BAB I

PENDAHULUAN

A. Latar Belakang

Teknologi perangkat mobile yang berkembang begitu cepat dapat digunakan dalam banyak aspek kehidupan, salah satunya adalah pendidikan. *Mobile learning* (*M-Learning*) adalah pembelajaran melalui perangkat bergerak. Dengannya, kita dapat dengan mudah mengakses informasi kapan saja dan di mana saja.

Komunitas IT Smansa (KITS) Parepare adalah organisasi sekolah yang didirikan di SMA Negeri 1 Parepare pada tanggal 4 Maret 2013 M, bertepatan dengan 22 Rabi'ul Akhir 1434 H. KITS Parepare didirikan sebagai kelompok belajar siswa TI. Tujuannya adalah untuk menyalurkan minat dan bakat siswa SMAN 1 Parepare dalam teknologi informasi.

Berdasarkan hasil observasi yang telah dilakukan, proses pembelajaran, pengolahan data dan pengarsipan pada Komunitas IT Smansa (KITS) Parepare masih dilaksanakan secara konvensional. Kegiatan pembelajaran dilakukan secara tatap muka di ruang kelas dengan waktu yang terbatas, sedangkan proses pencarian informasi dan pengolahan data memerlukan waktu yang cukup lama. Proses belajar mengajar menjadi kurang efektif karena beberapa kendala, salah satunya adalah keterlambatan kehadiran anggota komunitas yang mengakibatkan tertinggalnya materi pembelajaran dan keharusan mencatat ulang materi yang telah disampaikan.

Kondisi ini menyebabkan tidak optimalnya pemanfaatan waktu dalam setiap pertemuan pembelajaran yang relatif singkat.

Seiring dengan perkembangan teknologi, sistem informasi yang terkomputerisasi tentunya akan sangat membantu suatu organisasi dalam proses pengolahan data dan pengarsipan dengan menggunakan sebuah sistem. Sistem yang dimaksudkan untuk membantu pembelajaran pengolahan dan pengarsipan data digital dikenal sebagai aplikasi pembelajaran mobile (*Mobile learning*).

Aplikasi *Mobile learning* ini dilengkapi dengan berbagai fitur terintegrasi seperti pengelolaan master data, manajemen kurikulum, sistem monitoring, database anggota, pengelolaan materi pelatihan, sistem evaluasi/kuis, forum diskusi, presensi digital, dan sistem penilaian yang dapat diakses secara langsung oleh pengguna sesuai dengan tingkat aksesnya masing-masing.

Berdasarkan keadaan saat ini, penulis memutuskan untuk mengangkat judul "Rancang Bangun Aplikasi *Mobile learning* (*M-Learning*) Komunitas IT Smansa Parepare." Tujuan dari judul ini adalah untuk memanfaatkan teknologi mobile yang dapