teknologi belum dimanfaatkan secara optimal dalam proses belajar mengajar. Penyampaian materi dilakukan secara satu arah sehingga siswa cenderung pasif dan kurang terlibat secara aktif dalam pembelajaran.

Selain itu, keterbatasan waktu pembelajaran di kelas menyebabkan guru tidak selalu dapat menyampaikan materi secara mendalam. Siswa yang kurang memahami materi harus menunggu penjelasan ulang dari guru, sehingga proses belajar menjadi kurang efektif. Latihan soal juga masih diberikan secara tertulis, yang membuat proses evaluasi hasil belajar membutuhkan waktu lebih lama.

b. Evaluasi Sistem Berjalan

Berdasarkan analisa terhadap sistem pembelajaran yang berjalan, ditemukan beberapa kelemahan, antara lain kurangnya media pembelajaran yang interaktif, rendahnya motivasi belajar siswa, serta keterbatasan akses siswa terhadap materi pembelajaran di luar jam sekolah. Oleh karena itu, diperlukan sebuah media pembelajaran berbasis *Android* yang dapat membantu guru dalam menyampaikan materi dan memudahkan siswa dalam memahami pelajaran komputer secara mandiri.

Media pembelajaran berbasis *Android* diharapkan mampu menyajikan materi secara lebih menarik, interaktif, dan mudah diakses kapan saja serta di mana saja melalui perangkat mobile siswa. Dengan demikian, proses

pembelajaran dapat menjadi lebih efektif, efisien, dan sesuai dengan perkembangan teknologi saat ini.

4.1.2 Analisa Kebutuhan

a. Kebutuhan Pengguna (*User*)

Media pembelajaran berbasis *Android* ini digunakan oleh beberapa jenis pengguna dengan peran yang berbeda, yaitu:

1. Guru

Guru berperan sebagai pengelola materi pembelajaran. Guru dapat menginput, mengelola, dan memperbarui materi pembelajaran serta latihan soal yang disediakan dalam aplikasi. Selain itu, guru juga dapat memantau hasil latihan siswa untuk melihat tingkat pemahaman siswa.

2. Siswa

Siswa berperan sebagai pengguna utama aplikasi. Siswa dapat mengakses materi pembelajaran, melihat penjelasan materi komputer, serta mengerjakan latihan soal yang tersedia dalam aplikasi sebagai sarana evaluasi pemahaman.

b. Kebutuhan Data

Data yang dibutuhkan dalam aplikasi media pembelajaran meliputi:

1. Data Pengguna

Data pengguna berisi informasi akun yang digunakan untuk mengakses aplikasi, meliputi nama pengguna, *Username*, *password*, dan peran pengguna (guru atau siswa). Data ini digunakan untuk mengatur hak akses dan memastikan setiap pengguna dapat menggunakan fitur sesuai dengan perannya.

2. Data Materi Pembelajaran

Data materi pembelajaran berisi konten materi komputer yang disusun sesuai dengan kurikulum dan kebutuhan pembelajaran di MTs Ishlahul Ummah. Materi dapat berupa teks, dan gambar pendukung yang memudahkan pemahaman siswa.

3. Data Latihan Soal

Data latihan soal mencakup kumpulan soal yang disesuaikan dengan materi pembelajaran yang telah dipelajari oleh siswa. Data ini digunakan sebagai sarana evaluasi untuk mengukur tingkat pemahaman siswa terhadap materi komputer. Latihan soal dapat membantu siswa melatih kemampuan berpikir dan memperdalam pemahaman materi secara mandiri.

c. Kebutuhan fungsional

Kebutuhan fungsional merupakan fungsi utama yang harus dimiliki oleh media pembelajaran berbasis *Android*. Adapun kebutuhan fungsional sistem yang akan dibangun adalah sebagai berikut:

1. Sistem dapat menampilkan menu utama aplikasi media pembelajaran, yang berfungsi sebagai tampilan awal aplikasi dan memuat navigasi ke seluruh fitur utama, seperti menu materi pembelajaran, latihan soal, dan informasi aplikasi, sehingga pengguna dapat dengan mudah mengakses fitur yang tersedia.
2. Sistem dapat menampilkan daftar materi pembelajaran komputer, di mana materi disusun secara sistematis sesuai dengan urutan pembelajaran agar memudahkan siswa dalam memilih dan mempelajari materi yang diinginkan.
3. Sistem dapat menampilkan isi materi pembelajaran dalam bentuk teks dan gambar, dengan penyajian teks yang jelas serta dilengkapi gambar atau ilustrasi pendukung untuk membantu siswa memahami materi komputer dengan lebih baik.
4. Sistem dapat menampilkan latihan soal sesuai dengan materi yang dipelajari, sehingga siswa dapat langsung menguji pemahamannya setelah mempelajari suatu materi pembelajaran.
5. Sistem dapat menerima jawaban latihan soal dari siswa, baik dalam bentuk pilihan jawaban maupun isian sederhana yang diinput langsung melalui aplikasi.

6. Sistem dapat menampilkan hasil latihan soal berupa nilai kepada siswa sebagai umpan balik untuk mengetahui tingkat pemahaman siswa terhadap materi yang telah dipelajari.

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. Adapun kebutuhan non-fungsional sistem 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 perannya sebagai guru atau siswa. Data pengguna, materi pembelajaran, serta hasil latihan soal harus tersimpan dengan aman di dalam sistem dan tidak dapat diakses oleh pihak yang tidak berwenang.

2. Kinerja (*Performance*)

Sistem harus mampu menampilkan materi pembelajaran dan memproses latihan soal dengan waktu respon yang cepat. Aplikasi diharapkan tetap berjalan dengan stabil meskipun digunakan dalam jangka waktu lama serta tetap responsif ketika diakses oleh banyak pengguna secara bersamaan. Selain itu, sistem harus mampu mengelola penggunaan sumber daya perangkat secara efisien agar tidak menurunkan kinerja perangkat pengguna. Sistem juga diharapkan dapat meminimalkan terjadinya *error* atau kegagalan proses saat aplikasi sedang digunakan.

3. Kemudahan Penggunaan (*Usability*)

Antarmuka aplikasi harus dirancang sederhana, menarik, dan mudah dipahami oleh siswa maupun guru. Menu dan fitur disusun secara terstruktur agar pengguna dapat menggunakan aplikasi tanpa memerlukan pelatihan khusus..

4. **Keandalan (*Reliability*)**

Sistem harus dapat berjalan secara stabil dan meminimalkan terjadinya kesalahan saat digunakan. Materi pembelajaran dan data hasil latihan soal yang telah tersimpan tidak boleh hilang dan harus dapat ditampilkan kembali dengan benar.

5. **Ketersediaan (*Availability*)**

Aplikasi diharapkan dapat digunakan kapan saja dan di mana saja melalui perangkat *Android*, sehingga siswa dapat belajar baik di lingkungan sekolah maupun di luar jam pembelajaran.

6. **Kompatibilitas (*Compatibility*)**

Aplikasi harus dapat dijalankan pada berbagai perangkat smartphone *Android* dengan versi sistem operasi yang berbeda tanpa mengalami kendala berarti.

e. **Analisa Perangkat Keras & Lunak (*Hardware & Software*)**

Analisa perangkat keras dan perangkat lunak dilakukan untuk mengetahui spesifikasi minimum yang dibutuhkan agar media pembelajaran berbasis *Android* dapat berjalan dengan baik dan optimal. Analisa ini mencakup kebutuhan perangkat yang digunakan oleh pengembang maupun pengguna aplikasi, baik guru maupun siswa.

Perangkat keras (*hardware*) merupakan seluruh komponen fisik pada sistem komputer atau perangkat digital yang dapat dilihat dan digunakan secara langsung untuk menjalankan proses pengolahan data, seperti prosesor, memori, media penyimpanan, serta perangkat input dan output. Hardware berfungsi sebagai sarana utama yang mendukung berjalannya sistem dan aplikasi yang dikembangkan (Trisna et al., 2022).

Sementara itu, perangkat lunak (*software*) adalah sekumpulan program atau instruksi yang dirancang untuk mengendalikan dan mengatur kerja perangkat keras agar dapat menjalankan fungsi tertentu sesuai kebutuhan pengguna. Software tidak memiliki bentuk fisik, namun berperan penting sebagai penghubung antara pengguna dan perangkat keras dalam mengoperasikan sistem (Trisna et al., 2022). Adapun kebutuhan spesifikasi

minimum perangkat keras dan perangkat lunak pada sistem yang dikembangkan adalah sebagai berikut:

1. Perangkat Keras (*Hardware*)

Perangkat Keras yang digunakan

1. Laptop / Komputer
2. Processor : Intel Pentium atau yang setara
3. RAM : Minimal 4 GB
4. Storage : Minimal 256 GB
5. Keyboard dan Mouse
6. Handphone
7. OS *Android* 12
8. RAM : Minimal 4 GB
9. Storage : Minimal 128 GB

2. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan

1. Sistem Operasi : Microsoft Windows 10 64-bit
2. Bahasa Pemrograman : Dart 3.7.1
3. *Framework* : Flutter 3.28.1
4. *Database* : Fire Base
5. *Code Editor* : Visual Studio Code
6. Handphone Sebagai Emulator

4.2 Perancangan Sistem yang Diusulkan

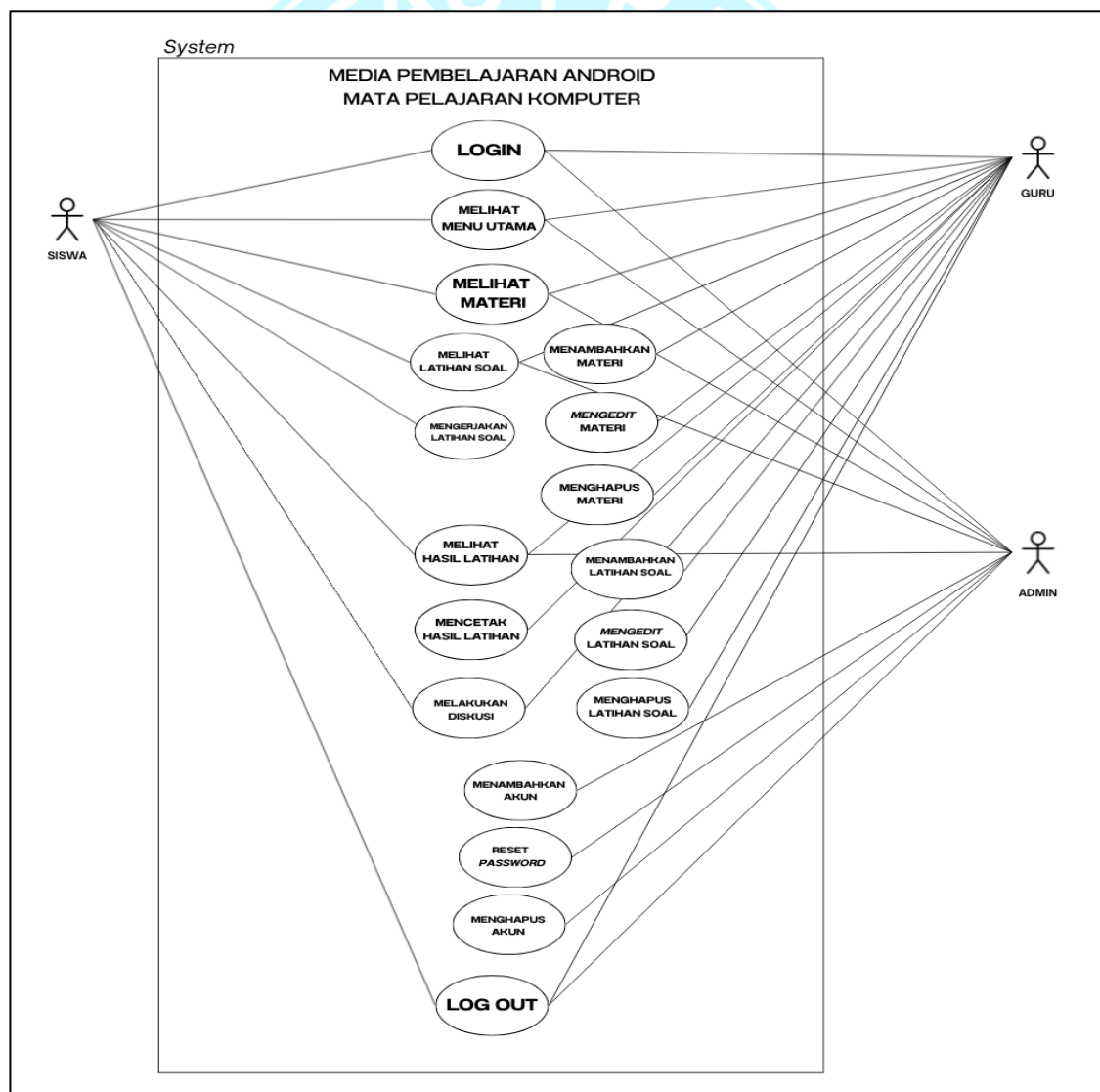
4.2.1 Perancangan Sistem (Logis)

Perancangan sistem logis merupakan tahapan yang bertujuan untuk menggambarkan cara kerja aplikasi media pembelajaran berbasis *Android* berdasarkan fungsi-fungsi yang tersedia, tanpa mempertimbangkan aspek teknis perangkat keras maupun perangkat lunak. Tahap ini menitikberatkan pada pemetaan alur penggunaan aplikasi, pola interaksi antara pengguna dengan sistem, serta layanan utama yang harus disediakan agar sesuai dengan hasil analisis kebutuhan. Perancangan sistem logis ini dituangkan ke dalam beberapa model,

yaitu *Use Case Diagram* yang menjelaskan peran dan interaksi aktor dengan sistem, *Activity Diagram* yang menggambarkan urutan aktivitas pengguna dalam menjalankan aplikasi pembelajaran, serta *Class Diagram* yang menunjukkan struktur kelas dan hubungan antar komponen sistem secara konseptual.

a. *Use Case Diagram*

Use Case Diagram digunakan untuk menggambarkan interaksi antara aktor (pengguna) dengan sistem media pembelajaran berbasis *Android*, seperti proses masuk ke aplikasi, mengakses materi pembelajaran, mengerjakan latihan soal, serta melihat hasil evaluasi. Diagram ini memberikan gambaran menyeluruh mengenai fungsi-fungsi utama yang disediakan oleh sistem sesuai dengan peran masing-masing pengguna.



Gambar 4. 1 Rancangan *Use Case Diagram*

Use Case Diagram menggambarkan interaksi antara aktor dengan sistem media pembelajaran berbasis *Android* pada mata pelajaran komputer. Pada sistem ini terdapat tiga aktor utama, yaitu Siswa, Guru, dan Admin, yang memiliki hak akses dan fungsi berbeda sesuai dengan perannya masing-masing.

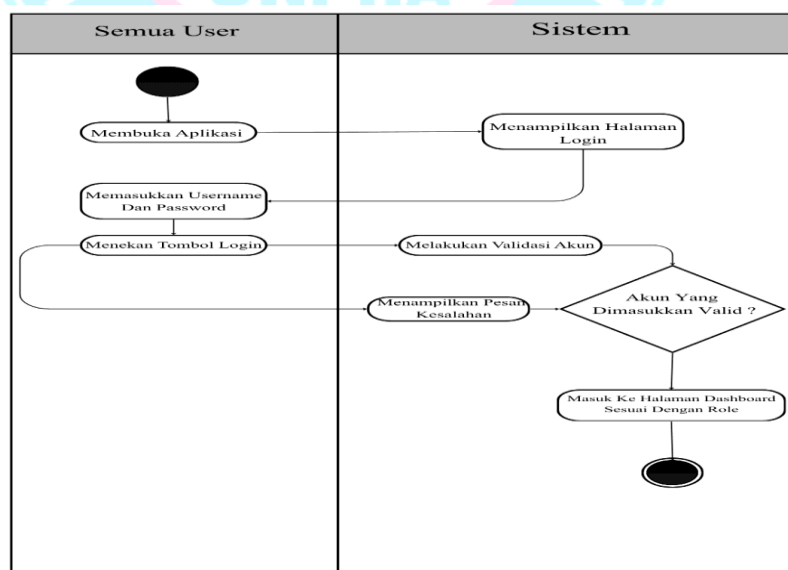
Aktor Siswa dapat melakukan *login*, melihat menu utama, melihat materi pembelajaran, melihat latihan soal, mengerjakan latihan soal, melihat dan mencetak hasil latihan, melakukan diskusi, serta *Logout*. Fitur tersebut membantu siswa belajar secara mandiri dan mengevaluasi pemahaman terhadap materi yang dipelajari.

Aktor Guru dapat melakukan *login*, melihat materi, menambahkan, mengedit, dan menghapus materi, mengelola latihan soal, melihat hasil latihan siswa, melakukan diskusi, serta *Logout*. Guru berperan dalam mengelola materi dan evaluasi pembelajaran pada sistem.

Aktor Admin dapat melakukan *login*, melihat materi, melihat latihan soal, melihat hasil latihan, menambahkan akun, melakukan reset *password*, menghapus akun, serta *Logout*. Admin bertanggung jawab dalam pengelolaan akun pengguna agar sistem dapat berjalan dengan baik dan teratur.

b. Activity Diagram

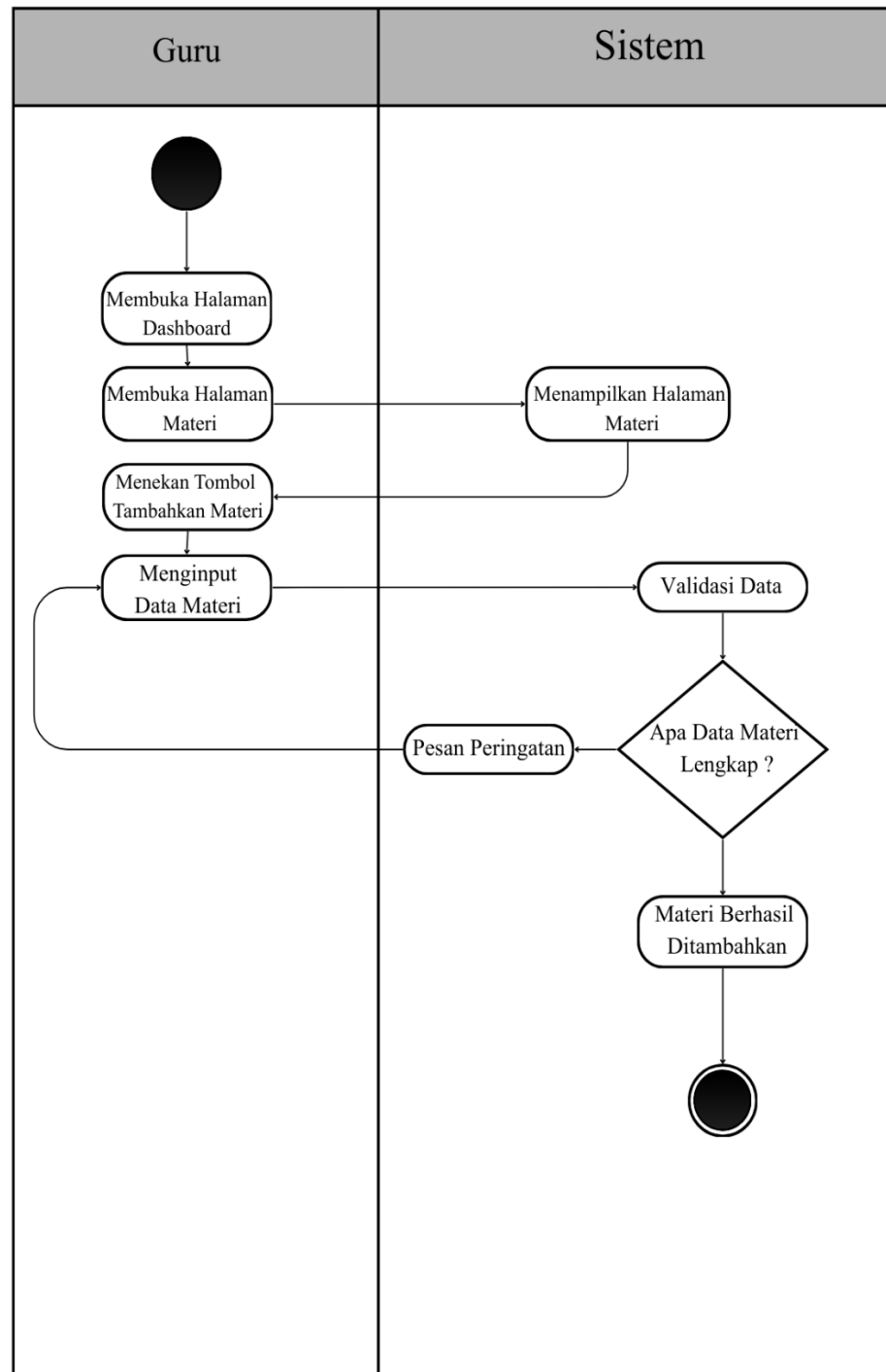
1. Activity Diagram Login Umum User



Sumber : data diolah oleh penulis (2026)

Gambar 4. 2 Login Activity

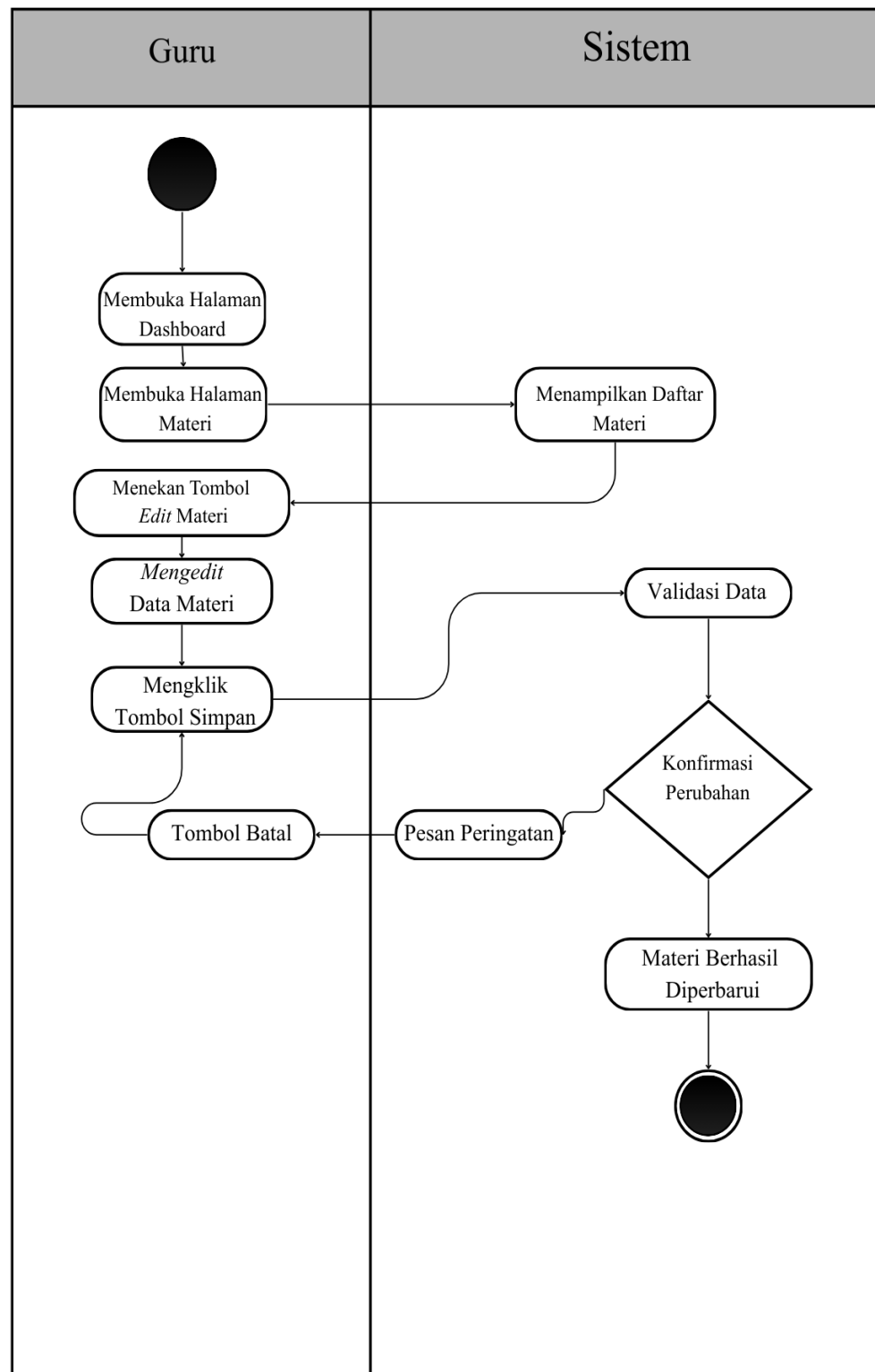
2. Activity Diagram Upload Materi



Sumber : data diolah oleh penulis (2026)

Gambar 4. 3 Upload Materi Activity

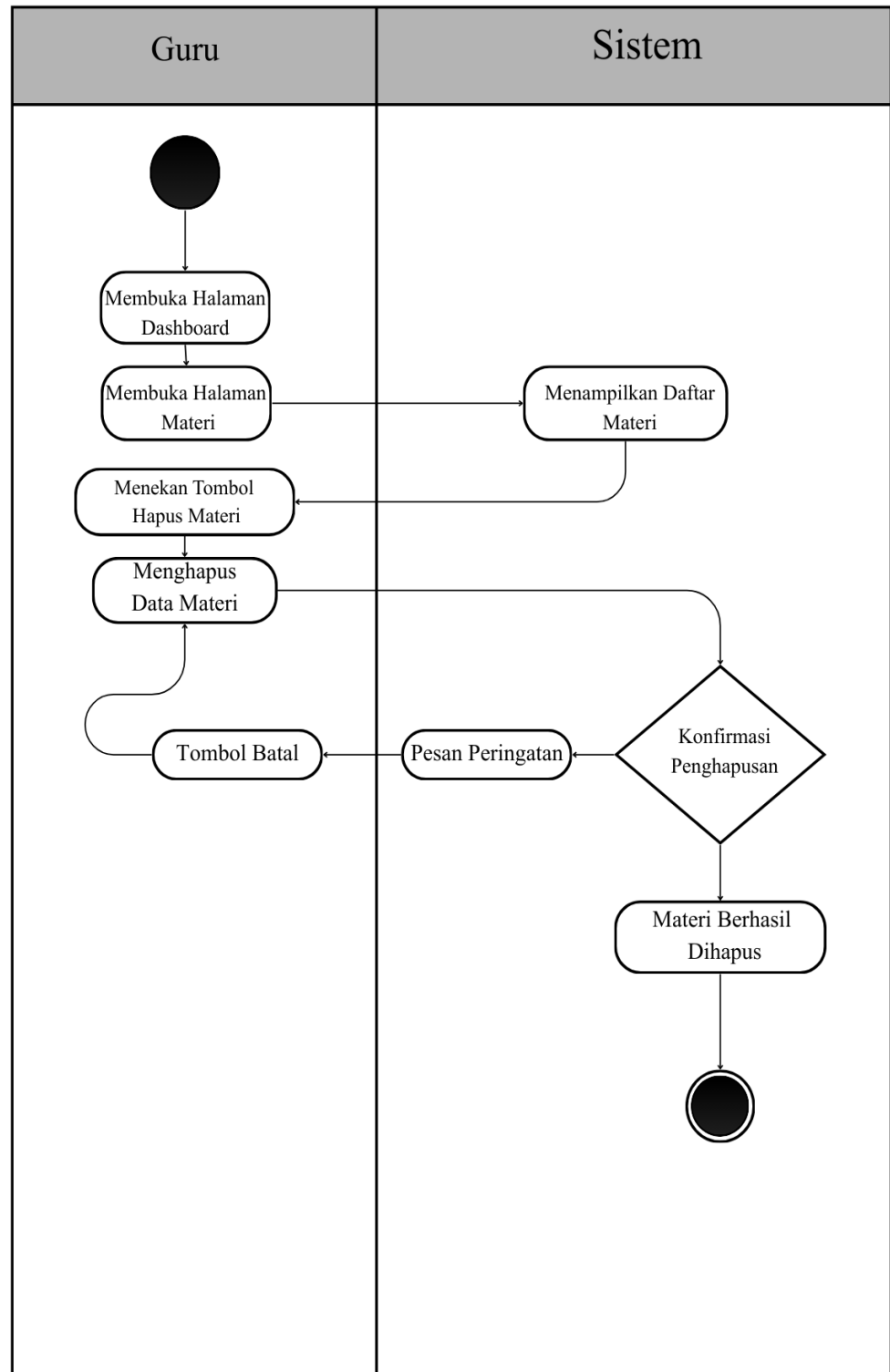
3. Activity Diagram Mengedit Materi



Sumber : data diolah oleh penulis (2026)

Gambar 4. 4 Mengedit Materi Activity

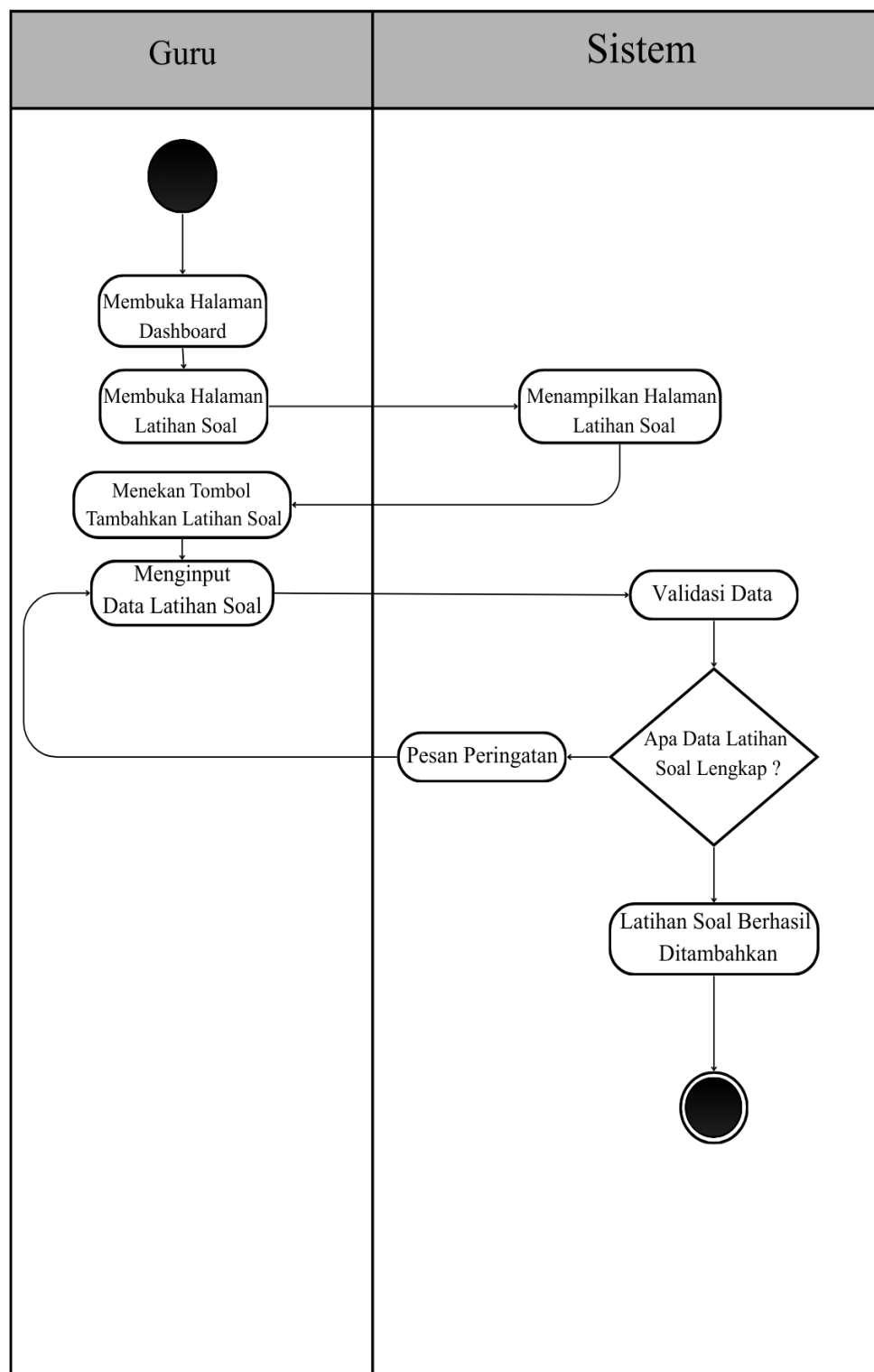
4. Activity Diagram Menghapus Materi



Sumber : data diolah oleh penulis (2026)

Gambar 4. 5 Menghapus Materi Activity

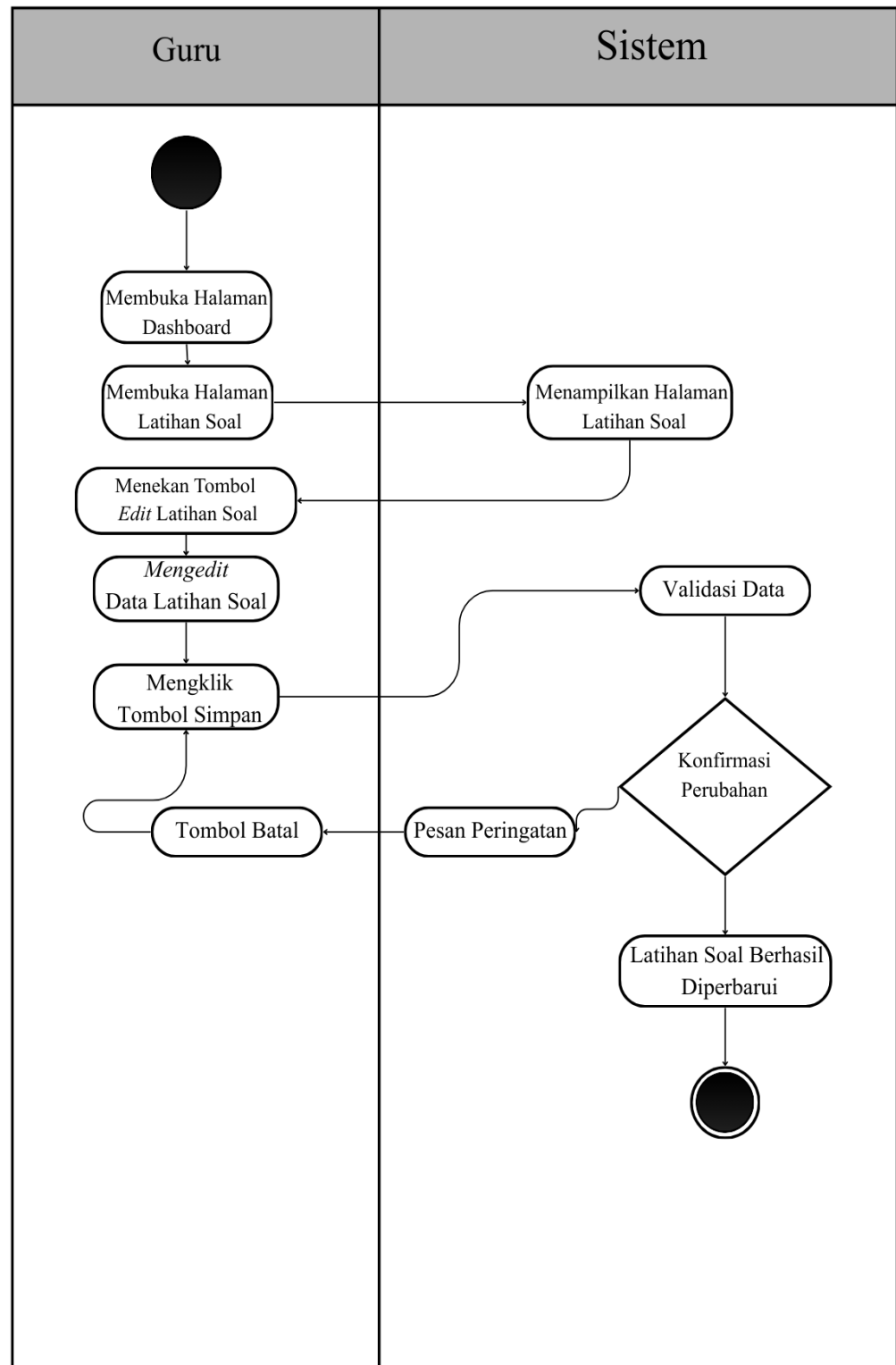
5. Activity Diagram Menambahkan Latihan Soal



Sumber : data diolah oleh penulis (2026)

Gambar 4. 6 Menambahkan Latihan Soal Activity

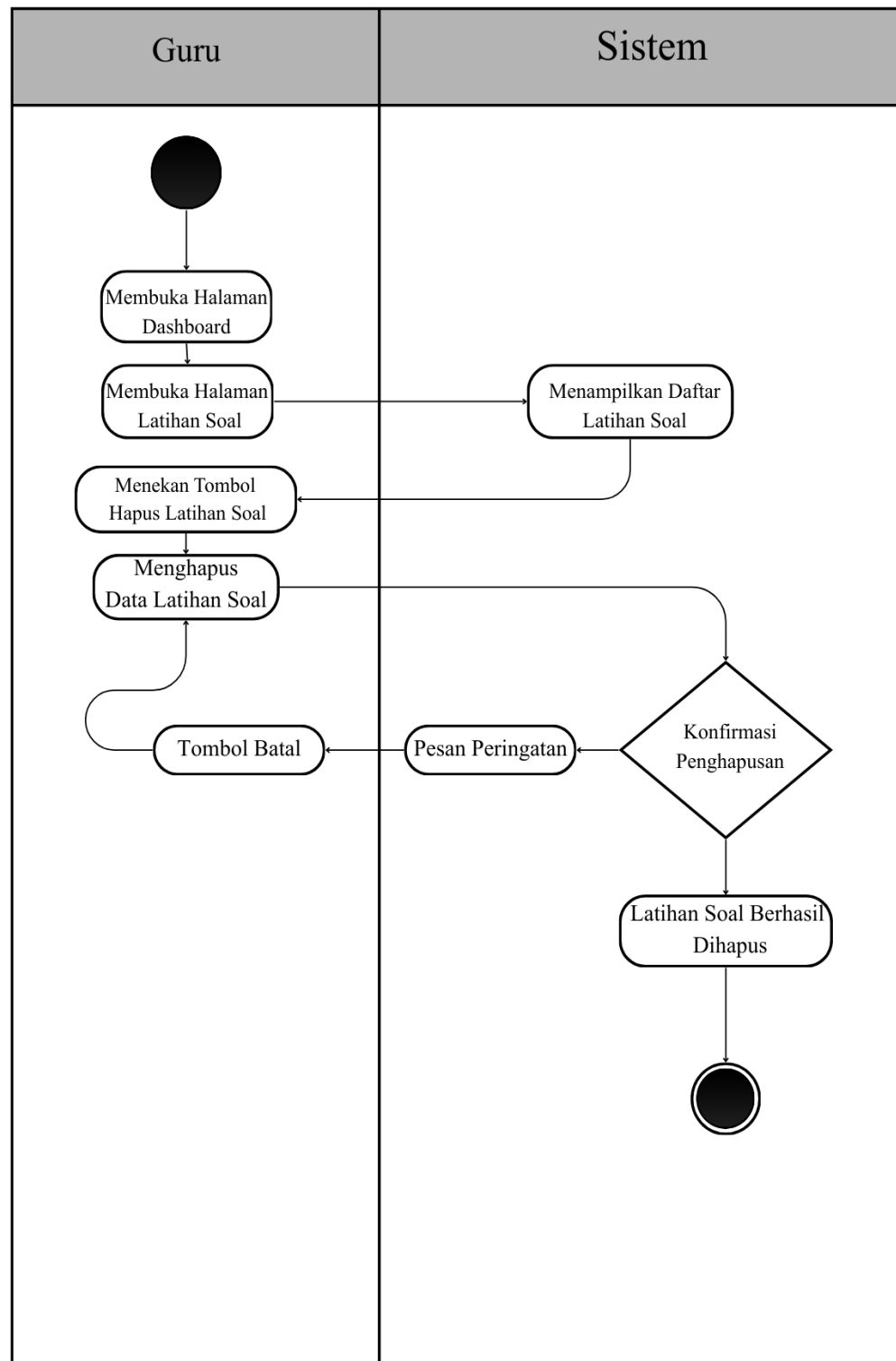
6. Activity Diagram Mengedit Latihan Soal



Sumber : data diolah oleh penulis (2026)

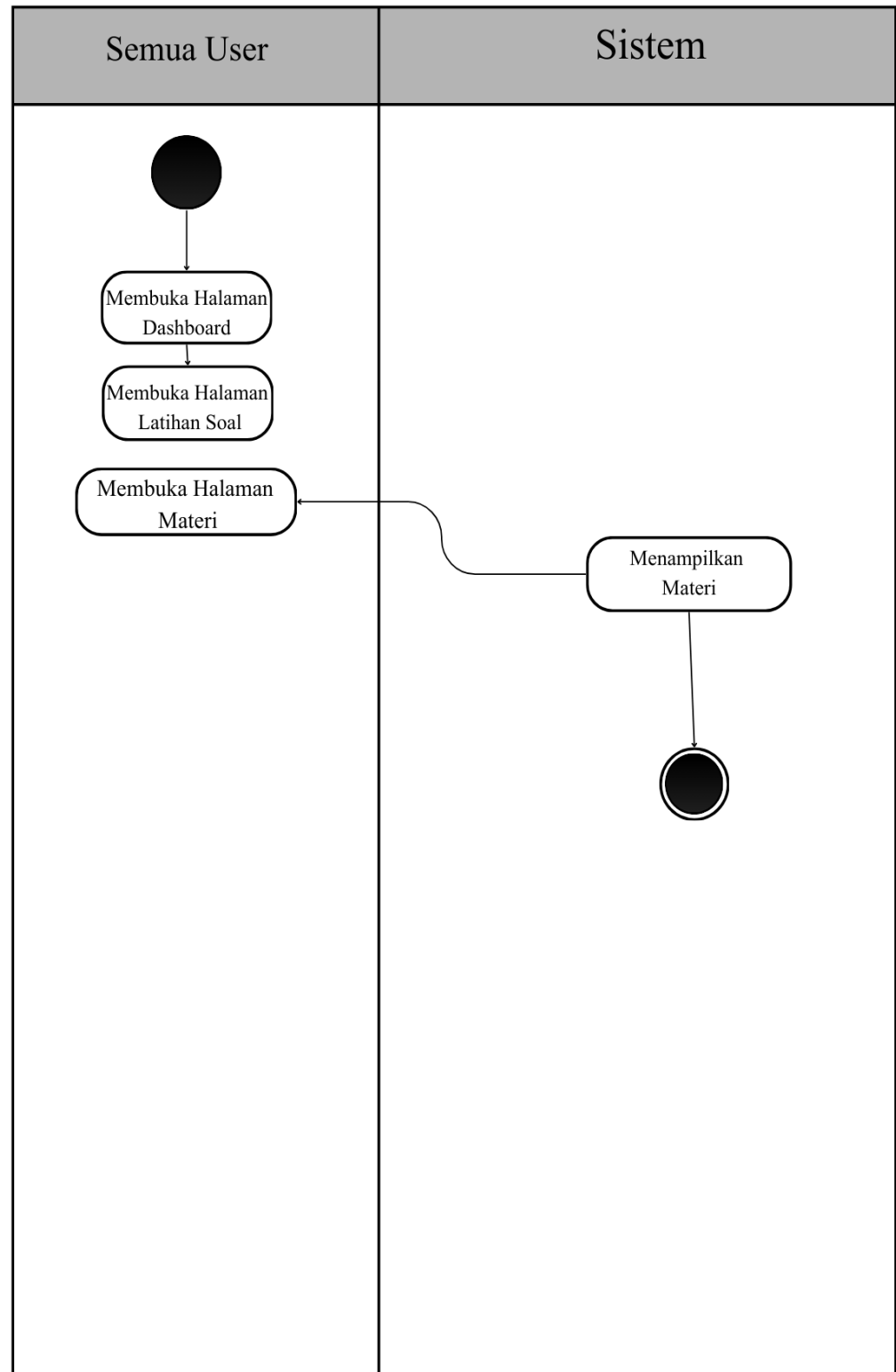
Gambar 4. 7 Edit Latihan Soal Activity

7. Activity Diagram Menghapus Latihan Soal



Sumber : data diolah oleh penulis (2026)

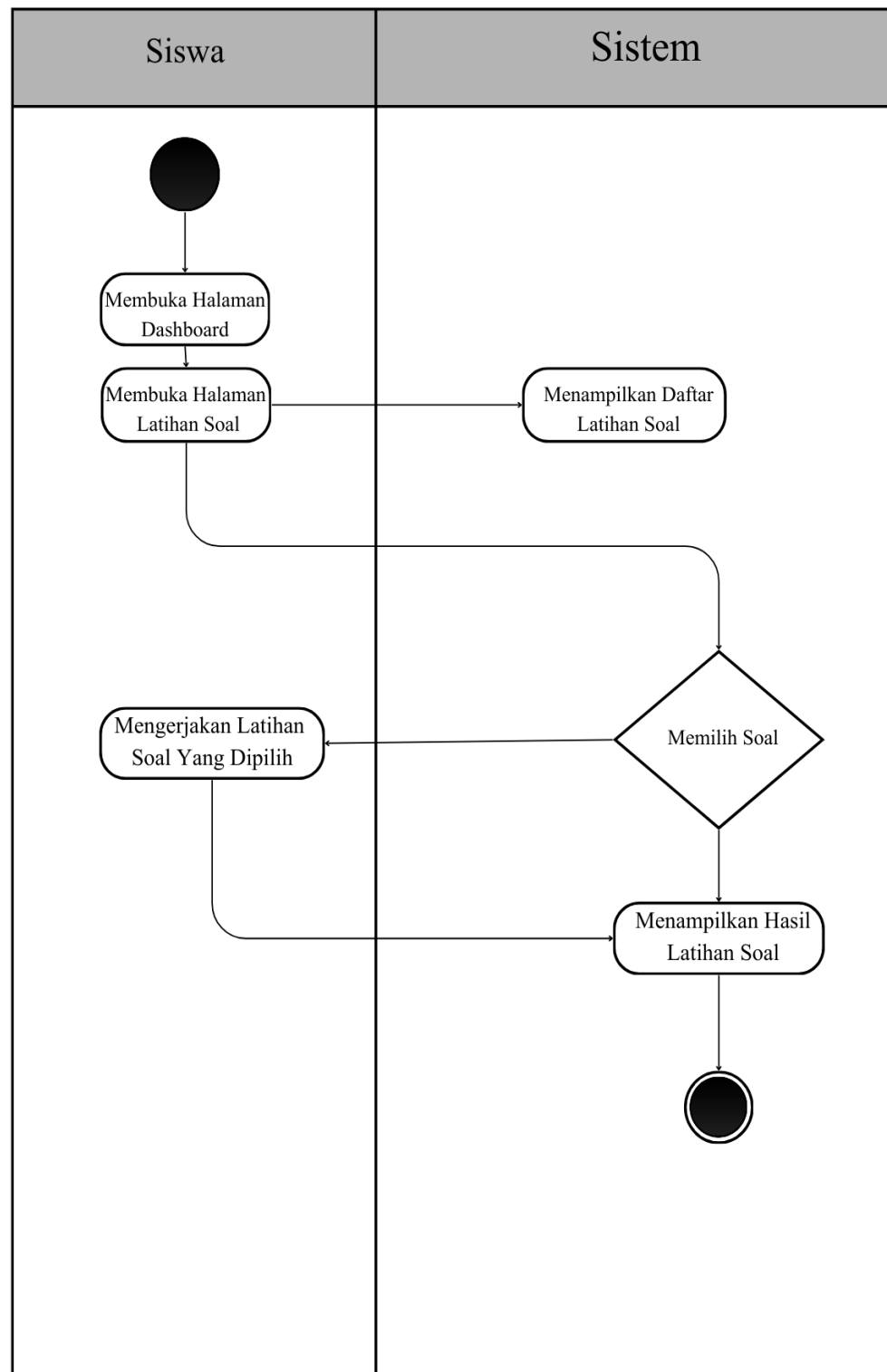
Gambar 4. 8 Hapus Latihan Soal *Activity*

8. *Activity Diagram Melihat Materi*

Sumber : data diolah oleh penulis (2026)

Gambar 4. 9 Melihat Materi *Activity*

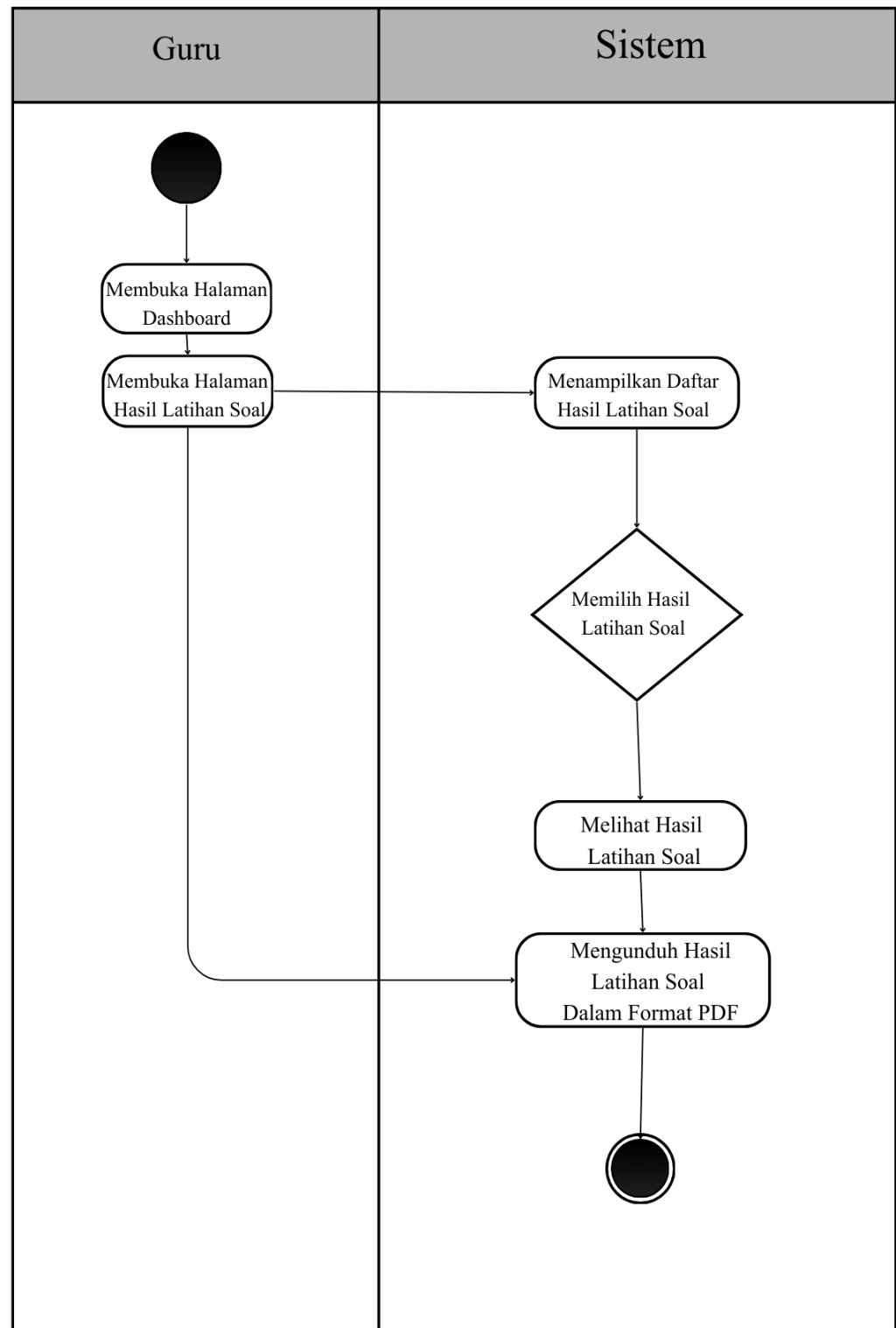
9. Activity Diagram Mengerjakan Latihan Soal



Sumber : data diolah oleh penulis (2026)

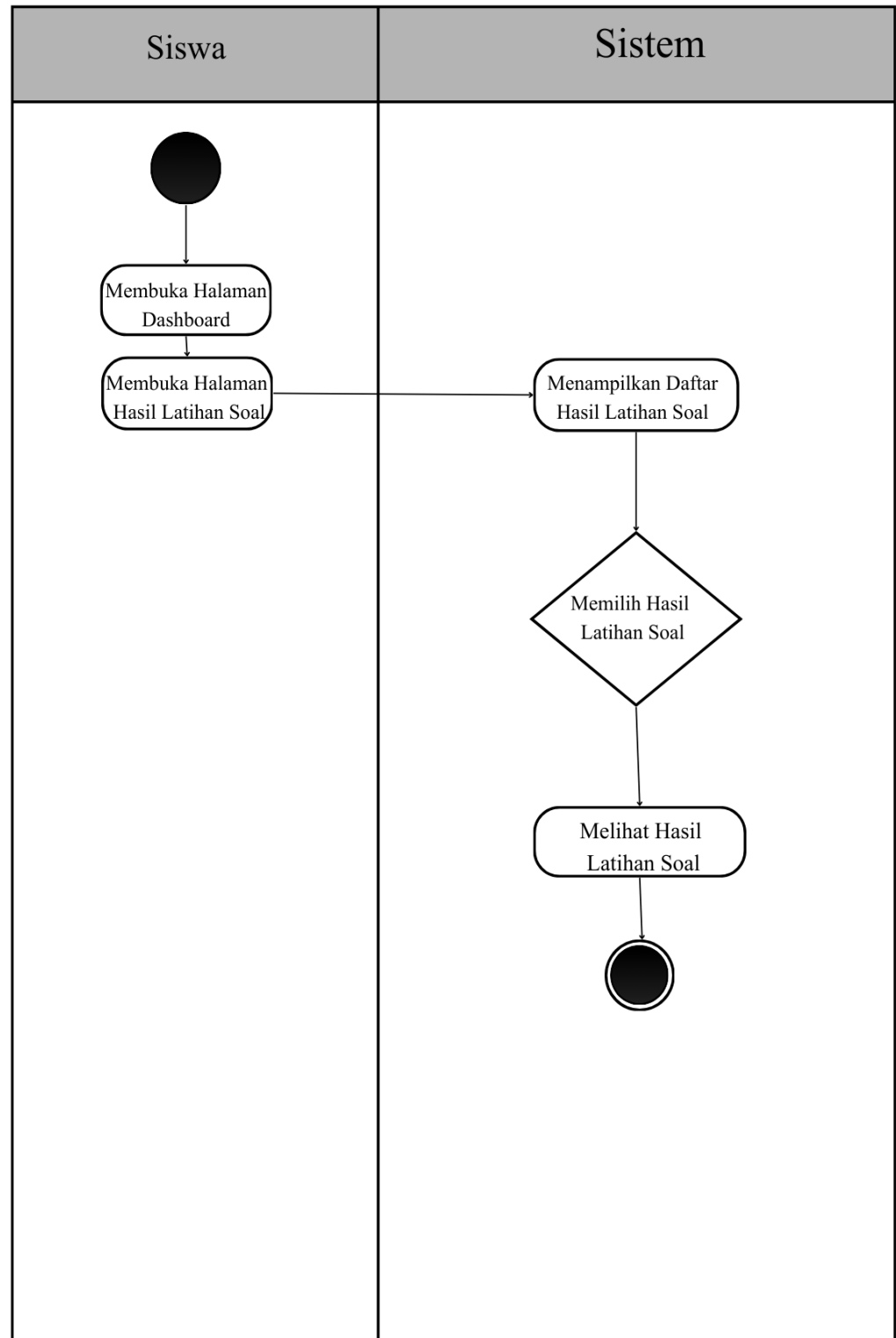
Gambar 4. 10 Mengerjakan Latihan Soal Activity

10. *Activity Diagram* Melihat dan Mengunduh Hasil Latihan Soal



Sumber : data diolah oleh penulis (2026)

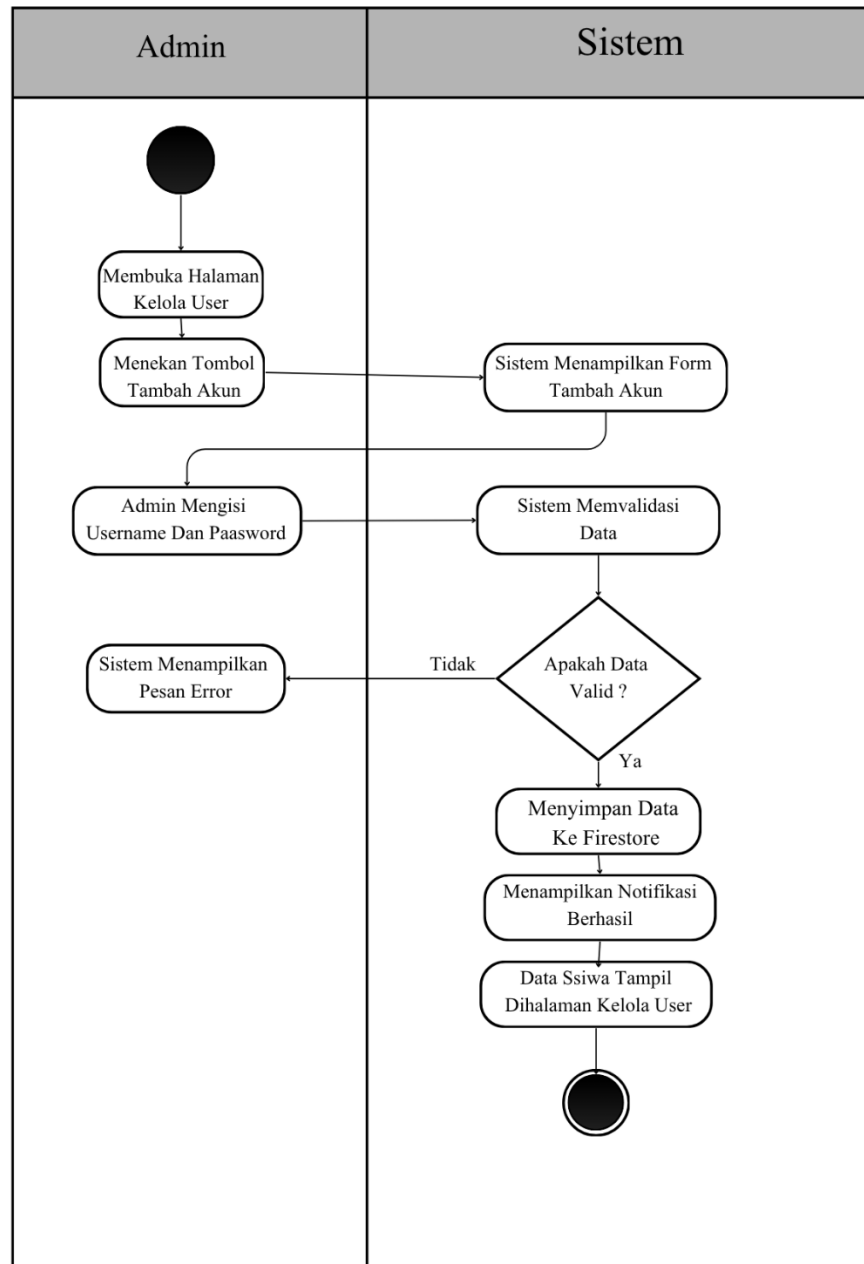
Gambar 4. 11 Melihat dan Mengunduh Hasil Latihan Soal *Activity*

11. *Activity Diagram* Melihat Hasil Latihan Soal

Sumber : data diolah oleh penulis (2026)

Gambar 4. 12 Melihat Hasil Latihan Soal *Activity*

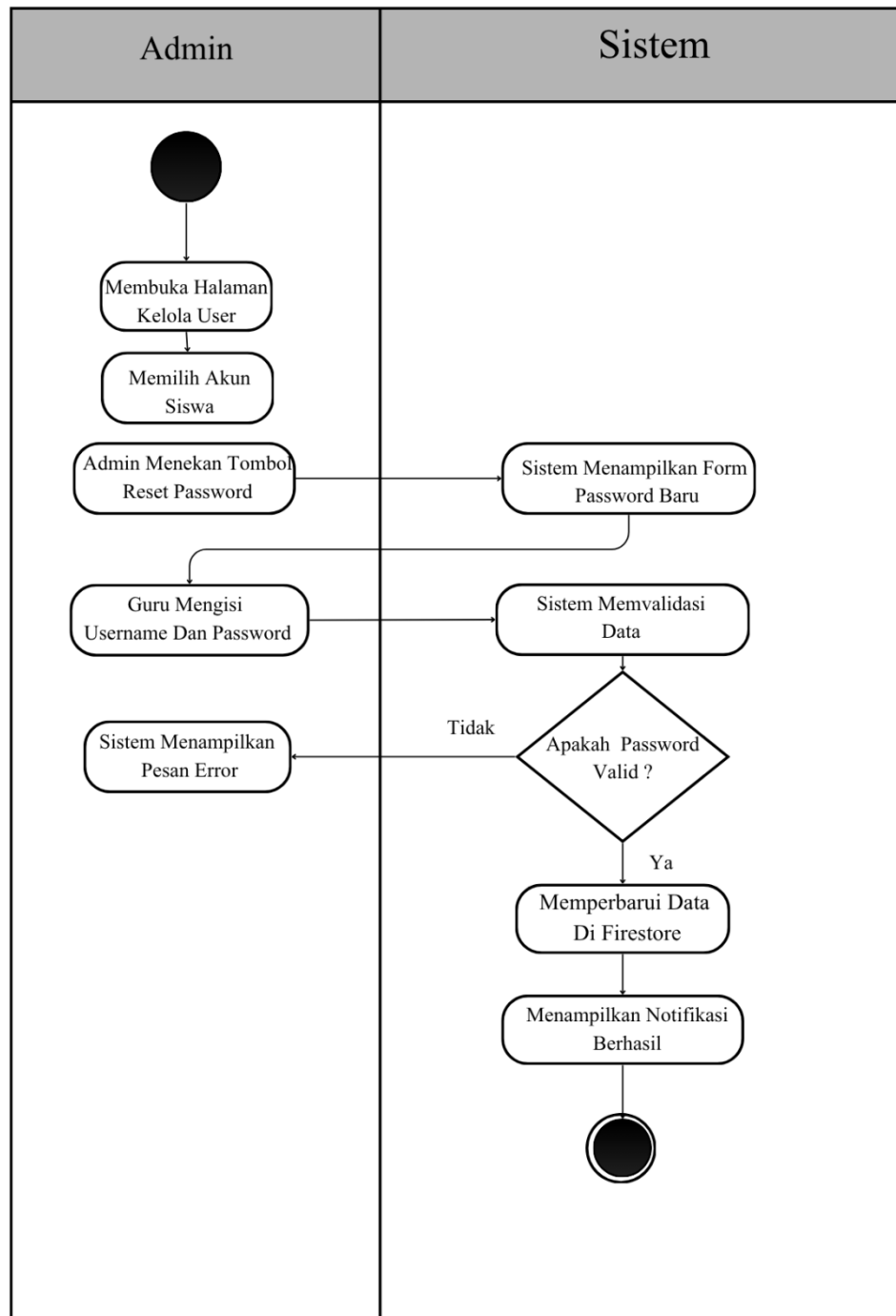
12. Activity Diagram Menambahkan Akun



Sumber : data diolah oleh penulis (2026)

Gambar 4. 13 Menambahkan Akun

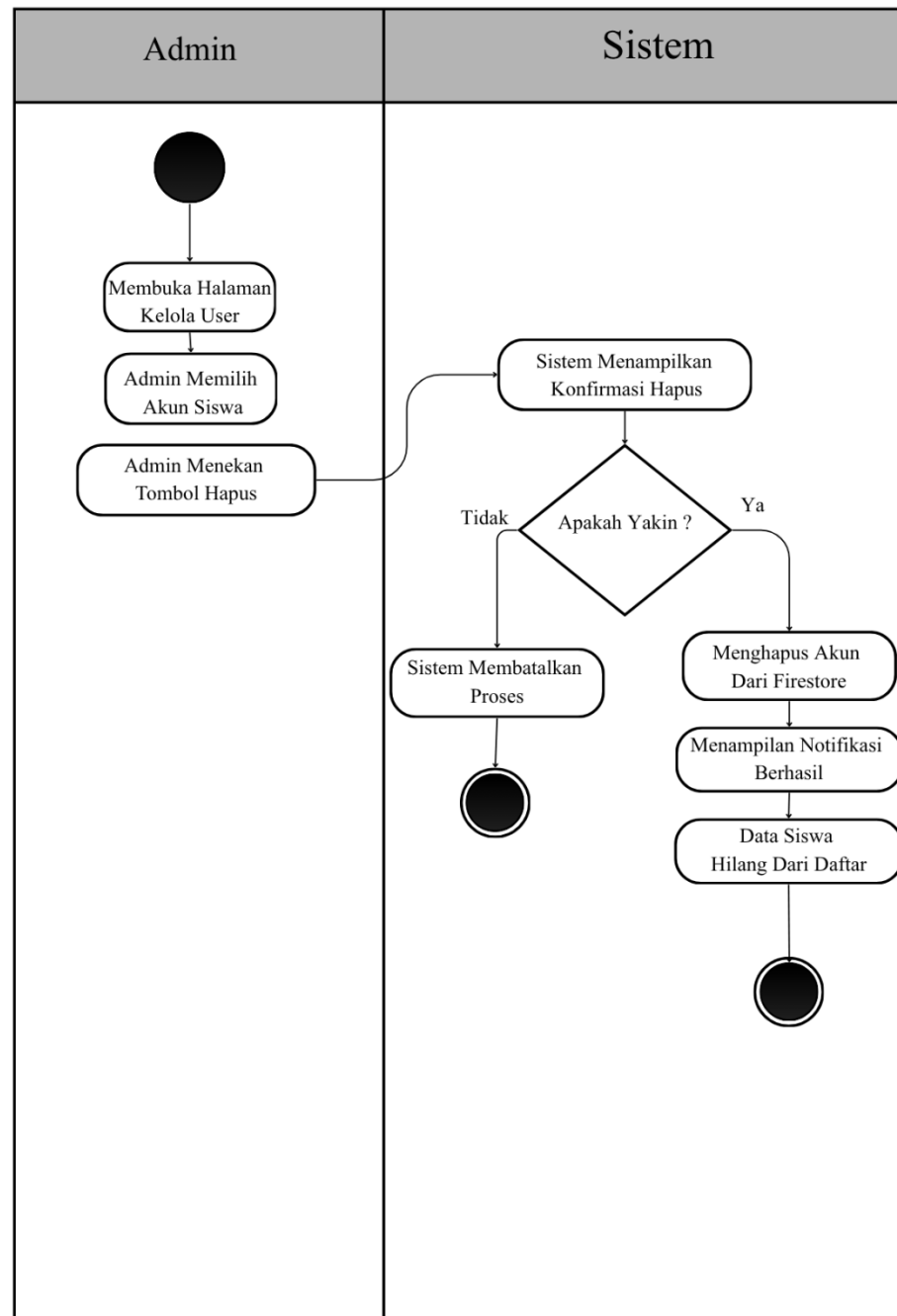
13. Activity Diagram Reset Password



Sumber : data diolah oleh penulis (2026)

Gambar 4. 14 *Reset Password*

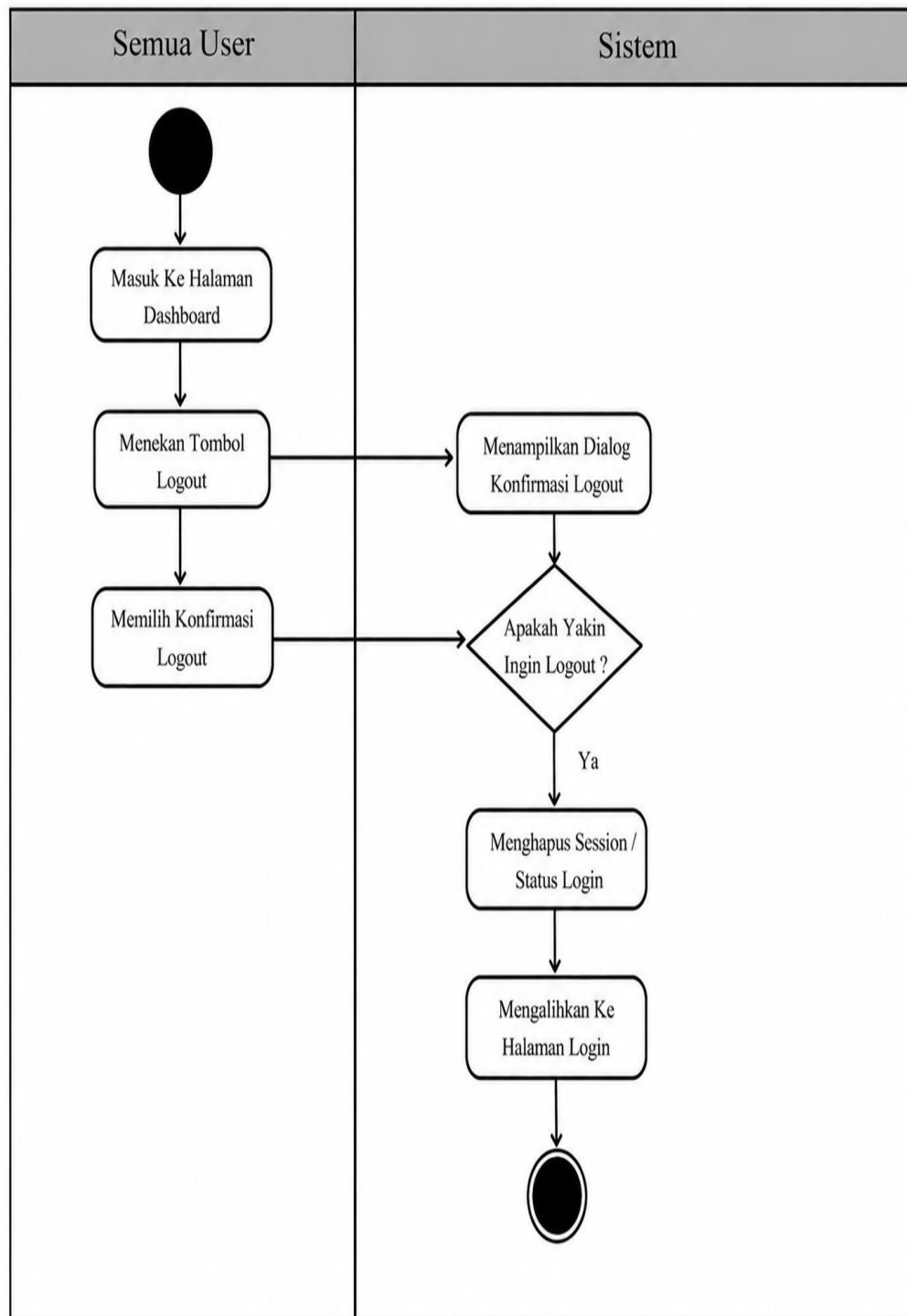
14. Activity Diagram Menghapus Akun



Sumber : data diolah oleh penulis (2026)

Gambar 4. 15 Menghapus Akun Siswa

15. Activity Diagram Log Out Umum User

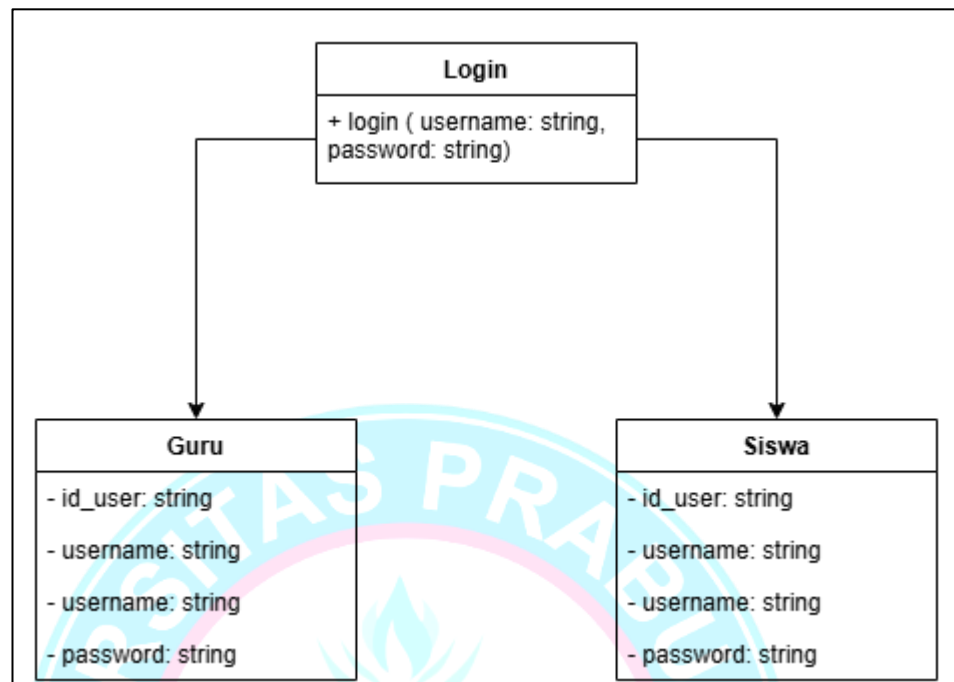


Sumber : data diolah oleh penulis (2026)

Gambar 4. 16 Log Out Umum User

c. *Class Diagram*

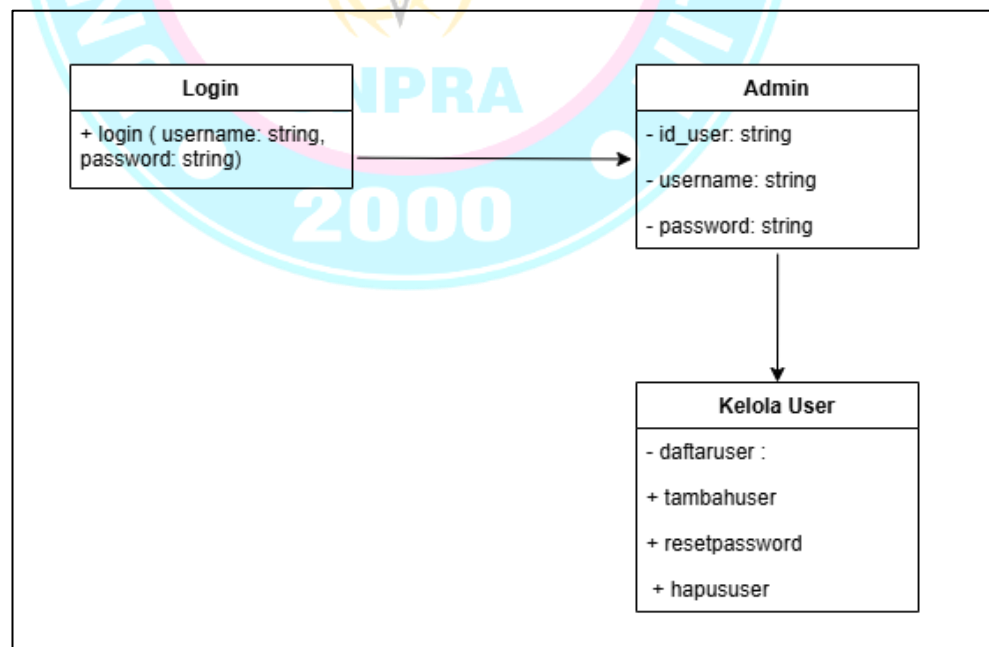
1. *Class Diagram Login Umum User*



Sumber : data diolah oleh penulis (2026)

Gambar 4. 17 *Class Diagram Login Umum*

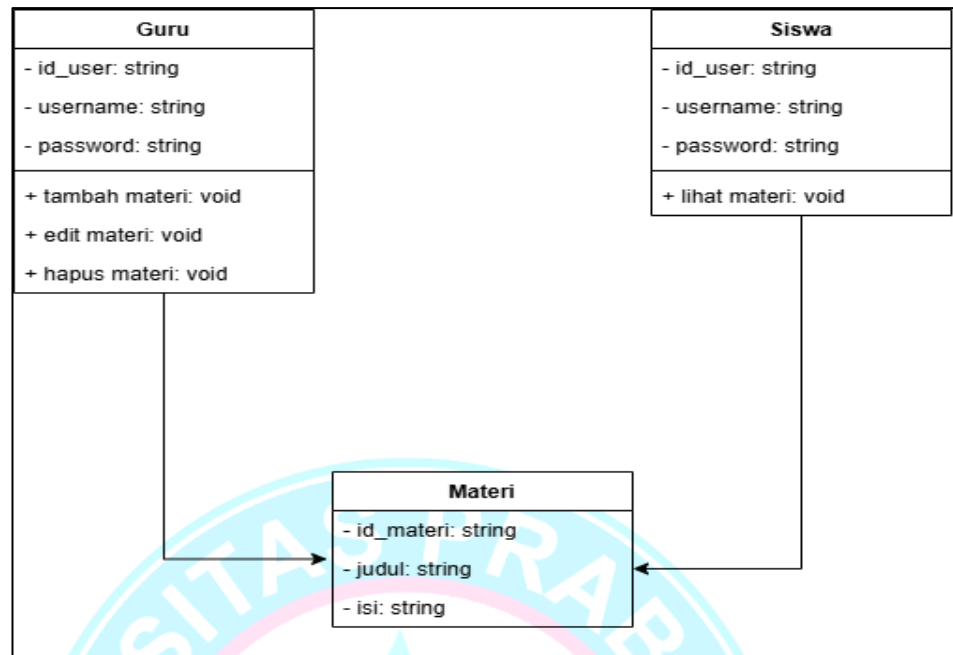
2. *Class Diagram Kelola User*



Sumber : data diolah oleh penulis (2026)

Gambar 4. 18 *Class Diagram Kelola User*

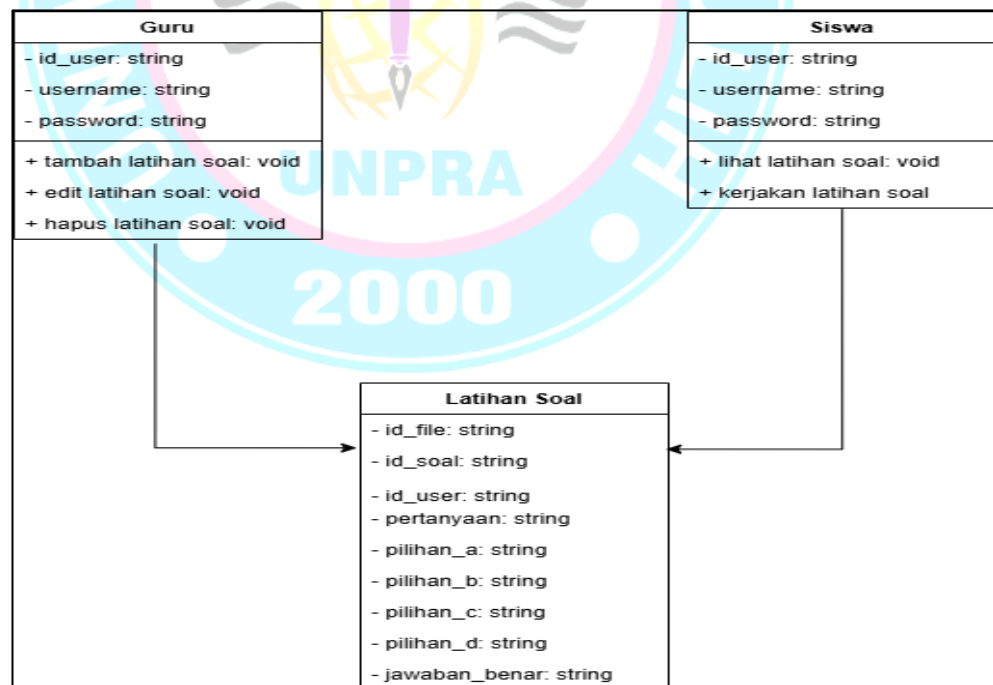
3. Class Diagram Materi



Sumber : data diolah oleh penulis (2026)

Gambar 4. 19 Class Diagram Materi

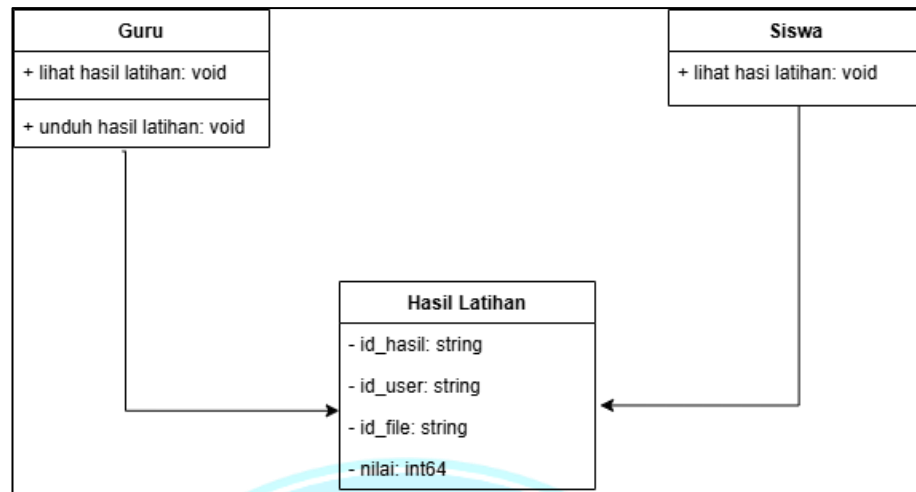
4. Class Diagram Latihan Soal



Sumber : data diolah oleh penulis (2026)

Gambar 4. 20 Class Diagram Latihan Soal

5. Class Diagram Hasil Latihan Soal



Sumber : data diolah oleh penulis (2026)

Gambar 4. 21 *Class Diagram* Hasil Latihan Soal

4.2.2 Perancangan Database (Fisik)

a. Tabel User

Tabel 4. 1 User

Nama Field	Tipe Data	Keterangan
<i>Id_User</i>	String	Primary Key (*)
<i>Username</i>	String	Nama Pengguna
<i>Password</i>	String	Disimpan <i>Firebase</i> Authentication
Role	String	Role Pengguna
created_at	Timestamp	Waktu akun dibuat

Sumber : data diolah oleh penulis (2026)

b. Tabel Materi

Tabel 4. 2 Materi

Nama Field	Tipe Data	Keterangan
id_materi	String	Primary Key (*)
Judul	String	
Isi	String	
gambar	String	URL

Created_at	Timestamp	Waktu materi dibuat
------------	-----------	---------------------

Sumber : data diolah oleh penulis (2026)

c. Tabel File Latihan Soal

Tabel 4. 3 File Latihan Soal

Nama Field	Tipe Data	Keterangan
id_file	String	Primary Key (*)
Judul	String	
Created_at	Timestamp	Waktu file dibuat

Sumber : data diolah oleh penulis (2026)

d. Tabel Latihan Soal

Tabel 4. 4 Latihan Soal

Nama Field	Tipe Data	Keterangan
id_soal	String	Primary Key (*)
Id_file	String	Relasi ke file latihan soal
pertanyaan	String	Isi pertanyaan
pilihan_a	String	Jawaban A
pilihan_b	String	Jawaban B
pilihan_c	String	Jawaban C
pilihan_d	String	Jawaban D
jawaban_benar	String	Jawaban Benar

Sumber : data diolah oleh penulis (2026)

e. Tabel Hasil Latihan Soal

Tabel 4. 5 Hasil Latihan Soal

Nama Field	Tipe Data	Keterangan
id_hasil	String	Primary Key (*)
id_User	String	Id Pengguna
id_file	String	Id File Latihan
Username	String	Nama Pengguna
Nilai	Int 64	Nilai Hasil
Total_soal	Int 64	Total Soal

Created_at	Timestamp	Waktu Hasil Disimpan
Sumber : data diolah oleh penulis (2026)		

4.2.3 Perancangan Antar Muka

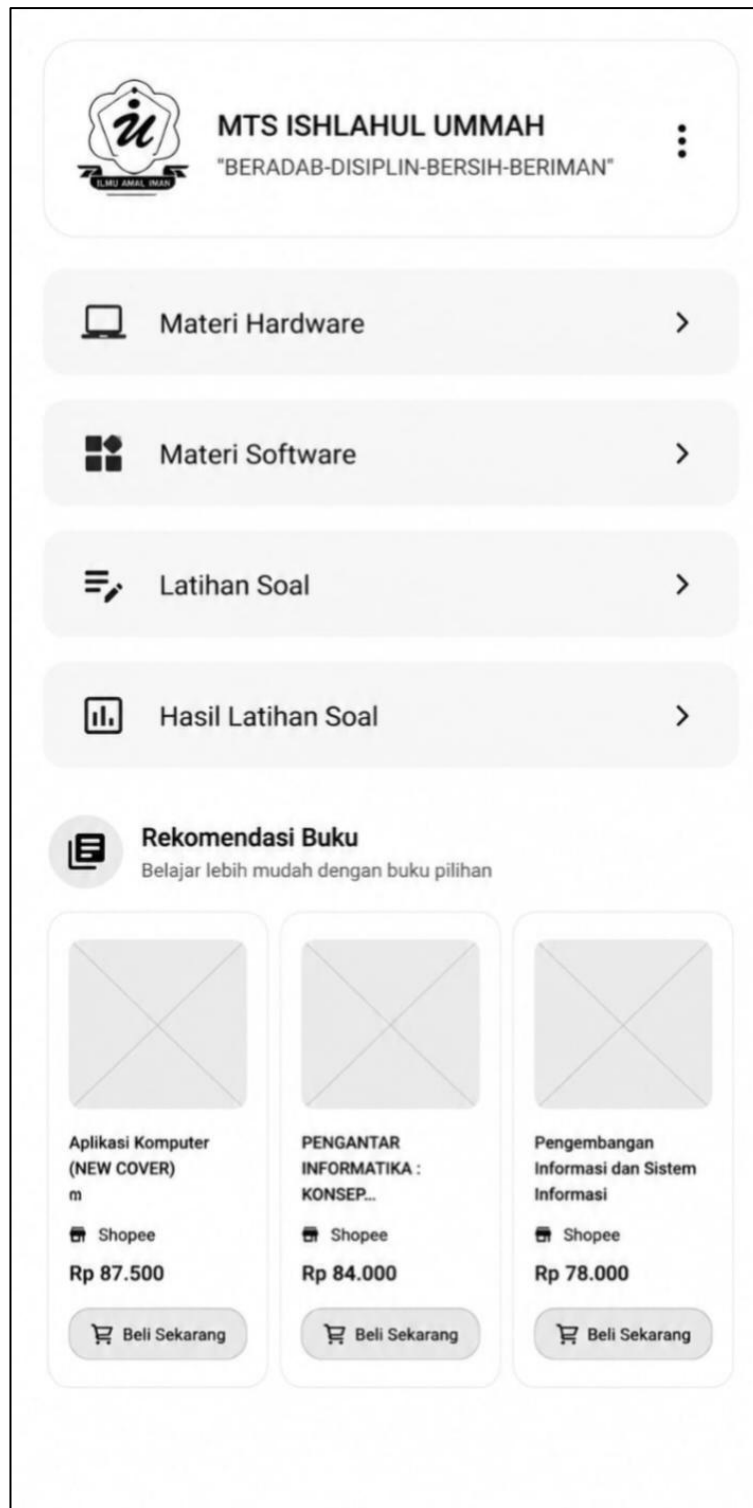
a. Halaman *Login*

The image shows a mobile app login screen. At the top right, there is a language selector showing 'Bahasa Indonesia' with a dropdown arrow. The main content area is a white rounded rectangle with a large black 'X' icon at the top center. Below the icon, the text 'MEDIA PEMBELAJARAN KOMPUTER' is displayed in bold, followed by the tagline 'BERADAB-DISIPLIN-BERSIH-BERIMAN' in a smaller font. There are two input fields: 'Username' with a person icon and 'Password' with a lock icon and a toggle icon for visibility. A 'Lupa Password?' link is located below the password field. A large black 'LOGIN' button is at the bottom of the form. The footer of the app shows the copyright '© afri.rhman'.

Sumber : data diolah oleh penulis (2026)

Gambar 4. 22 Halaman *Login*

b. Halaman Menu Utama



Sumber : data diolah oleh penulis (2026)

Gambar 4. 23 Halaman Menu Utama

c. Halaman Materi Guru



Sumber : data diolah oleh penulis (2026)

Gambar 4. 24 Halaman Materi Guru

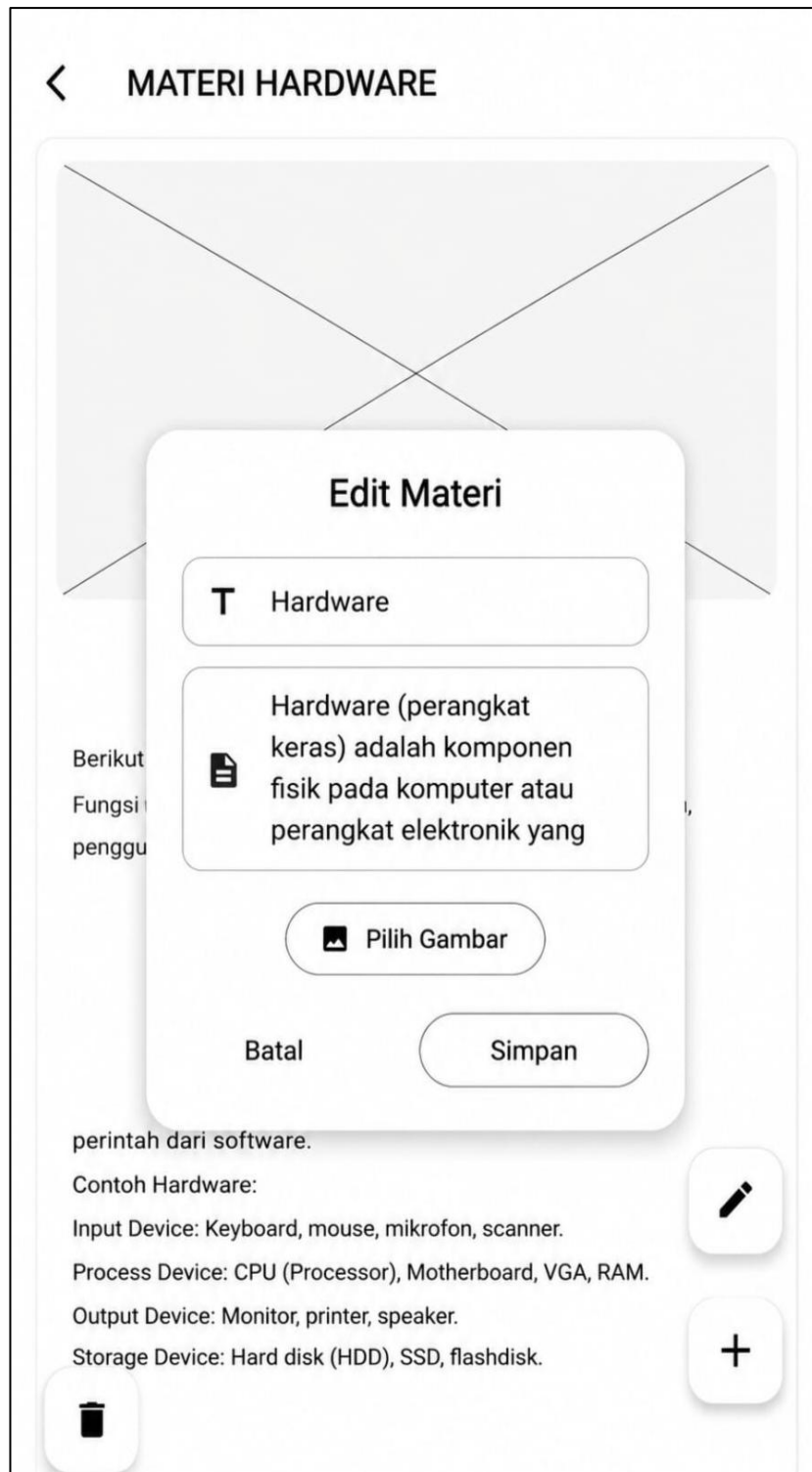
d. Halaman Menambahkan Materi

The screenshot displays a mobile application interface for adding hardware material. At the top, there is a header with a back arrow and the text "MATERI HARDWARE". Below the header, a modal titled "Tambah Materi" is open. The modal contains two input fields: "Judul" (Title) and "Isi" (Content). Below the "Isi" field, there is a "Pilih Gambar" (Select Image) button with a camera icon. At the bottom of the modal, there are two buttons: "Batal" (Cancel) and "Simpan" (Save). In the background, a list of hardware examples is visible, including "Berikut...", "Perikut...", "softwgb...", "Berikut...", "Fungsi...", and "penggu...". At the bottom of the screen, there is a section titled "Contoh Hardware:" with a list of hardware components: "Input Device: Keyboard, mouse, mikrofon, scanner.", "Process Device: CPU (Processor), Motherboard, VGA, RAM.", "Output Device: Monitor, printer, speaker.", and "Storage Device: Hard disk (HDD), SSD, flashdisk.".

Sumber : data diolah oleh penulis (2026)

Gambar 4. 25 Halaman Menambahkan Materi

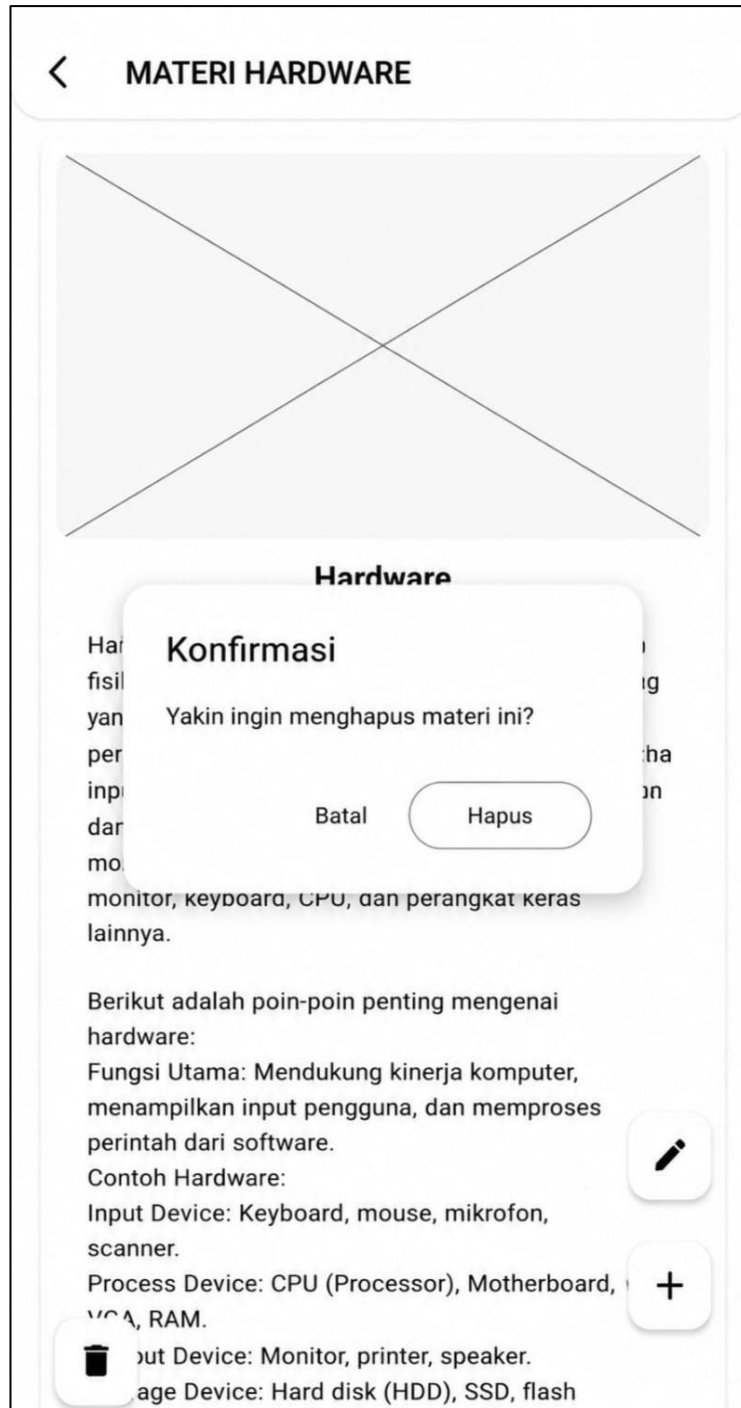
e. Halaman Edit Materi



Sumber : data diolah oleh penulis (2026)

Gambar 4. 26 Halaman Edit Materi

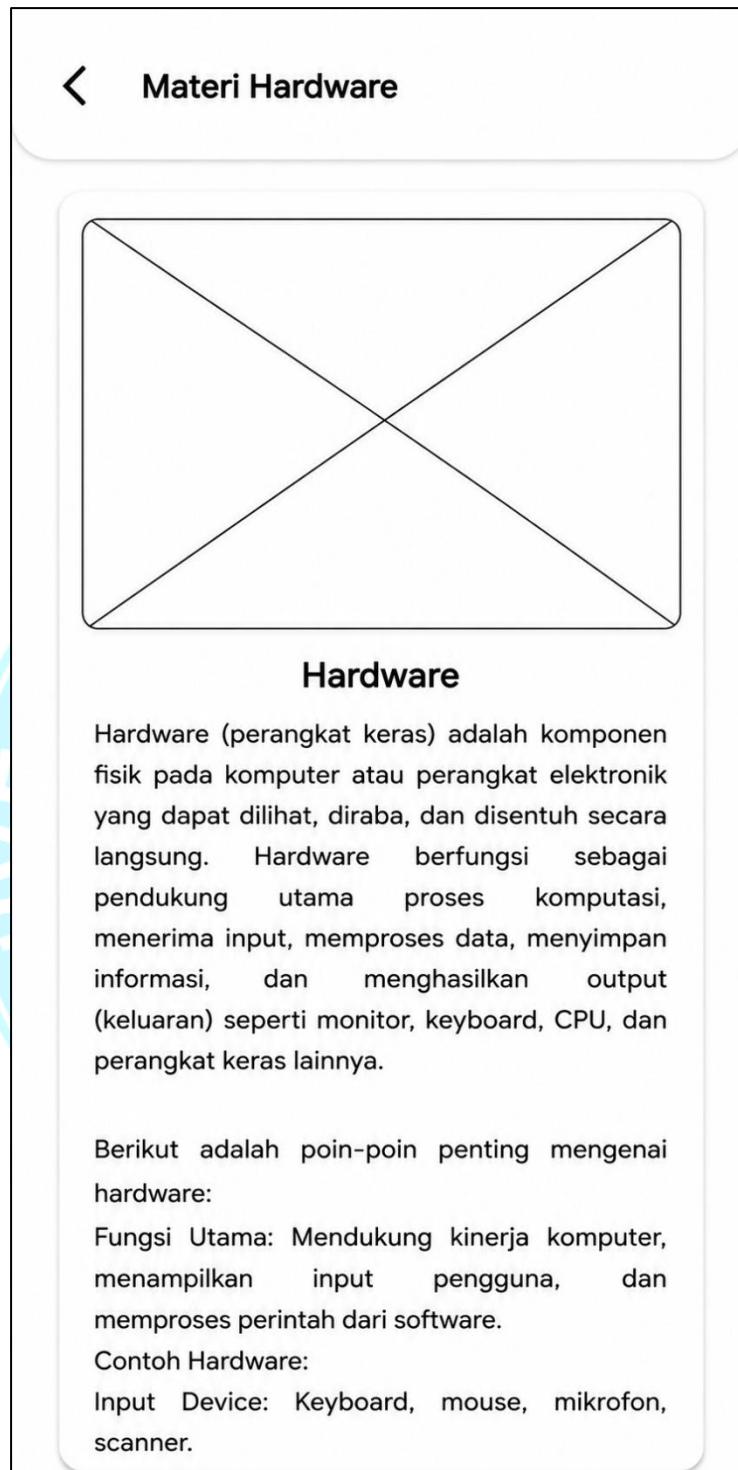
f. Halaman Hapus Materi



Sumber : data diolah oleh penulis (2026)

Gambar 4. 27 Halaman Hapus Materi

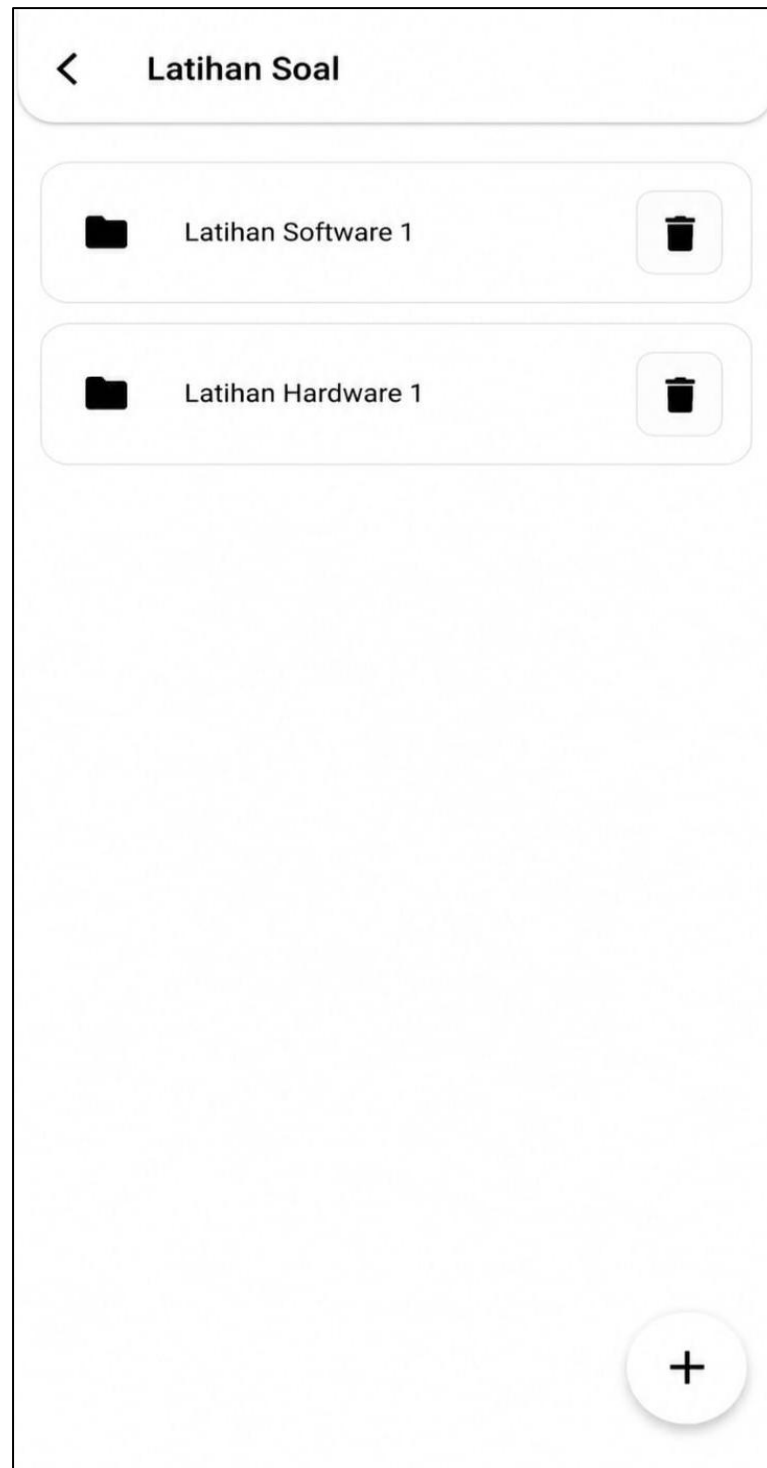
g. Halaman Materi Siswa



Sumber : data diolah oleh penulis (2026)

Gambar 4. 28 Halaman Materi Siswa

h. Halaman File Latihan Soal



Sumber : data diolah oleh penulis (2026)

Gambar 4. 29 Halaman File Latihan Soal

i. **Halaman Menambahkan File Latihan Soal**

< **Latihan Soal**

Latihan Software 1

Latihan Hardware 1

Tambah Latihan Baru

Nama Latihan

Masukkan nama latihan...

Batal Simpan

Sumber : data diolah oleh penulis (2026)

Gambar 4. 30 Halaman Menambahkan File Latihan Soal

j. Halaman Latihan Soal Guru

< Latihan Hardware 1

1/10

Soal 1

Perangkat keras komputer yang berfungsi sebagai otak komputer adalah

☐ A. RAM

☐ B. Harddisk

☐ C. Processor

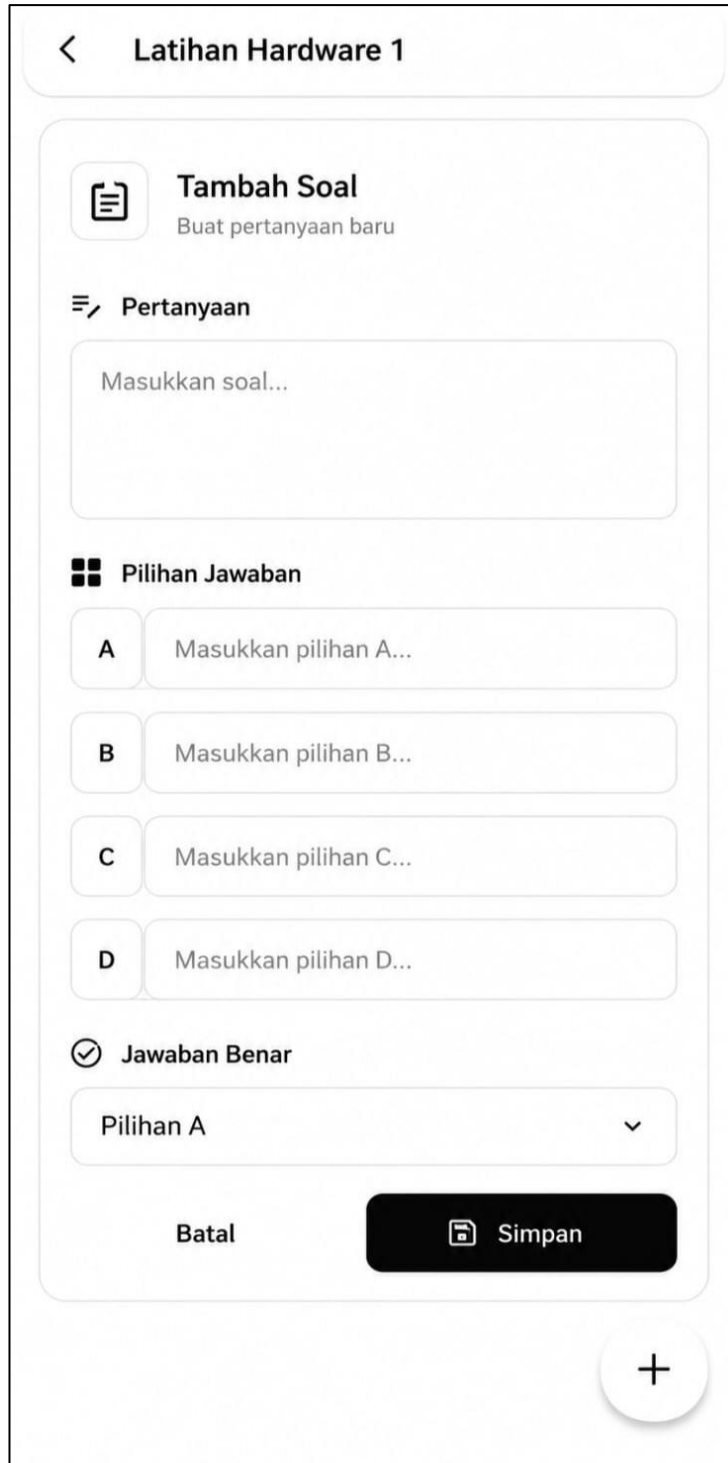
☐ D. Monitor

+

Sumber : data diolah oleh penulis (2026)

Gambar 4. 31 Halaman Latihan Soal Guru

k. Halaman Menambahkan Latihan Soal



< **Latihan Hardware 1**

 **Tambah Soal**
Buat pertanyaan baru

 **Pertanyaan**

Masukkan soal...

 **Pilihan Jawaban**

A Masukkan pilihan A...

B Masukkan pilihan B...

C Masukkan pilihan C...

D Masukkan pilihan D...

 **Jawaban Benar**

Pilihan A ▼

Batal  **Simpan**

+

Sumber : data diolah oleh penulis (2026)

Gambar 4. 32 Halaman Menambahkan Latihan Soal

1. Halaman Edit Latihan Soal

< **Latihan Hardware 1**

Edit Soal
Perbarui data soal

Pertanyaan

Perangkat keras komputer yang berfungsi sebagai otak komputer adalah

Pilihan Jawaban

A RAM

B Harddisk

C Processor

D Monitor

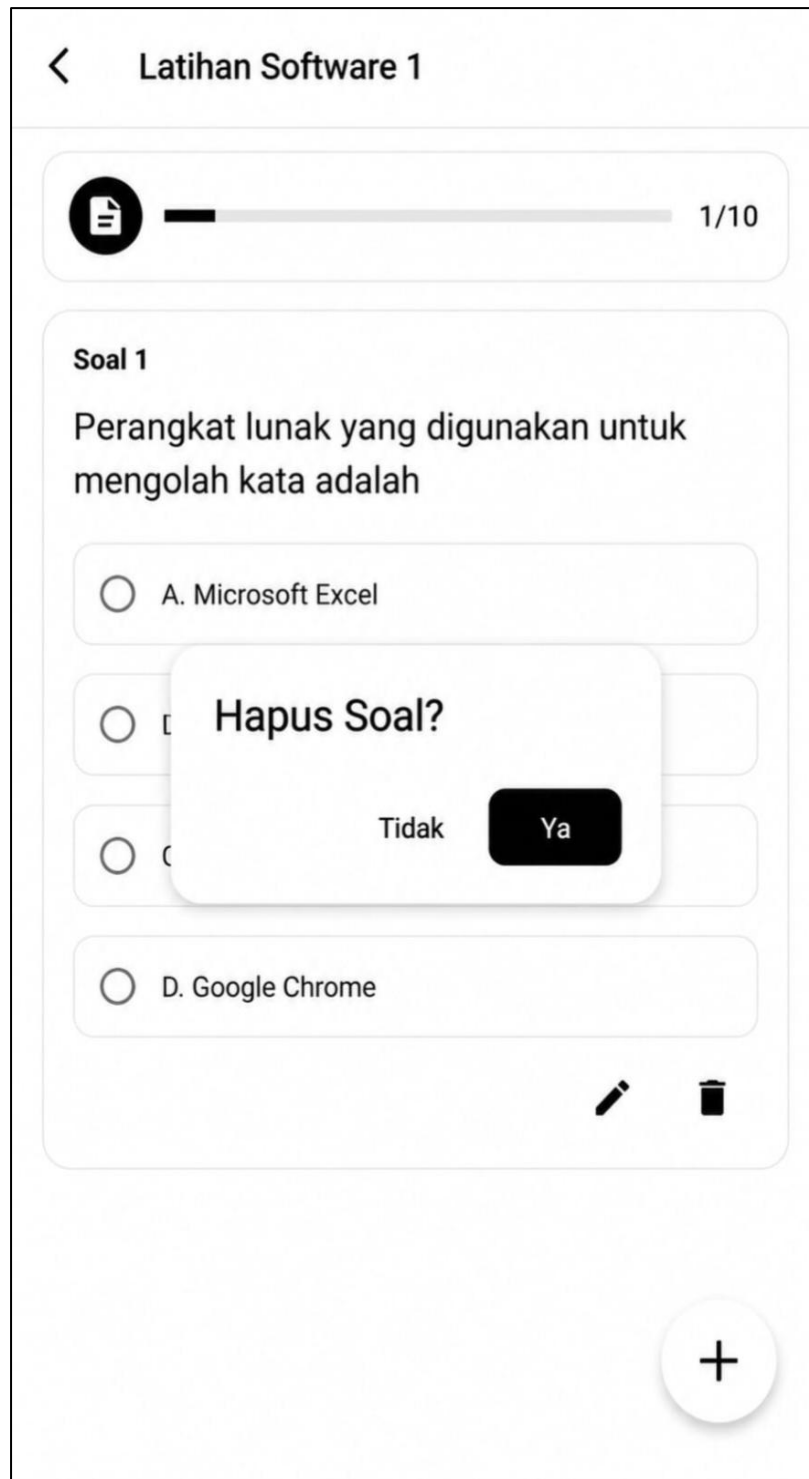
Jawaban Benar

Pilihan C

Batal **Simpan**

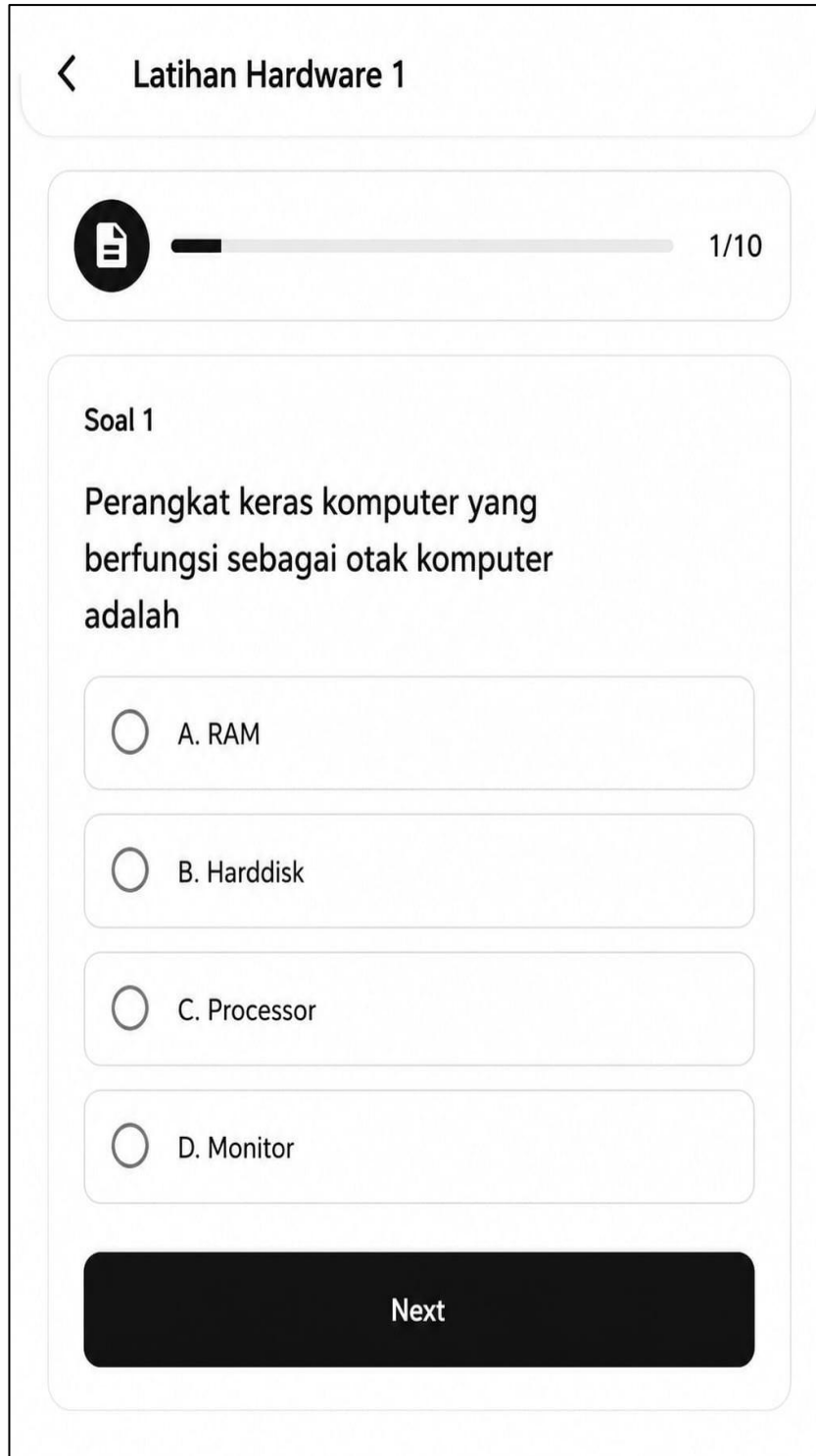
Sumber : data diolah oleh penulis (2026)

Gambar 4. 33 Halaman Edit Latihan Soal

m. Halaman Hapus Latihan Soal

Sumber : data diolah oleh penulis (2026)

Gambar 4. 34 Halaman Hapus Latihan Soal

n. Halaman Latihan Soal Siswa

< Latihan Hardware 1

 1/10

Soal 1

Perangkat keras komputer yang berfungsi sebagai otak komputer adalah

☐ A. RAM

☐ B. Harddisk

☐ C. Processor

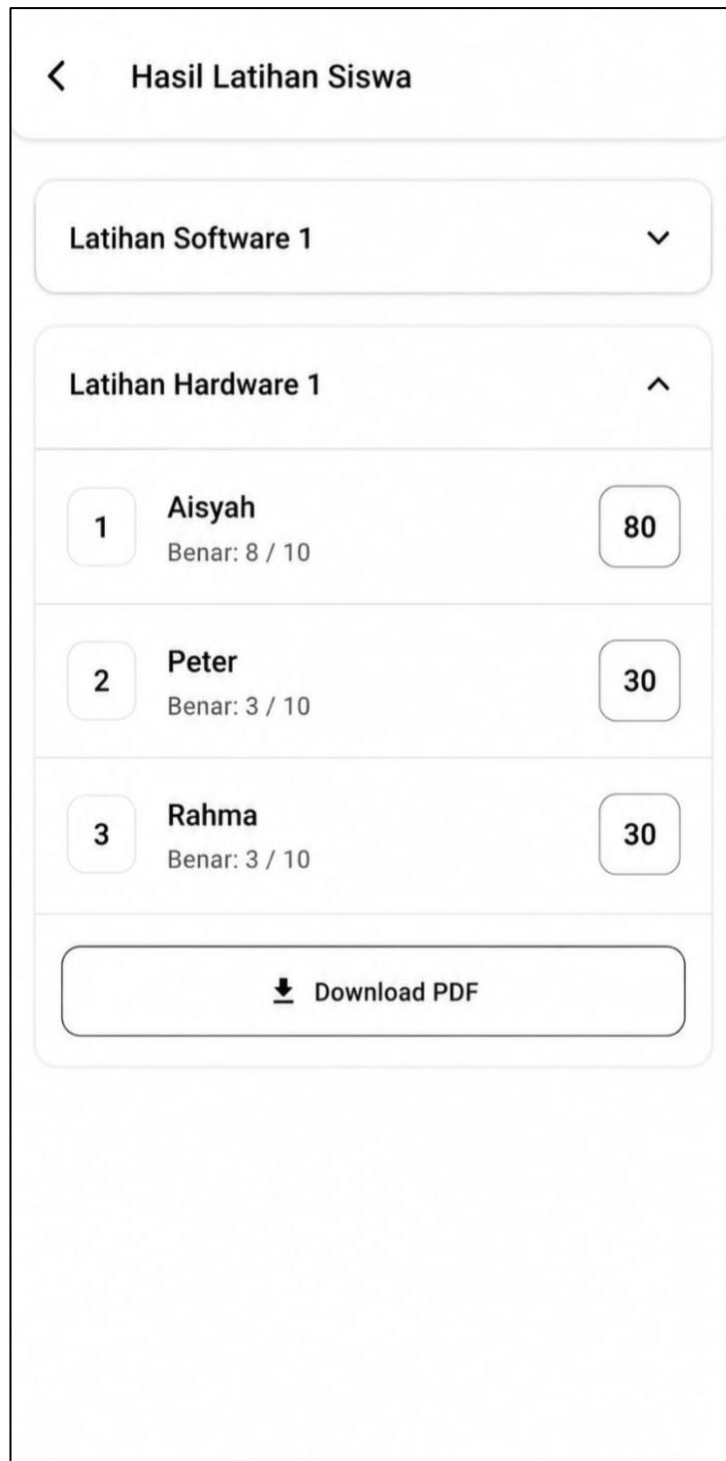
☐ D. Monitor

Next

Sumber : data diolah oleh penulis (2026)

Gambar 4. 35 Halaman Latihan Soal Siswa

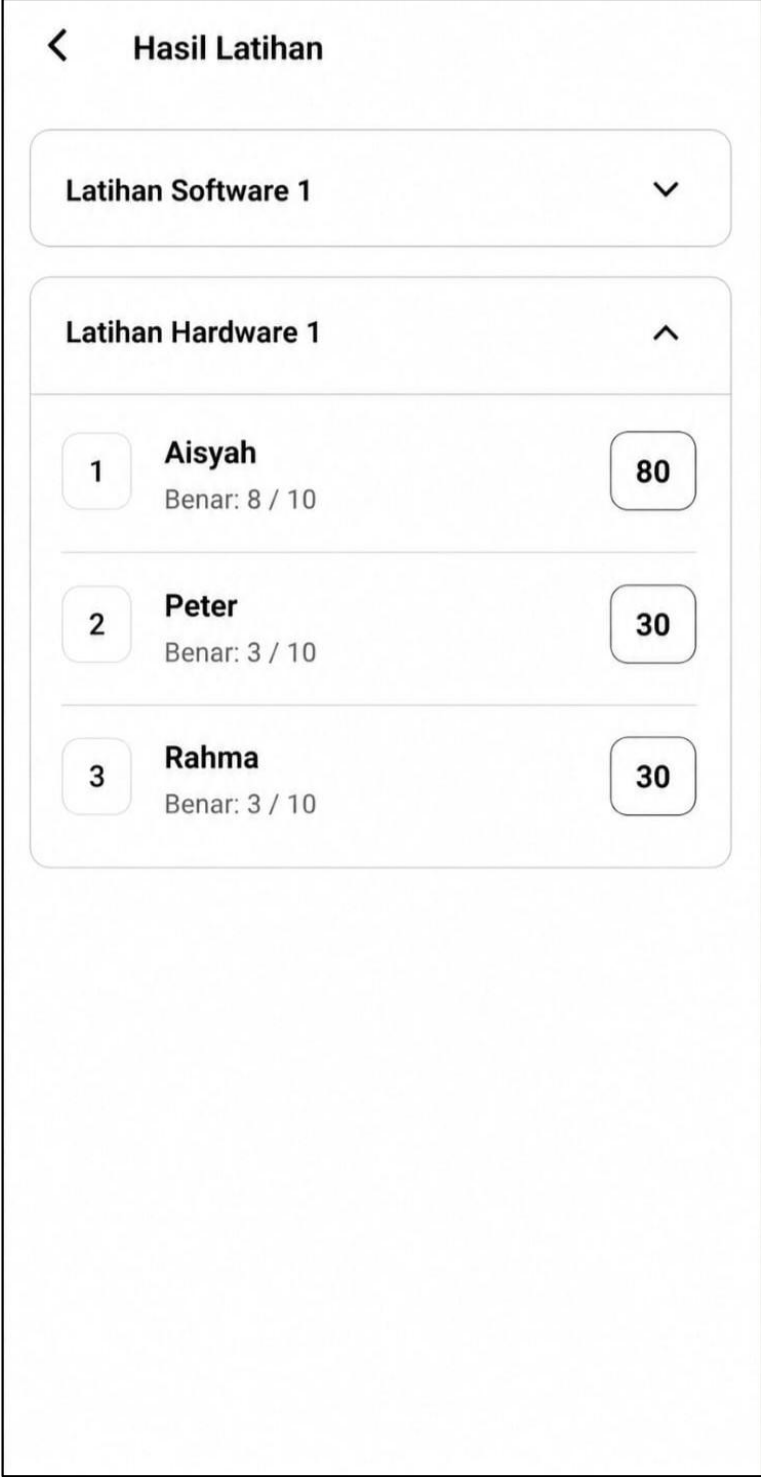
o. Halaman Hasil Latihan Soal Guru



Sumber : data diolah oleh penulis (2026)

Gambar 4. 36 Halaman Hasil Latihan Soal Guru

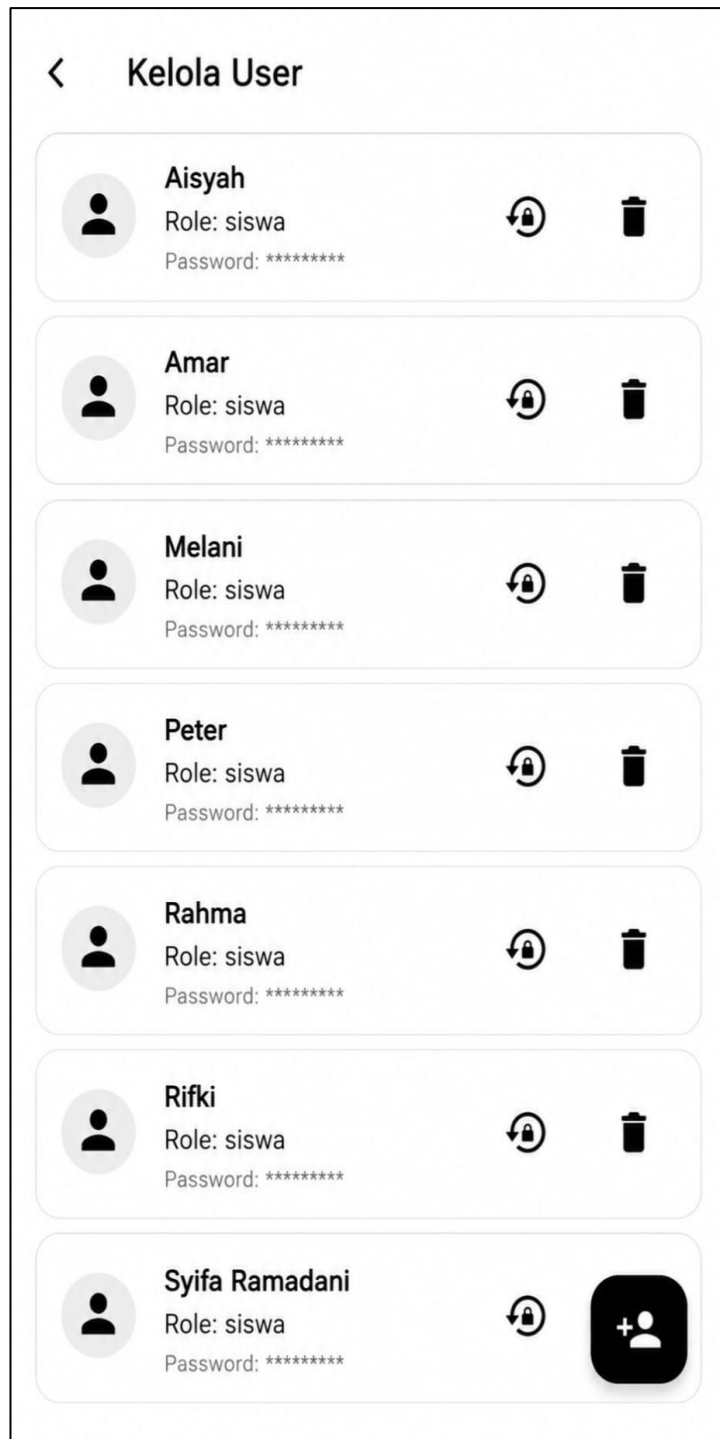
p. Halaman Hasil Latihan Soal Siswa



< Hasil Latihan		
Latihan Software 1		
Latihan Hardware 1		
1	Aisyah Benar: 8 / 10	80
2	Peter Benar: 3 / 10	30
3	Rahma Benar: 3 / 10	30

Sumber : data diolah oleh penulis (2026)

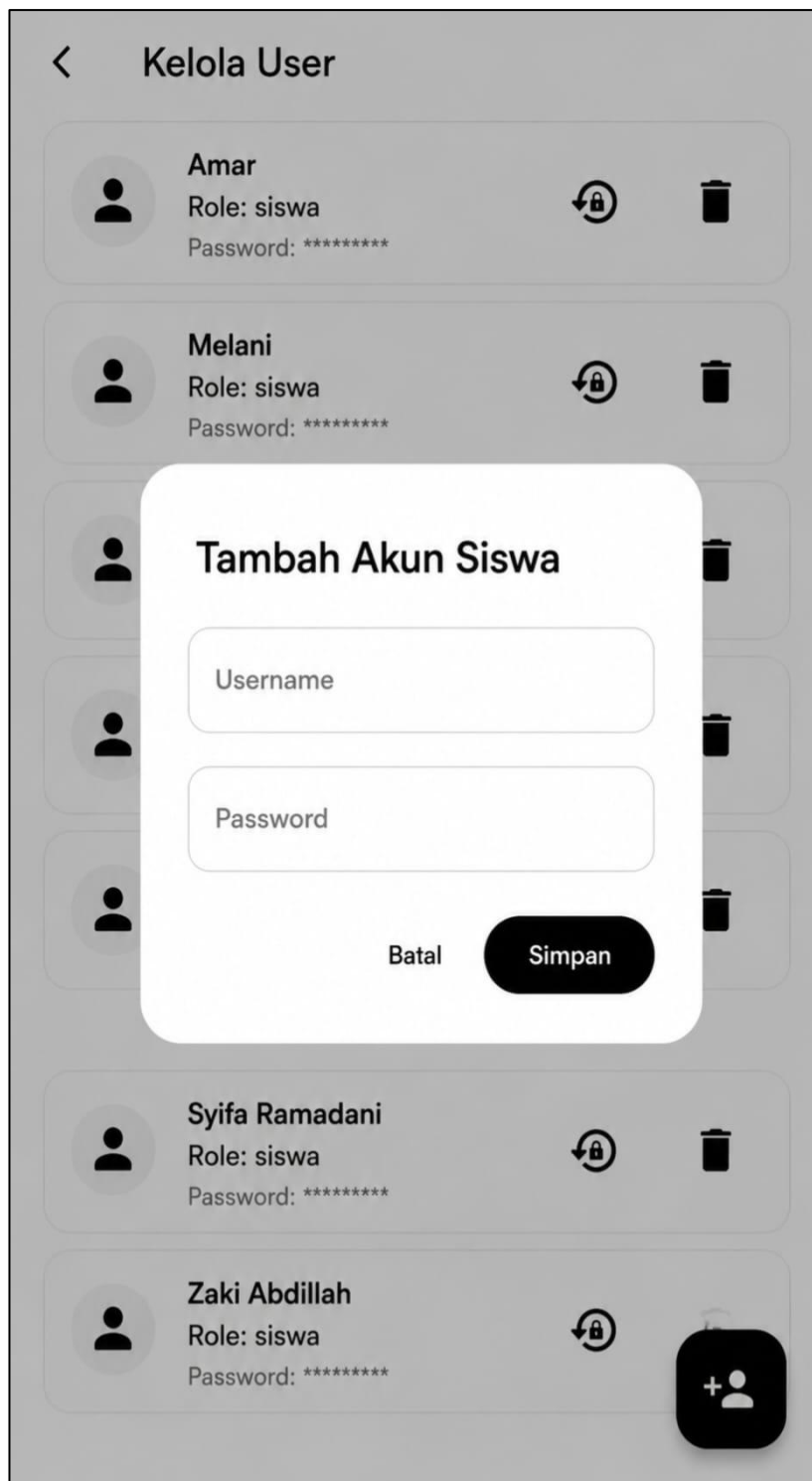
Gambar 4. 37 Halaman Hasil Latihan Soal Siswa

q. Halaman Kelola User

Sumber : data diolah oleh penulis (2026)

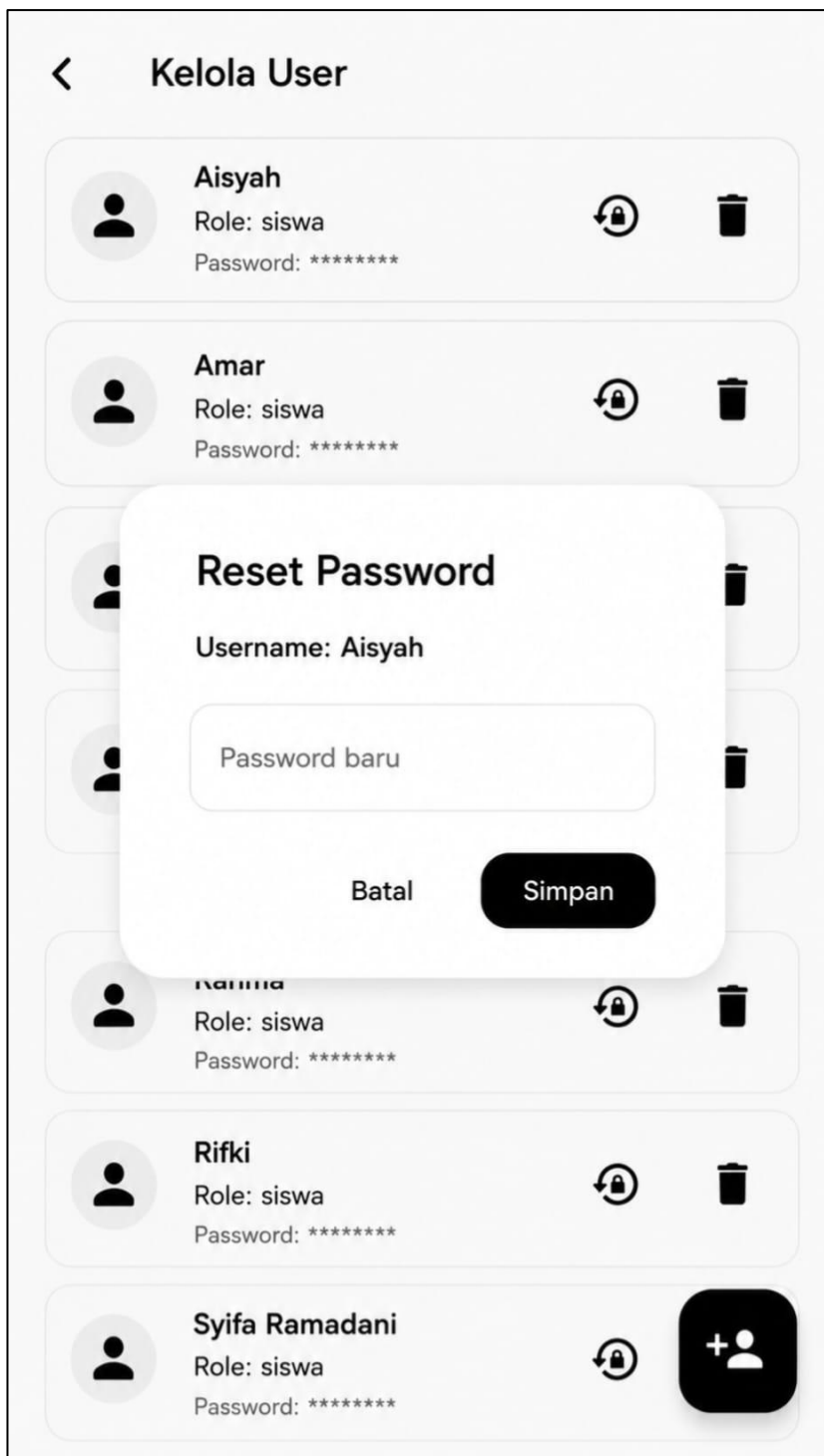
Gambar 4. 38 Halaman Kelola User

r. Halaman Tambah Akun Siswa



Sumber : data diolah oleh penulis (2026)

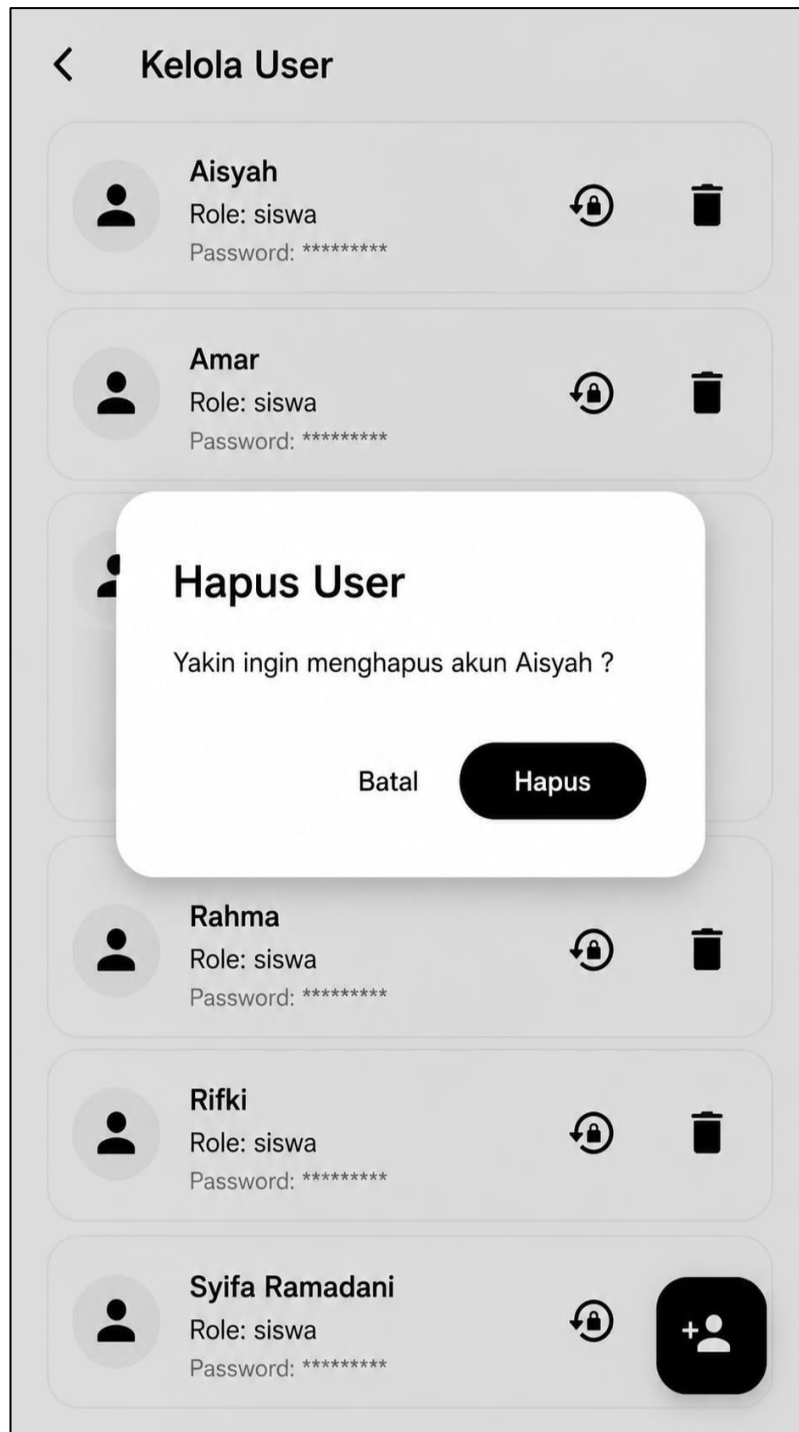
Gambar 4. 39 Halaman Tambah Akun Siswa

s. Halaman *Reset Password Akun*

Sumber : data diolah oleh penulis (2026)

Gambar 4. 40 Halaman *Reset Password Akun*

t. Halaman Hapus Akun



Sumber : data diolah oleh penulis (2026)

Gambar 4. 41 Halaman Hapus Akun

BAB V

HASIL DAN PEMBAHASAN

5.1 Spesifikasi Perangkat

Dalam penelitian ini, pengembangan aplikasi didukung oleh pemanfaatan berbagai perangkat pada setiap tahapannya. Perangkat tersebut berperan penting mulai dari proses perancangan, penulisan kode program, hingga tahap pengujian sistem untuk memastikan aplikasi berjalan dengan baik dan sesuai dengan kebutuhan pengguna. Penggunaan perangkat yang tepat dan sesuai kebutuhan tidak hanya meningkatkan efisiensi proses pengembangan, tetapi juga membantu meminimalkan kesalahan serta mempercepat waktu penyelesaian aplikasi. Selain itu, pemilihan perangkat yang optimal turut mendukung kestabilan dan performa sistem yang dihasilkan. Adapun perangkat yang digunakan mencakup dua kategori utama, yaitu perangkat keras (*hardware*) dan perangkat lunak (*software*), yang saling terintegrasi dan mendukung dalam keseluruhan proses pengembangan aplikasi.

5.1.1 Perangkat Keras yang Digunakan

Perangkat keras yang digunakan dalam penelitian ini merupakan perangkat yang secara langsung mendukung proses perancangan dan pengembangan media pembelajaran berbasis *Android* pada mata pelajaran komputer. Perangkat tersebut digunakan untuk menjalankan aplikasi, melakukan pengujian fitur, serta memastikan media pembelajaran dapat berjalan dengan baik pada perangkat *Android*. Kinerja perangkat keras yang digunakan sangat berpengaruh terhadap kelancaran proses pengembangan, kecepatan pengujian aplikasi, serta stabilitas saat media pembelajaran dijalankan oleh pengguna. Oleh karena itu, pemilihan perangkat keras yang sesuai menjadi faktor penting dalam menghasilkan aplikasi yang optimal. Adapun spesifikasi perangkat keras yang digunakan dalam penelitian ini adalah sebagai berikut.

1. *Processor* : Intel Core i3 Gen 11
2. *RAM* : 8 GB
3. *Storage* : 512 Gb
4. Keyboard
5. Mouse
6. Handphone *Android* OS 14

5.1.2 Perangkat Lunak yang Digunakan

Perangkat lunak dalam penelitian ini berfungsi sebagai komponen utama yang mendukung proses perancangan, pengembangan, dan implementasi sistem. Perangkat lunak tersebut digunakan dalam kegiatan penulisan kode program, pengolahan data, serta pengujian aplikasi yang telah dikembangkan. Pemilihan perangkat lunak yang tepat memungkinkan sistem dapat berjalan secara optimal dan stabil sesuai dengan kebutuhan pengembangan. Oleh karena itu, perangkat lunak memiliki peran penting dalam memastikan seluruh fitur aplikasi dapat berfungsi dengan baik. Adapun perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Sistem Operasi : *Microsoft Windows 11 Home*
2. Bahasa Pemrograman : *Dart SDK 3.7.2*
3. *Framework* : *Flutter 3.29.3*
4. *Firebase_core*: *3.6.0*
5. *Cloud_firestore*: *5.4.4*
6. *Firebase_auth*: *5.3.1*
7. *Code Editor* : *Visual Studio Code 1.120.0*
8. Alat Bantu Design: *Figma 124.5.6*
9. *Cloudinary* : *5.4.4*

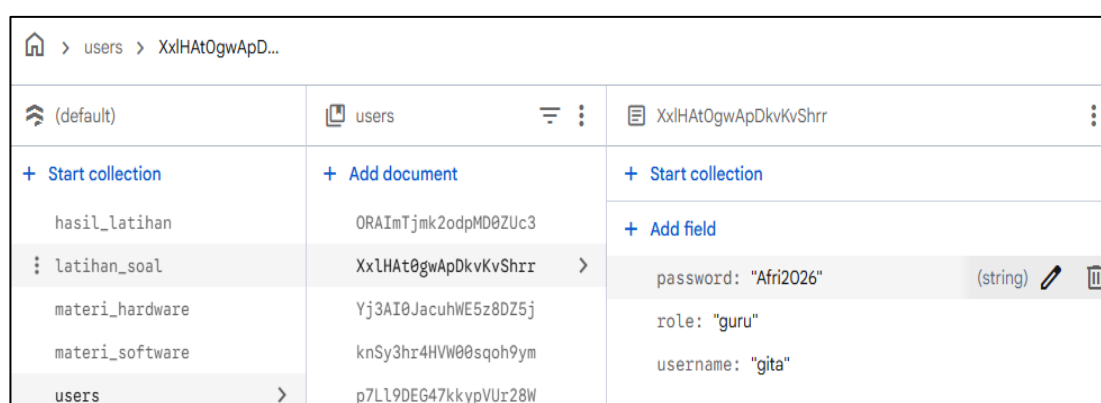
5.2 Implementasi Database

Database pada aplikasi media pembelajaran komputer ini dibuat untuk menyimpan dan mengelola seluruh data yang digunakan dalam sistem, seperti data pengguna, materi pembelajaran, latihan soal, dan hasil latihan siswa.

Struktur database disusun secara terorganisir agar proses pengolahan data dapat berjalan dengan lebih cepat, mudah, dan terhubung antar bagian sistem.

Penggunaan database ini juga membantu sistem dalam mengurangi terjadinya data yang berulang serta mempermudah proses pengelolaan informasi pada aplikasi. Pada penelitian ini digunakan *Firebase Firestore* sebagai media penyimpanan database, sedangkan *Cloudinary* digunakan untuk menyimpan file gambar yang terdapat pada materi pembelajaran.

5.2.1 Implementasi Tabel *Users*



Collection	Document ID	Fields
users	XxlHAt0gwApDkvKvShrr	password: "Afri2026" (string) role: "guru" username: "gita"

Sumber : data diolah oleh penulis (2026)

Gambar 5. 1 Tabel *Users*

Tabel *Users* pada basis data *Firebase Firestore* digunakan untuk menyimpan data pengguna yang memiliki hak akses ke dalam aplikasi media pembelajaran. Pada koleksi ini, setiap pengguna disimpan dalam bentuk dokumen dengan identitas unik berupa Document ID yang dibuat secara otomatis oleh *Firebase*, sehingga memudahkan proses pengelolaan serta pencarian data pengguna di dalam sistem.

Struktur data pada koleksi *Users* terdiri dari beberapa atribut penting, seperti *Username*, *password*, dan *role*. Atribut *Username* digunakan sebagai identitas pengguna saat melakukan login ke aplikasi, sedangkan atribut *password* digunakan dalam proses autentikasi pengguna. Selain itu, atribut *role* berfungsi untuk menentukan hak akses setiap pengguna di dalam sistem, misalnya sebagai guru maupun pengguna lainnya.

Berdasarkan struktur tersebut, sistem dapat mengelola proses autentikasi dan pembagian hak akses pengguna dengan lebih teratur. Penggunaan koleksi

Users juga memungkinkan aplikasi membedakan tampilan serta fitur yang dapat diakses oleh masing-masing pengguna sesuai dengan perannya di dalam sistem.

5.2.2 Implementasi Tabel Materi Hardware

(default)	materi_hardwar...	U4A7PTp9iUqviAQ3903u
+ Start collection hasil_latihan latihan_soal materi_hardwar > materi_software users	+ Add document U4A7PTp9iUqviAQ3903u > URr9P390QQYbz8L7huIc VcBooDbINItKpViQ08c4 fo780yBkuU7Nh3SsQePP lmWSAAmbkYq6TPK5zA1A qBW7yypQl7ehLHn0JESm	+ Start collection + Add field created: May 4, 2026 at 3:59:20 PM UTC+7 gambar: "https://res.cloudinary.com/dydxoymyk/image/upload/ isi: "Hardware (perangkat keras) adalah komponen fisik pada komputer atau perangkat elektronik yang dapat dilihat, diraba, dan disentuh secara langsung. Hardware berfungsi sebagai pendukung utama proses komputasi, menerima input, memproses data, menyimpan informasi, dan menghasilkan output

Sumber : data diolah oleh penulis (2026)

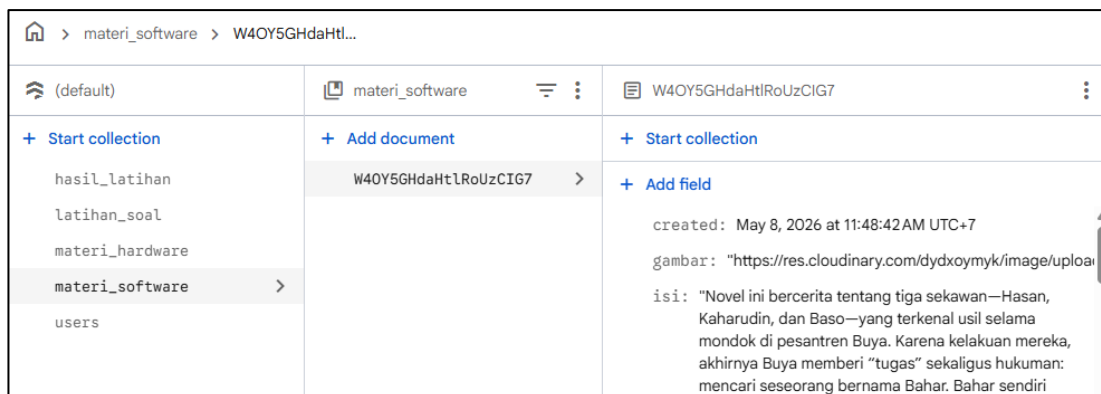
Gambar 5. 2 Table Materi Hardware

Tabel *materi hardware* pada basis data *Firebase Firestore* digunakan untuk menyimpan data materi pembelajaran mengenai perangkat keras komputer pada aplikasi media pembelajaran. Setiap data materi disimpan dalam bentuk dokumen dengan Document ID unik yang dibuat secara otomatis oleh *Firebase*, sehingga memudahkan proses pengelolaan, pencarian, dan pembaruan data materi di dalam sistem.

Struktur data pada koleksi *materi hardware* terdiri dari beberapa atribut penting, seperti gambar, isi, dan *created*. Atribut gambar digunakan untuk menyimpan tautan gambar materi yang berasal dari layanan penyimpanan *cloud*, sehingga gambar dapat ditampilkan pada aplikasi. Atribut isi berfungsi untuk menyimpan penjelasan atau materi pembelajaran mengenai perangkat keras komputer yang akan dipelajari oleh pengguna. Selain itu, atribut *created* digunakan untuk mencatat waktu penambahan data materi ke dalam sistem.

Dengan adanya struktur tersebut, sistem dapat mengelola materi pembelajaran perangkat keras secara terorganisir dan mudah diakses oleh pengguna. Koleksi *materi hardware* juga memungkinkan admin atau guru untuk menambahkan, mengubah, maupun menghapus materi pembelajaran sesuai kebutuhan aplikasi.

5.2.3 Implementasi Tabel Materi Software



(default)	materi_software	W4OY5GHdaHtlRoUzCIG7
+ Start collection hasil_latihan latihan_soal materi_hardware materi_software > users	+ Add document W4OY5GHdaHtlRoUzCIG7 >	+ Start collection + Add field created: May 8, 2026 at 11:48:42 AM UTC+7 gambar: "https://res.cloudinary.com/dydxoymyk/image/upload..." isi: "Novel ini bercerita tentang tiga sekawan—Hasan, Kaharudin, dan Baso—yang terkenal usil selama mondok di pesantren Buya. Karena kelakuan mereka, akhirnya Buya memberi 'tugas' sekaligus hukuman: mencari seseorang bernama Bahar. Bahar sendiri."

Sumber : data diolah oleh penulis (2026)

Gambar 5. 3 Tabel Materi *Software*

Tabel *materi_software* pada basis data *Firebase Firestore* digunakan untuk menyimpan data materi pembelajaran mengenai perangkat lunak komputer pada aplikasi media pembelajaran. Setiap data materi disimpan dalam bentuk dokumen dengan Document ID unik yang dibuat secara otomatis oleh *Firebase* sehingga memudahkan proses pengelolaan, pencarian, serta pembaruan data di dalam sistem. Penyimpanan data menggunakan model dokumen pada *Firestore* juga membuat proses pengembangan aplikasi menjadi lebih fleksibel karena data dapat diakses secara *real-time* dan terintegrasi langsung dengan aplikasi.

Struktur data pada koleksi *materi_software* terdiri dari atribut gambar, isi, dan created. Atribut gambar digunakan untuk menyimpan tautan gambar materi agar tampilan materi lebih menarik dan mudah dipahami oleh pengguna. Atribut isi digunakan untuk menyimpan penjelasan materi perangkat lunak yang akan ditampilkan pada halaman pembelajaran aplikasi. Sedangkan atribut *created* digunakan untuk mencatat waktu penambahan data materi ke dalam sistem sehingga memudahkan proses pengelolaan data dan pengurutan materi berdasarkan waktu input. Dengan adanya struktur data tersebut, sistem dapat menampilkan materi *software* secara terorganisir, mudah diakses, serta mendukung proses pembelajaran pengguna dengan lebih efektif.

5.2.4 Implementasi Tabel File Latihan Soal

🏠 > latihan_soal > LW4PUF6NFuV...		
🏠 (default)	📄 latihan_soal	📄 LW4PUF6NFuVhZFeE41V8
+ Start collection	+ Add document	+ Start collection
hasil_latihan	LW4PUF6NFuVhZFeE41V8 >	+ Add field
⋮ latihan_soal >	y3AGEHxVIIUgdJruk1AZ	created: May 8, 2026 at 12:48:19 PM UTC+7
materi_hardware		title: "Latihan Software 1"
materi_software		
users		

Sumber : data diolah oleh penulis (2026)

Gambar 5. 4 Table File Latihan Soal

Tabel *latihan_soal* pada basis data *Firebase Firestore* digunakan untuk menyimpan data latihan soal yang tersedia pada aplikasi media pembelajaran. Setiap data latihan soal disimpan dalam bentuk dokumen dengan *Document ID* unik yang dibuat secara otomatis oleh *Firebase* sehingga memudahkan proses pengelolaan dan pencarian data di dalam sistem. Penggunaan *Firestore* juga memungkinkan data latihan soal dapat diakses secara cepat dan tersinkronisasi dengan aplikasi.

Struktur data pada koleksi *latihan_soal* terdiri dari atribut *title* dan *created*. Atribut *title* digunakan untuk menyimpan nama atau judul latihan soal, sedangkan atribut *created* digunakan untuk mencatat waktu penambahan data latihan soal ke dalam sistem. Dengan struktur tersebut, sistem dapat mengelola data latihan soal dengan lebih teratur dan memudahkan pengguna dalam mengakses latihan yang tersedia pada aplikasi. Selain itu, data latihan soal dapat ditampilkan secara rapi sehingga membantu pengguna memilih latihan sesuai kebutuhan pembelajaran.

5.2.5 Implementasi Tabel Soal

latihan_soal > y3AGEHxVIIUgd... soal > wUOODKsmfHb...		
y3AGEHxVIIUgdJruk1AZ + Start collection soal + Add field created: May 8, 2026 at 12:07:08 PM UTC+7 title: "Latihan Hardware 1"	soal + Add document 81EEfnyXqQgNQY8Iq0Jm ACHV47YK3gYiskaQr0UN BFL7SIeeM7Qr9eaDfuUn Bo0ts3uBYBkmwSgreKB C06vrKNyLGqIOZf4Iy4w N7LYmBHFS48N2sZUGJ7z NSoq6gD20nBpegX2LCCB PgKjLL8GLNRevH67CKUd XEKVcwBLhnq2MgMjrQ9C wUOODKsmfHb825Sf3VwU	wUOODKsmfHb825Sf3VwU + Start collection + Add field A: "Flashdisk" B: "Speaker" C: "Webcam" D: "Mikrofon" created: May 8, 2026 at 12:11:11 PM UTC+7 jawaban: "A" soal: "Perangkat berikut yang termasuk media penyimpanan adalah"

Sumber : data diolah oleh penulis (2026)

Gambar 5. 5 Tabel Soal

Tabel soal pada basis data *Firebase Firestore* digunakan untuk menyimpan data pertanyaan latihan soal yang terdapat pada setiap kategori latihan di aplikasi media pembelajaran. Data soal disimpan di dalam subkoleksi soal yang berada pada koleksi latihan_soal, sehingga setiap latihan dapat memiliki kumpulan soal yang berbeda sesuai materi yang dipelajari.

Struktur data pada koleksi soal terdiri dari atribut soal, A, B, C, D, jawaban, dan created. Atribut soal digunakan untuk menyimpan pertanyaan, sedangkan atribut A, B, C, dan D digunakan untuk menyimpan pilihan jawaban. Atribut jawaban digunakan untuk menyimpan kunci jawaban yang benar, dan atribut *created* digunakan untuk mencatat waktu penambahan soal ke dalam sistem. Dengan struktur tersebut, sistem dapat mengelola data soal latihan secara teratur dan memudahkan proses pelaksanaan latihan soal pada aplikasi.

5.2.6 Implementasi Tabel Hasil Latihan

🏠 > hasil_latihan > TSX0jKLX3wio...		
(default) + Start collection hasil_latihan > latihan_soal materi_hardware materi_software users	hasil_latihan + Add document 0Tj0cZPQ6MLwMbd4kgHo G69C70nB0HsGp2r7DRLj TSX0jKLX3wioLNHdffa >	TSX0jKLX3wioLNHdffa + Start collection + Add field created: May 8, 2026 at 1:42:57 PM UTC+7 docId: "y3AGEHxVIIUgdJruk1AZ" judul: "Latihan Hardware 1" nilai: 8 role: "siswa" total_soal: 10 uid: "Yj3AI0JacuhWE5z8DZ5j" username: "Aisyah"

Sumber : data diolah oleh penulis (2026)

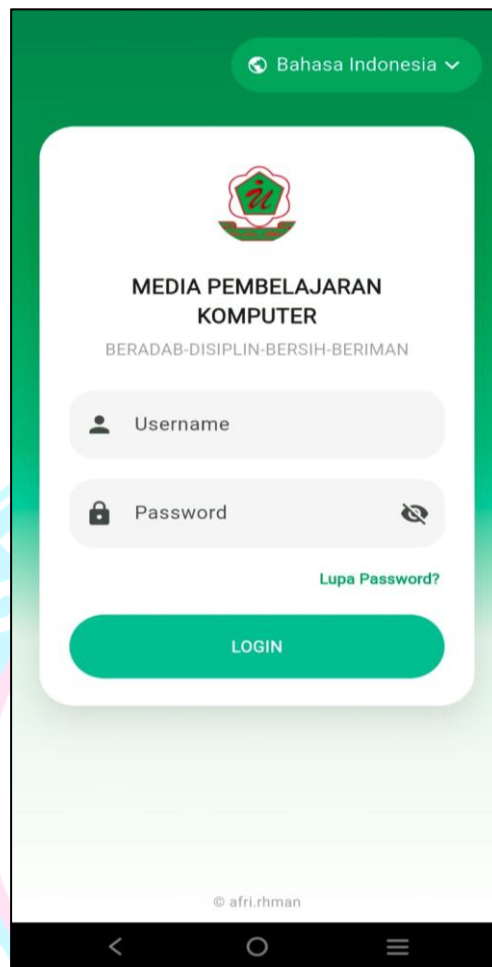
Gambar 5. 6 Tabel Hasil Latihan

Tabel hasil_latihan pada basis data *Firebase Firestore* digunakan untuk menyimpan data hasil latihan soal yang telah dikerjakan oleh pengguna pada aplikasi media pembelajaran. Setiap data hasil latihan disimpan dalam bentuk dokumen dengan Document ID unik yang dibuat secara otomatis oleh *Firebase* sehingga memudahkan proses pengelolaan dan pencarian data hasil latihan di dalam sistem.

Struktur data pada koleksi hasil_latihan terdiri dari atribut *Username*, *uid*, *judul*, *docId*, *nilai*, *total_soal*, *role*, dan *created*. Atribut *Username* digunakan untuk menyimpan nama pengguna, sedangkan *uid* digunakan sebagai identitas pengguna. Atribut *judul* dan *docId* digunakan untuk menyimpan informasi latihan soal yang dikerjakan. Selain itu, atribut *nilai* digunakan untuk menyimpan skor hasil latihan, *total_soal* untuk jumlah soal, *role* untuk menyimpan peran pengguna, dan *created* untuk mencatat waktu penyimpanan data hasil latihan. Dengan struktur tersebut, sistem dapat mengelola dan menampilkan hasil latihan pengguna secara teratur pada aplikasi.

5.3 Tampilan Antarmuka

5.3.1 Tampilan Halaman Login



Sumber : data diolah oleh penulis (2026)

Gambar 5. 7 Halaman *Login*

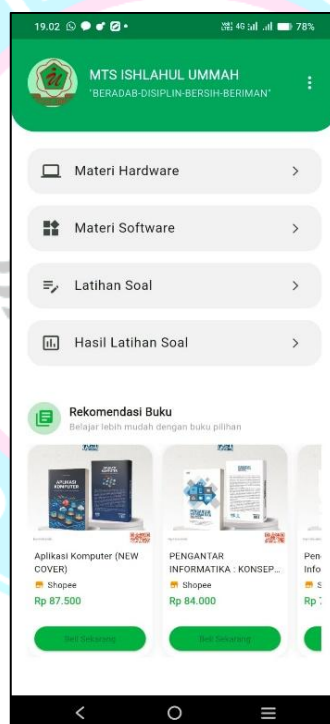
Halaman pertama yang ditampilkan kepada pengguna saat membuka aplikasi media pembelajaran komputer adalah halaman *login*. Tampilan antarmuka pada halaman ini dirancang sederhana dan informatif untuk memudahkan pengguna dalam melakukan proses *autentikasi*. Pada bagian atas halaman terdapat fitur pemilihan bahasa, seperti Bahasa Indonesia dan Bahasa Arab, sehingga pengguna dapat menyesuaikan bahasa tampilan aplikasi sesuai kebutuhan.

Pada bagian tengah halaman ditampilkan logo sekolah sebagai identitas aplikasi, diikuti dengan nama aplikasi yaitu Media Pembelajaran Komputer serta slogan sekolah yang mencerminkan nilai disiplin, kebersihan, dan keimanan.

Selanjutnya terdapat formulir *login* yang terdiri dari *field Username* dan *password*. *Field* tersebut digunakan sebagai proses autentikasi agar hanya pengguna yang memiliki hak akses yang dapat masuk ke dalam aplikasi. Pada *field password* juga tersedia ikon untuk menampilkan atau menyembunyikan kata sandi.

Di bagian bawah formulir terdapat tombol *Login* atau *Masuk* yang digunakan untuk memproses data pengguna. Apabila data yang dimasukkan benar, maka pengguna akan diarahkan menuju halaman utama aplikasi sesuai hak akses yang dimiliki. Tampilan halaman *login* menggunakan kombinasi warna hijau dan putih yang memberikan kesan bersih, sederhana, dan nyaman digunakan.

5.3.2 Tampilan Halaman Menu Utama



Sumber : data diolah oleh penulis (2026)

Gambar 5. 8 Halaman Menu Utama

Setelah pengguna berhasil *login*, sistem akan menampilkan halaman utama aplikasi media pembelajaran komputer. Pada bagian atas halaman terdapat logo sekolah, nama sekolah, serta ikon menu sebagai identitas dan navigasi aplikasi.

Pada bagian tengah halaman tersedia beberapa menu utama, yaitu Materi *Hardware*, Materi *Software*, Latihan Soal, dan Hasil Latihan Soal. Menu tersebut digunakan untuk mengakses materi pembelajaran dan melihat hasil latihan pengguna.

Di bagian bawah halaman terdapat fitur Rekomendasi Buku yang menampilkan buku pembelajaran komputer beserta gambar dan tombol pembelian. Tampilan halaman menggunakan kombinasi warna hijau dan putih sehingga terlihat sederhana dan nyaman digunakan.

5.3.3 Tampilan Halaman Materi Guru



Sumber : data diolah oleh penulis (2026)

Gambar 5. 9 Halaman Materi

Setelah guru memilih menu Materi, sistem akan menampilkan halaman materi *hardware* khusus guru. Halaman ini dirancang sederhana dan mudah digunakan agar guru dapat mengelola materi pembelajaran dengan lebih mudah.

Pada bagian atas halaman terdapat tombol kembali dan judul sebagai identitas halaman. Di bagian tengah ditampilkan gambar materi, judul materi, dan

isi materi pembelajaran *hardware* dalam bentuk kartu agar terlihat rapi dan mudah dibaca.

Pada bagian bawah halaman tersedia tombol tambah, edit, dan hapus yang digunakan guru untuk mengelola materi pembelajaran. Tampilan halaman menggunakan kombinasi warna hijau dan putih sehingga memberikan kesan sederhana, nyaman, dan mudah dipahami.

5.3.4 Tampilan Menambahkan Materi



Sumber : data diolah oleh penulis (2026)

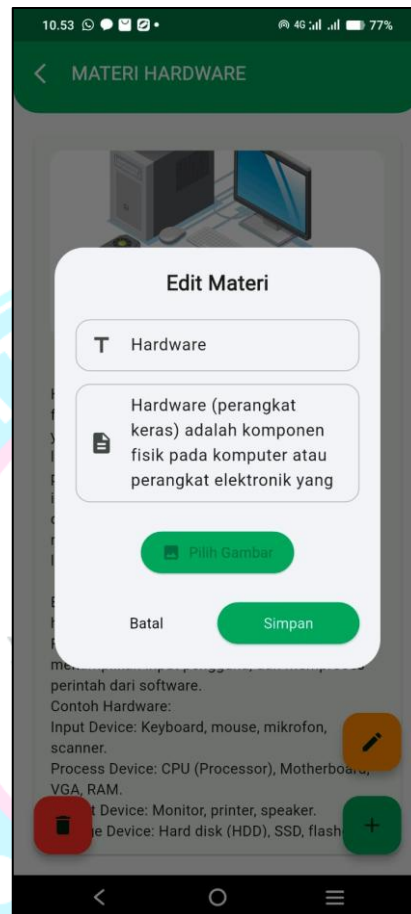
Gambar 5. 10 Menambahkan Materi

Setelah guru menekan tombol tambah materi, sistem akan menampilkan halaman tambah materi *hardware* atau *software*. Halaman ini digunakan untuk menambahkan data materi pembelajaran baru ke dalam aplikasi.

Pada halaman tersebut tersedia beberapa form *input*, yaitu form judul materi, form isi materi, dan form gambar untuk mengunggah gambar materi pembelajaran. Selain itu, terdapat tombol batal untuk membatalkan proses

penambahan materi dan tombol simpan untuk menyimpan data materi ke dalam sistem. Tampilan halaman dibuat sederhana dan mudah digunakan agar guru dapat menambahkan materi pembelajaran dengan lebih nyaman dan efisien.

5.3.5 Tampilan Edit Materi

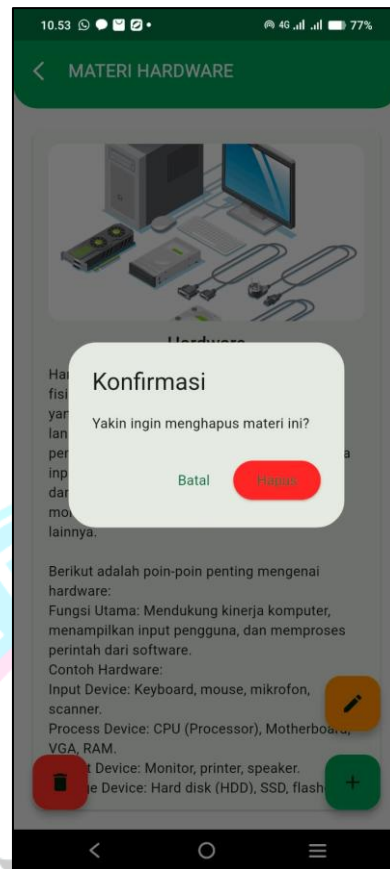


Sumber : data diolah oleh penulis (2026)

Gambar 5. 11 Edit Materi

Tombol Edit digunakan untuk mengubah atau memperbarui data yang telah tersimpan pada sistem. Ketika tombol Edit dipilih, pengguna akan diarahkan ke halaman perubahan data sehingga informasi yang sebelumnya tersimpan dapat diperbaiki atau diperbarui sesuai kebutuhan. Fitur ini memudahkan pengguna dalam melakukan pengelolaan data tanpa perlu menambahkan data baru.

5.3.6 Tampilan Hapus Materi



Sumber : data diolah oleh penulis (2026)

Gambar 5. 12 Hapus Materi

Saat guru menekan tombol hapus materi, sistem akan menampilkan pesan konfirmasi untuk memastikan tindakan penghapusan data. Pesan konfirmasi tersebut bertujuan agar guru tidak salah menghapus materi yang masih diperlukan.

Pada pesan konfirmasi tersedia tombol batal untuk membatalkan proses penghapusan dan tombol hapus untuk melanjutkan penghapusan materi dari sistem. Tampilan konfirmasi dibuat sederhana dan mudah dipahami sehingga membantu guru dalam mengelola data materi pembelajaran dengan lebih aman.

5.3.7 Tampilan Halaman Materi Siswa



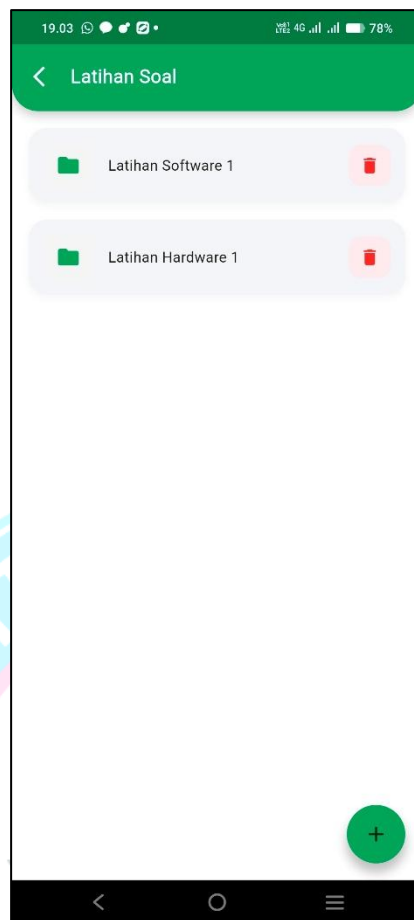
Sumber : data diolah oleh penulis (2026)

Gambar 5. 13 Halaman Materi Siswa

Setelah siswa memilih menu Materi *Hardware* atau *Software*, sistem akan menampilkan halaman materi pembelajaran. Halaman ini digunakan untuk membantu siswa membaca dan memahami materi yang telah ditambahkan oleh guru.

Pada halaman tersebut ditampilkan gambar materi, judul materi, dan isi materi pembelajaran dalam bentuk tampilan yang rapi dan mudah dibaca. Siswa hanya dapat melihat dan mempelajari materi beserta gambar yang tersedia tanpa dapat menambah, mengubah, maupun menghapus data materi.

5.3.8 Tampilan Halaman File Latihan Soal

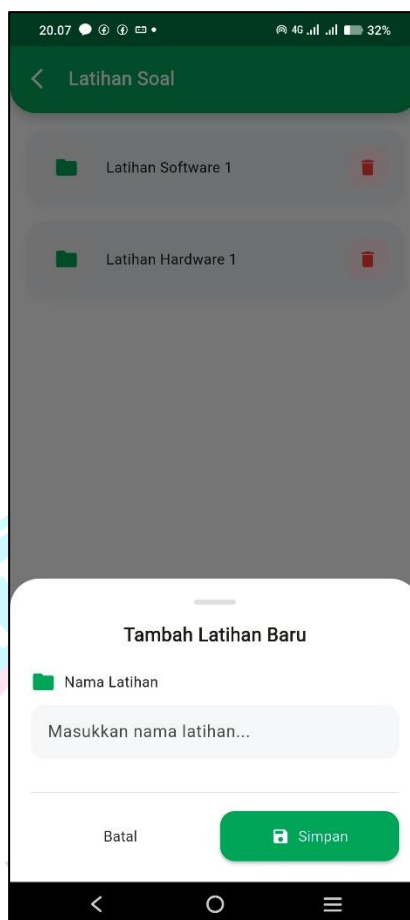


Sumber : data diolah oleh penulis (2026)

Gambar 5. 14 File Latihan Soal

Setelah guru memilih menu Latihan Soal, sistem akan menampilkan halaman daftar latihan soal yang digunakan untuk mengelola file latihan soal pembelajaran. Halaman ini dirancang sederhana dan mudah digunakan agar guru dapat mengatur latihan soal dengan lebih efisien.

5.3.9 Tampilan Menambahkan File Latihan Soal



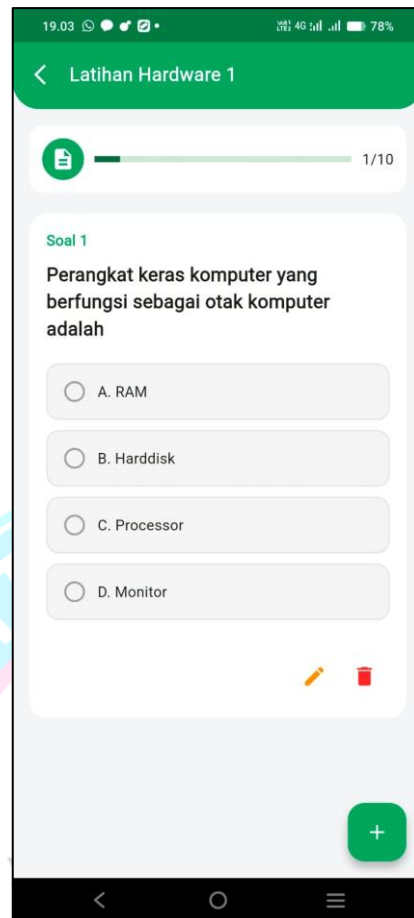
Sumber : data diolah oleh penulis (2026)

Gambar 5. 15 Menambahkan File Latihan Soal

Guru dapat menambahkan file latihan soal baru melalui tombol tambah yang terdapat pada bagian bawah halaman. Selain itu, guru juga dapat menghapus file latihan soal menggunakan tombol hapus yang tersedia pada setiap data latihan soal.

Sedangkan pada tampilan siswa, halaman latihan soal hanya digunakan untuk memilih file latihan soal yang ingin dikerjakan. Siswa tidak memiliki akses untuk menambahkan maupun menghapus file latihan soal sehingga hanya dapat mengerjakan soal yang telah disediakan oleh guru.

5.3.10 Tampilan Halaman Soal Guru



Sumber : data diolah oleh penulis (2026)

Gambar 5. 16 Halaman Latihan Soal Guru

Setelah guru membuat dan memilih file latihan soal, sistem akan menampilkan halaman soal latihan. Pada awalnya halaman ini masih kosong karena belum terdapat soal yang ditambahkan ke dalam file latihan tersebut. Halaman ini digunakan guru untuk mengelola soal-soal latihan *hardware* atau *software* dengan tampilan yang sederhana dan mudah digunakan.

Guru dapat menambahkan soal baru melalui tombol tambah yang terdapat pada bagian bawah halaman. Setelah soal ditambahkan, sistem akan menampilkan pertanyaan, pilihan jawaban, serta indikator jumlah soal. Selain itu, guru juga dapat mengedit dan menghapus soal yang telah dibuat sesuai kebutuhan pembelajaran.

5.3.11 Tampilan Menambahkan Soal

Sumber : data diolah oleh penulis (2026)

Gambar 5. 17 Menambahkan Soal

Saat guru menekan tombol tambah soal, sistem akan menampilkan halaman tambah soal yang digunakan untuk membuat soal latihan baru. Halaman ini dirancang sederhana dan mudah digunakan agar guru dapat menambahkan soal dengan lebih cepat dan efisien.

Pada halaman tersebut tersedia form pertanyaan, pilihan jawaban A, B, C, dan D, serta pilihan jawaban benar untuk menentukan kunci jawaban soal. Selain itu, terdapat tombol batal untuk membatalkan proses penambahan soal dan tombol simpan untuk menyimpan soal ke dalam file latihan yang dipilih.

5.3.12 Tampilan Edit Latihan Soal

The screenshot displays the 'Edit Soal' interface. At the top, there's a title bar with a back arrow and 'Latihan Hardware 1'. Below it, a green header contains an edit icon and the text 'Edit Soal' with a subtitle 'Perbarui data soal'. The main content area includes a 'Pertanyaan' (Question) section with the text: 'Perangkat keras komputer yang berfungsi sebagai otak komputer adalah'. Below this is a 'Pilihan Jawaban' (Multiple Choice) section with four options: A. RAM, B. Harddisk, C. Processor, and D. Monitor. A 'Jawaban Benar' (Correct Answer) section shows a dropdown menu currently set to 'Pilihan C'. At the bottom, there are two buttons: 'Batal' (Cancel) in red and 'Simpan' (Save) in green.

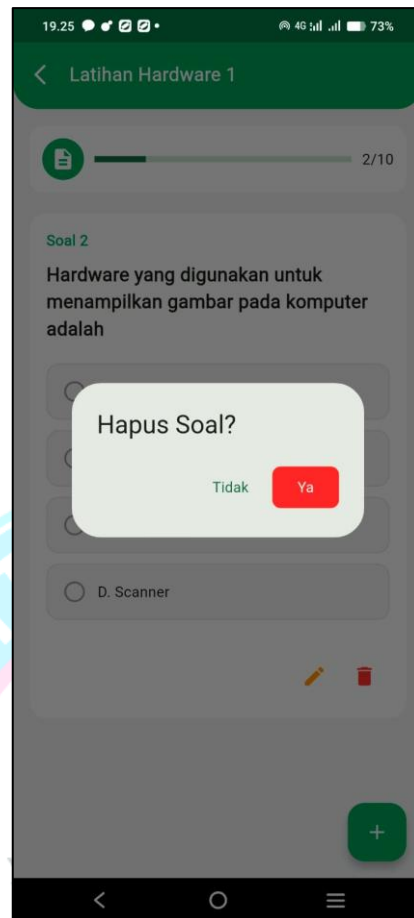
Sumber : data diolah oleh penulis (2026)

Gambar 5. 18 *Edit* Latihan Soal

Saat guru menekan tombol edit soal, sistem akan menampilkan halaman edit soal yang digunakan untuk mengubah data soal latihan yang sudah tersedia. Halaman ini memudahkan guru dalam memperbaiki atau memperbarui isi soal sesuai kebutuhan pembelajaran.

Pada halaman tersebut ditampilkan form pertanyaan, pilihan jawaban A, B, C, dan D, serta pilihan jawaban benar yang sudah terisi sesuai data sebelumnya. Guru dapat mengubah isi soal maupun kunci jawaban, kemudian menyimpan perubahan menggunakan tombol simpan atau membatalkannya melalui tombol batal.

5.3.13 Tampilan Hapus Latihan Soal



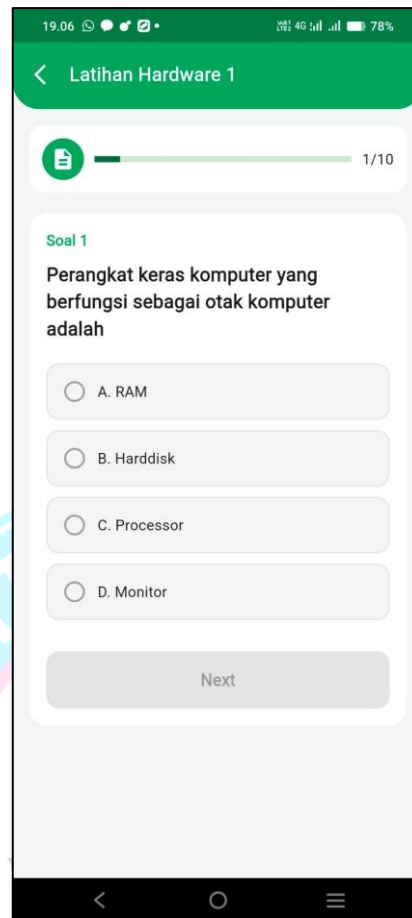
Sumber : data diolah oleh penulis (2026)

Gambar 5. 19 Hapus Latihan Soal

Saat guru menekan tombol hapus soal, sistem akan menampilkan pesan konfirmasi penghapusan soal. Halaman ini bertujuan untuk memastikan bahwa soal yang dipilih benar-benar ingin dihapus sehingga mengurangi risiko kesalahan saat mengelola data latihan soal.

Pada pesan konfirmasi tersedia tombol batal untuk membatalkan proses penghapusan dan tombol hapus untuk menghapus soal dari file latihan secara permanen. Tampilan dibuat sederhana dan mudah dipahami agar guru dapat menggunakan fitur dengan lebih aman dan nyaman.

5.3.14 Tampilan Halaman Latihan Soal Siswa



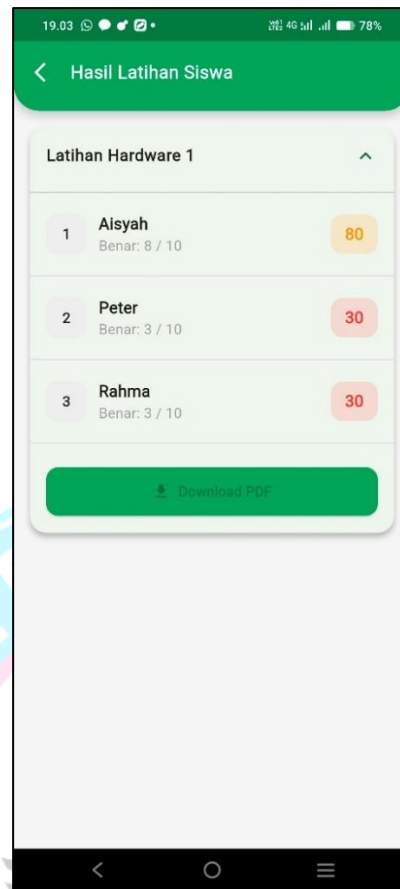
Sumber : data diolah oleh penulis (2026)

Gambar 5. 20 Halaman Soal Siswa

Setelah siswa memilih file latihan soal, sistem akan menampilkan halaman soal latihan yang digunakan untuk mengerjakan soal secara bertahap. Pada halaman tersebut ditampilkan pertanyaan, pilihan jawaban, serta indikator nomor soal agar siswa dapat mengetahui progres pengerjaan soal.

Siswa hanya dapat melanjutkan ke soal berikutnya menggunakan tombol *next* yang tersedia pada halaman. Sistem tidak menyediakan tombol untuk kembali ke soal sebelumnya sehingga jawaban yang telah dipilih tidak dapat diubah kembali. Tampilan halaman dibuat sederhana dan mudah dipahami agar siswa dapat lebih fokus dalam mengerjakan latihan soal.

5.3.15 Tampilan Hasil Latihan Soal Guru



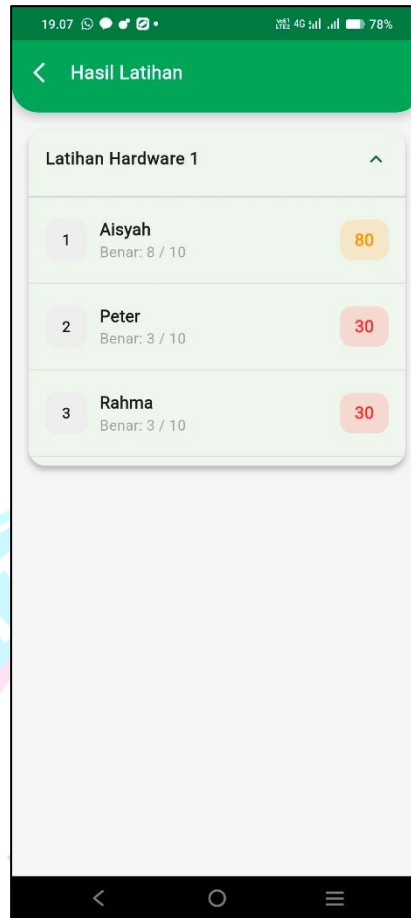
Sumber : data diolah oleh penulis (2026)

Gambar 5. 21 Halaman Hasil Latihan Guru

Setelah guru memilih menu Hasil Latihan, sistem akan menampilkan halaman hasil latihan siswa yang berisi data nilai dari latihan soal yang telah dikerjakan. Halaman ini digunakan guru untuk melihat dan memantau hasil belajar siswa dengan lebih mudah.

Pada halaman tersebut ditampilkan daftar hasil latihan siswa yang telah diurutkan berdasarkan nilai yang diperoleh. Selain melihat hasil nilai, guru juga dapat mengunduh data hasil latihan siswa untuk digunakan sebagai dokumentasi atau evaluasi pembelajaran. Tampilan halaman dibuat sederhana dan rapi sehingga memudahkan guru dalam mengelola data hasil latihan.

5.3.16 Tampilan Hasil Latihan Soal Siswa



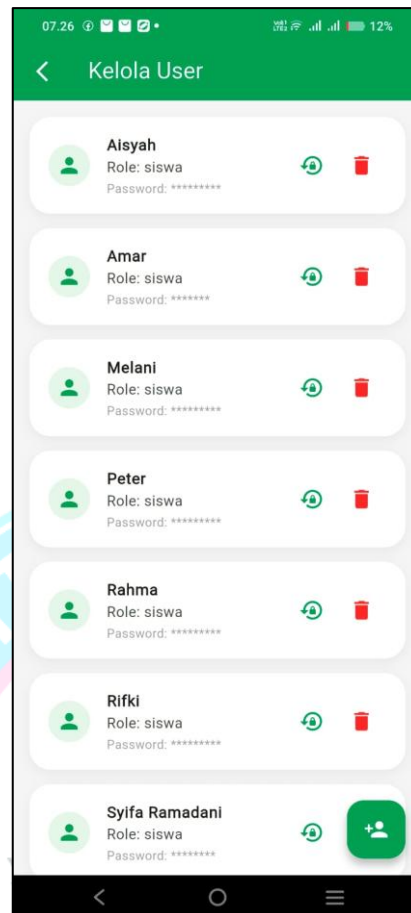
Sumber : data diolah oleh penulis (2026)

Gambar 5. 22 Halaman Hasil Latihan Siswa

Setelah siswa memilih menu Hasil Latihan, sistem akan menampilkan halaman hasil latihan yang berisi nilai dari latihan soal yang telah dikerjakan. Halaman ini digunakan siswa untuk melihat hasil pengerjaan latihan soal dengan lebih mudah.

Pada halaman tersebut, hasil latihan dikelompokkan berdasarkan file soal yang telah dikerjakan, seperti latihan *hardware* atau *software*. Selain itu, sistem juga menampilkan peringkat siswa berdasarkan nilai yang diperoleh sehingga siswa dapat mengetahui posisi hasil belajarnya dibandingkan pengguna lain. Tampilan halaman dibuat sederhana dan mudah dipahami agar siswa dapat melihat hasil latihan dengan nyaman.

5.3.17 Tampilan Halaman Kelola User



Sumber : data diolah oleh penulis (2026)

Gambar 5. 23 Halaman Kelola User

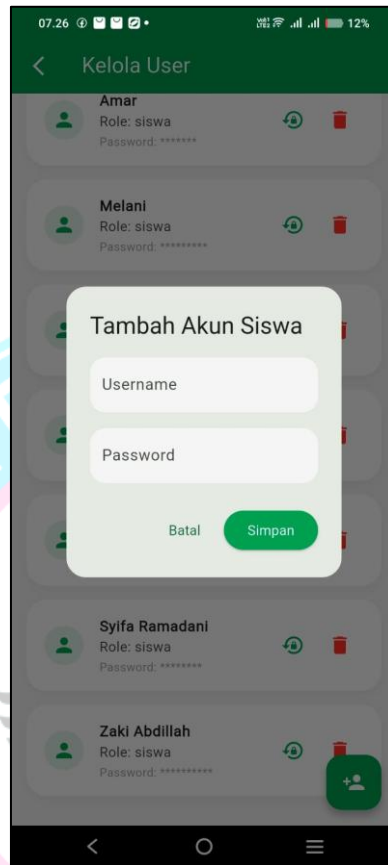
Halaman Kelola User digunakan untuk menampilkan dan mengelola data pengguna yang terdaftar pada aplikasi media pembelajaran. Pada halaman ini ditampilkan daftar pengguna beserta informasi nama pengguna, *role* pengguna, dan kata sandi yang disamarkan. Data pengguna ditampilkan dalam bentuk kartu agar terlihat rapi dan mudah dibaca oleh admin.

Setiap data pengguna dilengkapi dengan tombol ubah kata sandi dan tombol hapus untuk memudahkan proses pengelolaan akun. Tombol ubah kata sandi digunakan untuk mengganti *password* pengguna, sedangkan tombol hapus digunakan untuk menghapus data pengguna dari sistem.

Pada bagian kanan bawah halaman terdapat tombol tambah pengguna yang digunakan untuk menambahkan akun baru ke dalam aplikasi. Tampilan halaman

menggunakan desain sederhana dengan kombinasi warna hijau dan putih sehingga mudah digunakan dalam proses pengelolaan data pengguna.

5.3.18 Tampilan Halaman Tambah Akun



Sumber : data diolah oleh penulis (2026)

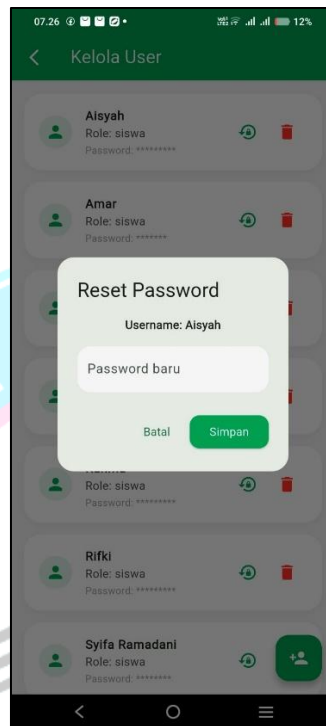
Gambar 5. 24 Halaman Tambah Akun Siswa

Halaman Tambah Akun Siswa digunakan untuk menambahkan data pengguna baru ke dalam aplikasi media pembelajaran. Halaman ini ditampilkan dalam bentuk *popup dialog* yang muncul setelah admin menekan tombol tambah pengguna pada halaman *Kelola User*.

Pada halaman ini tersedia dua kolom input, yaitu *Username* dan *Password*. Kolom *Username* digunakan untuk memasukkan nama akun pengguna, sedangkan kolom *Password* digunakan untuk memasukkan kata sandi akun yang akan dibuat. Di bagian bawah formulir terdapat tombol *Batal* dan *Simpan*. Tombol *Batal* digunakan untuk menutup formulir tanpa menyimpan data, sedangkan tombol *Simpan* digunakan untuk menyimpan data akun baru ke dalam sistem.

Tampilan halaman dibuat sederhana dengan kombinasi warna hijau dan putih sehingga mudah dipahami dan digunakan oleh admin saat melakukan pengelolaan data pengguna.

5.3.19 Tampilan Halaman *Reset Password Akun*



Sumber : data diolah oleh penulis (2026)

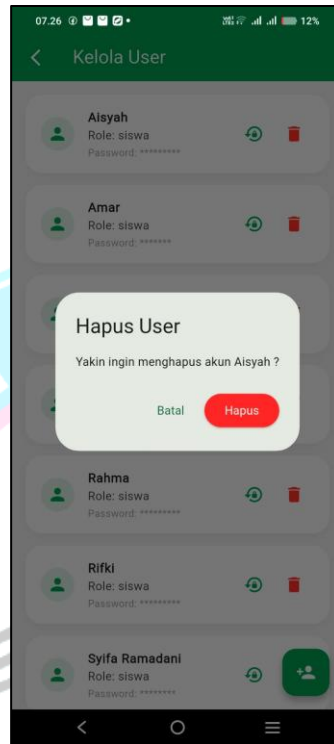
Gambar 5. 25 *Reset Password*

Halaman *Reset Password* digunakan untuk mengubah kata sandi pengguna pada aplikasi media pembelajaran. Halaman ini ditampilkan dalam bentuk *popup dialog* setelah admin menekan tombol *reset password* pada data pengguna yang dipilih.

Pada halaman ini ditampilkan informasi *Username* pengguna yang akan diubah kata sandinya serta kolom input *Password baru* untuk memasukkan *password* pengganti. Di bagian bawah formulir terdapat tombol *Batal* dan *Simpan*. Tombol *Batal* digunakan untuk menutup formulir tanpa melakukan perubahan data, sedangkan tombol *Simpan* digunakan untuk menyimpan *password* baru ke dalam sistem.

Tampilan halaman dibuat sederhana dengan kombinasi warna hijau dan putih sehingga memudahkan admin dalam melakukan proses pengelolaan akun pengguna.

5.3.20 Tampilan Halaman Hapus Akun



Sumber : data diolah oleh penulis (2026)

Gambar 5. 26 Halaman Hapus Akun

Halaman *Hapus User* digunakan untuk melakukan konfirmasi penghapusan data pengguna pada aplikasi media pembelajaran. Halaman ini ditampilkan dalam bentuk *popup dialog* setelah admin menekan tombol hapus pada data pengguna yang dipilih.

Pada halaman ini sistem menampilkan pesan konfirmasi untuk memastikan apakah admin benar-benar ingin menghapus akun pengguna dari sistem. Di bagian bawah dialog terdapat tombol *Batal* dan *Hapus*. Tombol *Batal* digunakan untuk membatalkan proses penghapusan akun, sedangkan tombol *Hapus* digunakan untuk menghapus data pengguna secara permanen dari sistem.

Tampilan halaman dibuat sederhana dengan kombinasi warna hijau, putih, dan merah sehingga memudahkan admin dalam melakukan proses pengelolaan data pengguna.

5.4 Hasil Pengujian Sistem

Pengujian sistem pada penelitian ini dilakukan menggunakan metode whitebox testing untuk menganalisis alur logika program pada aplikasi media pembelajaran berbasis *Android* pada mata pelajaran komputer. Pengujian dilakukan dengan menganalisis *source code* pada beberapa proses utama sistem, seperti proses *login* pengguna, pengelolaan materi pembelajaran, latihan soal, penilaian hasil latihan, pengaturan bahasa, hingga proses *Logout* aplikasi. 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. Pengujian pada penelitian ini menggunakan *tools* bawaan Flutter yaitu flutter test untuk membantu pengujian fungsi program berdasarkan *source code* aplikasi.

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, seperti validasi *login*, navigasi menu, pemilihan materi, pengerjaan latihan soal, serta penyimpanan hasil latihan pengguna.

Tahap akhir pengujian dilakukan menggunakan *Statement Coverage* untuk mengevaluasi apakah seluruh *Statement* pada *source code* telah berhasil dieksekusi selama proses pengujian berlangsung sehingga aplikasi dapat berjalan dengan baik dan sesuai kebutuhan pengguna.

5.4.1 Pengujian Fungsi *Login*

a. Potongan *Source Code* Fungsi *Login*

Fungsi *login* pada aplikasi media pembelajaran berbasis *Android* digunakan untuk melakukan proses autentikasi pengguna sebelum masuk ke dalam sistem. Proses yang dilakukan meliputi validasi input *Username* dan *password*, validasi panjang *password*, pengecekan kombinasi huruf dan angka pada *password*, pencarian data pengguna pada *Firebase Firestore*, penyimpanan session pengguna, hingga navigasi menuju halaman utama aplikasi apabila *login* berhasil dilakukan. Selain itu, fungsi ini juga menangani kondisi *error* apabila *Username* atau *password* tidak sesuai. Berikut adalah potongan *source code* fungsi login yang digunakan dalam pengujian:

```
void login() async {
  setState(() => _error = False);

  final password = _password.text.trim();

  if (_Username.text.isEmpty || password.isEmpty) {
    setState(() => _error = True);
    return;
  }

  if (password.length > 10) {
    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(
        content: Text(
          'Password maksimal 10 karakter',
        ),
      ),
    );
    return;
  }
```

```

final regex = RegExp(
  r'^(?=.*[A-Za-z])(?=.*\d)[A-Za-z\d]+$'
);

if (!regex.hasMatch(password)) {
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(
      content: Text(
        'Password wajib mengandung huruf dan angka',
      ),
    ),
  );
  return;
}

setState(() => _loading = True);

try {
  final result = await FirebaseFirestore.instance
    .collection('Users')
    .where(
      'Username',
      isEqualTo: _Username.text,
    )
    .where(
      'password',
      isEqualTo: _password.text,
    )
    .get();

  setState(() => _loading = False);
}

```

```

if (result.docs.isEmpty) {

    final doc = result.docs.first;
    final User = doc.data();
    final role = User['role'];

    Session.uid = doc.id.toString();
    Session.Username =
        User['Username'].toString();
    Session.role =
        User['role'].toString();

    Navigator.pushReplacement(
        context,
        MaterialPageRoute(
            builder: (_) => HomePage(
                isGuru: role == 'guru',
                language: _lang,
            ),
        ),
    );

} else {
    setState(() => _error = True);
}

} catch (e) {

    setState() {
        _loading = False;
        _error = True;
    });
}

```

```

    }
}

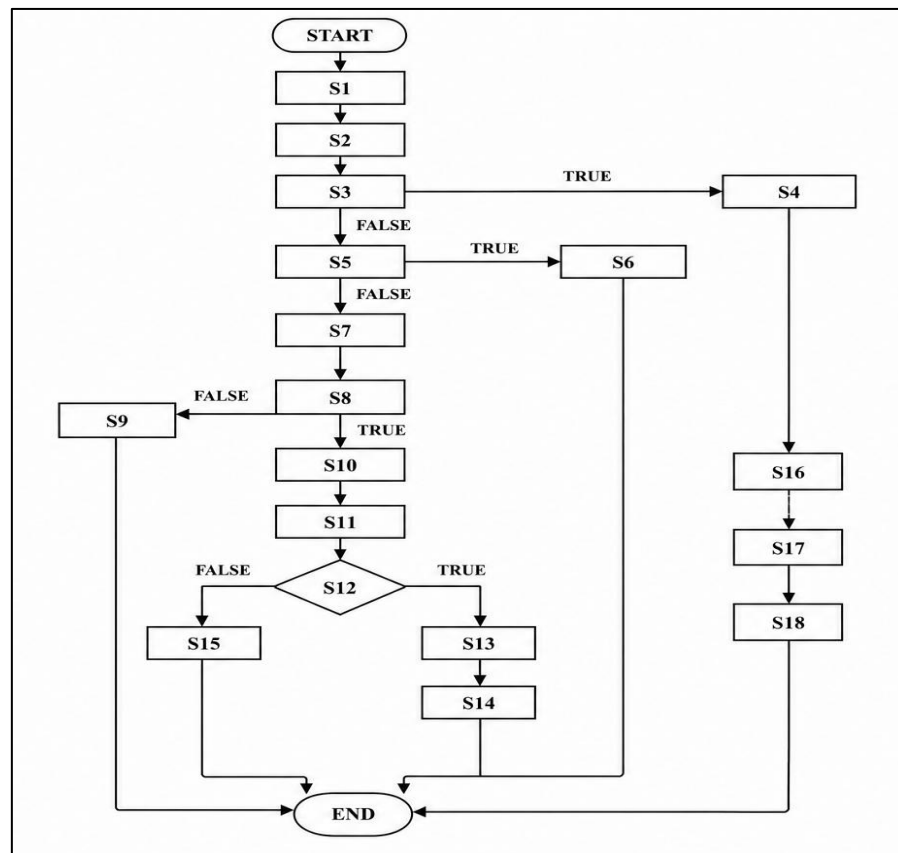
```

b. Identifikasi *Statement* Fungsi Login

Tabel 5. 1 Identifikasi *Statement* Fungsi Login

<i>Statement</i>	Source Code	Keterangan
S1	<code>setState() => _error = False)</code>	Reset status <i>error</i>
S2	<code>password = _password.text.trim()</code>	Mengambil input <i>password</i>
S3	<code>`if (_Username.text.isEmpty</code>	
S4	<code>setState() => _error = True)</code>	Menampilkan <i>error</i> input kosong
S5	<code>if (password.length > 10)</code>	Validasi panjang <i>password</i>
S6	<code>SnackBar('Password maksimal 10 karakter')</code>	Menampilkan pesan <i>password</i> terlalu panjang
S7	<code>regex = Regexp(...)</code>	Membuat validasi <i>regex</i> <i>password</i>
S8	<code>if (!regex.hasMatch(password))</code>	Validasi kombinasi huruf dan angka
S9	<code>SnackBar('Password wajib mengandung huruf dan angka')</code>	Menampilkan pesan validasi <i>password</i>
S10	<code>setState() => _loading = True)</code>	Mengaktifkan loading
S11	<code>FirebaseFirestore.instance.collection()</code>	Mengambil data <i>User Firebase</i>
S12	<code>if (result.docs.isNotEmpty)</code>	Memeriksa <i>User</i> ditemukan
S13	<code>Session.uid = doc.id.toString()</code>	Menyimpan session <i>User</i>
S14	<code>Navigator.pushReplacement(...)</code>	Navigasi ke halaman home
S15	<code>setState() => _error = True)</code>	Menampilkan login gagal
S16	<code>catch (e)</code>	Menangkap <i>error</i> system
S17	<code>_loading = False</code>	Menghentikan loading
S18	<code>_error = True</code>	Menampilkan <i>error</i> sistem

Sumber : data diolah oleh penulis (2026)

c. *Flowchart Fungsi Login*

Sumber : data diolah oleh penulis (2026)

Gambar 5. 27 *Flowchart Fungsi Login*

Berdasarkan diagram alur pada gambar *Flowchart Fungsi Login* di atas, urutan logika kerja fungsi *login* dapat dijabarkan sebagai berikut:

1. S1 dan S2 (Inisialisasi dan Input *Password*)

Sistem memulai proses *login* dengan melakukan reset status *error* menggunakan `setState() => _error = False` kemudian mengambil input *password* dari pengguna menggunakan `_password.text.trim()`.

2. S3 dan S4 (Validasi Input Kosong)

Sistem melakukan pengecekan apakah *Username* atau *password* masih kosong. Jika kondisi bernilai *True*, maka sistem menjalankan S4 untuk menampilkan status *error login* dan proses berakhir.

3. S5 dan S6 (Validasi Panjang *Password*)

Jika input tidak kosong, sistem melanjutkan pengecekan panjang *password* pada S5. Apabila *password* lebih dari 10 karakter (*True*), maka

sistem menjalankan S6 untuk menampilkan pesan peringatan bahwa *password* maksimal 10 karakter, kemudian proses dihentikan.

4. S7 dan S8 (Validasi Format *Password*)

Jika *password* memenuhi batas panjang karakter, sistem membuat pola validasi *regex* pada S7 dan melakukan pengecekan kecocokan format *password* pada S8. Sistem memastikan *password* mengandung huruf dan angka.

5. S9 (*Password* Tidak Valid)

Jika hasil pengecekan pada S8 bernilai *False*, maka sistem menjalankan S9 untuk menampilkan pesan bahwa *password* wajib mengandung huruf dan angka, kemudian proses berakhir.

6. S10 dan S11 (Proses Login *Firebase*)

Jika *password* valid (*True*), sistem mengaktifkan status loading pada S10 kemudian menjalankan proses *query Firebase Firestore* pada S11 untuk mencocokkan *Username* dan *password* pengguna dengan data pada *database*.

7. S12 (Pengecekan Data *User*)

Setelah *query* selesai dilakukan, sistem memeriksa apakah data pengguna ditemukan di *database*. Pada simbol keputusan S12, sistem menentukan keberhasilan login berdasarkan hasil *query Firebase*.

8. S15 (*Login* Gagal)

Jika hasil pengecekan pada S12 bernilai *False*, maka sistem menjalankan S15 untuk menampilkan status *login* gagal karena *Username* atau *password* tidak sesuai.

9. S13 dan S14 (*Login* Berhasil)

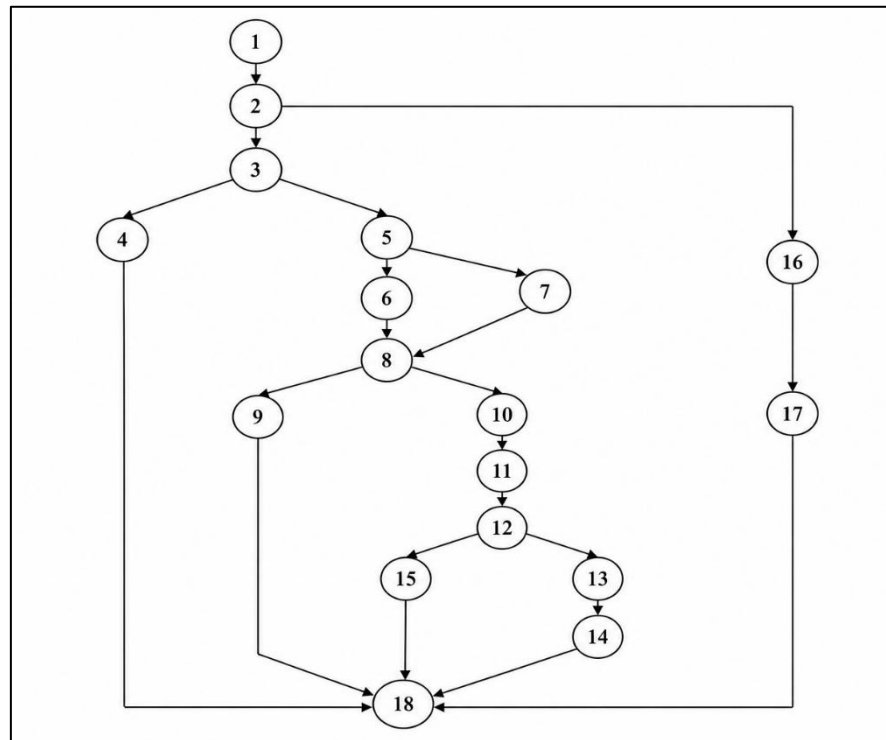
Jika data pengguna ditemukan (*True*), sistem menjalankan S13 untuk menyimpan *session* pengguna seperti *uid*, *Username*, dan *role*. Selanjutnya pada S14 sistem mengarahkan pengguna menuju halaman *HomePage* sebagai tanda login berhasil.

10. S16, S17, dan S18 (Penanganan *Error* Sistem)

Jika terjadi kesalahan sistem atau gangguan saat proses *login* berlangsung, maka sistem menjalankan S16 untuk menangkap *error* menggunakan

catch(e), kemudian pada S17 status *loading* dinonaktifkan dan pada S18 sistem menampilkan status *error login* kepada pengguna.

d. *Flowgraph Fungsi Login*



Sumber : data diolah oleh penulis (2026)

Gambar 5. 28 *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 18 node yang saling terhubung sesuai dengan alur logika pada *source code* fungsi login. Setiap node menggambarkan proses atau percabangan yang terjadi selama proses autentikasi pengguna berlangsung, mulai dari validasi input hingga proses login berhasil ataupun gagal. Node terakhir yaitu Node 18 digunakan sebagai terminal node (end process) yang menandai berakhirnya seluruh alur eksekusi program.

Node 1 dan Node 2 merepresentasikan proses awal sistem, yaitu reset status *error* serta pengambilan input *password* dari pengguna. Selanjutnya Node 3 merupakan titik keputusan pertama yang digunakan untuk memeriksa apakah *Username* atau *password* masih kosong. Apabila kondisi bernilai benar, maka alur

program menuju Node 4 untuk menampilkan status *error* kemudian langsung berakhir pada Node 18.

Jika input tidak kosong, maka alur program dilanjutkan menuju Node 5 yang digunakan untuk melakukan validasi panjang *password*. Apabila *password* melebihi 10 karakter, sistem akan menuju Node 6 untuk menampilkan pesan peringatan bahwa *password* terlalu panjang, kemudian proses berakhir pada Node 18. Namun jika kondisi tersebut tidak terpenuhi, maka alur program dilanjutkan menuju Node 7 untuk membentuk pola validasi *regex password*.

Node 8 merupakan titik keputusan berikutnya yang digunakan untuk memeriksa apakah *password* telah memenuhi format kombinasi huruf dan angka. Jika *password* tidak valid, maka alur program menuju Node 9 untuk menampilkan pesan kesalahan validasi *password*, kemudian proses berakhir pada Node 18. Sebaliknya, jika *password* valid maka sistem melanjutkan proses login menuju Node 10 dan Node 11 yang digunakan untuk mengaktifkan status loading serta menjalankan query *Firebase Firestore* untuk mencari data pengguna berdasarkan *Username* dan *password* yang dimasukkan.

Selanjutnya pada Node 12 sistem melakukan pengecekan apakah data pengguna berhasil ditemukan di dalam database. Jika data tidak ditemukan, maka alur program menuju Node 15 untuk menampilkan status login gagal sebelum berakhir pada Node 18. Namun apabila data pengguna ditemukan, maka sistem akan melanjutkan proses menuju Node 13 untuk menyimpan session pengguna seperti uid, *Username*, dan role, kemudian menuju Node 14 untuk mengarahkan pengguna ke halaman HomePage sebagai tanda login berhasil.

Selain itu, *Flowgraph* juga memiliki jalur penanganan *error* yang ditunjukkan oleh Node 16 dan Node 17. Jalur ini digunakan ketika terjadi kesalahan sistem saat proses login berlangsung. Pada kondisi tersebut sistem akan menangkap *error* menggunakan blok catch, kemudian menghentikan status loading dan menampilkan status *error* sebelum akhirnya seluruh alur program berakhir pada Node 18 sebagai terminal node.

e. *Cyclomatic Complexity Fungsi Login*

Berdasarkan *Flowgraph* yang telah dibentuk, terdapat 4 predicate node (P), yaitu pada Node 3 yang melakukan pengecekan *Username* atau *password* kosong, Node 5 yang melakukan validasi panjang *password*, Node 8 yang melakukan pengecekan format *password* menggunakan *regex*, dan Node 12 yang melakukan pengecekan keberadaan data pengguna pada database *Firebase*. Oleh karena itu, nilai *Cyclomatic Complexity* dapat dihitung menggunakan rumus berikut:

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

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

$$V(G)=5$$

Dari perhitungan tersebut diperoleh nilai *Cyclomatic Complexity* sebesar 5. Nilai tersebut menunjukkan bahwa terdapat lima jalur independen dalam proses eksekusi fungsi login yang merepresentasikan seluruh kemungkinan alur logika berdasarkan kondisi yang terjadi pada program.

Nilai *Cyclomatic Complexity* tersebut 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 whitebox testing. Setiap jalur independen merepresentasikan kemungkinan alur eksekusi program yang berbeda berdasarkan kondisi percabangan pada proses validasi input, validasi *password*, serta pengecekan data pengguna pada database *Firebase*. Dengan demikian, setiap kemungkinan skenario pada fungsi login dapat dianalisis dan diuji secara menyeluruh untuk memastikan seluruh logika program berjalan

dengan benar. Sehingga didapatkan 5 jalur independen (*independent path*) berikut ini

1. Path 1:

$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 18$

(*Username* atau *password* kosong sehingga sistem menampilkan *error* input)

2. Path 2:

$1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 18$

(*Password* melebihi 10 karakter sehingga sistem menampilkan peringatan *password* terlalu panjang)

3. Path 3:

$1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 18$

(*Password* tidak memenuhi format kombinasi huruf dan angka)

4. Path 4:

$1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 8 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 15 \rightarrow 18$

(Data pengguna tidak ditemukan pada database sehingga login gagal)

5. Path 5:

$1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 8 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 13 \rightarrow 14 \rightarrow 18$

(Data pengguna ditemukan dan login berhasil sehingga pengguna diarahkan ke halaman HomePage)

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

Test Case	Username	Password	Jalur Eksekusi	Output
-----------	----------	----------	----------------	--------

TC1	Kosong	Kosong		1 → 2 → 3 → 4 → 18	Error input kosong
TC2	Username valid	Password lebih dari 10 karakter		1 → 2 → 3 → 5 → 6 → 18	Password terlalu panjang
TC3	Username valid	Password kombinasi huruf dan angka	tanpa huruf	1 → 2 → 3 → 5 → 7 → 8 → 9 → 18	Password tidak valid
TC4	Username tidak terdaftar	Password valid		1 → 2 → 3 → 5 → 7 → 8 → 10 → 11 → 12 → 15 → 18	Login gagal
TC5	Username valid	Password benar		1 → 2 → 3 → 5 → 7 → 8 → 10 → 11 → 12 → 13 → 14 → 18	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 pada masing-masing *Test Case* adalah sebagai berikut:

1. TC1:
S1, S2, S3, S4
2. TC2:
S1, S2, S3, S5, S6
3. TC3:
S1, S2, S3, S5, S7, S8, S9
4. TC4:
S1, S2, S3, S5, S7, S8, S10, S11, S12, S15
5. TC5:
S1, S2, S3, S5, S7, S8, S10, S11, S12, S13, S14

Berdasarkan hasil pengujian yang telah dilakukan, seluruh *Statement* pada fungsi login yaitu S1 hingga S18 berhasil dieksekusi melalui kombinasi *Test Case*

TC1, TC2, TC3, TC4, dan TC5. Dengan demikian diperoleh nilai *Statement Coverage* sebesar:

$$\frac{18}{18} \times 100\% = 100\%$$

Nilai tersebut menunjukkan bahwa seluruh *Statement* pada fungsi login telah diuji dan seluruh kemungkinan alur eksekusi program berhasil dicakup dalam proses pengujian whitebox.

5.4.2 Pengujian Fungsi *Logout*

a. Potongan *Source Code* Fungsi *Logout*

Fungsi *Logout* merupakan salah satu fungsi utama yang akan diuji dalam penelitian media pembelajaran berbasis *Android* ini untuk memastikan proses keluar aplikasi berjalan dengan baik sesuai alur program yang telah dirancang. Komponen di dalamnya mencakup proses penampilan dialog konfirmasi *Logout*, pengecekan pilihan pengguna, penutupan dialog, hingga penghapusan seluruh halaman sebelumnya dan pengalihan pengguna menuju halaman *login*. Jika pengguna memilih *Logout*, maka sistem akan mengarahkan kembali pengguna ke halaman *LoginPage* sebagai tanda bahwa sesi penggunaan aplikasi telah berakhir. Berikut adalah potongan *source code* fungsi *Logout* yang digunakan dalam tahap pengujian:

```
void _Logout(BuildContext context) {
  final t = text[widget.language]!;

  showDialog(
    context: context,
    builder: (_) => AlertDialog(
      title: Text(t['Logout']!),
      content: Text(t['Logout_confirm']!),
      actions: [

        TextButton(
          onPressed: () => Navigator.pop(context),
```

```

        child: Text(t['cancel']!),
      ),

      ElevatedButton(
        style: ElevatedButton.styleFrom(
          backgroundColor: Colors.red,
          foregroundColor: Colors.white,
        ),

        onPressed: () {

          Navigator.pop(context);

          Navigator.pushAndRemoveUntil(
            context,
            MaterialPageRoute(
              builder: (_) => const LoginPage(),
            ),
            (route) => False,
          );
        },

        child: Text(t['yes']!),
      ),
    ],
  ),
);
}

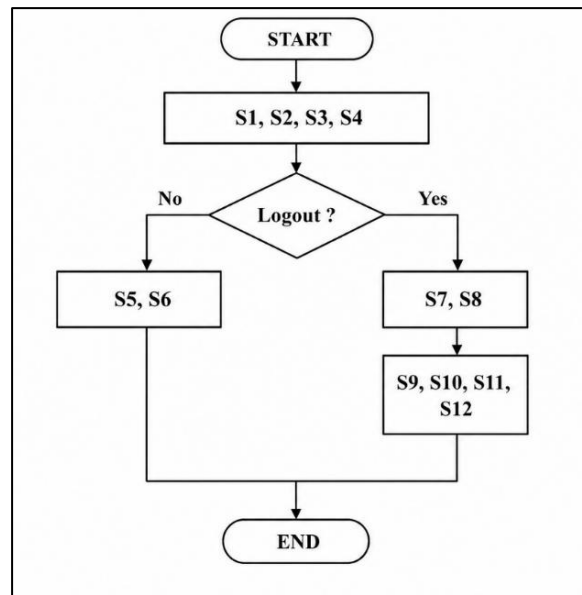
```

b. Identifikasi *Statement* Fungsi *Logout*Tabel 5. 3 Identifikasi *Statement* Fungsi *Logout*

<i>Statement</i>	Source Code	Keterangan
S1	<code>void _Logout(BuildContext context)</code>	Sistem menjalankan fungsi <i>Logout</i>
S2	<code>final t = text[<i>widget.language</i>];</code>	Sistem mengambil data bahasa sesuai pilihan pengguna
S3	<code>showDialog(...)</code>	Sistem menampilkan dialog konfirmasi <i>Logout</i>
S4	<code>builder: (_) => AlertDialog(...)</code>	Sistem menampilkan isi dialog <i>Logout</i>
S5	<code>onPressed: () => Navigator.pop(context)</code>	Pengguna memilih tombol batal
S6	<code>Navigator.pop(context)</code>	Sistem menutup dialog <i>Logout</i>
S7	<code>onPressed: () { ... }</code>	Pengguna memilih tombol <i>Logout</i>
S8	<code>Navigator.pop(context)</code>	Sistem menutup dialog konfirmasi
S9	<code>Navigator.pushAndRemoveUntil(...)</code>	Sistem menghapus seluruh halaman sebelumnya
S10	<code>builder: (_) => const LoginPage()</code>	Sistem mengarahkan pengguna menuju halaman <i>LoginPage</i>
S11	<code>(route) => False</code>	Sistem mengakhiri seluruh halaman sebelumnya
S12	<code>child: Text(t['yes']!)</code>	Proses <i>Logout</i> selesai

Sumber : data diolah oleh penulis(2026)

c. *Flowchart Fungsi Logout*



Sumber : data diolah oleh penulis (2026)

Gambar 5. 29 *Flowchart Fungsi Logout*

Berdasarkan diagram alur pada *Flowchart* fungsi *Logout* diatas, urutan logika kerja fungsi *Logout* dapat dijabarkan sebagai berikut:

1. S1 dan S2 (Menjalankan Fungsi *Logout*):
Sistem memulai proses dengan menjalankan fungsi *log out* serta mengambil data bahasa sesuai pengaturan bahasa yang sedang digunakan oleh pengguna pada aplikasi.
2. S3 dan S4 (Menampilkan Dialog Konfirmasi):
Sistem menampilkan dialog konfirmasi log out berupa AlertDialog yang berisi pilihan batal atau *Logout* untuk memastikan tindakan pengguna sebelum keluar dari aplikasi.
3. S5 dan S6 (Membatalkan *Logout*):
Apabila pengguna memilih tombol batal, maka sistem akan menutup dialog konfirmasi *log out* menggunakan `Navigator.pop(context)`, kemudian proses berakhir dan pengguna tetap berada pada halaman `HomePage`.
4. S7 dan S8 (Melanjutkan *Logout*):
Apabila pengguna memilih tombol *Logout*, maka sistem akan menutup

dialog konfirmasi terlebih dahulu sebelum menjalankan proses *Logout* dari aplikasi.

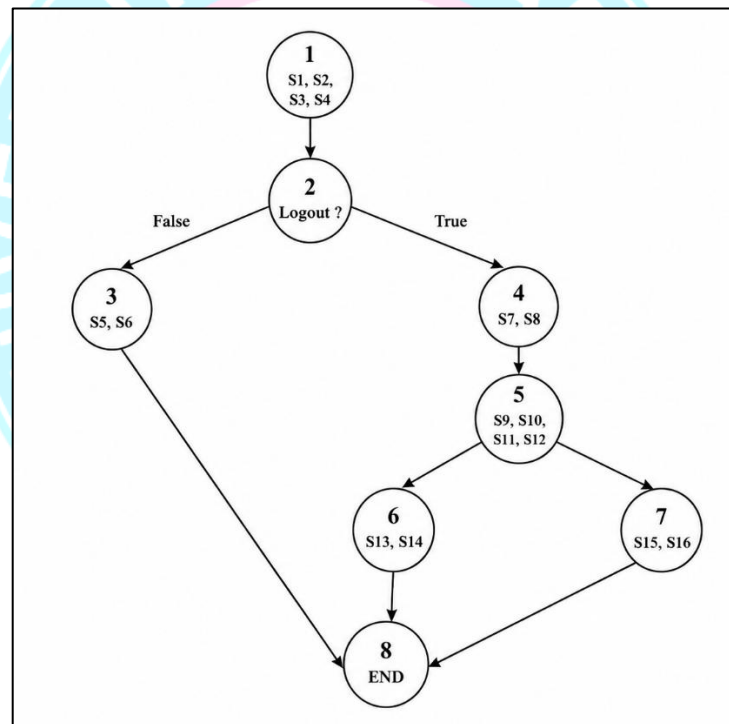
5. S9, S10, dan S11 (Menghapus Halaman Sebelumnya):

Sistem menjalankan `Navigator.pushAndRemoveUntil()` untuk menghapus seluruh halaman yang sebelumnya aktif sehingga pengguna tidak dapat kembali ke halaman `HomePage` setelah *Logout* dilakukan.

6. S12 (Kembali ke Halaman Login):

Setelah seluruh halaman sebelumnya dihapus, sistem mengarahkan pengguna menuju halaman `LoginPage` sebagai tanda bahwa proses *Logout* berhasil dilakukan dan sesi pengguna telah berakhir.

d. *Flowgraph* Fungsi *Logout*



Sumber : data diolah oleh penulis (2026)

Gambar 5. 30 *Flowgraph* Fungsi *Logout*

Flowgraph pada fungsi *Logout* ini merepresentasikan alur eksekusi program berdasarkan hasil pengelompokan *Statement* yang telah diidentifikasi sebelumnya. *Flowgraph* yang dibentuk terdiri dari 6 node dan 7 edge, dengan penambahan Node 6 sebagai terminal node (end process) yang berfungsi untuk

menandai berakhirnya seluruh alur eksekusi program. Node pertama (Node 1) terdiri dari S1 hingga S4 yang mencakup proses menjalankan fungsi *Logout*, mengambil data bahasa sesuai pilihan pengguna, menampilkan dialog konfirmasi *Logout*, serta menampilkan isi dialog kepada pengguna. Seluruh proses tersebut masih berada pada alur linear tanpa adanya percabangan sehingga digabungkan ke dalam satu node.

Selanjutnya, Node 2 merepresentasikan titik keputusan yang menentukan pilihan pengguna terhadap proses *Logout*. Apabila pengguna memilih batal (*False*), maka alur program akan menuju Node 3 yang berisi S5 dan S6 untuk menutup dialog konfirmasi *Logout* dan mengakhiri proses tanpa keluar dari aplikasi. Sebaliknya, apabila pengguna memilih *Logout* (*True*), maka alur program dilanjutkan menuju Node 4 yang berisi S7 dan S8 untuk menutup dialog konfirmasi sebelum proses *Logout* dijalankan.

Tahap berikutnya dilanjutkan menuju Node 5 yang terdiri dari S9 hingga S12, yaitu proses penghapusan seluruh halaman sebelumnya menggunakan *Navigator.pushAndRemoveUntil()* serta pengalihan pengguna menuju halaman *LoginPage* sebagai tanda *Logout* berhasil dilakukan. Seluruh alur proses tersebut kemudian bermuara pada Node 6 sebagai node akhir (end process).

e. *Cyclomatic Complexity* Fungsi *Logout*

Berdasarkan *Flowgraph* yang telah dibentuk, terdapat 1 predicate node (P), yaitu pada Node 2 yang melakukan pengecekan pilihan pengguna terhadap proses *Logout*, apakah pengguna memilih batal atau melanjutkan *Logout*. Oleh karena itu, nilai *Cyclomatic Complexity* dapat dihitung menggunakan rumus berikut:

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

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

$$V(G) = 2$$

Dari perhitungan tersebut diperoleh nilai *Cyclomatic Complexity* sebesar 2. Nilai *Cyclomatic Complexity* tersebut menunjukkan bahwa terdapat dua jalur independen dalam proses eksekusi fungsi *Logout*, yang merepresentasikan seluruh kemungkinan alur logika berdasarkan pilihan pengguna pada dialog konfirmasi

Logout. 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 *Logout* telah ditangani dengan benar.

f. *Independent path* Fungsi *Logout*

Berdasarkan hasil perhitungan *Cyclomatic Complexity* yang diperoleh dari *Flowgraph* fungsi *Logout*, nilai tersebut digunakan sebagai dasar untuk menentukan jumlah jalur independen yang harus diuji dalam proses whitebox testing. Setiap jalur independen merepresentasikan kemungkinan alur eksekusi program yang berbeda berdasarkan pilihan pengguna pada dialog konfirmasi *Logout*. Dengan demikian, setiap kemungkinan skenario pada fungsi *Logout* dapat dianalisis dan diuji secara menyeluruh untuk memastikan seluruh logika program berjalan dengan benar. Sehingga didapatkan 2 jalur independen (*independent path*) berikut ini:

1. Path 1:

$1 \rightarrow 2 \rightarrow 3 \rightarrow 6$

(Pengguna memilih batal sehingga proses *Logout* dibatalkan dan pengguna tetap berada pada halaman *HomePage*)

2. Path 2:

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

(Pengguna memilih *Logout* sehingga sistem menghapus seluruh halaman sebelumnya dan mengarahkan pengguna menuju halaman *LoginPage*)

g. *Test Case* Fungsi *Logout*

Pengujian dilakukan berdasarkan jalur independen (*independent path*) yang telah diperoleh dari hasil analisis *Flowgraph* fungsi *Logout*. Setiap jalur independen direpresentasikan ke dalam beberapa *Test Case* berdasarkan pilihan pengguna pada dialog konfirmasi *Logout* 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 *Logout* secara menyeluruh.

Tabel 5. 4 *Test Case* Fungsi *Logout*

Test Case	Kondisi Pengujian	Jalur Eksekusi	Output
TC1	Pengguna memilih tombol batal	1 → 2 → 3 → 6	Dialog <i>Logout</i> tertutup dan pengguna tetap berada pada halaman <i>HomePage</i>
TC2	Pengguna memilih tombol <i>Logout</i>	1 → 2 → 4 → 5 → 6	Sistem berhasil <i>Logout</i> dan berpindah ke halaman <i>LoginPage</i>

Sumber : data diolah oleh penulis (2026)

h. *Statement Coverage* Fungsi *Logout*

Statement Coverage digunakan untuk mengukur sejauh mana seluruh *Statement* pada fungsi *Logout* 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
2. TC2:
S1, S2, S3, S4, S7, S8, S9, S10, S11, S12

Berdasarkan hasil pengujian yang telah dilakukan, seluruh *Statement* pada fungsi *Logout* yaitu S1 hingga S12 berhasil dieksekusi melalui kombinasi *Test Case* TC1 dan TC2. Dengan demikian diperoleh nilai *Statement Coverage* sebesar:

$$\frac{12}{12} \times 100\% = 100\%$$

Nilai tersebut menunjukkan bahwa seluruh *Statement* pada fungsi *Logout* telah diuji dan seluruh kemungkinan alur eksekusi program berhasil dicakup dalam proses pengujian whitebox.

5.4.3 Pengujian Fungsi Simpan Materi

a. Potongan *Source Code* Fungsi Simpan Materi

Fungsi simpan materi digunakan untuk melakukan proses penyimpanan data materi hardware ke *database Firebase Firestore*. Proses yang dilakukan meliputi validasi input judul dan isi materi, proses upload gambar ke *Cloudinary* apabila pengguna memilih gambar, serta proses penambahan atau pembaruan data materi ke database. Selain itu, fungsi ini juga menangani proses reset form dan penutupan dialog setelah data berhasil disimpan. Berikut adalah potongan *source code* fungsi simpan materi yang digunakan dalam tahap pengujian:

```
Future<void> _simpanMateri() async {
  if (_judulController.text.isEmpty ||
    _isiController.text.isEmpty) {
    return;
  }

  String? imageUrl;

  if (_gambar != null) {
    imageUrl =
      await uploadToCloudinary(_gambar!);
  }

  try {
    if (_editDocId == null) {
      await FirebaseFirestore.instance

        .collection('materi硬件')
        .add({

          'judul': _judulController.text,
          'isi': _isiController.text,
          'gambar': imageUrl,
```

```

        'created':
            FieldValue.serverTimestamp(),
    });
} else {
    await FirebaseFirestore.instance

        .collection('materi_hardware')
        .doc(_editDocId)
        .update({
            'judul': _judulController.text
        },
        'isi': _isiController.text,
        'gambar': imageUrl,
    ));
}

_resetForm();

if (!mounted) return;

Navigator.pop(context);
} catch (e) {

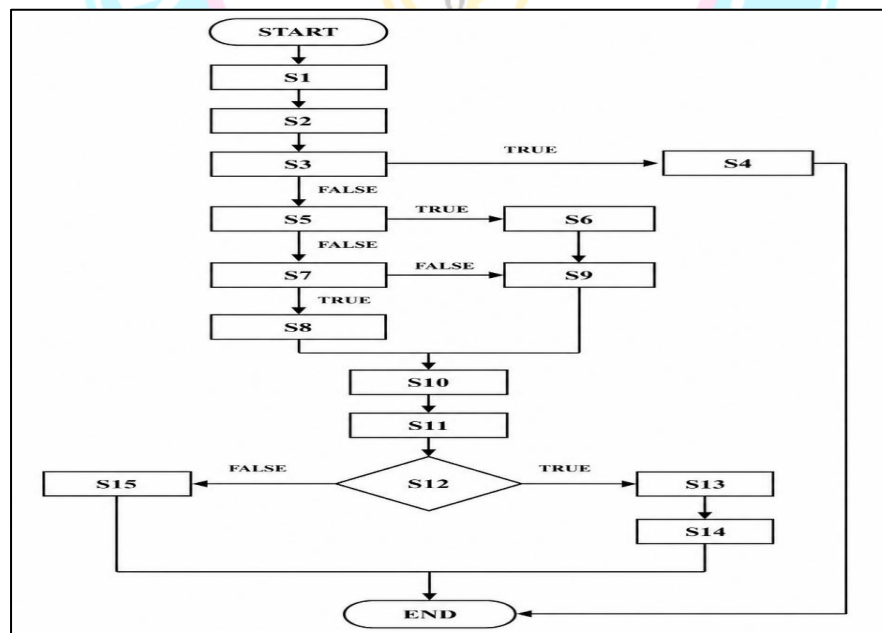
    debugPrint("Firestore error: $e");
}
}

```

b. Identifikasi *Statement* Fungsi Simpan MateriTabel 5. 5 Identifikasi *Statement* Fungsi Simpan Materi

<i>Statement</i>	Source Code	Keterangan
S1	<code>_judulController.text.isEmpty</code>	Memeriksa input judul kosong
S2	<code>_isiController.text.isEmpty</code>	Memeriksa input isi kosong
S3	<code>Return</code>	Menghentikan proses jika input kosong
S4	<code>String? imageUrl;</code>	Membuat variabel URL gambar
S5	<code>if (_gambar != null)</code>	Memeriksa apakah gambar dipilih
S6	<code>uploadToCloudinary(_gambar!)</code>	Upload gambar ke Cloudinary
S7	<code>if (_editDocId == null)</code>	Memeriksa proses tambah atau edit
S8	<code>.add({...})</code>	Menambahkan data materi baru
S9	<code>.update({...})</code>	Memperbarui data materi
S10	<code>_resetForm()</code>	Mengosongkan form
S11	<code>if (!mounted) return</code>	Memeriksa <i>widget</i> masih aktif
S12	<code>Navigator.pop(context)</code>	Menutup dialog
S13	<code>catch (e)</code>	Menangani <i>error</i> Firestore
S14	<code>debugPrint(...)</code>	Menampilkan log <i>error</i>

Sumber : data diolah oleh penulis(2026)

c. *Flowchart* Fungsi Simpan Materi

Sumber : data diolah oleh penulis (2026)

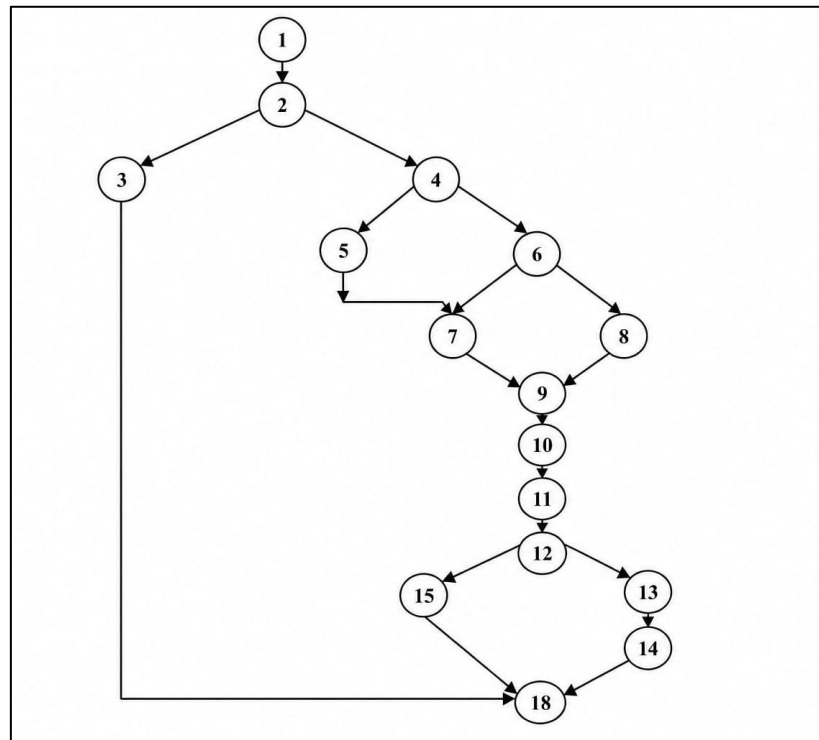
Gambar 5. 31 *Flowchart* Fungsi Simpan Materi

Berdasarkan diagram alur pada *Flowchart* Fungsi Simpan Materi di atas, urutan logika kerja fungsi simpan materi dapat dijabarkan sebagai berikut:

1. S1 dan S2 (Validasi Input): Sistem memulai proses dengan memeriksa apakah input judul materi dan isi materi masih kosong. Validasi ini dilakukan untuk memastikan seluruh data penting telah diisi oleh pengguna sebelum proses penyimpanan dilakukan.
2. S3 (Penghentian Proses): Jika salah satu input masih kosong (*True*), maka sistem akan menghentikan proses penyimpanan menggunakan perintah *return* sehingga data tidak akan diproses lebih lanjut.
3. S4 dan S5 (Pengecekan Gambar): Jika seluruh input telah terisi (*False*), sistem melanjutkan proses dengan membuat variabel *imageUrl* dan memeriksa apakah pengguna memilih gambar materi.
4. S6 (Upload Gambar): Jika pengguna memilih gambar (*True*), maka sistem akan melakukan proses upload gambar ke Cloudinary menggunakan fungsi *uploadToCloudinary()* untuk memperoleh URL gambar yang akan disimpan ke database.
5. S7 (Pengecekan Edit Data): Setelah proses upload selesai atau tidak ada gambar yang dipilih, sistem akan memeriksa apakah proses yang dilakukan merupakan tambah data baru atau edit data materi.
6. S8 (Tambah Data): Jika data yang diproses merupakan data baru (*False*), maka sistem akan menambahkan data materi ke *Firebase Firestore* menggunakan fungsi *add()*.
7. S9 (Edit Data): Jika data yang diproses merupakan edit data (*True*), maka sistem akan memperbarui data materi yang sudah ada menggunakan fungsi *update()*.
8. S10, S11, dan S12 (Reset dan Penutupan Dialog): Setelah proses penyimpanan berhasil dilakukan, sistem akan mengosongkan form input, memeriksa status *mounted widget*, kemudian menutup dialog atau halaman input materi menggunakan *Navigator.pop(context)*.
9. S13 dan S14 (Penanganan *Error*): Jika terjadi kesalahan pada proses penyimpanan data ke *Firebase Firestore*, sistem akan menangkap *error*

menggunakan blok catch dan menampilkan pesan kesalahan melalui `debugPrint()`.

d. *Flowgraph* Fungsi Simpan Materi



Sumber : data diolah oleh penulis (2026)

Gambar 5. 32 *Flowgraph* Fungsi Simpan Materi

Flowgraph pada fungsi simpan materi ini merepresentasikan alur eksekusi program berdasarkan hasil pengelompokan *Statement* yang telah diidentifikasi sebelumnya. *Flowgraph* yang dibentuk terdiri dari 18 node dan 21 edge, dengan penambahan Node 18 sebagai terminal node (end process) yang berfungsi untuk menandai berakhirnya seluruh alur eksekusi program. Node pertama (Node 1) merupakan awal proses program, kemudian dilanjutkan ke Node 2 yang merepresentasikan proses validasi input judul dan isi materi. Pada tahap ini sistem melakukan pengecekan apakah data yang dimasukkan pengguna masih kosong atau tidak.

Selanjutnya, Node 3 merupakan titik keputusan hasil validasi input. Jika kondisi bernilai *True* atau input masih kosong, maka alur program langsung menuju Node 4 yang berfungsi untuk menghentikan proses penyimpanan data.

Sebaliknya, jika kondisi bernilai *False* atau input valid, maka alur dilanjutkan menuju Node 5 yang digunakan untuk memeriksa apakah pengguna memilih gambar materi atau tidak.

Apabila pengguna memilih gambar, maka proses akan dilanjutkan ke Node 6 untuk melakukan upload gambar ke Cloudinary. Setelah proses upload selesai, alur akan menuju Node 8. Namun apabila pengguna tidak memilih gambar, maka alur langsung menuju Node 7 sebelum diteruskan ke Node 8 sebagai proses lanjutan penyimpanan data.

Node 8 merepresentasikan titik keputusan untuk menentukan apakah proses yang dilakukan merupakan tambah data baru atau edit data materi. Jika kondisi bernilai *False*, maka alur program menuju Node 9 untuk menambahkan data materi baru ke *Firestore*. Sebaliknya, jika kondisi bernilai *True*, maka alur akan menuju Node 10 dan Node 11 yang merepresentasikan proses pembaruan data materi yang telah ada sebelumnya.

Selanjutnya, Node 12 merupakan titik keputusan untuk memeriksa kondisi *mounted widget* sebelum dialog ditutup. Jika kondisi bernilai *True*, maka alur program menuju Node 13 dan Node 14 yang merepresentasikan proses penutupan dialog menggunakan *Navigator.pop(context)*. Sedangkan apabila kondisi bernilai *False*, maka program akan menuju Node 15 untuk menghentikan proses tanpa melakukan penutupan dialog.

Seluruh kemungkinan alur program pada akhirnya akan bermuara pada Node 18 sebagai terminal node atau akhir proses eksekusi program fungsi simpan materi.

e. *Cyclomatic Complexity* Fungsi Simpan Materi

Cyclomatic Complexity digunakan untuk mengukur tingkat kompleksitas logika pada suatu program berdasarkan jumlah jalur independen yang terdapat pada *Flowgraph*. Perhitungan ini bertujuan untuk mengetahui banyaknya jalur pengujian minimum yang harus dilakukan agar seluruh alur program dapat dieksekusi.

Berdasarkan *Flowgraph* fungsi simpan materi yang telah dibentuk, diperoleh jumlah node sebanyak 18 dan jumlah edge sebanyak 21. Maka nilai *Cyclomatic Complexity* dapat dihitung menggunakan rumus berikut:

$$V(G) = E - N + 2$$

Keterangan:

$$V(G) = \text{Cyclomatic Complexity}$$

E = Jumlah edge

N = Jumlah node

Perhitungan *Cyclomatic Complexity*:

$$V(G) = 21 - 18 + 2$$

$$V(G) = 5$$

Selain menggunakan rumus edge dan node, *Cyclomatic Complexity* juga dapat dihitung menggunakan jumlah predicate node (P). Pada *Flowgraph* fungsi simpan materi terdapat 4 predicate node, yaitu:

Node 3 : Validasi input kosong

Node 5 : Pengecekan gambar

Node 8 : Pengecekan tambah atau edit data

Node 12 : Pengecekan mounted *widget*

Maka perhitungannya sebagai 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*) yang harus diuji untuk memastikan seluruh kemungkinan alur logika pada fungsi simpan materi dapat berjalan dengan baik.

f. *Independen Path* Fungsi Simpan Materi

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

program yang berbeda berdasarkan kondisi percabangan, khususnya pada proses validasi input, pengecekan gambar, proses tambah atau edit data, serta pengecekan kondisi mounted *widget*. Dengan demikian, setiap kemungkinan skenario pada fungsi simpan materi dapat dianalisis dan diuji secara menyeluruh untuk memastikan seluruh logika program berjalan dengan benar. Sehingga didapatkan 5 jalur independen (*independent path*) berikut ini:

1. Path 1:

1 → 2 → 3 → 4 → 18

(Input judul atau isi materi kosong sehingga proses penyimpanan dihentikan)

2. Path 2:

1 → 2 → 3 → 5 → 6 → 8 → 9 → 18

(Input valid, gambar dipilih, kemudian data materi baru ditambahkan)

3. Path 3:

1 → 2 → 3 → 5 → 7 → 8 → 9 → 18

(Input valid, tanpa gambar, kemudian data materi baru ditambahkan)

4. Path 4:

1 → 2 → 3 → 5 → 6 → 8 → 10 → 11 → 12 → 13 → 14 → 18

(Input valid, gambar dipilih, proses edit data berhasil, dan dialog ditutup)

5. Path 5:

1 → 2 → 3 → 5 → 6 → 8 → 10 → 11 → 12 → 15 → 18

(Input valid, gambar dipilih, proses edit data dilakukan, namun *widget* tidak mounted sehingga proses dihentikan)

g. *Test Case* Fungsi Simpan Materi

Berdasarkan hasil identifikasi *independent path* pada fungsi simpan materi, maka dibuat beberapa skenario pengujian (*test case*) untuk menguji seluruh kemungkinan alur eksekusi program. Setiap *Test Case* dirancang untuk memastikan bahwa proses validasi input, upload gambar, penambahan data, pembaruan data, serta penanganan kondisi tertentu dapat berjalan sesuai dengan logika program yang telah dirancang.

Adapun *Test Case* pada fungsi simpan materi adalah sebagai berikut:

Tabel 5. 6 *Test Case* Fungsi Simpan Materi

Test Case	Jalur Independen	Skenario Pengujian	Hasil Yang Diharapkan
TC1	1 → 2 → 3 → 4 → 18	Input judul atau isi materi kosong	Sistem menghentikan proses penyimpanan dan data tidak tersimpan
TC2	1 → 2 → 3 → 5 → 6 → 8 → 9 → 18	Input valid dan gambar dipilih saat tambah data	Sistem berhasil upload gambar dan menambahkan data materi
TC3	1 → 2 → 3 → 5 → 7 → 8 → 9 → 18	Input valid tanpa gambar saat tambah data	Sistem berhasil menambahkan data materi tanpa gambar
TC4	1 → 2 → 3 → 5 → 6 → 8 → 10 → 11 → 12 → 13 → 14 → 18	Input valid, gambar dipilih, dan proses edit data berhasil	Sistem berhasil memperbarui data materi dan menutup dialog
TC5	1 → 2 → 3 → 5 → 6 → 8 → 10 → 11 → 12 → 15 → 18	Input valid, proses edit data dilakukan, namun <i>widget</i> tidak mounted	Sistem menghentikan proses tanpa menutup dialog

Sumber : data diolah oleh penulis (2026)

h. *Statement Coverage* Fungsi Simpan Materi

Statement Coverage digunakan untuk mengukur sejauh mana seluruh *Statement* pada fungsi simpan materi 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:

TC1:

S1, S2, S3

TC2:

S1, S2, S4, S5, S6, S7, S8, S10, S11, S12

TC3:

S1, S2, S4, S5, S7, S8, S10, S11, S12

TC4:

S1, S2, S4, S5, S6, S7, S9, S10, S11, S12

TC5:

S1, S2, S4, S5, S6, S7, S9, S10, S11

Berdasarkan hasil pengujian yang telah dilakukan, seluruh *Statement* pada fungsi simpan materi yaitu S1 hingga S14 berhasil dieksekusi melalui kombinasi *Test Case* TC1, TC2, TC3, TC4, dan TC5. Dengan demikian diperoleh nilai *Statement Coverage* sebesar:

$$14/14 \times 100\% = 100\%$$

Nilai tersebut menunjukkan bahwa seluruh *Statement* pada fungsi simpan materi telah diuji dan seluruh kemungkinan alur eksekusi program berhasil dicakup dalam proses pengujian whitebox.

5.4.4 Pengujian Fungsi Hapus Materi

a. Potongan *Source Code* Fungsi Hapus Materi

Fungsi hapus materi digunakan untuk menghapus data materi *hardware/software* pada *Firebase Firestore*. Proses dimulai dengan menampilkan dialog konfirmasi kepada pengguna. Jika pengguna membatalkan proses, dialog akan ditutup tanpa menghapus data. Namun jika pengguna memilih hapus, sistem akan menghapus data berdasarkan *docId* kemudian menutup dialog setelah proses berhasil dilakukan.

```
void _hapusMateri(String docId) {
    final t =

    text[widget.language] ?? text['ID'];
    showDialog(
```

```

context: context,
builder: (_) => AlertDialog(
  title: Text(t['konfirmasi']!),
  content: Text(t['yakin']!),
  actions: [

    TextButton(
      onPressed: () =>

        Navigator.pop(context),
        child: Text(t['batal']!),
    ),
    ElevatedButton(
      style: ElevatedButton.styleFrom(
        backgroundColor: Colors.red,
      ),
      onPressed: () async {
        await FirebaseFirestore.instance
          .collection(

            'materi_hardware')
          .doc(docId)
          .delete();

        Navigator.pop(context);
      },
      child: Text(t['hapus']!),
    ),
  ],

```

```

    ),
  );
}

```

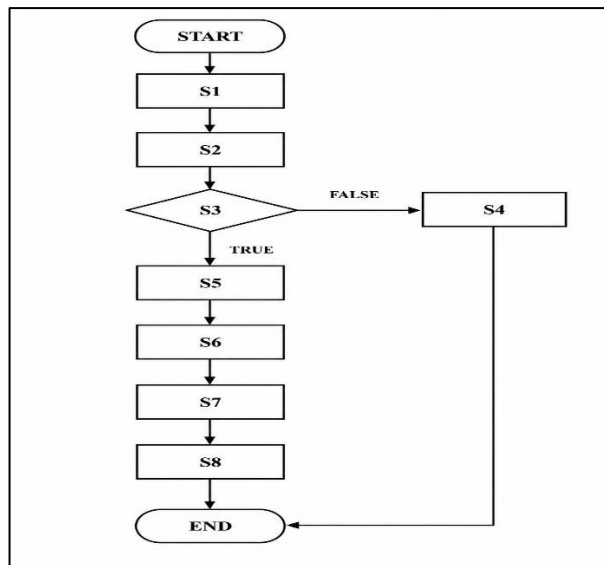
b. Identifikasi *Statement* Fungsi Hapus Materi

Tabel 5. 7 Identifikasi *Statement* Fungsi Hapus Materi

<i>Statement</i>	Source Code	Keterangan
S1	final t = text[<i>widget.language</i>] ?? text['ID']!;	Mengambil teks sesuai bahasa
S2	showDialog(...)	Menampilkan dialog konfirmasi
S3	Navigator.pop(context)	Menutup dialog saat batal
S4	.collection('materi_hardware')	Mengakses collection Firestore
S5	.doc(docId)	Mengambil dokumen berdasarkan ID
S6	.delete()	Menghapus data materi
S7	Navigator.pop(context)	Menutup dialog setelah hapus berhasil

Sumber : data diolah oleh penulis (2026)

c. *Flowchart* Fungsi Hapus Materi



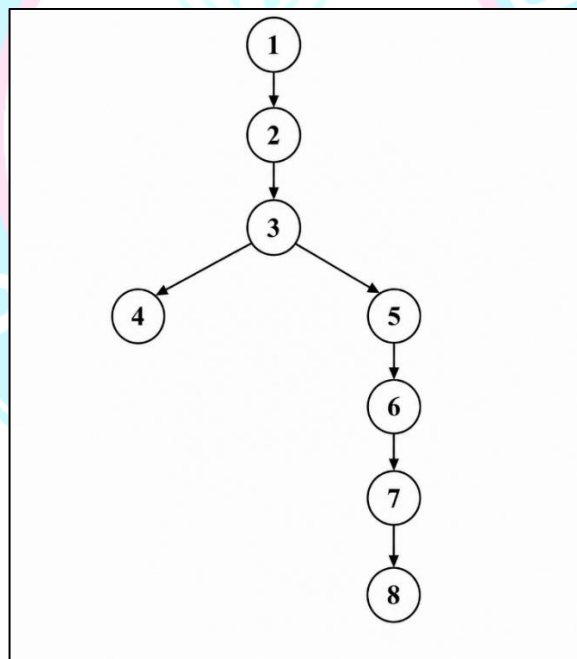
Sumber : data diolah oleh penulis (2026)

Gambar 5. 33 *Flowchart* Fungsi Hapus Materi

Berdasarkan diagram alur pada *Flowchart* fungsi hapus materi di atas, urutan logika kerja fungsi hapus dapat dijabarkan sebagai berikut:

1. S1 (Inisialisasi Bahasa)
Sistem memulai proses dengan mengambil data teks berdasarkan bahasa yang sedang digunakan pada aplikasi. Data teks ini digunakan untuk menampilkan isi dialog konfirmasi sesuai bahasa pengguna.
2. S2 (Menampilkan Dialog Konfirmasi)
Setelah data bahasa diperoleh, sistem menampilkan dialog konfirmasi penghapusan materi kepada pengguna. Dialog ini berisi pilihan batal atau hapus.
3. S3 (Keputusan Pengguna)
Pada tahap ini sistem menunggu tindakan pengguna, apakah pengguna menekan tombol hapus atau membatalkan proses penghapusan.
4. S4 (Batal Hapus)
Jika hasil keputusan pada S3 adalah *False*, berarti pengguna memilih batal. Sistem akan menutup dialog konfirmasi tanpa melakukan perubahan data, kemudian proses berakhir.

5. S5 (Mengakses Collection Firestore)
Jika hasil keputusan pada S3 adalah *True*, sistem melanjutkan proses dengan mengakses collection materi_*hardware* pada *Firestore*.
 6. S6 (Mengambil Dokumen Berdasarkan docId)
Sistem mengambil dokumen materi berdasarkan docId yang dipilih pengguna sebagai identitas data yang akan dihapus.
 7. S7 (Menghapus Dokumen)
Setelah dokumen ditemukan, sistem menjalankan proses delete() untuk menghapus data materi dari database *Firestore*.
 8. S8 (Menutup Dialog Setelah Hapus Berhasil)
Setelah proses penghapusan berhasil dilakukan, sistem menutup dialog konfirmasi dan proses berakhir.
- d. *Flowgraph* Fungsi Hapus Materi



Sumber : data diolah oleh penulis (2026)

Gambar 5. 34 *Flowgraph* Fungsi Hapus Materi

Flowgraph pada fungsi hapus materi merepresentasikan alur eksekusi program berdasarkan hasil pengelompokan *Statement* yang telah diidentifikasi sebelumnya. *Flowgraph* yang dibentuk terdiri dari 8 node dan 8 edge. Node pertama (Node 1) merepresentasikan proses inisialisasi bahasa yang digunakan

aplikasi untuk menentukan teks dialog konfirmasi. Selanjutnya Node 2 menggambarkan proses penampilan dialog konfirmasi hapus materi kepada pengguna.

Node 3 merupakan titik keputusan yang digunakan untuk menentukan tindakan pengguna, apakah membatalkan atau melanjutkan proses penghapusan data. Apabila pengguna memilih batal (*False*), maka alur program akan menuju Node 4 dan proses berakhir tanpa adanya perubahan data pada database.

Sebaliknya, apabila pengguna memilih hapus (*True*), maka alur program akan dilanjutkan menuju Node 5 yang merepresentasikan proses akses collection materi pada *Firebase Firestore*. Setelah itu program bergerak ke Node 6 untuk mengambil dokumen berdasarkan docId yang dipilih pengguna.

Node 7 merepresentasikan proses penghapusan dokumen menggunakan fungsi `delete()`. Setelah proses penghapusan berhasil dilakukan, alur program berakhir pada Node 8 yang menunjukkan proses penutupan dialog konfirmasi setelah data berhasil dihapus dari database *Firebase Firestore*.

e. *Cyclomatic Complexity* Fungsi Hapus Materi

Cyclomatic Complexity digunakan untuk mengukur tingkat kompleksitas logika program berdasarkan jumlah jalur independen pada *Flowgraph*. Perhitungan *Cyclomatic Complexity* dilakukan menggunakan rumus:

$$V(G)=E-N+2$$

Keterangan:

(E) = jumlah edge

(N) = jumlah node

Berdasarkan *Flowgraph* fungsi hapus materi:

Jumlah node ((N)) = 8

Jumlah edge ((E)) = 8

Maka perhitungan *Cyclomatic Complexity* adalah:

$$V(G)=8-8+2=2$$

Hasil perhitungan menunjukkan bahwa nilai *Cyclomatic Complexity* pada fungsi hapus materi adalah 2, yang berarti terdapat 2 jalur independen (*independent path*) yang harus diuji pada proses whitebox testing fungsi hapus materi.

f. *Independent path* Fungsi Hapus Materi

Berdasarkan *Flowgraph* fungsi hapus materi, diperoleh beberapa jalur independen (*independent path*) yang digunakan dalam proses whitebox testing. Jalur independen merupakan jalur unik yang dilalui program dari awal hingga akhir proses.

Adapun *independent path* pada fungsi hapus materi adalah sebagai berikut:

1. Path 1:

$1 \rightarrow 2 \rightarrow 3 \rightarrow 4$

Jalur ini terjadi ketika pengguna memilih membatalkan proses penghapusan materi sehingga sistem menutup dialog tanpa menghapus data.

2. Path 2:

$1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8$

Jalur ini terjadi ketika pengguna memilih melanjutkan proses penghapusan sehingga sistem mengakses *Firebase Firestore*, mengambil dokumen berdasarkan docId, menghapus data materi, dan menutup dialog setelah proses berhasil dilakukan.

g. *Test Case* Fungsi Hapus Materi

Test Case digunakan untuk menguji seluruh jalur independen pada fungsi hapus materi berdasarkan *Flowgraph* yang telah dibuat. Pengujian dilakukan pada kondisi pengguna membatalkan penghapusan dan pengguna melanjutkan proses penghapusan data.

Tabel 5. 8 *Test Case* Fungsi Hapus Materi

Test Case	Kondisi Pengujian	Jalur Eksekusi	Hasil yang Diharapkan
TC1	Pengguna memilih tombol batal pada dialog konfirmasi	$1 \rightarrow 2 \rightarrow 3 \rightarrow 4$	Dialog tertutup tanpa menghapus data
TC2	Pengguna memilih tombol hapus pada dialog konfirmasi	$1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8$	Data berhasil dihapus dari <i>Firebase Firestore</i>

Sumber : data diolah oleh penulis (2026)

h. *Statement Coverage* Fungsi Hapus Materi

Statement Coverage digunakan untuk memastikan seluruh *Statement* pada fungsi hapus materi telah dieksekusi minimal satu kali selama proses pengujian.

Pada *Test Case* TC1, *Statement* yang dieksekusi adalah proses menampilkan dialog konfirmasi dan membatalkan penghapusan data. Sedangkan pada *Test Case* TC2, *Statement* yang dieksekusi meliputi proses akses collection Firestore, pengambilan dokumen berdasarkan docId, hingga proses penghapusan data materi.

Berdasarkan hasil pengujian, seluruh *Statement* pada fungsi hapus materi berhasil dieksekusi melalui kombinasi *Test Case* TC1 dan TC2 sehingga diperoleh nilai *Statement Coverage* sebesar:

$$\frac{8}{8} \times 100\% = 100\%$$

Hasil tersebut menunjukkan bahwa seluruh *Statement* pada fungsi hapus materi telah berhasil diuji dan seluruh alur program dapat berjalan dengan baik.

5.4.5 Pengujian Fungsi Tambah Latihan

a. Potongan *Source Code* Fungsi *Login*

Fungsi tambah latihan digunakan untuk menambahkan data latihan baru ke dalam database *Firebase Firestore*. Proses yang dilakukan meliputi validasi input nama latihan, penyimpanan data ke database, serta penutupan form setelah data berhasil disimpan. Berikut merupakan potongan source code fungsi tambah latihan yang digunakan pada proses pengujian:

```
onPressed: () async {

    if (titleController.text
        .trim()
        .isEmpty) {
        return;
    }

    await _firestore
```

```

.collection(

    'latihan_soal')
.add({
'title':
    titleController.text
    .trim(),

'created': FieldValue
    .serverTimestamp(),
});

if (!mounted) return;

Navigator.pop(context);
},

```

b. Identifikasi *Statement* Fungsi Tambah Latihan

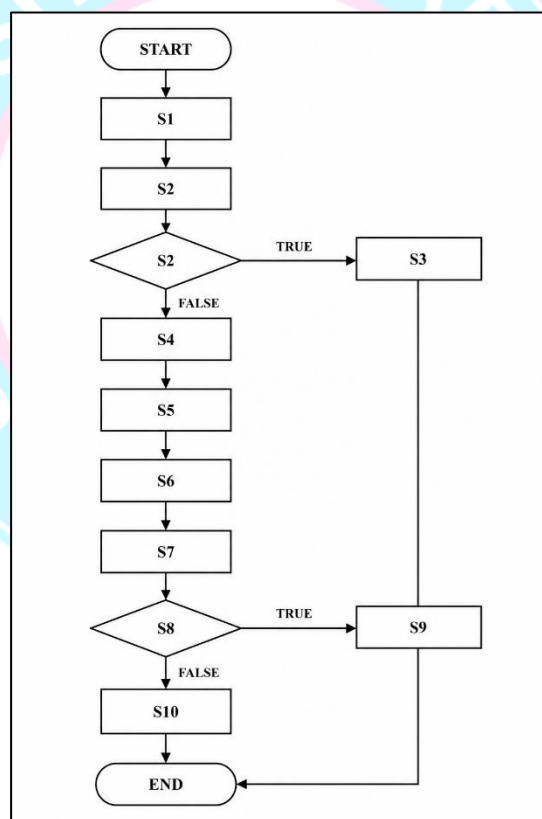
Tabel 5. 9 Identifikasi *Statement* Fungsi Tambah Latihan

<i>Statement</i>	Source Code	Keterangan
S1	titleController.text.trim()	Mengambil input nama latihan
S2	if (titleController.text.trim().isEmpty)	Memeriksa apakah input kosong
S3	return;	Menghentikan proses jika input kosong
S4	_firestore.collection('latihan_soal')	Mengakses collection latihan_soal
S5	.add({...})	Menambahkan data Latihan

S6	'title': titleController.text.trim()	Menyimpan nama Latihan
S7	'created': FieldValue.serverTimestamp()	Menyimpan waktu pembuatan
S8	if (!mounted)	Memeriksa status <i>widget</i>
S9	return;	Menghentikan proses jika <i>widget</i> tidak aktif
S10	Navigator.pop(context)	Menutup form tambah Latihan

Sumber : data diolah oleh penulis(2026)

c. *Flowchart* Fungsi Tambah Latihan



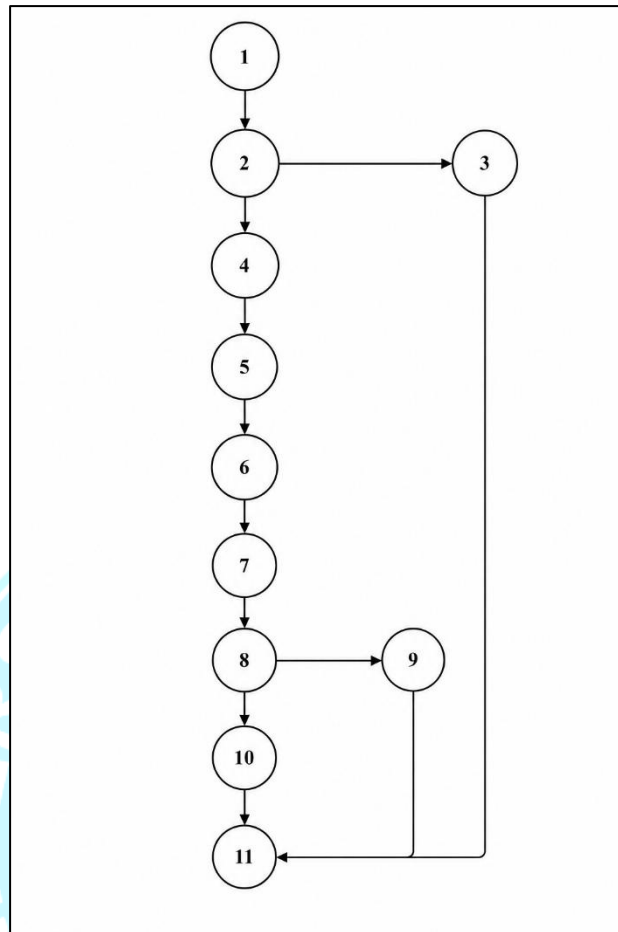
Sumber : data diolah oleh penulis (2026)

Gambar 5. 35 *Flowchart* Fungsi Tambah Latihan

Berdasarkan diagram alur pada gambar *Flowchart* Fungsi Tambah Latihan di atas, urutan logika kerja fungsi tambah latihan dapat dijabarkan sebagai berikut:

1. S1 dan S2 (Input dan Validasi Data):
Sistem memulai proses dengan mengambil input nama latihan dari pengguna melalui field input. Setelah itu, sistem melakukan pengecekan apakah input yang dimasukkan kosong atau tidak.
2. S3 (Pengecekan Input Kosong):
Pada simbol keputusan S3, sistem memeriksa kondisi apakah input nama latihan bernilai kosong (*isEmpty*). Proses ini digunakan untuk memastikan data yang dimasukkan valid sebelum disimpan ke database.
3. S4 (Proses Dihentikan):
Jika hasil pengecekan pada S3 bernilai *True*, maka sistem akan menjalankan proses return sehingga proses tambah latihan dihentikan dan data tidak disimpan.
4. S5, S6, dan S7 (Penyimpanan Data):
Jika input tidak kosong (*False*), sistem melanjutkan proses dengan mengakses collection latihan_soal pada *Firebase Firestore*. Selanjutnya sistem menambahkan data latihan berupa title dan created timestamp ke database.
5. S8 (Pengecekan Status *Widget*):
Setelah proses penyimpanan data selesai dilakukan, sistem memeriksa kondisi *widget* menggunakan *if (!mounted)* untuk memastikan *widget* masih aktif sebelum menjalankan proses berikutnya.
6. S9 (*Widget* Tidak Aktif):
Jika hasil pengecekan pada S8 bernilai *True*, maka sistem menjalankan return sehingga proses dihentikan karena *widget* sudah tidak aktif.
7. S10 (Menutup Form Tambah Latihan):
Jika *widget* masih aktif (*False*), sistem menjalankan *Navigator.pop(context)* untuk menutup halaman form tambah latihan dan proses berakhir.

d. *Flowgraph* Fungsi Tambah Latihan



Sumber : data diolah oleh penulis (2026)

Gambar 5. 36 *Flowgraph* Fungsi Tambah Latihan

Flowgraph pada fungsi tambah latihan 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 Node 11 sebagai terminal node (end process) yang berfungsi untuk menandai berakhirnya seluruh alur eksekusi program. Node pertama (Node 1) terdiri dari S1 dan S2 yang mencakup proses pengambilan input nama latihan serta validasi awal input pada sistem. Seluruh proses tersebut masih berada pada alur linear sehingga digabungkan ke dalam satu node.

Selanjutnya, Node 2 merepresentasikan *Statement* S2 yang merupakan titik keputusan untuk memeriksa apakah input nama latihan kosong (isEmpty). Apabila kondisi bernilai *True*, maka alur program akan menuju Node 3 yang berisi S3 (return) sehingga proses tambah latihan dihentikan. Namun apabila kondisi

bernilai *False*, maka alur program dilanjutkan menuju Node 4, Node 5, Node 6, dan Node 7 yang mencakup proses akses collection *Firebase Firestore*, proses penambahan data latihan, penyimpanan title, serta penyimpanan timestamp created.

Setelah proses penyimpanan data selesai dilakukan, alur program menuju Node 8 yang merepresentasikan *Statement* S8 untuk memeriksa status *widget* menggunakan kondisi *if (!mounted)*. Jika kondisi bernilai *True*, maka alur program akan menuju Node 9 yang berisi S9 (return) sehingga proses dihentikan karena *widget* sudah tidak aktif. Sebaliknya, jika kondisi bernilai *False* maka sistem melanjutkan proses menuju Node 10 yang berisi S10 (*Navigator.pop(context)*) untuk menutup halaman form tambah latihan. Seluruh kemungkinan alur eksekusi tersebut kemudian bermuara pada Node 11 sebagai node akhir (end process).

e. *Cyclomatic Complexity* Fungsi Tambah Latihan

Berdasarkan *Flowgraph* yang telah dibentuk, terdapat 2 predicate node (P), yaitu pada Node 2 yang melakukan pengecekan apakah input nama latihan kosong (*isEmpty*), dan Node 8 yang melakukan pengecekan status *widget* menggunakan kondisi *if (!mounted)*. 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 diperoleh nilai *Cyclomatic Complexity* sebesar 3. Nilai tersebut menunjukkan bahwa terdapat tiga jalur independen dalam proses eksekusi fungsi tambah latihan, yang merepresentasikan seluruh kemungkinan alur logika program berdasarkan kondisi validasi input dan status *widget*. Nilai ini kemudian digunakan sebagai dasar dalam menentukan jumlah *independent path* yang akan diuji pada tahap pengujian berikutnya sehingga seluruh alur eksekusi program dapat diverifikasi dengan baik.

f. *Independen Path* Fungsi Tambah Latihan

Berdasarkan hasil perhitungan *Cyclomatic Complexity* yang diperoleh dari *Flowgraph* fungsi tambah latihan, nilai tersebut digunakan sebagai dasar untuk menentukan jumlah jalur independen yang harus diuji dalam proses whitebox testing. Setiap jalur independen merepresentasikan kemungkinan alur eksekusi program yang berbeda berdasarkan kondisi percabangan, khususnya pada validasi input nama latihan dan pengecekan status *widget*. Dengan demikian, setiap kemungkinan alur pada fungsi tambah latihan dapat dianalisis dan diuji secara menyeluruh untuk memastikan seluruh logika program berjalan dengan benar. Sehingga diperoleh 3 jalur independen (*independent path*) berikut ini:

1. Path 1:

$1 \rightarrow 2 \rightarrow 3 \rightarrow 11$

(Input nama latihan kosong sehingga proses dihentikan)

2. Path 2:

$1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 11$

(Input valid, data berhasil disimpan, namun *widget* tidak aktif)

3. Path 3:

$1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 10 \rightarrow 11$

(Input valid, data berhasil disimpan, dan form tambah latihan ditutup)

g. *Test Case* Fungsi Tambah Latihan

Pengujian dilakukan berdasarkan jalur independen (*independent path*) yang telah diperoleh dari hasil analisis *Flowgraph* fungsi tambah latihan. Setiap jalur independen direpresentasikan ke dalam beberapa *Test Case* dengan kombinasi input nama latihan dan kondisi *widget* 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 tambah latihan secara menyeluruh.

Tabel 5. 10 *Test Case* Fungsi Tambah Latihan

Test Case	Input Nama Latihan	Kondisi Widget	Jalur Eksekusi	Output
TC1	Kosong	Aktif	1 → 2 → 3 → 11	Proses dihentikan dan data tidak disimpan
TC2	Valid	Tidak aktif	1 → 2 → 4 → 5 → 6 → 7 → 8 → 9 → 11	Data berhasil disimpan namun proses dihentikan
TC3	Valid	Aktif	1 → 2 → 4 → 5 → 6 → 7 → 8 → 10 → 11	Data berhasil disimpan dan form ditutup

Sumber : data diolah oleh penulis (2026)

h. *Statement Coverage* Fungsi Tambah Latihan

Statement Coverage digunakan untuk mengukur sejauh mana seluruh *Statement* pada fungsi tambah latihan 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
2. TC2:
S1, S2, S4, S5, S6, S7, S8, S9
3. TC3:
S1, S2, S4, S5, S6, S7, S8, S10

Berdasarkan hasil pengujian yang telah dilakukan, seluruh *Statement* pada fungsi tambah latihan yaitu S1 hingga S10 berhasil dieksekusi melalui kombinasi *Test Case* TC1, TC2, dan TC3. Dengan demikian diperoleh nilai *Statement Coverage* sebesar:

$$\frac{10}{10} \times 100\% = 100\%$$

Nilai tersebut menunjukkan bahwa seluruh *Statement* pada fungsi tambah latihan telah diuji dan seluruh kemungkinan alur eksekusi program berhasil dicakup dalam proses pengujian whitebox.

5.4.6 Pengujian Fungsi Hapus Latihan

a. Potongan *Source Code* Fungsi Hapus Latihan

Fungsi hapus latihan digunakan untuk menghapus data latihan dari *Firebase Firestore* berdasarkan *docId* yang dipilih pengguna. Proses ini diawali dengan menampilkan dialog konfirmasi penghapusan, kemudian sistem akan menjalankan proses delete data apabila pengguna memilih tombol “Ya”. Berikut merupakan potongan *source code* fungsi hapus latihan yang digunakan pada tahap pengujian:

```
ElevatedButton(
  onPressed: () async {
    await _firestore
      .collection('latihan_soal')
      .doc(docId)
      .delete();
    Navigator.pop(context);
  },
  child: Text(t['ya']!),
),
```

b. Identifikasi *Statement* Fungsi Hapus Latihan

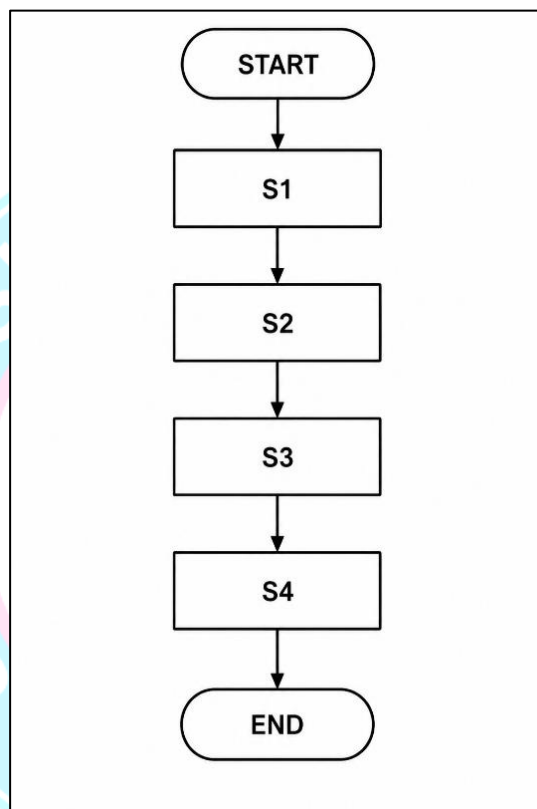
Tabel 5. 11 Identifikasi *Statement* Fungsi Hapus Latihan

<i>Statement</i>	<i>Source Code</i>	<i>Keterangan</i>
S1	<code>_firestore.collection('latihan_soal')</code>	Mengakses collection latihan_soal
S2	<code>.doc(docId)</code>	Mengambil document berdasarkan docId

S3	.delete()	Menghapus data Latihan
S4	Navigator.pop(context)	Menutup dialog konfirmasi

Sumber : data diolah oleh penulis (2026)

c. *Flowchart* Fungsi Hapus Latihan



Sumber : data diolah oleh penulis (2026)

Gambar 5. 37 *Flowchart* Fungsi Hapus Latihan

Berdasarkan diagram alir pada gambar *Flowchart* Fungsi Hapus Latihan di atas, urutan logika kerja fungsi hapus latihan dapat dijabarkan sebagai berikut:

1. S1 (Akses Collection):

Sistem memulai proses dengan mengakses collection latihan_soal pada *Firestore*.

2. S2 (Mengambil Document):

Sistem mengambil document latihan berdasarkan docId yang dipilih oleh pengguna.

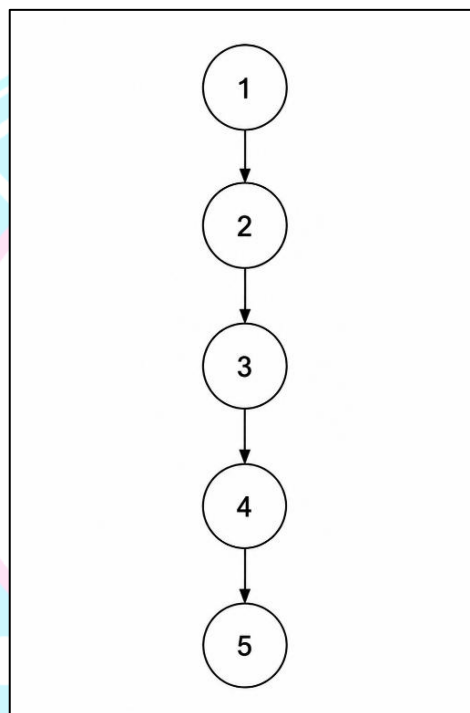
3. S3 (Proses Hapus Data):

Setelah document ditemukan, sistem menjalankan proses `.delete()` untuk menghapus data latihan dari *Firebase Firestore*.

4. S4 (Menutup Dialog):

Setelah proses penghapusan berhasil dilakukan, sistem menjalankan `Navigator.pop(context)` untuk menutup dialog konfirmasi penghapusan dan proses program berakhir.

d. *Flowgraph* Fungsi Hapus Latihan



Sumber : data diolah oleh penulis (2026)

Gambar 5. 38 *Flowgraph* Hapus Latihan

Flowgraph pada fungsi hapus latihan merepresentasikan alur eksekusi program berdasarkan hasil identifikasi *Statement* yang telah dilakukan sebelumnya. *Flowgraph* yang dibentuk terdiri dari 5 node dan 4 edge, dengan penambahan Node 5 sebagai terminal node (end process) yang berfungsi untuk menandai berakhirnya seluruh alur eksekusi program.

Node pertama (Node 1) merepresentasikan *Statement* S1 yang mencakup proses akses collection latihan_soal pada *Firebase Firestore*. Selanjutnya, alur

program menuju Node 2 yang berisi *Statement* S2 untuk mengambil document berdasarkan docId dari data latihan yang dipilih pengguna.

Setelah document berhasil diperoleh, alur program dilanjutkan menuju Node 3 yang merepresentasikan *Statement* S3 untuk menjalankan proses `.delete()` sehingga data latihan dihapus dari database *Firestore*. Selanjutnya, sistem menuju Node 4 yang berisi *Statement* S4 (`Navigator.pop(context)`) untuk menutup dialog konfirmasi penghapusan data.

Seluruh alur eksekusi program tersebut kemudian bermuara pada Node 5 sebagai node akhir (end process) yang menandai bahwa proses hapus latihan telah selesai dijalankan.

e. *Cyclomatic Complexity* Fungsi Hapus Latihan

Berdasarkan *Flowgraph* yang telah dibentuk, fungsi hapus latihan tidak memiliki predicate node (P) atau percabangan kondisi karena seluruh alur program berjalan secara linear dari proses akses collection hingga proses penutupan dialog konfirmasi. Oleh karena itu, nilai *Cyclomatic Complexity* dapat dihitung menggunakan rumus berikut:

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

$$V(G) = 0 + 1$$

$$V(G) = 1$$

Dari hasil perhitungan tersebut diperoleh nilai *Cyclomatic Complexity* sebesar 1. Nilai tersebut menunjukkan bahwa fungsi hapus latihan hanya memiliki satu jalur independen (*independent path*) karena seluruh proses eksekusi program berjalan tanpa adanya percabangan kondisi atau keputusan logika. Dengan demikian, pengujian whitebox pada fungsi hapus latihan cukup dilakukan pada satu jalur pengujian untuk mencakup seluruh alur eksekusi program.

f. *Independen Path* Fungsi Hapus Latihan

Berdasarkan hasil perhitungan *Cyclomatic Complexity* yang diperoleh dari *Flowgraph* fungsi hapus latihan, nilai tersebut digunakan sebagai dasar untuk menentukan jumlah jalur independen yang harus diuji dalam proses whitebox testing. Karena fungsi hapus latihan tidak memiliki percabangan kondisi dan

seluruh alur program berjalan secara linear, maka hanya terdapat satu jalur independen yang merepresentasikan seluruh proses eksekusi program. Jalur independen tersebut meliputi proses akses collection *Firebase Firestore*, pengambilan document berdasarkan docId, proses penghapusan data latihan, hingga penutupan dialog konfirmasi penghapusan. Dengan demikian diperoleh satu jalur independen (*independent path*) sebagai berikut:

1. Path 1:

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

(Proses hapus latihan berhasil dilakukan hingga dialog konfirmasi ditutup)

g. *Test Case* Fungsi Hapus Latihan

Pengujian dilakukan berdasarkan jalur independen (*independent path*) yang telah diperoleh dari hasil analisis *Flowgraph* fungsi hapus latihan. Karena fungsi hapus latihan hanya memiliki satu jalur independen tanpa percabangan kondisi, maka pengujian dilakukan untuk memastikan seluruh proses penghapusan data latihan berjalan dengan baik mulai dari akses collection *Firebase Firestore* hingga penutupan dialog konfirmasi penghapusan.

Tabel 5. 12 *Test Case* Fungsi Hapus Latihan

Test Case	Kondisi Data	Jalur Eksekusi	Output
TC1	Data latihan tersedia	$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$	Data latihan berhasil dihapus dan dialog konfirmasi ditutup

Sumber : data diolah oleh penulis (2026)

h. *Statement Coverage* Fungsi Hapus Latihan

Statement Coverage digunakan untuk mengukur sejauh mana seluruh *Statement* pada fungsi hapus latihan 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

Berdasarkan hasil pengujian yang telah dilakukan, seluruh *Statement* pada fungsi hapus latihan yaitu S1 hingga S4 berhasil dieksekusi melalui *Test Case* TC1. Dengan demikian diperoleh nilai *Statement Coverage* sebesar:

$$\frac{4}{4} \times 100\% = 100\%$$

Nilai tersebut menunjukkan bahwa seluruh *Statement* pada fungsi hapus latihan telah diuji dan seluruh alur eksekusi program berhasil dicakup dalam proses pengujian whitebox.

5.4.7 Pengujian Fungsi Tambah Soal

a. Potongan *Source Code* Fungsi Tambah Soal

Fungsi tambah soal digunakan untuk menambahkan data soal baru ke dalam *Firebase Firestore*. Proses yang dilakukan meliputi validasi input soal, penyusunan data soal beserta pilihan jawaban, hingga proses penyimpanan data ke database. Berikut merupakan potongan source code fungsi tambah soal yang digunakan pada tahap pengujian:

```
onPressed: () async {
  if (_soalController
    .text
    .isEmpty) {
    return;
  }

  final data = {
    'soal':
      _soalController.text,
    'A':
      _aController.text,
    'B':
      _bController.text,
```

```

'C':
    _cController.text,
'D':
    _dController.text,
'jawaban':
    _jawabanBenar,
'created':
    FieldValue
        .serverTimestamp(),
};

if (isEdit) {
    await soalRef
        .doc(soalList[
            index]['id'])
        .update(data);
} else {
    await soalRef
        .add(data);
}

Navigator.pop(
    context);
},

```

b. Identifikasi *Statement* Fungsi Tambah Soal

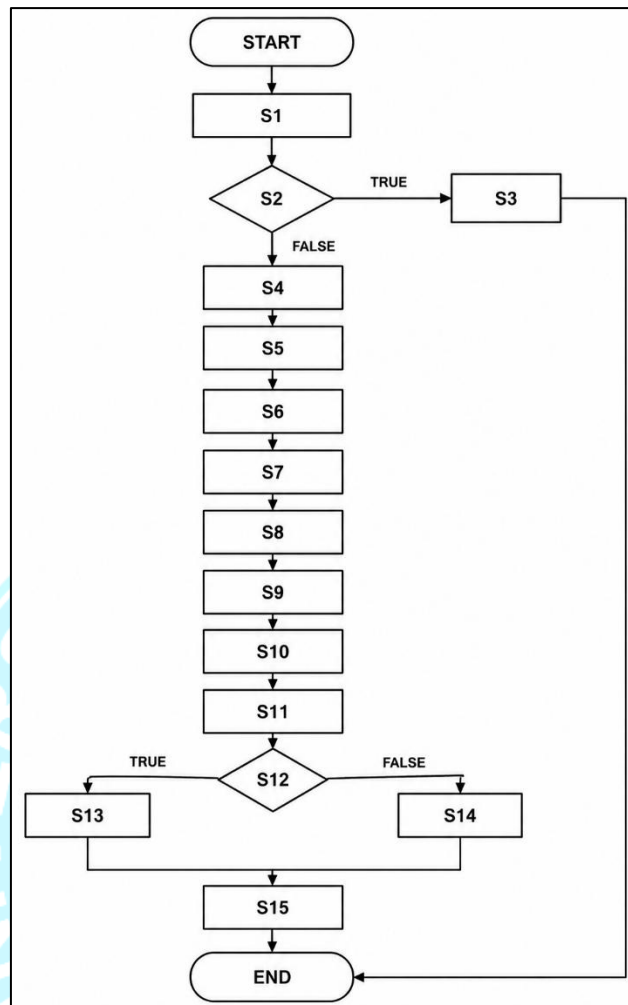
Tabel 5. 13 Identifikasi *Statement* Fungsi Tambah Soal

<i>Statement</i>	Source Code	Keterangan
S1	<code>_soalController.text</code>	Mengambil input soal

S2	if (_soalController.text.isEmpty)	Memeriksa apakah input soal kosong
S3	return;	Menghentikan proses jika input kosong
S4	final data = {...}	Membuat data soal
S5	'soal': _soalController.text	Menyimpan pertanyaan soal
S6	'A': _aController.text	Menyimpan pilihan A
S7	'B': _bController.text	Menyimpan pilihan B
S8	'C': _cController.text	Menyimpan pilihan C
S9	'D': _dController.text	Menyimpan pilihan D
S10	'jawaban': _jawabanBenar	Menyimpan jawaban benar
S11	'created': FieldValue.serverTimestamp()	Menyimpan waktu pembuatan
S12	if (isEdit)	Memeriksa mode edit atau tambah
S13	.update(data)	Memperbarui data soal
S14	.add(data)	Menambahkan data soal
S15	Navigator.pop(context)	Menutup form tambah soal

Sumber : data diolah oleh penulis(2026)

c. *Flowchart Fungsi Tambah Soal*



Sumber : data diolah oleh penulis (2026)

Gambar 5. 39 *Flowchart Fungsi Tambah Soal*

Berdasarkan diagram alur pada *Flowchart Fungsi Tambah Soal*, urutan logika kerja fungsi tambah soal dapat dijabarkan sebagai berikut:

1. S1 dan S2 (Input dan Validasi Soal):

Sistem memulai proses dengan mengambil input soal dari pengguna melalui `_soalController`. Selanjutnya sistem melakukan pengecekan apakah input soal kosong atau tidak.

2. S3 (Input Kosong):

Jika hasil pengecekan pada S2 bernilai *True*, maka sistem menjalankan return sehingga proses tambah soal dihentikan dan data tidak disimpan.

3. S4 hingga S11 (Penyusunan Data Soal):

Jika input soal tidak kosong (*False*), sistem melanjutkan proses dengan membuat object data yang berisi pertanyaan soal, pilihan jawaban A, B, C, D, jawaban benar, serta timestamp created.

4. S12 (Pengecekan Mode Form):

Setelah data soal berhasil disusun, sistem memeriksa kondisi *isEdit* untuk menentukan apakah proses yang dilakukan merupakan edit soal atau tambah soal baru.

5. S13 (Update Data Soal):

Jika kondisi *isEdit* bernilai *True*, maka sistem menjalankan proses `.update(data)` untuk memperbarui data soal yang sudah ada pada *Firebase Firestore*.

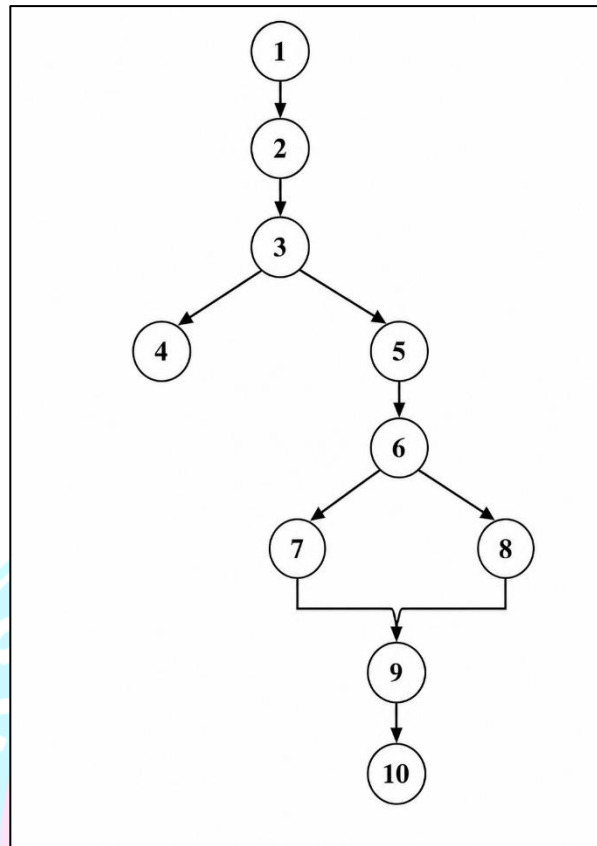
6. S14 (Tambah Data Soal):

Jika kondisi *isEdit* bernilai *False*, maka sistem menjalankan proses `.add(data)` untuk menambahkan data soal baru ke *Firebase Firestore*.

7. S15 (Menutup Form):

Setelah proses update atau tambah data berhasil dilakukan, sistem menjalankan `Navigator.pop(context)` untuk menutup form tambah soal dan proses program berakhir.

d. *Flowgraph* Fungsi Tambah Soal



Sumber : data diolah oleh penulis (2026)

Gambar 5. 40 *Flowgraph* Fungsi Tambah Soal

Flowgraph pada fungsi tambah soal ini merepresentasikan alur eksekusi program berdasarkan hasil pengelompokan *Statement* yang telah diidentifikasi sebelumnya. *Flowgraph* yang dibentuk terdiri dari 10 node dan 11 edge, dengan penambahan Node 10 sebagai terminal node (end process) yang berfungsi untuk menandai berakhirnya seluruh alur eksekusi program. Node pertama (Node 1) merepresentasikan awal proses program dimulai. Selanjutnya, Node 2 mencakup proses pengambilan input soal dari pengguna melalui controller yang digunakan pada form tambah soal.

Node 3 merepresentasikan proses validasi input untuk memeriksa apakah data soal yang dimasukkan kosong atau tidak. Apabila kondisi bernilai *True* atau input kosong, maka alur program akan menuju Node 4 yang berisi proses return atau penghentian program sehingga data tidak disimpan. Namun apabila kondisi bernilai *False* atau input tidak kosong, maka alur dilanjutkan ke Node 5 yang

berisi proses penyusunan data soal, meliputi pertanyaan, pilihan jawaban A, B, C, D, jawaban benar, dan timestamp created.

Setelah data berhasil disusun, alur program menuju Node 6 yang merupakan titik keputusan untuk memeriksa kondisi *isEdit*. Kondisi ini digunakan untuk menentukan apakah proses yang dilakukan merupakan update data soal atau penambahan data soal baru. Apabila kondisi bernilai *True*, maka alur program akan menuju Node 7 yang menjalankan proses *update(data)* untuk memperbarui data soal pada *Firebase Firestore*. Sebaliknya, apabila kondisi bernilai *False*, maka alur program akan menuju Node 8 yang menjalankan proses *add(data)* untuk menambahkan data soal baru ke *Firebase Firestore*.

Baik proses update maupun tambah data selanjutnya akan menuju Node 9 yang menjalankan *Navigator.pop(context)* untuk menutup form tambah soal. Setelah seluruh proses selesai dijalankan, alur program akan berakhir pada Node 10 sebagai terminal node (end process).

e. *Cyclomatic Complexity* Fungsi Tambah Soal

Perhitungan *Cyclomatic Complexity* pada fungsi tambah soal dilakukan untuk mengetahui tingkat kompleksitas logika program berdasarkan jumlah keputusan (decision point) yang terdapat pada *Flowgraph*. Berdasarkan *Flowgraph* yang telah dibentuk, diketahui bahwa fungsi tambah soal memiliki:

Jumlah Node (N) = 10

Jumlah Edge (E) = 11

Jumlah Predicate Node (P) = 2

Sehingga perhitungan *Cyclomatic Complexity* menggunakan rumus:

$$V(G) = E - N + 2$$

$$V(G) = 11 - 10 + 2 = 3$$

Hasil perhitungan menunjukkan bahwa nilai *Cyclomatic Complexity* pada fungsi tambah soal adalah 3. Nilai tersebut menunjukkan bahwa terdapat 3 jalur independen (*independent path*) yang dapat dilalui pada proses eksekusi program. Jalur independen tersebut terdiri dari:

1. Jalur ketika input soal kosong sehingga proses dihentikan.

2. Jalur ketika data soal berhasil diperbarui (update data).
3. Jalur ketika data soal berhasil ditambahkan sebagai data baru.

Dengan nilai *Cyclomatic Complexity* sebesar 3, maka fungsi tambah soal termasuk dalam kategori kompleksitas rendah sehingga logika program masih mudah dipahami, diuji, dan dipelihara.

f. *Independen Path* Fungsi Tambah Soal

Berdasarkan hasil perhitungan *Cyclomatic Complexity* yang diperoleh dari *Flowgraph* fungsi tambah soal, nilai tersebut digunakan sebagai dasar untuk menentukan jumlah jalur independen yang harus diuji dalam proses whitebox testing. Karena fungsi tambah soal memiliki percabangan kondisi pada proses validasi input soal dan pengecekan kondisi *isEdit*, maka terdapat beberapa jalur independen yang merepresentasikan kemungkinan alur eksekusi program yang berbeda. Jalur independen tersebut meliputi proses validasi input soal, penyusunan data soal, proses update data, serta proses penambahan data baru ke *Firebase Firestore*. Dengan demikian diperoleh tiga jalur independen (*independent path*) sebagai berikut:

1. Path 1:

1 → 2 → 3 → 4 → 10

(Input soal kosong sehingga proses dihentikan dan data tidak disimpan)

2. Path 2:

1 → 2 → 3 → 5 → 6 → 7 → 9 → 10

(Input soal valid dan kondisi *isEdit* bernilai *True* sehingga sistem melakukan update data soal)

3. Path 3:

1 → 2 → 3 → 5 → 6 → 8 → 9 → 10

(Input soal valid dan kondisi *isEdit* bernilai *False* sehingga sistem menambahkan data soal baru)

g. *Test Case* Fungsi Tambah Soal

Berdasarkan *independent path* yang telah diperoleh pada fungsi tambah soal, maka disusun beberapa *Test Case* untuk menguji seluruh kemungkinan alur eksekusi program. Pengujian dilakukan untuk memastikan proses validasi input, update data, dan penambahan data soal berjalan sesuai dengan logika program yang telah dirancang. Adapun *Test Case* fungsi tambah soal adalah sebagai berikut:

Tabel 5. 14 *Test Case* Fungsi Tambah Soal

No	<i>Independent path</i>	Skenario Pengujian	Input	Output yang Diharapkan	Hasil
1	1 → 2 → 3 → 4 → 10	Menguji kondisi ketika input soal kosong	Field soal kosong	Sistem menjalankan return dan data tidak disimpan	Berhasil
2	1 → 2 → 3 → 5 → 6 → 7 → 9 → 10	Menguji proses update data soal	Input soal valid dan isEdit = <i>True</i>	Data soal berhasil diperbarui pada <i>Firestore</i> dan form tertutup	Berhasil
3	1 → 2 → 3 → 5 → 6 → 8 → 9 → 10	Menguji proses tambah data soal baru	Input soal valid dan isEdit = <i>False</i>	Data soal baru berhasil ditambahkan ke <i>Firestore</i> dan form tertutup	Berhasil

Sumber : data diolah oleh penulis (2026)

h. *Statement Coverage* Fungsi Tambah Soal

Statement Coverage digunakan untuk mengukur sejauh mana seluruh *Statement* pada fungsi tambah soal 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:

TC1:

S1, S2, S3

TC2:

S1, S2, S4, S5, S6, S7, S8, S9, S10, S11, S12, S13, S15

TC3:

S1, S2, S4, S5, S6, S7, S8, S9, S10, S11, S12, S14, S15

Berdasarkan hasil pengujian yang telah dilakukan, seluruh *Statement* pada fungsi tambah soal yaitu S1 hingga S15 berhasil dieksekusi melalui *Test Case* TC1, TC2, dan TC3. Dengan demikian diperoleh nilai *Statement Coverage* sebesar:

$$\frac{15}{15} \times 100\% = 100\%$$

Nilai tersebut menunjukkan bahwa seluruh *Statement* pada fungsi tambah soal telah diuji dan seluruh alur eksekusi program berhasil dicakup dalam proses pengujian whitebox.

5.4.8 Pengujian Fungsi Hapus Soal

a. Potongan *Source Code* Fungsi Hapus Soal

Fungsi hapus soal digunakan untuk menghapus data soal yang terdapat pada *Firestore*. Proses diawali dengan menampilkan dialog konfirmasi kepada pengguna sebelum data dihapus. Jika pengguna memilih tombol “Ya”, maka sistem akan menjalankan proses penghapusan data soal berdasarkan docId yang dipilih. Setelah data berhasil dihapus, dialog konfirmasi akan ditutup menggunakan `Navigator.pop(context)`. Berikut adalah potongan source code fungsi hapus soal yang digunakan dalam tahap pengujian:

```
void _hapusSoal(int i) {
  final t = text[widget.language]!;
  showDialog(
    context: context,

    builder: (_) => AlertDialog(
      title: Text(t['hapus']!),
      actions: [
```

```

IconButton(
  onPressed: () =>
    Navigator.pop(context),
  child: Text(t['tidak']!),
),
ElevatedButton(
  onPressed: () async {
    await soalRef
      .doc(soalList[i]['id'])
      .delete();
    Navigator.pop(context);
  },
  child: Text(t['ya']!),
),
),
);

```

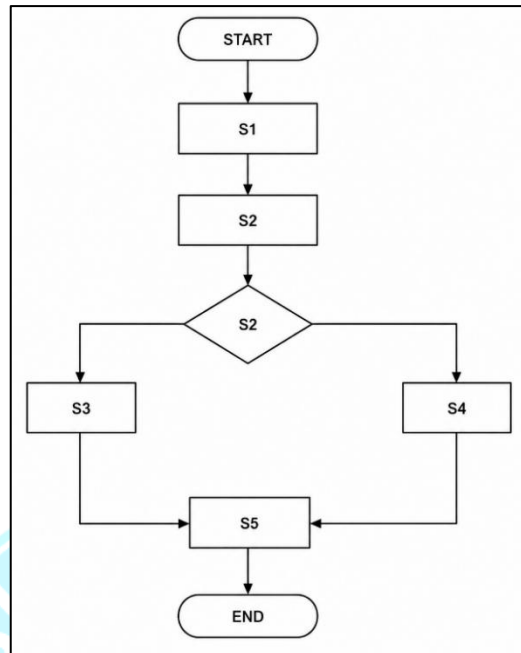
b. Identifikasi *Statement* Fungsi Hapus Soal

Tabel 5. 15 Identifikasi *Statement* Fungsi Hapus Soal

<i>Statement</i>	Source Code	Keterangan
S1	showDialog(...)	Menampilkan dialog konfirmasi hapus
S2	IconButton(...)	Tombol batal penghapusan
S3	Navigator.pop(context)	Menutup dialog tanpa menghapus data
S4	await soalRef.doc(...).delete()	Menghapus data soal dari Firestore
S5	Navigator.pop(context)	Menutup dialog setelah hapus berhasil

Sumber : data diolah oleh penulis (2026)

c. *Flowchart* Fungsi Hapus Soal



Sumber : data diolah oleh penulis (2026)

Gambar 5. 41 *Flowchart* Fungsi Hapus Soal

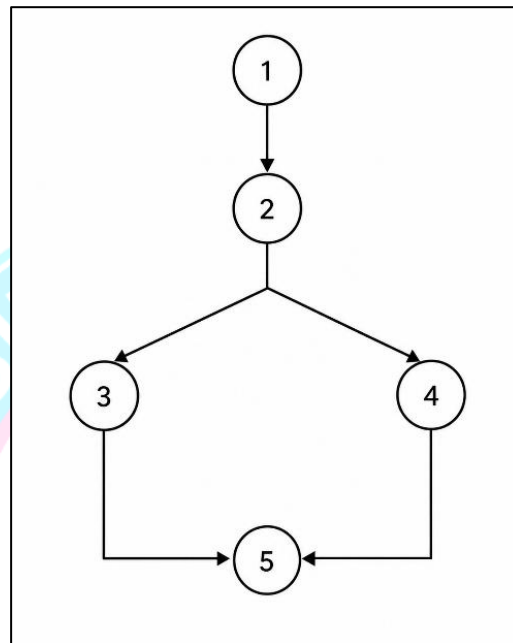
Berdasarkan diagram alir pada *Flowchart* Fungsi Hapus Soal, urutan logika kerja fungsi hapus soal dapat dijabarkan sebagai berikut:

1. S1(Menampilkan Dialog Konfirmasi):
Sistem memulai proses dengan menampilkan dialog konfirmasi penghapusan soal kepada pengguna.
2. S2 (Pilihan Pengguna):
Sistem menunggu pilihan pengguna apakah ingin membatalkan atau melanjutkan proses penghapusan data soal.
3. S3 (Batal Hapus):
Jika pengguna memilih tombol “Tidak”, maka sistem menjalankan `Navigator.pop(context)` untuk menutup dialog dan proses program berakhir tanpa menghapus data.
4. S4 (Hapus Data Soal):
Jika pengguna memilih tombol “Ya”, maka sistem menjalankan proses `.delete()` untuk menghapus data soal pada *Firestore* berdasarkan `docId` yang dipilih.

5. S5 (Menutup Dialog):

Setelah proses penghapusan berhasil dilakukan, sistem menjalankan `Navigator.pop(context)` untuk menutup dialog konfirmasi dan proses program berakhir.

d. *Flowgraph* Fungsi Hapus Soal



Sumber : data diolah oleh penulis (2026)

Gambar 5. 42 *Flowgraph* Fungsi Hapus Soal

Flowgraph pada fungsi hapus soal ini merepresentasikan alur eksekusi program berdasarkan hasil pengelompokan *Statement* yang telah diidentifikasi sebelumnya. *Flowgraph* yang dibentuk terdiri dari 5 node dan 5 edge, dengan penambahan Node 5 sebagai terminal node (end process) yang berfungsi untuk menandai berakhirnya seluruh alur eksekusi program.

Node pertama (Node 1) terdiri dari S1 yang mencakup proses penampilan dialog konfirmasi penghapusan soal. Selanjutnya, Node 2 merepresentasikan proses pilihan pengguna pada dialog konfirmasi. Apabila pengguna memilih tombol “Tidak”, maka alur program akan menuju Node 3 (S3) yang menjalankan `Navigator.pop(context)` untuk menutup dialog tanpa menghapus data.

Namun apabila pengguna memilih tombol “Ya”, maka alur program akan menuju Node 4 yang berisi proses `.delete()` untuk menghapus data soal pada

Firebase Firestore serta menjalankan `Navigator.pop(context)` setelah proses penghapusan selesai dilakukan. Seluruh alur program kemudian bermuara pada Node 5 sebagai node akhir (end process).

e. *Cyclomatic Complexity* Fungsi Hapus Soal

Berdasarkan *Flowgraph* yang telah dibentuk, terdapat 1 predicate node (P), yaitu pada Node 2 yang melakukan pengecekan pilihan pengguna pada dialog konfirmasi hapus soal. Oleh karena itu, nilai *Cyclomatic Complexity* dapat dihitung menggunakan rumus berikut:

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

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

Dari perhitungan tersebut diperoleh nilai *Cyclomatic Complexity* sebesar 2. Nilai tersebut menunjukkan bahwa terdapat dua jalur independen dalam proses eksekusi fungsi hapus soal, yang merepresentasikan seluruh kemungkinan alur logika berdasarkan pilihan pengguna pada dialog konfirmasi penghapusan data.

f. *Independen Path* Fungsi Hapus Soal

Berdasarkan hasil perhitungan *Cyclomatic Complexity* yang diperoleh dari *Flowgraph* fungsi hapus soal, nilai tersebut digunakan sebagai dasar untuk menentukan jumlah jalur independen yang harus diuji dalam proses whitebox testing. Karena fungsi hapus soal memiliki percabangan kondisi pada pilihan pengguna di dialog konfirmasi, maka terdapat dua jalur independen yang merepresentasikan kemungkinan alur eksekusi program yang berbeda. Dengan demikian diperoleh 2 jalur independen (*independent path*) berikut ini:

1. Path 1:

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 5$$

(Pengguna membatalkan proses penghapusan soal)

2. Path 2:

$$1 \rightarrow 2 \rightarrow 4 \rightarrow 5$$

(Proses hapus soal berhasil dilakukan hingga dialog konfirmasi ditutup)

g. *Test Case* Fungsi Hapus Soal

Pengujian dilakukan berdasarkan jalur independen (*independent path*) yang telah diperoleh dari hasil analisis *Flowgraph* fungsi hapus soal. Setiap jalur independen direpresentasikan ke dalam beberapa *Test Case* dengan kombinasi pilihan pengguna 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 hapus soal secara menyeluruh.

Tabel 5. 16 *Test Case* Fungsi Hapus Soal

Test Case	Aksi Pengguna	Jalur Eksekusi	Output
TC1	Memilih tombol “Tidak”	1 → 2 → 3 → 5	Dialog tertutup dan data soal tidak dihapus
TC2	Memilih tombol “Ya”	1 → 2 → 4 → 5	Data soal berhasil dihapus dari <i>Firebase Firestore</i>

Sumber : data diolah oleh penulis (2026)

h. *Statement Coverage* Fungsi Hapus Soal

Statement Coverage digunakan untuk mengukur sejauh mana seluruh *Statement* pada fungsi hapus soal 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
2. TC2:
S1, S2, S4, S5

Berdasarkan hasil pengujian yang telah dilakukan, seluruh *Statement* pada fungsi hapus soal yaitu S1 hingga S5 berhasil dieksekusi melalui kombinasi *Test Case* TC1 dan TC2. Dengan demikian diperoleh nilai *Statement Coverage* sebesar:

$$\frac{5}{5} \times 100\% = 100\%$$

Nilai tersebut menunjukkan bahwa seluruh *Statement* pada fungsi hapus soal telah diuji dan seluruh kemungkinan alur eksekusi program berhasil dicakup dalam proses pengujian whitebox.

5.4.9 Pengujian Fungsi Next

a. Potongan *Source Code* Fungsi Next

Fungsi next digunakan untuk melanjutkan proses pengerjaan soal ke soal berikutnya pada halaman latihan soal. Sistem akan memeriksa apakah masih terdapat soal berikutnya yang dapat ditampilkan. Jika masih tersedia, maka index soal akan ditambah sehingga pengguna berpindah ke soal berikutnya. Namun apabila soal yang sedang dikerjakan merupakan soal terakhir, maka sistem akan menampilkan hasil latihan dan mengakhiri proses pengerjaan soal. Berikut adalah potongan *source code* fungsi next yang digunakan dalam tahap pengujian:

```
void nextQuestion() {
    if (currentIndex < soalList.length - 1) {
        setState(() {
            currentIndex++;
        });
    } else {
        tampilHasil();
    }
}
```

b. Identifikasi *Statement* Fungsi Next

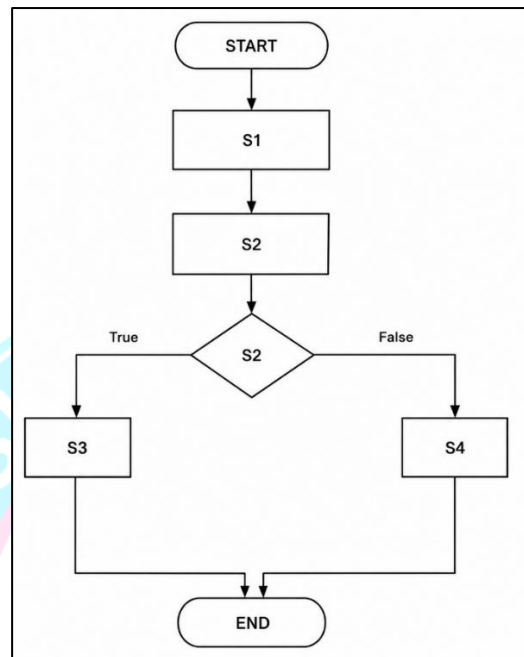
Tabel 5. 17 Identifikasi *Statement* Fungsi Next

<i>Statement</i>	<i>Source Code</i>	<i>Keterangan</i>
S1	nextQuestion()	Menjalankan fungsi next
S2	if(currentIndex < soalList.length - 1)	Pengecekan apakah masih terdapat soal berikutnya
S3	currentIndex++	Berpindah ke soal berikutnya

S4	tampilHasil()	Menampilkan latihan	hasil
----	---------------	------------------------	-------

Sumber : data diolah oleh penulis (2026)

c. *Flowchart* Fungsi Next



Sumber : data diolah oleh penulis (2026)

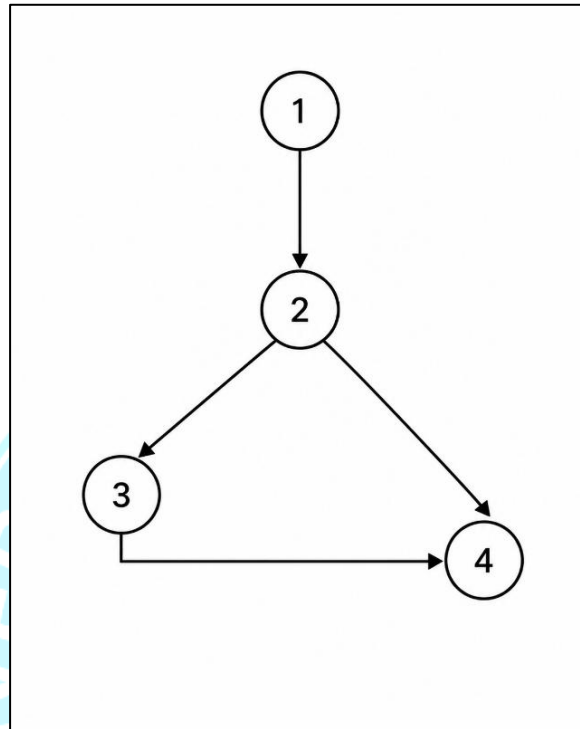
Gambar 5. 43 *Flowchart* Fungsi Next

Berdasarkan diagram alur pada *Flowchart* Fungsi Next, urutan logika kerja fungsi next dapat dijabarkan sebagai berikut:

1. S1 (Menjalankan Fungsi Next):
Sistem memulai proses dengan menjalankan fungsi next untuk melanjutkan pengerjaan soal.
2. S2 (Pengecekan Soal Berikutnya):
Sistem melakukan pengecekan apakah masih terdapat soal berikutnya berdasarkan nilai `currentIndex` dan jumlah soal yang tersedia.
3. S3 (Menampilkan Soal Berikutnya):
Jika kondisi bernilai *True*, maka sistem menjalankan proses `currentIndex++` sehingga soal berikutnya ditampilkan kepada pengguna.
4. S4 (Menampilkan Hasil Latihan):
Jika kondisi bernilai *False*, maka sistem menjalankan fungsi `tampilHasil()`

untuk menampilkan hasil latihan karena pengguna telah mencapai soal terakhir.

d. *Flowgraph* Fungsi Next



Sumber : data diolah oleh penulis (2026)

Gambar 5. 44 *Flowgraph* Fungsi Next

Flowgraph pada fungsi next ini merepresentasikan alur eksekusi program berdasarkan hasil pengelompokan *Statement* yang telah diidentifikasi sebelumnya. *Flowgraph* yang dibentuk terdiri dari 4 node dan 4 edge, dengan penambahan Node 4 sebagai terminal node (end process) yang berfungsi untuk menandai berakhirnya seluruh alur eksekusi program.

Node pertama (Node 1) terdiri dari S1 yang mencakup proses menjalankan fungsi next. Selanjutnya, Node 2 merepresentasikan proses pengecekan apakah masih terdapat soal berikutnya atau tidak berdasarkan nilai *currentIndex*.

Apabila kondisi bernilai *True*, maka alur program akan menuju Node 3 (S3) yang menjalankan proses perpindahan ke soal berikutnya menggunakan

currentIndex++. Namun apabila kondisi bernilai *False*, maka alur program akan langsung menuju Node 4 (S4) yang menjalankan fungsi tampilHasil() untuk menampilkan hasil latihan dan mengakhiri proses program. Seluruh alur program kemudian bermuara pada Node 4 sebagai node akhir (end process).

e. *Cyclomatic Complexity* Fungsi Next

Berdasarkan *Flowgraph* yang telah dibentuk, terdapat 1 predicate node (P), yaitu pada Node 2 yang melakukan pengecekan kondisi soal berikutnya. Oleh karena itu, nilai *Cyclomatic Complexity* dapat dihitung menggunakan rumus berikut:

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

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

Dari perhitungan tersebut diperoleh nilai *Cyclomatic Complexity* sebesar 2. Nilai tersebut menunjukkan bahwa terdapat dua jalur independen dalam proses eksekusi fungsi next yang merepresentasikan seluruh kemungkinan alur logika program.

f. *Independen Path* Fungsi Next

Berdasarkan hasil perhitungan *Cyclomatic Complexity* yang diperoleh dari *Flowgraph* fungsi next, nilai tersebut digunakan sebagai dasar untuk menentukan jumlah jalur independen yang harus diuji dalam proses whitebox testing. Karena fungsi next memiliki percabangan kondisi pada proses pengecekan soal berikutnya, maka terdapat dua jalur independen yang merepresentasikan kemungkinan alur eksekusi program yang berbeda. Dengan demikian diperoleh 2 jalur independen (*independent path*) berikut ini:

1. Path 1:

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 4$$

(Proses perpindahan ke soal berikutnya berhasil dilakukan)

2. Path 2:

$$1 \rightarrow 2 \rightarrow 4$$

(Pengguna telah mencapai soal terakhir sehingga sistem menampilkan hasil latihan)

g. *Test Case* Fungsi Next

Pengujian dilakukan berdasarkan jalur independen (*independent path*) yang telah diperoleh dari hasil analisis *Flowgraph* fungsi next. Setiap jalur independen direpresentasikan ke dalam beberapa *Test Case* dengan kondisi 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 next secara menyeluruh.

Tabel 5. 18 *Test Case* Fungsi Next

Test Case	Aksi Pengguna	Jalur Eksekusi	Output
TC1	Menekan tombol next saat masih terdapat soal berikutnya	1 → 2 → 3 → 4	Sistem menampilkan soal berikutnya
TC2	Menekan tombol next pada soal terakhir	1 → 2 → 4	Sistem menampilkan hasil Latihan

Sumber : data diolah oleh penulis (2026)

h. *Statement Coverage* Fungsi Next

Statement Coverage digunakan untuk mengukur sejauh mana seluruh *Statement* pada fungsi next 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
2. TC2:
S1, S2, S4

Berdasarkan hasil pengujian yang telah dilakukan, seluruh *Statement* pada fungsi next yaitu S1 hingga S4 berhasil dieksekusi melalui kombinasi *Test Case* TC1 dan TC2. Dengan demikian diperoleh nilai *Statement Coverage* sebesar:

$$\frac{4}{4} \times 100\% = 100\%$$

Nilai tersebut menunjukkan bahwa seluruh *Statement* pada fungsi next telah diuji dan seluruh kemungkinan alur eksekusi program berhasil dicakup dalam proses pengujian whitebox.

5.4.10 Pengujian Fungsi Hitung Nilai

a. Potongan *Source Code* Fungsi Hitung Nilai

Fungsi hitung nilai digunakan untuk menghitung persentase nilai siswa berdasarkan jumlah jawaban benar dan total soal yang tersedia. Proses dilakukan dengan membagi jumlah jawaban benar dengan total soal, kemudian hasilnya dikalikan 100 dan dibulatkan menggunakan fungsi `.round()`. Berikut adalah potongan source code fungsi hitung nilai yang digunakan dalam tahap pengujian:

```
int hitungNilai(int benar, int total) { return ((benar / total) * 100).round();
```

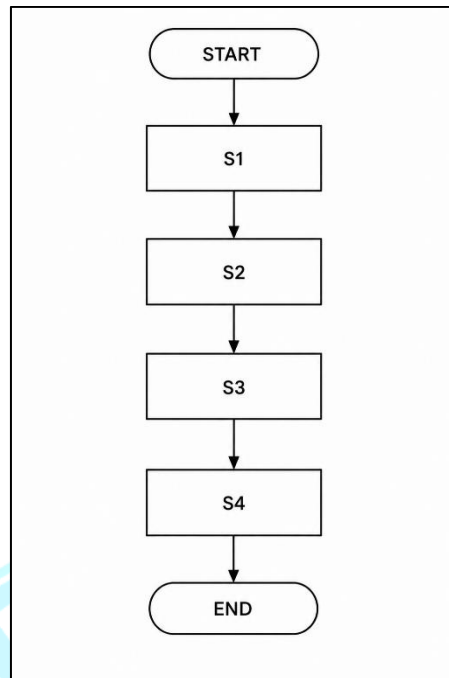
b. Identifikasi *Statement* Fungsi Hitung Nilai

Tabel 5. 19 Identifikasi *Statement* Fungsi Hitung Nilai

<i>Statement</i>	<i>Source Code</i>	<i>Keterangan</i>
S1	hitungNilai(int benar, int total)	Menjalankan fungsi hitung nilai
S2	(benar / total)	Menghitung rasio jawaban benar
S3	* 100	Mengubah hasil menjadi persentase
S4	.round()	Membulatkan hasil nilai

Sumber : data diolah oleh penulis (2026)

c. *Flowchart* Fungsi Hitung Nilai



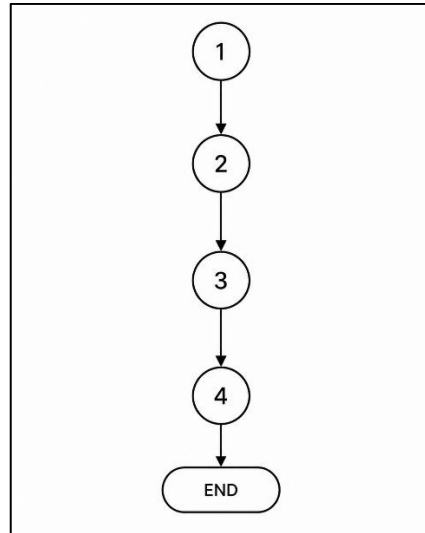
Sumber : data diolah oleh penulis (2026)

Gambar 5. 45 *Flowchart* Fungsi Hitung Nilai

Berdasarkan diagram alur pada *Flowchart* Fungsi Hitung Nilai, urutan logika kerja fungsi hitung nilai dapat dijabarkan sebagai berikut:

1. S1 (Menjalankan Fungsi Hitung Nilai):
Sistem memulai proses dengan menjalankan fungsi hitung nilai menggunakan parameter jumlah jawaban benar dan total soal.
2. S2 (Menghitung Rasio Nilai):
Sistem melakukan proses pembagian jumlah jawaban benar dengan total soal untuk memperoleh rasio nilai siswa.
3. S3 (Mengubah Menjadi Persentase):
Hasil rasio kemudian dikalikan dengan 100 untuk menghasilkan nilai dalam bentuk persentase.
4. S4 (Membulatkan Nilai):
Sistem menjalankan fungsi `.round()` untuk membulatkan hasil perhitungan nilai sebelum ditampilkan kepada pengguna.

d. *Flowgraph* Fungsi Hitung Nilai



Sumber : data diolah oleh penulis (2026)

Gambar 5. 46 *Flowgraph* Fungsi Hitung Nilai

Flowgraph pada fungsi hitung nilai ini merepresentasikan alur eksekusi program berdasarkan hasil pengelompokan *Statement* yang telah diidentifikasi sebelumnya. *Flowgraph* yang dibentuk terdiri dari 4 node dan 3 edge, dengan penambahan Node 4 sebagai terminal node (end process) yang berfungsi untuk menandai berakhirnya seluruh alur eksekusi program.

Node pertama (Node 1) terdiri dari S1 yang mencakup proses menjalankan fungsi hitung nilai. Selanjutnya, Node 2 merepresentasikan proses pembagian jumlah jawaban benar dengan total soal untuk memperoleh rasio nilai siswa.

Node 3 merepresentasikan proses pengubahan hasil perhitungan menjadi persentase dengan mengalikan hasil rasio dengan angka 100. Setelah itu, program menuju Node 4 yang menjalankan fungsi `.round()` untuk membulatkan nilai dan mengakhiri proses program sebagai node akhir (end process).

e. *Cyclomatic Complexity* Fungsi Hitung Nilai

Berdasarkan *Flowgraph* yang telah dibentuk, fungsi hitung nilai tidak memiliki predicate node atau percabangan kondisi sehingga seluruh proses program berjalan secara linear. Oleh karena itu, nilai *Cyclomatic Complexity* dapat dihitung menggunakan rumus berikut:

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

$$V(G) = 0 + 1 = 1$$

Dari perhitungan tersebut diperoleh nilai *Cyclomatic Complexity* sebesar 1. Nilai tersebut menunjukkan bahwa hanya terdapat satu jalur independen dalam proses eksekusi fungsi hitung nilai.

f. *Independen Path* Fungsi Hitung Nilai

Berdasarkan hasil perhitungan *Cyclomatic Complexity* yang diperoleh dari *Flowgraph* fungsi hitung nilai, nilai tersebut digunakan sebagai dasar untuk menentukan jumlah jalur independen yang harus diuji dalam proses whitebox testing. Karena fungsi hitung nilai tidak memiliki percabangan kondisi dan seluruh alur program berjalan secara linear, maka hanya terdapat satu jalur independen yang merepresentasikan seluruh proses eksekusi program. Dengan demikian diperoleh 1 jalur independen (*independent path*) berikut ini:

1. Path 1:

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 4$$

(Proses perhitungan nilai berhasil dilakukan hingga hasil dibulatkan)

g. *Test Case* Fungsi Hitung Nilai

Pengujian dilakukan berdasarkan jalur independen (*independent path*) yang telah diperoleh dari hasil analisis *Flowgraph* fungsi hitung nilai. Jalur eksekusi pada *Test Case* diperoleh dari node-node yang dilalui pada *Flowgraph* sehingga proses pengujian dapat memverifikasi fungsi hitung nilai secara menyeluruh.

Tabel 5. 20 *Test Case* Fungsi Hitung Nilai

Test Case	Input	Jalur Eksekusi	Output
TC1	benar = 8, total = 10	$1 \rightarrow 2 \rightarrow 3$ $\rightarrow 4$	Nilai = 80
TC2	benar = 10, total = 10	$1 \rightarrow 2 \rightarrow 3$ $\rightarrow 4$	Nilai = 100
TC3	benar = 7, total = 9	$1 \rightarrow 2 \rightarrow 3$ $\rightarrow 4$	Nilai = 78

Sumber : data diolah oleh penulis (2026)

h. *Statement Coverage* Fungsi Hitung Nilai

Statement Coverage digunakan untuk mengukur sejauh mana seluruh *Statement* pada fungsi hitung nilai 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
2. TC2:
S1, S2, S3, S4
3. TC3:
S1, S2, S3, S4

Berdasarkan hasil pengujian yang telah dilakukan, seluruh *Statement* pada fungsi hitung nilai yaitu S1 hingga S4 berhasil dieksekusi melalui kombinasi *Test Case* TC1, TC2, dan TC3. Dengan demikian diperoleh nilai *Statement Coverage* sebesar:

$$\frac{4}{4} \times 100\% = 100\%$$

Nilai tersebut menunjukkan bahwa seluruh *Statement* pada fungsi hitung nilai telah diuji dan seluruh kemungkinan alur eksekusi program berhasil dicakup dalam proses pengujian whitebox.

5.4.11 Pengujian Fungsi *Generate PDF*

a. Potongan *Source Code* Fungsi *Generate PDF*

Fungsi *Generate PDF* digunakan untuk membuat file PDF hasil latihan siswa berdasarkan data yang diperoleh dari *Firebase Firestore*. Proses dimulai dengan melakukan sorting data nilai siswa, kemudian sistem membuat dokumen PDF yang berisi tabel hasil latihan siswa dan menampilkan file PDF

menggunakan `Printing.layoutPdf()`. Berikut adalah potongan source code fungsi *Generate PDF* yang digunakan dalam tahap pengujian.

```
void _GeneratePdf(
    String namaLatihan,
    List<Map<String, dynamic>> hasilSiswa,

) async {
    final pdf = pw.Document();

    final List<Map<String, dynamic>> sortedHasil =

    List<Map<String, dynamic>>.from(hasilSiswa)
        ..sort((a, b) {

            final nilaiA = a['nilai'];
            final nilaiB = b['nilai'];
            return nilaiB.compareTo(nilaiA);
        });

    pdf.addPage(
        pw.Page(

        build: (pw.Context context) {
            return pw.Column(
                children: [

                pw.Table.fromTextArray(
                    headers: [

                    'No',
                    'Nama',
                    'Benar',
```

```

        'Nilai'
    ],
    ),
    ],
    );
    },
    ),
);
await Printing.layoutPdf(

```

```

onLayout: (format) async =>
    pdf.save(),
);

```

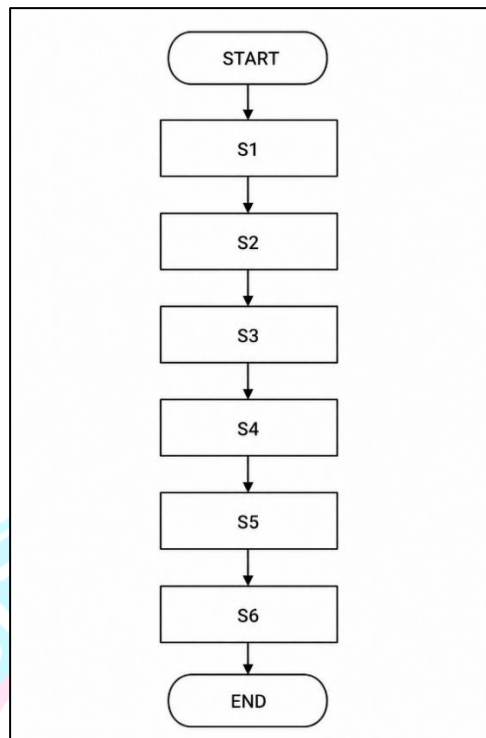
b. Identifikasi *Statement* Fungsi *Generate* PDF

Tabel 5. 21 Identifikasi *Statement* Fungsi *Generate* PDF

<i>Statement</i>	<i>Source Code</i>	<i>Keterangan</i>
S1	<code>_GeneratePdf()</code>	Menjalankan fungsi <i>Generate</i> PDF
S2	<code>pw.Document()</code>	Membuat dokumen PDF
S3	<code>sortedHasil.sort()</code>	Mengurutkan data nilai siswa
S4	<code>pdf.addPage()</code>	Menambahkan halaman PDF
S5	<code>pw.Table.fromTextArray()</code>	Membuat tabel hasil latihan
S6	<code>Printing.layoutPdf()</code>	Menampilkan atau mengunduh file PDF

Sumber : data diolah oleh penulis (2026)

c. *Flowchart Fungsi Generate PDF*



Sumber : data diolah oleh penulis (2026)

Gambar 5. 47 *Flowchart Fungsi Generate PDF*

Berdasarkan diagram alur pada *Flowchart Fungsi Generate PDF*, urutan logika kerja fungsi *Generate PDF* dapat dijabarkan sebagai berikut:

1. S1 (Menjalankan Fungsi *Generate PDF*):
Sistem memulai proses dengan menjalankan fungsi *Generate PDF* menggunakan data hasil latihan siswa.
2. S2 (Membuat Dokumen PDF):
Sistem membuat object dokumen PDF menggunakan `pw.Document()`.
3. S3 (Mengurutkan Data Nilai):
Sistem melakukan sorting data nilai siswa berdasarkan nilai tertinggi menggunakan `sort()`.
4. S4 (Menambahkan Halaman PDF):
Sistem menjalankan `pdf.addPage()` untuk membuat halaman dokumen PDF.

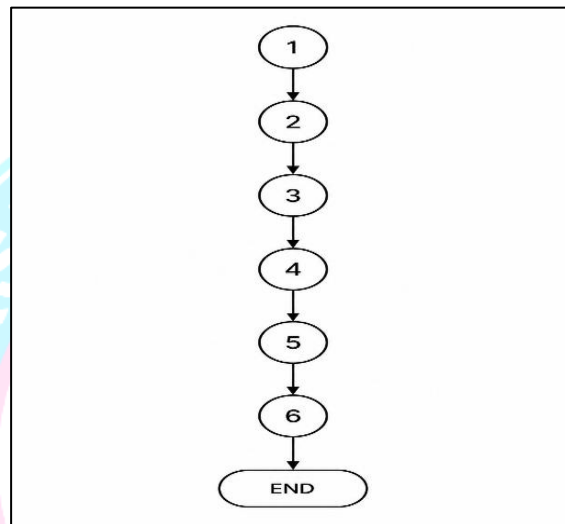
5. S5 (Membuat Tabel PDF):

Sistem membuat tabel hasil latihan siswa menggunakan `pw.Table.fromTextArray()`.

6. S6 (Menampilkan PDF):

Setelah proses pembuatan dokumen selesai, sistem menjalankan `Printing.layoutPdf()` untuk menampilkan atau mengunduh file PDF.

d. *Flowgraph Fungsi Generate PDF*



Sumber : data diolah oleh penulis (2026)

Gambar 5. 48 *Flowgraph Fungsi Generate PDF*

Flowgraph pada fungsi *Generate PDF* ini merepresentasikan alur eksekusi program berdasarkan hasil pengelompokan *Statement* yang telah diidentifikasi sebelumnya. *Flowgraph* yang dibentuk terdiri dari 6 node dan 5 edge, dengan penambahan Node 6 sebagai terminal node (end process) yang berfungsi untuk menandai berakhirnya seluruh alur eksekusi program.

Node pertama (Node 1) terdiri dari S1 yang mencakup proses menjalankan fungsi *Generate PDF*. Selanjutnya, Node 2 merepresentasikan proses pembuatan dokumen PDF menggunakan `pw.Document()`.

Node 3 merepresentasikan proses sorting data hasil latihan siswa berdasarkan nilai tertinggi menggunakan `sort()`. Setelah itu, program menuju Node 4 yang menjalankan `pdf.addPage()` untuk menambahkan halaman PDF.

Node 5 merepresentasikan proses pembuatan tabel hasil latihan menggunakan `pw.Table.fromTextArray()`. Selanjutnya program menuju Node 6 yang menjalankan `Printing.layoutPdf()` untuk menampilkan atau mengunduh file PDF sekaligus mengakhiri proses program sebagai node akhir (end process).

e. *Cyclomatic Complexity* Fungsi *Generate PDF*

Berdasarkan *Flowgraph* yang telah dibentuk, fungsi *Generate PDF* tidak memiliki predicate node atau percabangan kondisi sehingga seluruh proses program berjalan secara linear. Oleh karena itu, nilai *Cyclomatic Complexity* dapat dihitung menggunakan rumus berikut:

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

$$V(G) = 0 + 1 = 1$$

Dari perhitungan tersebut diperoleh nilai *Cyclomatic Complexity* sebesar 1. Nilai tersebut menunjukkan bahwa hanya terdapat satu jalur independen dalam proses eksekusi fungsi *Generate PDF*.

f. *Independen Path* Fungsi *Generate PDF*

Berdasarkan hasil perhitungan *Cyclomatic Complexity* yang diperoleh dari *Flowgraph* fungsi *Generate PDF*, nilai tersebut digunakan sebagai dasar untuk menentukan jumlah jalur independen yang harus diuji dalam proses whitebox testing. Karena fungsi *Generate PDF* tidak memiliki percabangan kondisi dan seluruh alur program berjalan secara linear, maka hanya terdapat satu jalur independen yang merepresentasikan seluruh proses eksekusi program. Dengan demikian diperoleh 1 jalur independen (*independent path*) berikut ini:

1. Path 1:

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

(Proses *Generate PDF* berhasil dilakukan hingga file PDF ditampilkan)

g. *Test Case Fungsi Generate PDF*

Pengujian dilakukan berdasarkan jalur independen (*independent path*) yang telah diperoleh dari hasil analisis *Flowgraph* fungsi *Generate PDF*. Jalur eksekusi pada *Test Case* diperoleh dari node-node yang dilalui pada *Flowgraph* sehingga proses pengujian dapat memverifikasi fungsi *Generate PDF* secara menyeluruh.

Tabel 5. 22 *Test Case Fungsi Generate PDF*

Test Case	Input	Jalur Eksekusi	Output
TC1	Data hasil latihan siswa tersedia	1 → 2 → 3 → 4 → 5 → 6	File PDF berhasil dibuat dan ditampilkan

Sumber : data diolah oleh penulis (2026)

h. *Statement Coverage Fungsi Generate PDF*

Statement Coverage digunakan untuk mengukur sejauh mana seluruh *Statement* pada fungsi *Generate PDF* 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

Berdasarkan hasil pengujian yang telah dilakukan, seluruh *Statement* pada fungsi *Generate PDF* yaitu S1 hingga S6 berhasil dieksekusi melalui *Test Case* TC1. Dengan demikian diperoleh nilai *Statement Coverage* sebesar:

$$\frac{6}{6} \times 100\% = 100\%$$

Nilai tersebut menunjukkan bahwa seluruh *Statement* pada fungsi *Generate PDF* telah diuji dan seluruh alur eksekusi program berhasil dicakup dalam proses pengujian whitebox.

5.4.12 Pengujian Fungsi Tambah Akun User

a. Potongan *Source Code* Fungsi Tambah Akun *User*

Fungsi tambah akun *User* digunakan untuk menambahkan akun siswa baru ke dalam database *Firebase Firestore*. Proses dimulai dengan pengambilan input *Username* dan *password* dari pengguna, kemudian sistem melakukan validasi data seperti pengecekan input kosong, panjang *password*, format *password*, serta pengecekan *Username* yang sudah digunakan. Jika seluruh validasi berhasil, maka data akun siswa akan disimpan ke *Firebase Firestore*. Berikut adalah potongan *source code* fungsi tambah akun *User* yang digunakan dalam tahap pengujian.

```
void tambahUser() {

    final UsernameController =
        TextEditingController();

    final passwordController =
        TextEditingController();

    showDialog(
        context: context,
        builder: (context) {

            return AlertDialog(
                actions: [

                    ElevatedButton(

                        onPressed: () async {

                            final Username =
                                UsernameController.text
                                    .trim();
```

```

final password =
    passwordController.text
        .trim();

if (Username.isEmpty ||
    password.isEmpty) {
    return;
}
if (password.length > 10) {
    return;
}
final regex = RegExp(
    r'^(?=.*[A-Za-z])(?=.*\d)[A-Za-z\d]+$',
);
if (!regex.hasMatch(
    password,
)) {
    return;
}
final checkUser =

await FirebaseFirestore
    .instance
    .collection('Users')

    .where(
        'Username',
        isEqualTo: Username,
    )
    .get();

```

```

    if (checkUser
      .docs
      .isEmpty) {
      return;
    }
    await FirebaseFirestore

      .instance
      .collection('Users')
      .add({

        'Username': Username,
        'password': password,
        'role': 'siswa',
      });
    Navigator.pop(context);
  },
),
],
);
},
);
}

```

b. Identifikasi *Statement* Fungsi Tambah Akun *User*

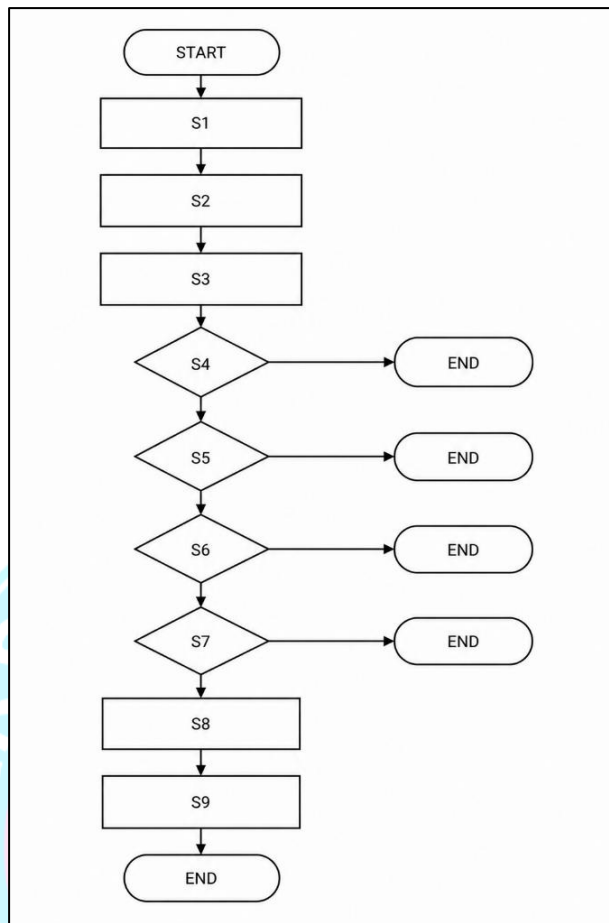
Tabel 5. 23 Identifikasi *Statement* Fungsi Tambah Akun *User*

<i>Statement</i>	Source Code	Keterangan
S1	tambahUser()	Menjalankan fungsi tambah akun <i>User</i>

S2	<code>UsernameController.text.trim()</code>	Mengambil input <i>Username</i>
S3	<code>passwordController.text.trim()</code>	Mengambil input <i>password</i>
S4	<code>if(Username.isEmpty password.isEmpty)</code>	Validasi input kosong
S5	<code>if(password.length > 10)</code>	Validasi panjang <i>password</i>
S6	<code>regex.hasMatch(password)</code>	Validasi format <i>password</i>
S7	<code>checkUser.docs.isNotEmpty</code>	Pengecekan <i>Username</i> sudah digunakan
S8	<code>FirebaseFirestore.instance.collection('Users').add()</code>	Menambahkan data <i>User</i> ke <i>Firebase</i>
S9	<code>Navigator.pop(context)</code>	Menutup dialog

Sumber : data diolah oleh penulis (2026)

c. *Flowchart Fungsi Tambah Akun User*



Sumber : data diolah oleh penulis (2026)

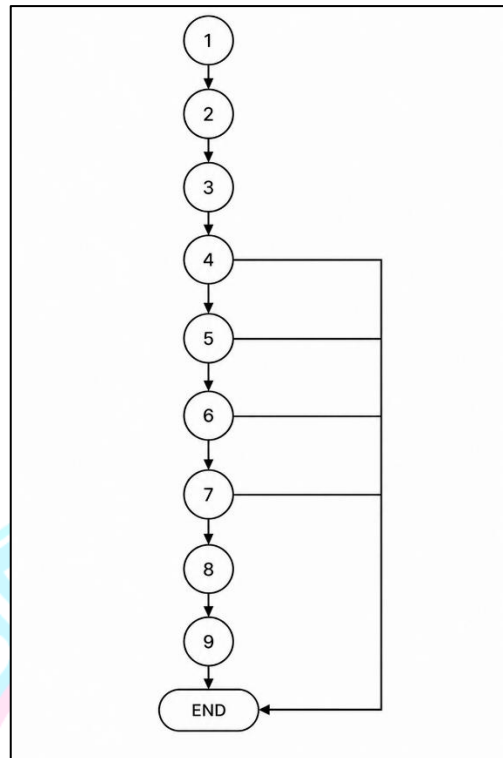
Gambar 5. 49 *Flowchart Fungsi Tambah Akun User*

Berdasarkan diagram alur pada *Flowchart Fungsi Tambah Akun User*, urutan logika kerja fungsi tambah akun *User* dapat dijabarkan sebagai berikut:

1. S1 (Menjalankan Fungsi Tambah *User*):
Sistem memulai proses dengan menjalankan fungsi tambah akun *User*.
2. S2 (Mengambil Input *Username*):
Sistem mengambil input *Username* yang dimasukkan pengguna melalui *UsernameController*.
3. S3 (Mengambil Input *Password*):
Sistem mengambil input *password* yang dimasukkan pengguna melalui *passwordController*.

4. S4 (Validasi Input Kosong):
Sistem melakukan pengecekan apakah *Username* atau *password* kosong. Jika kondisi bernilai *True*, maka proses program dihentikan.
5. S5 (Validasi Panjang *Password*):
Sistem melakukan pengecekan apakah *password* melebihi 10 karakter. Jika kondisi bernilai *True*, maka proses program dihentikan.
6. S6 (Validasi Format *Password*):
Sistem melakukan pengecekan apakah *password* mengandung huruf dan angka menggunakan *regex*. Jika kondisi bernilai *False*, maka proses program dihentikan.
7. S7 (Pengecekan *Username*):
Sistem memeriksa apakah *Username* sudah digunakan pada *Firebase Firestore*. Jika kondisi bernilai *True*, maka proses program dihentikan.
8. S8 (Menambahkan Data *User*):
Jika seluruh validasi berhasil, maka sistem menjalankan proses `.add()` untuk menyimpan data akun siswa ke *Firebase Firestore*.
9. S9 (Menutup Dialog):
Setelah proses penambahan akun berhasil dilakukan, sistem menjalankan `Navigator.pop(context)` untuk menutup dialog dan proses program berakhir.

d. *Flowgraph* Fungsi Tambah Akun *User*



Sumber : data diolah oleh penulis (2026)

Gambar 5. 50 *Flowgraph* Fungsi Tambah Akun *User*

Flowgraph pada fungsi tambah akun *User* ini merepresentasikan alur eksekusi program berdasarkan hasil pengelompokan *Statement* yang telah diidentifikasi sebelumnya. *Flowgraph* yang dibentuk terdiri dari 9 node dan 12 edge, dengan penambahan Node 9 sebagai terminal node (end process) yang berfungsi untuk menandai berakhirnya seluruh alur eksekusi program.

Node pertama (Node 1) terdiri dari S1 yang mencakup proses menjalankan fungsi tambah akun *User*. Selanjutnya, Node 2 dan Node 3 merepresentasikan proses pengambilan input *Username* dan *password* dari pengguna.

Node 4 merepresentasikan proses validasi input kosong. Apabila kondisi bernilai *True*, maka proses program akan langsung berakhir pada Node 9. Jika kondisi bernilai *False*, maka program melanjutkan ke Node 5 untuk melakukan validasi panjang *password*.

Pada Node 5, sistem memeriksa apakah *password* melebihi 10 karakter. Jika kondisi bernilai *True*, maka program akan berakhir pada Node 9. Namun jika

kondisi bernilai *False*, maka program melanjutkan ke Node 6 untuk melakukan validasi format *password* menggunakan *regex*.

Node 6 merepresentasikan proses pengecekan apakah *password* mengandung huruf dan angka. Jika kondisi bernilai *False*, maka program akan berakhir pada Node 9. Namun apabila kondisi bernilai *True*, maka program melanjutkan ke Node 7 untuk memeriksa apakah *Username* sudah digunakan.

Pada Node 7, sistem melakukan pengecekan *Username* pada *Firestore*. Jika *Username* sudah digunakan, maka program akan berakhir pada Node 9. Namun apabila *Username* belum digunakan, maka program melanjutkan ke Node 8 yang menjalankan proses *.add()* untuk menambahkan data akun *User* ke *Firestore*.

Setelah proses penambahan data berhasil dilakukan, program menuju Node 9 yang menjalankan *Navigator.pop(context)* sekaligus menjadi node akhir (end process).

e. *Cyclomatic Complexity* Fungsi Tambah Akun *User*

Berdasarkan *Flowgraph* yang telah dibentuk, fungsi tambah akun *User* memiliki beberapa predicate node atau percabangan kondisi yang berasal dari proses validasi input kosong, validasi panjang *password*, validasi format *password*, serta pengecekan *Username* pada *Firestore*. Oleh karena itu, nilai *Cyclomatic Complexity* dapat dihitung menggunakan rumus berikut:

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

$$V(G) = 4 + 1 = 5$$

Dari perhitungan tersebut diperoleh nilai *Cyclomatic Complexity* sebesar 5. Nilai tersebut menunjukkan bahwa terdapat 5 jalur independen pada fungsi tambah akun *User* yang harus diuji dalam proses whitebox testing.

f. *Independen Path* Fungsi Tambah Akun *User*

Berdasarkan hasil perhitungan *Cyclomatic Complexity* yang diperoleh dari *Flowgraph* fungsi tambah akun *User*, nilai tersebut digunakan sebagai dasar untuk menentukan jumlah jalur independen yang harus diuji dalam proses whitebox

testing. Setiap jalur independen merepresentasikan kemungkinan alur eksekusi program yang berbeda berdasarkan hasil validasi input dan pengecekan data pada *Firebase Firestore*. Dengan demikian diperoleh 5 jalur independen (*independent path*) berikut ini:

1. Path 1:

1 → 2 → 3 → 4 → END

(*Username* atau *password* kosong)

2. Path 2:

1 → 2 → 3 → 4 → 5 → END

(*Password* melebihi 10 karakter)

3. Path 3:

1 → 2 → 3 → 4 → 5 → 6 → END

(Format *password* tidak sesuai *regex*)

4. Path 4:

1 → 2 → 3 → 4 → 5 → 6 → 7 → END

(*Username* sudah digunakan)

5. Path 5:

1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → END

(Proses tambah akun *User* berhasil dilakukan)

g. *Test Case* Fungsi Tambah Akun *User*

Pengujian dilakukan berdasarkan jalur independen (*independent path*) yang telah diperoleh dari hasil analisis *Flowgraph* fungsi tambah akun *User*. Jalur eksekusi pada *Test Case* diperoleh dari node-node yang dilalui pada *Flowgraph* sehingga proses pengujian dapat memverifikasi fungsi tambah akun *User* secara menyeluruh.

Tabel 5. 24 *Test Case* Fungsi Tambah Akun *User*

Test Case	Input	Jalur Eksekusi	Output
TC1	<i>Username</i> dan <i>password</i> kosong	1 → 2 → 3 → 4 → END	Proses tambah akun dihentikan

TC2	<i>Password</i> lebih dari 10 karakter	1 → 2 → 3 → 4 → 5 → END	Proses tambah akun dihentikan
TC3	<i>Password</i> tidak sesuai <i>regex</i>	1 → 2 → 3 → 4 → 5 → 6 → END	Proses tambah akun dihentikan
TC4	<i>Username</i> sudah digunakan	1 → 2 → 3 → 4 → 5 → 6 → 7 → END	Proses tambah akun dihentikan
TC5	Seluruh input valid	1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → END	Data akun berhasil ditambahkan

Sumber : data diolah oleh penulis (2026)

h. *Statement Coverage* Fungsi Tambah Akun *User*

Statement Coverage digunakan untuk mengukur sejauh mana seluruh *Statement* pada fungsi tambah akun *User* 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:

TC1:

S1, S2, S3, S4

TC2:

S1, S2, S3, S4, S5

TC3:

S1, S2, S3, S4, S5, S6

TC4:

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

TC5:

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

Berdasarkan hasil pengujian yang telah dilakukan, seluruh *Statement* pada fungsi tambah akun *User* yaitu S1 hingga S9 berhasil dieksekusi melalui *Test Case* TC1 sampai TC5. Dengan demikian diperoleh nilai *Statement Coverage* sebesar:

$$\frac{9}{9} \times 100\% = 100\%$$

Nilai tersebut menunjukkan bahwa seluruh *Statement* pada fungsi tambah akun *User* telah diuji dan seluruh alur eksekusi program berhasil dicakup dalam proses pengujian whitebox.

5.4.13 Pengujian Fungsi Reset *Password*

a. Potongan *Source Code* Fungsi Reset *Password*

Fungsi reset *password* digunakan untuk memperbarui *password* akun *User* pada *Firebase Firestore*. Proses dimulai dengan pengambilan input *password* baru dari pengguna, kemudian sistem melakukan validasi *password* seperti pengecekan input kosong, panjang *password*, dan format *password* menggunakan *regex*. Jika seluruh validasi berhasil, maka sistem menjalankan proses update *password* pada *Firebase Firestore*. Berikut adalah potongan *source code* fungsi reset *password* yang digunakan dalam tahap pengujian.

```
void resetPassword(String docId) {
```

```
    final passwordController =  
        TextEditingController();
```

```
    showDialog(  
        context: context,  
        builder: (context) {
```

```
        return AlertDialog(  
            actions: [
```

```
                ElevatedButton(  


```

```
                    onPressed: () async {
```

```
                        final newPassword =  
                            passwordController.text  
                                .trim();
```

```

    if (newPassword.isEmpty) {
        return;
    }

    if (newPassword.length > 10) {
        return;
    }

    final regex = RegExp(
        r'^(?=.*[A-Za-z])(?=.*\d)[A-Za-z\d]+$'
    );

    if (!regex.hasMatch(
        newPassword,
    )) {
        return;
    }

    await FirebaseFirestore
        .instance
        .collection('Users')
        .doc(docId)
        .update({
            'password':
                newPassword,
        });

    Navigator.pop(context);
},
),
],
);
},

```

```
);
}
```

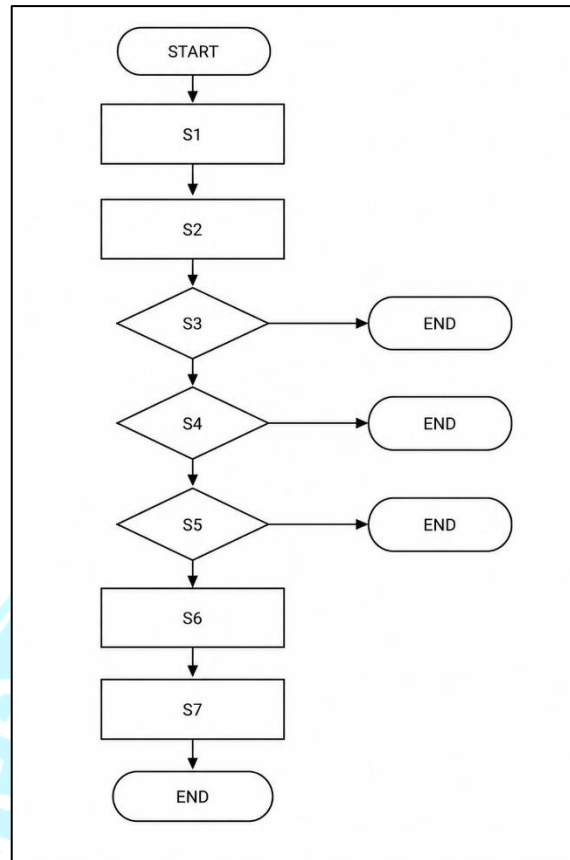
b. Identifikasi *Statement* Fungsi Reset *Password*

Tabel 5. 25 Identifikasi *Statement* Fungsi Reset *Password*

<i>Statement</i>	Source Code	Keterangan
S1	<code>resetPassword()</code>	Menjalankan fungsi reset <i>password</i>
S2	<code>passwordController.text.trim()</code>	Mengambil input <i>password</i> baru
S3	<code>if(newPassword.isEmpty)</code>	Validasi <i>password</i> kosong
S4	<code>if(newPassword.length > 10)</code>	Validasi panjang <i>password</i>
S5	<code>regex.hasMatch(newPassword)</code>	Validasi format <i>password</i>
S6	<code>Firestore.instance.collection('Users').doc(docId).update()</code>	Memperbarui <i>password</i> <i>User</i>
S7	<code>Navigator.pop(context)</code>	Menutup dialog

Sumber : data diolah oleh penulis (2026)

c. *Flowchart Fungsi Reset Password*



Sumber : data diolah oleh penulis (2026)

Gambar 5. 51 *Flowchart Fungsi Reset Password*

Berdasarkan diagram alur pada *Flowchart Fungsi Reset Password*, urutan logika kerja fungsi reset *password* dapat dijabarkan sebagai berikut:

1. S1 (Menjalankan Fungsi Reset *Password*):
Sistem memulai proses dengan menjalankan fungsi reset *password*.
2. S2 (Mengambil Input *Password* Baru):
Sistem mengambil input *password* baru yang dimasukkan pengguna melalui *password* Controller.
3. S3 (Validasi *Password* Kosong):
Sistem melakukan pengecekan apakah *password* baru kosong. Jika kondisi bernilai *True*, maka proses program dihentikan.
4. S4 (Validasi Panjang *Password*):
Sistem melakukan pengecekan apakah *password* melebihi 10 karakter. Jika kondisi bernilai *True*, maka proses program dihentikan.

5. S5 (Validasi Format *Password*):

Sistem melakukan pengecekan apakah *password* mengandung huruf dan angka menggunakan *regex*. Jika kondisi bernilai *False*, maka proses program dihentikan.

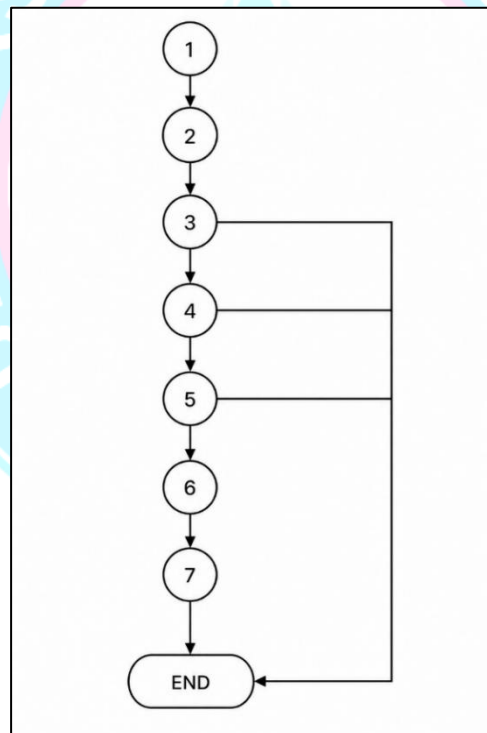
6. S6 (Update *Password User*):

Jika seluruh validasi berhasil, maka sistem menjalankan proses `.update()` untuk memperbarui *password User* pada *Firebase Firestore*.

7. S7 (Menutup Dialog):

Setelah proses reset *password* berhasil dilakukan, sistem menjalankan `Navigator.pop(context)` untuk menutup dialog dan proses program berakhir.

d. *Flowgraph* Fungsi Reset *Password*



Sumber : data diolah oleh penulis (2026)

Gambar 5. 52 *Flowgraph* Fungsi Reset *Password*

Flowgraph pada fungsi reset *password* ini merepresentasikan alur eksekusi program berdasarkan hasil pengelompokan *Statement* yang telah diidentifikasi sebelumnya. *Flowgraph* yang dibentuk terdiri dari 7 node dan 9

edge, dengan penambahan Node 7 sebagai terminal node (end process) yang berfungsi untuk menandai berakhirnya seluruh alur eksekusi program.

Node pertama (Node 1) terdiri dari S1 yang mencakup proses menjalankan fungsi reset *password*. Selanjutnya, Node 2 merepresentasikan proses pengambilan input *password* baru dari pengguna.

Node 3 merepresentasikan proses validasi *password* kosong. Apabila kondisi bernilai *True*, maka proses program akan langsung berakhir pada Node 7. Jika kondisi bernilai *False*, maka program melanjutkan ke Node 4 untuk melakukan validasi panjang *password*.

Pada Node 4, sistem memeriksa apakah *password* melebihi 10 karakter. Jika kondisi bernilai *True*, maka program akan berakhir pada Node 7. Namun jika kondisi bernilai *False*, maka program melanjutkan ke Node 5 untuk melakukan validasi format *password* menggunakan *regex*.

Node 5 merepresentasikan proses pengecekan apakah *password* mengandung huruf dan angka. Jika kondisi bernilai *False*, maka program akan berakhir pada Node 7. Namun apabila kondisi bernilai *True*, maka program melanjutkan ke Node 6 yang menjalankan proses *.update()* untuk memperbarui *password* User pada *Firebase Firestore*.

Setelah proses update *password* berhasil dilakukan, program menuju Node 7 yang menjalankan *Navigator.pop(context)* sekaligus menjadi node akhir (end process).

e. *Cyclomatic Complexity* Fungsi Reset Password

Berdasarkan *Flowgraph* yang telah dibentuk, fungsi reset *password* memiliki beberapa predicate node atau percabangan kondisi yang berasal dari proses validasi *password* kosong, validasi panjang *password*, serta validasi format *password* menggunakan *regex*. Oleh karena itu, nilai *Cyclomatic Complexity* dapat dihitung menggunakan rumus berikut:

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

$$V(G) = 3 + 1 = 4$$

Dari perhitungan tersebut diperoleh nilai *Cyclomatic Complexity* sebesar 4. Nilai tersebut menunjukkan bahwa terdapat 4 jalur independen pada fungsi reset *password* yang harus diuji dalam proses whitebox testing.

f. *Independen Path Fungsi Reset Password*

Berdasarkan hasil perhitungan *Cyclomatic Complexity* yang diperoleh dari *Flowgraph* fungsi reset *password*, nilai tersebut digunakan sebagai dasar untuk menentukan jumlah jalur independen yang harus diuji dalam proses whitebox testing. Setiap jalur independen merepresentasikan kemungkinan alur eksekusi program yang berbeda berdasarkan hasil validasi *password* dan proses update data pada *Firebase Firestore*. Dengan demikian diperoleh 4 jalur independen (*independent path*) berikut ini:

1. Path 1:

$1 \rightarrow 2 \rightarrow 3 \rightarrow 7$

(*Password* baru kosong)

2. Path 2:

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

(*Password* melebihi 10 karakter)

3. Path 3:

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

(Format *password* tidak sesuai *regex*)

4. Path 4:

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

(Proses reset *password* berhasil dilakukan)

g. *Test Case Fungsi Reset Password*

Pengujian dilakukan berdasarkan jalur independen (*independent path*) yang telah diperoleh dari hasil analisis *Flowgraph* fungsi reset *password*. Jalur eksekusi pada *Test Case* diperoleh dari node-node yang dilalui pada *Flowgraph* sehingga proses pengujian dapat memverifikasi fungsi reset *password* secara menyeluruh.

Tabel 5. 26 *Test Case* Fungsi Reset Password

Test Case	Input	Jalur Eksekusi	Output
TC1	Password kosong	1 → 2 → 3 → 7	Proses reset password dihentikan
TC2	Password lebih dari 10 karakter	1 → 2 → 3 → 4 → 7	Proses reset password dihentikan
TC3	Password tidak sesuai regex	1 → 2 → 3 → 4 → 5 → 7	Proses reset password dihentikan
TC4	Password valid	1 → 2 → 3 → 4 → 5 → 6 → 7	Password berhasil diperbarui

Sumber : data diolah oleh penulis (2026)

h. *Statement Coverage* Fungsi Reset Password

Statement Coverage digunakan untuk mengukur sejauh mana seluruh *Statement* pada fungsi reset password 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:

TC1:

S1, S2, S3

TC2:

S1, S2, S3, S4

TC3:

S1, S2, S3, S4, S5

TC4:

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

Berdasarkan hasil pengujian yang telah dilakukan, seluruh *Statement* pada fungsi reset password yaitu S1 hingga S7 berhasil dieksekusi melalui *Test Case* TC1 sampai TC4. Dengan demikian diperoleh nilai *Statement Coverage* sebesar:

$$\frac{7}{7} \times 100\% = 100\%$$

Nilai tersebut menunjukkan bahwa seluruh *Statement* pada fungsi reset *password* telah diuji dan seluruh alur eksekusi program berhasil dicakup dalam proses pengujian whitebox.

5.4.14 Pengujian Fungsi Hapus Akun *User*

a. Potongan *Source Code* Fungsi Hapus Akun *User*

Fungsi hapus akun *User* digunakan untuk menghapus data akun pengguna pada *Firebase Firestore* berdasarkan *docId* yang dipilih. Proses dimulai dengan menampilkan dialog konfirmasi penghapusan akun kepada admin. Apabila admin memilih melanjutkan proses penghapusan, maka sistem menjalankan proses *.delete()* pada *Firebase Firestore* kemudian menutup dialog konfirmasi. Berikut adalah potongan *source code* fungsi hapus akun *User* yang digunakan dalam tahap pengujian.

```
void hapusUser(String docId) {

showDialog(

context: context,
builder: (context) {

return AlertDialog(

actions: [
ElevatedButton(

onPressed: () async {

await FirebaseFirestore
.instance

.collection('Users')
```

```

        .doc(docId)
        .delete();

        Navigator.pop(context);
    },
    ),
],
);
},
); }

```

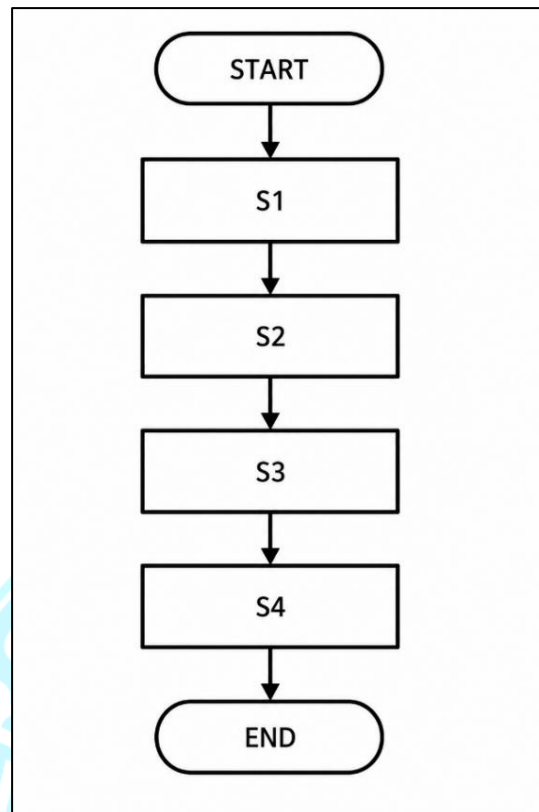
b. Identifikasi *Statement* Fungsi Hapus Akun *User*

Tabel 5. 27 Identifikasi *Statement* Fungsi Hapus Akun *User*

<i>Statement</i>	Source Code	Keterangan
S1	<code>hapusUser()</code>	Menjalankan fungsi hapus akun <i>User</i>
S2	<code>showDialog()</code>	Menampilkan dialog konfirmasi
S3	<code>FirebaseFirestore.instance.collection('Users').doc(docId).delete()</code>	Menghapus data akun <i>User</i>
S4	<code>Navigator.pop(context)</code>	Menutup dialog

Sumber : data diolah oleh penulis (2026)

c. *Flowchart Fungsi Hapus Akun User*



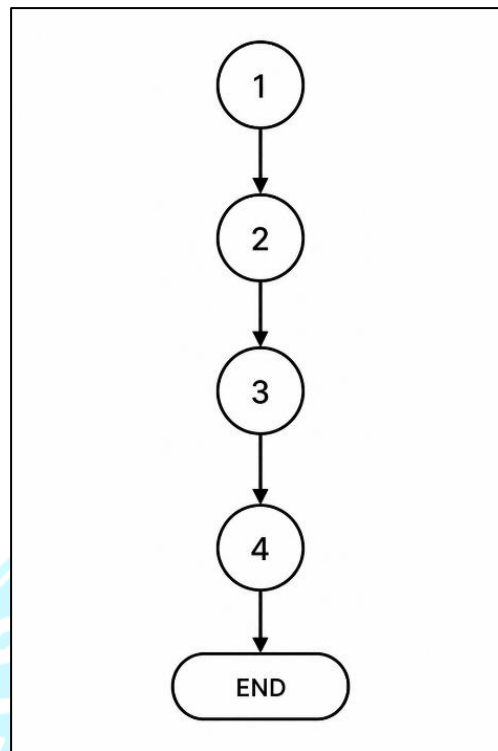
Sumber : data diolah oleh penulis (2026)

Gambar 5. 53 *Flowchart Fungsi Hapus Akun User*

Berdasarkan diagram alir pada *Flowchart Fungsi Hapus Akun User*, urutan logika kerja fungsi hapus akun *User* dapat dijabarkan sebagai berikut:

1. S1 (Menjalankan Fungsi Hapus *User*):
Sistem memulai proses dengan menjalankan fungsi hapus akun *User*.
2. S2 (Menampilkan Dialog Konfirmasi):
Sistem menampilkan dialog konfirmasi penghapusan akun *User* kepada admin.
3. S3 (Menghapus Data *User*):
Sistem menjalankan proses `.delete()` untuk menghapus data akun *User* pada *Firebase Firestore* berdasarkan `docId` yang dipilih.
4. S4 (Menutup Dialog):
Setelah proses penghapusan berhasil dilakukan, sistem menjalankan `Navigator.pop(context)` untuk menutup dialog dan proses program berakhir.

d. *Flowgraph* Fungsi Hapus Akun *User*



Sumber : data diolah oleh penulis (2026)

Gambar 5. 54 *Flowgraph* Fungsi Hapus Akun *User*

Flowgraph pada fungsi hapus akun *User* ini merepresentasikan alur eksekusi program berdasarkan hasil pengelompokan *Statement* yang telah diidentifikasi sebelumnya. *Flowgraph* yang dibentuk terdiri dari 4 node dan 3 edge, dengan penambahan Node 4 sebagai terminal node (end process) yang berfungsi untuk menandai berakhirnya seluruh alur eksekusi program.

Node pertama (Node 1) terdiri dari S1 yang mencakup proses menjalankan fungsi hapus akun *User*. Selanjutnya, Node 2 merepresentasikan proses penampilan dialog konfirmasi penghapusan akun *User*.

Node 3 merepresentasikan proses `.delete()` untuk menghapus data akun *User* pada *Firebase Firestore* berdasarkan `docId` yang dipilih. Setelah proses penghapusan selesai dilakukan, program menuju Node 4 yang menjalankan `Navigator.pop(context)` sekaligus menjadi node akhir (end process).

e. *Cyclomatic Complexity* Fungsi Hapus Akun *User*

Berdasarkan *Flowgraph* yang telah dibentuk, fungsi hapus akun *User* tidak memiliki predicate node atau percabangan kondisi karena seluruh alur program berjalan secara linear. Oleh karena itu, nilai *Cyclomatic Complexity* dapat dihitung menggunakan rumus berikut:

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

$$V(G) = 0 + 1 = 1$$

Dari perhitungan tersebut diperoleh nilai *Cyclomatic Complexity* sebesar 1. Nilai tersebut menunjukkan bahwa hanya terdapat 1 jalur independen pada fungsi hapus akun *User* yang harus diuji dalam proses whitebox testing.

f. *Independen Path Fungsi Hapus Akun User*

Berdasarkan hasil perhitungan *Cyclomatic Complexity* yang diperoleh dari *Flowgraph* fungsi hapus akun *User*, nilai tersebut digunakan sebagai dasar untuk menentukan jumlah jalur independen yang harus diuji dalam proses whitebox testing. Karena fungsi hapus akun *User* tidak memiliki percabangan kondisi dan seluruh alur program berjalan secara linear, maka hanya terdapat satu jalur independen yang merepresentasikan seluruh proses eksekusi program. Jalur independen tersebut meliputi proses menampilkan dialog konfirmasi, penghapusan data akun *User* pada *Firebase Firestore*, hingga penutupan dialog konfirmasi. Dengan demikian diperoleh satu jalur independen (*independent path*) sebagai berikut:

1. Path1:

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 4$$

(Proses hapus akun *User* berhasil dilakukan hingga dialog konfirmasi ditutup)

g. *Test Case Fungsi Hapus Akun User*

Pengujian dilakukan berdasarkan jalur independen (*independent path*) yang telah diperoleh dari hasil analisis *Flowgraph* fungsi hapus akun *User*. Jalur eksekusi pada *Test Case* diperoleh dari node-node yang dilalui pada *Flowgraph* sehingga proses pengujian dapat memverifikasi fungsi hapus akun *User* secara menyeluruh.

Tabel 5. 28 *Test Case* Fungsi Hapus Akun *User*

Test Case	Input	Jalur Eksekusi	Output
TC1	Memilih hapus akun <i>User</i>	1 → 2 → 3 → 4	Data akun <i>User</i> berhasil dihapus

Sumber : data diolah oleh penulis (2026)

h. *Statement Coverage* Fungsi Hapus Akun *User*

Statement Coverage digunakan untuk mengukur sejauh mana seluruh *Statement* pada fungsi hapus akun *User* 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:

TC1:

S1, S2, S3, S4

Berdasarkan hasil pengujian yang telah dilakukan, seluruh *Statement* pada fungsi hapus akun *User* yaitu S1 hingga S4 berhasil dieksekusi melalui *Test Case* TC1. Dengan demikian diperoleh nilai *Statement Coverage* sebesar:

$$\frac{4}{4} \times 100\% = 100\%$$

Nilai tersebut menunjukkan bahwa seluruh *Statement* pada fungsi hapus akun *User* telah diuji dan seluruh alur eksekusi program berhasil dicakup dalam proses pengujian whitebox.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil penelitian, perancangan, implementasi, serta pengujian sistem yang telah dilakukan pada penelitian “Perancangan Media Pembelajaran Berbasis *Android* Pada Mata Pelajaran Komputer”, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Aplikasi media pembelajaran berbasis *Android* yang dirancang berhasil dikembangkan sebagai media pendukung proses belajar mengajar mata pelajaran komputer di MTs Ishlahul Ummah. Aplikasi ini menyediakan materi pembelajaran *hardware* dan *software*, latihan soal, serta hasil latihan yang dapat membantu guru dan siswa dalam kegiatan pembelajaran secara lebih efektif dan terstruktur.
2. Media pembelajaran berbasis *Android* yang dibangun mampu memberikan kemudahan bagi siswa dalam memahami materi komputer melalui tampilan yang interaktif, sederhana, dan mudah digunakan. Penggunaan gambar, materi digital, dan latihan soal pada aplikasi dapat meningkatkan motivasi belajar siswa serta membantu proses pembelajaran menjadi lebih menarik dibandingkan metode konvensional.
3. Aplikasi yang dikembangkan dapat diakses menggunakan perangkat *mobile* berbasis *Android* sehingga memudahkan siswa untuk belajar kapan saja dan di mana saja. Dengan memanfaatkan teknologi *Android*, proses pembelajaran menjadi lebih fleksibel dan sesuai dengan perkembangan teknologi digital saat ini.
4. Implementasi *Firebase Firestore* sebagai media penyimpanan *database* berhasil mendukung pengelolaan data pengguna, materi pembelajaran, latihan soal, serta hasil latihan secara terorganisir dan terintegrasi dengan aplikasi. Selain itu, sistem hak akses berdasarkan peran pengguna (guru dan siswa) mampu menjaga keamanan data serta mendukung pengelolaan aplikasi dengan lebih baik.

6.2 Saran

Meskipun aplikasi media pembelajaran berbasis *Android* ini telah berhasil dirancang dan diimplementasikan sesuai dengan kebutuhan penelitian, masih terdapat beberapa hal yang dapat dikembangkan untuk meningkatkan kualitas sistem dan pengalaman pengguna di masa mendatang. Oleh karena itu, beberapa saran yang dapat diberikan untuk pengembangan selanjutnya adalah sebagai berikut:

1. Penambahan Fitur Multimedia Pembelajaran

Untuk meningkatkan daya tarik dan pemahaman siswa terhadap materi komputer, disarankan agar aplikasi dikembangkan dengan menambahkan fitur multimedia seperti video pembelajaran, animasi interaktif, audio penjelasan materi, serta simulasi penggunaan perangkat komputer. Dengan adanya fitur tersebut, proses pembelajaran diharapkan menjadi lebih menarik, interaktif, dan mudah dipahami oleh siswa.

2. Pengembangan Fitur Evaluasi Pembelajaran

Pada pengembangan berikutnya, aplikasi dapat ditambahkan fitur evaluasi yang lebih lengkap seperti *timer* pengerjaan soal, pembahasan jawaban otomatis, peringkat nilai siswa, serta grafik perkembangan hasil belajar. Fitur tersebut dapat membantu guru dalam memantau kemampuan siswa dan meningkatkan motivasi belajar peserta didik.

3. Pengembangan Pengujian Sistem

Penelitian ini masih berfokus pada pengembangan dan implementasi aplikasi media pembelajaran berbasis *Android*. Oleh karena itu, untuk penelitian selanjutnya disarankan melakukan pengujian sistem secara lebih mendalam menggunakan metode *User Acceptance Testing (UAT)* agar dapat diketahui tingkat kelayakan, kemudahan penggunaan, serta kepuasan pengguna terhadap aplikasi yang telah dikembangkan.

4. Pengembangan Sistem Berbasis *Online* dan *Multi Platform*

Aplikasi saat ini masih difokuskan pada platform *Android*. Untuk pengembangan selanjutnya, aplikasi dapat dikembangkan menjadi sistem berbasis *online* atau *multi platform* agar dapat diakses tidak hanya

5. melalui *Android*, tetapi juga melalui *website* maupun perangkat lainnya. Dengan demikian, akses pembelajaran akan menjadi lebih fleksibel dan menjangkau lebih banyak pengguna.

6. Peningkatan Keamanan dan Penyimpanan Data

Untuk menjaga keamanan data pengguna dan materi pembelajaran, pengembangan selanjutnya dapat menambahkan sistem keamanan yang lebih baik seperti enkripsi data, autentikasi berlapis, serta sistem *backup database* otomatis. Hal ini bertujuan untuk mengurangi risiko kehilangan data dan menjaga keamanan informasi dalam aplikasi.



DAFTAR PUSTAKA

- Abdussamad, Z. (2021). Metode Penelitian Kualitatif. In *Syakir Media Press*.
- Afriyadi, H., Hery, T., Noor, H., Supardi, Laila, Sinta, N., Prakasa, Yana, F., Hasibuan, R. P. A., & Asyhar, A. D. A. (2023). Media Pembelajaran Berbasis Digital (Teori & Praktik). In *PT. Sonpedia Publishing Indonesia* (Issue 1). https://repository.uinmataram.ac.id/2683/1/Media_pembelajaran_berbasis_digital.pdf
- Ahmad, N., Widarti, ErniRahayu, T., Lina, T. N., & Seta, Henki BayuManuhutu, AbrahamErnawati, IinHidayanto, S. W. (2023). Pengantar Ilmu Komputer. In *Insight Mediatama*. http://repository.uinsu.ac.id/8611/1/Modul_Perkuliahan_PIK_%283%29.pdf
- Andriyanto, S. (2022). *Pemodelan Perangkat Lunak Behavior Diagram* (M. Ayu (ed.); 1st ed.). Politeknik Manufaktur Negeri Bangka Belitung.
- Fatqurhohman, Paling, S., Makmur, A., Yati, Albar, M., Susetyo, A. M., Putra, Y. W. S., Rajiman, W., Djamilah, S., Ratnadewi, Suhendi, H. Y., & Irvani, A. I. (2023). *MEDIA PEMBELAJARAN DIGITAL Penulis: (Issue September)*. TOHAR MEDIA.
- Gunawan, A. (2024). *Mobile Programming Menggunakan Flutter dan Visual Studio Code Untuk Pemula*. www.penerbitlitnus.co.id
- Haryadi, O., Bunda, Y. P., & Rouza, E. (2024). Media Pembelajaran Pengenalan Hardware dan Software Berbasis *Android* pada SMK. *REMIK: Riset Dan E-Jurnal* ..., 8, 130–138. <https://www.polgan.ac.id/jurnal/index.php/remik/article/view/13265%0Ahttps://www.polgan.ac.id/jurnal/index.php/remik/article/download/13265/2264>
- Iwan Ady Prabowo, & Wijayanto, Hendro Yudanto, B. W. S. N. (2021). Buku Ajar Pemrograman Mobile Berbasis *Android*. In *Angewandte Chemie International Edition*. https://eprints.sinus.ac.id/762/1/Buku_Ajar-Pemrograman_Android.pdf
- Lantana, D. A., & Fauziah. (2025). DESIGN THINKING DAN INOVASI. In

- Maharani, M., Rini, F., & Pratama, A. (2023). Pengembangan Media Pembelajaran Berbasis *Android* pada Mata Pelajaran Informatika Kelas X di SMK Nusantara Padang. *Jurnal Pustaka Data (Pusat Akses Kajian Database, Analisa Teknologi, Dan Arsitektur Komputer)*, 3(1), 19–24. <https://doi.org/10.55382/jurnalpustakadata.v3i1.416>
- Ramdany, S. W., Kaidar, S. A., Aguchino, B., Amelia, C., & Putri, A. (2024). *Penerapan UML Class Diagram dalam Perancangan Sistem Informasi Perpustakaan Berbasis Web*. 5(1).
- Resa Ayu, R., Efriyanti, L., & Benfani, R. (2022). Perancangan Media Pembelajaran Berbasis *Android* Menggunakan Aplikasi Andromo Pada Mata Pelajaran Bimbingan TIK. *Intellect : Indonesian Journal of Learning and Technological Innovation*, 1(2), 260–267. <https://doi.org/10.57255/intellect.v1i2.108>
- Saputra, D., Darmawan, W. S., Syarif, M., & Risdiansyah, D. (2023). Analisis & Perancangan Sistem Informasi. In *Insan Cendekia Mandiri*.
- Trisna, N., Elva, Y., & Jamhur, A. I. (2022). *Sosialisasi Pengenalan Hardware dan Software Komputer pada Siswa SMP SAHARA Padang*. 2(4), 208–213.
- Tugiman, Setyawati, AriestaNovita, LinaSyamiya, Estu NianaMaryani, LeniWahab, A. S. L., & Nurislamiah, SitiSaputra, Erwin RahayuHasanah, HaniMunawati, SitiUtami, Wanda YuliaAtrisia, Marchela IndahAhmad, Deni NasirFajrin, VerawatiFarliana, NinaQadarsi, J. (2024). *Media Pembelajaran Berbasis Digital*. Arr Rad Pratama.
- Wahyuni, S., Suarman, & Kartikowati, S. (2022). Pengembangan Media Pembelajaran Berbasis *Android* (MEKOID). In *Scaffolding: Jurnal Pendidikan Islam dan Multikulturalisme* (Vol. 4, Issue 3).
- Wayahdi, M. R., & Ruziq, F. (2023). Pemodelan Sistem Penerimaan Anggota Baru dengan Unified Modeling Language (UML) (Studi Kasus: Programmer Association of Battuta). *Jurnal Minfo Polgan*, 12(1), 1514–1521. <https://doi.org/10.33395/jmp.v12i1.12870>
- Wijaya, K., Kartina, R., & H, N. A. (2024). Implementasi Metode Forward Chaining Dalam Pengembangan Sistem Pakar Diagnosis Awal Demam

- Berdarah (Dbd) Studi Kasus Puskesmas Belida Darat. *Jurnal Sistem Komputer Musi Rawas*, 9(2), 98–107.
- Wijaya, K., Kartina, R., Yandi, J., & Hermizahadiwidastra. (2021). *Pemrograman Mobile*. LITERASI NUSANTARA.
- Wintana, D., Pribadi, D., & Nurhadi, M. Y. (2022). Analisis Perbandingan Efektifitas White-Box Testing dan Black-Box Testing. *Jurnal Larik: Ladang Artikel Ilmu Komputer*, 2(1), 8–16. <https://doi.org/10.31294/larik.v2i1.1382>



LAMPIRAN

Lampiran 1 Surat Permohonan Penelitian

 FAKULTAS ILMU KOMPUTER UNIVERSITAS PRABUMULIH	
Prabumulih, 03 November 2025	
Nomor	: 029 /FASILKOM/UNPRA/Inf-S1/XI/2025
Lampiran	: -
Perihal	: Permohonan Penelitian dan Pengambilan Data
Kepada Yth, Kepala Sekolah Mts. Islahul Ummah Di Tempat	
Dalam rangka pelaksanaan skripsi mahasiswa Program Studi Informatika Fakultas Ilmu Komputer Universitas Prabumulih, maka kami harapkan bantuan dari Bpk/Ibu kiranya mengizinkan mahasiswa:	
1. Nama	: M. Afri Rahman
Nim	: 2022230037
Program Studi	: Informatika
Untuk melaksanakan penelitian dan pengambilan data ditempat Bpk/Ibu Pimpin guna menyelesaikan skripsi. Atas bantuan dan kerjasamanya kami ucapkan terima kasih.	
 Ketua Program Studi, Informatika <u>Iwan Setiawan, S.Kom., M.Kom.</u> 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 e-mail : mod@umpra.ac.id	

Lampiran 2 Surat Balasan Permohonan Penelitian



YAYASAN ISHLAHUL UMMAH
MADRASAH TSANAWIYAH ISHLAHUL UMMAH
Akte Notaris HAMIDAYATI HAMID, SH No. 21
Alamat : Jl. Raya Baturaja Desa Beringin Kec. Lubai
Kabupaten Muara Enim

Beringin, 4 November 2025

SURAT IZIN PENELITIAN

No : 05/MTs.IU-DB/XI/2025

Kepada
Yth. Ka, Prodi Informatika
Universitas Prabumulih
Di

Tempat

Menindak lanjuti surat nomor : 029 / FASILKOM /UNPRA/Inf - S1 / XI/2025, tanggal 03 november 2025 perihal permohonan penelitian dan pengambilan data. Sehubungan dengan surat tersebut diatas bersama ini kami memberikan izin kepada mahasiswa ;

Nama ; M. Afri Rahman

Nim ; 20222 30037

Program studi ; Informatika

Untuk Melakukan penelitian dan pengambilan data di MTs Ishlahul Ummah Sampai dengan Selesai.

Demikianlah surat Izin ini dibuat agar dapat dipergunakan sebagaimana mestinya atas perhatian diucapkan terima kasih.



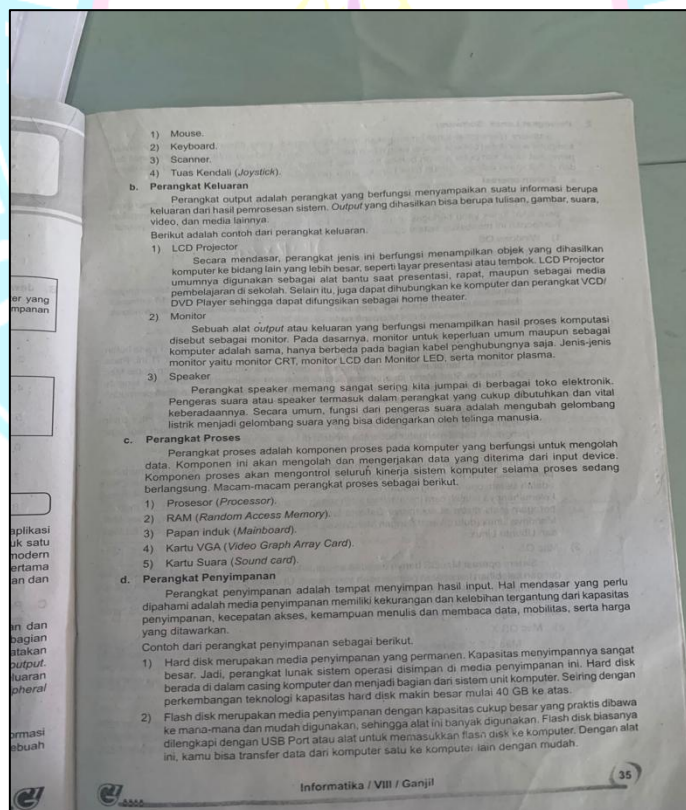
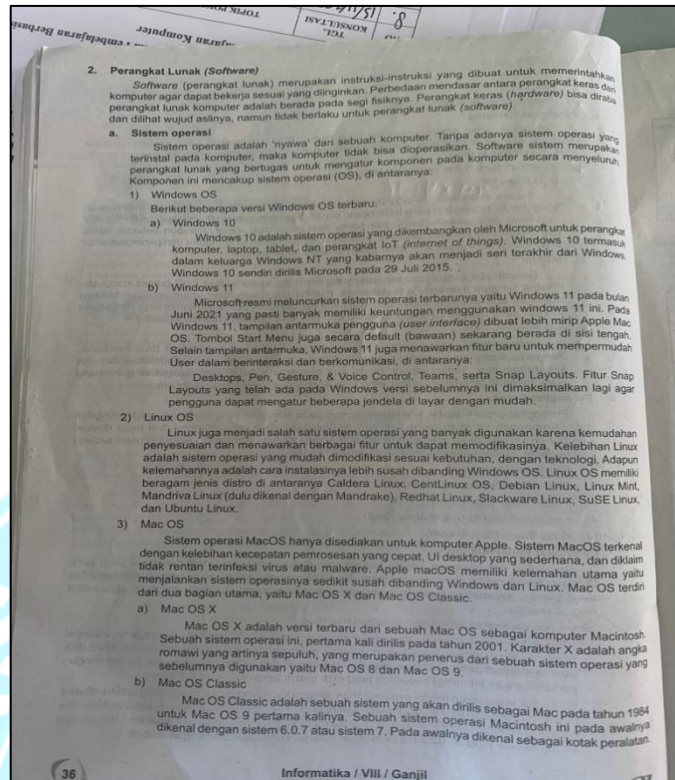
Kepala madrasah

Kurnia Sari, S.Pd.I

Lampiran 3 Dokumentasi Wawancara & Observasi







Lampiran 4 Daftar Pertanyaan Wawancara

Draft Wawancara

Nama Narasumber : Gita Umaroh, S.Pd.

Jabatan : Guru

Tempat Wawancara : Ruang Guru MTs Ishlahul Ummah

Tanggal Wawancara : 5 November 2025

Pewawancara : M. Afri Rahman

No	Pertanyaan	Jawaban
1	Apakah bisa ibu jelaskan sejarah singkat dari MTs Ishlahul Ummah ?	Madrasah ini resmi didirikan pada tahun 2007 di bawah naungan Yayasan Ishlahul Ummah dan merupakan sekolah Islam swasta. Sejak awal berdiri, sekolah berkembang secara bertahap berkat dukungan tokoh masyarakat dan wali murid
2	Apa visi, misi serta tujuan MTs Ishlahul Ummah ini?	Untuk visi, misi serta tujuan dapat dilihat di reklame sekolah yang ada didepan gedung kantor kepala sekolah agar lebih valid dan tepat.
3	Bagaimana struktur organisasi di MTs Ishlahul Ummah ?	Struktur organisasi di sekolah terdiri dari kepala madrasah, wakil kepala bidang kurikulum dan wakil kepala bidang kesiswaan, para guru, serta staf administrasi,
4	Bisa dijelaskan mengenai jabatan dan tanggung jawab ibu sebagai guru di sekolah ini?	Saya bertugas sebagai wali kelas dan juga sebagai guru mata pelajaran komputer. Selain mengajar, saya membimbing siswa dalam memahami

		teknologi serta terlibat dalam berbagai kegiatan sekolah yang berkaitan dengan kebutuhan digital
5	Bagaimana proses pembelajaran di MTs Ishlahul Ummah ini?	Proses pembelajaran berjalan cukup baik. Saat ini, proses belajar masih menggunakan buku sebagai sumber utama, dan guru menjadi penyampai materi secara langsung. Kami menggabungkan metode ceramah dan diskusi sederhana, dan sebisa mungkin menyesuaikan penyampaian materi agar siswa lebih mudah memahami pelajaran, meskipun fasilitas pembelajaran masih bersifat konvensional.
6	Apakah Saat ini sudah ada sistem berupa media pembelajaran berbasis <i>Android</i> atau sebagainya ?	Sejauh ini belum ada media pembelajaran berbasis <i>Android</i> , ataupun website dan sejenisnya. Kami masih menggunakan metode biasa seperti buku, modul, dan penjelasan langsung dari guru dalam menyampaikan materi kepada siswa.
7	Apakah kondisi pembelajaran saat ini berpengaruh pada semangat siswa dalam belajar?	Ya, saya melihat beberapa siswa masih kurang termotivasi, terutama jika materi cukup abstrak atau tidak disertai contoh yang menarik. Media pembelajaran yang lebih interaktif sebenarnya bisa membantu meningkatkan minat mereka.
8	Bagaimana pendapat ibu jika peneliti mengajukan pada pihak sekolah untuk melakukan	Menurut saya itu ide yang sangat baik. Aplikasi media pembelajaran yang dapat digunakan di smartphone

pembuatan aplikasi media pembelajaran berbasis <i>Android</i> khusus untuk saat ini pada mata pelajaran komputer?	itu bisa membuat pembelajaran lebih menarik dan memudahkan siswa memahami materi. Saya pribadi sangat mendukung jika ada inovasi seperti itu di sekolah kami.
9 Apakah Ibu bersedia apabila di kemudian hari peneliti melakukan uji coba aplikasi yang dikembangkan bersama siswa/i di kelas Ibu?	tentu, saya bersedia. Saya akan senang jika penelitian ini bisa memberikan manfaat untuk siswa saya. Selama prosesnya jelas dan tidak mengganggu jadwal pembelajaran, saya siap membantu pelaksanaannya.

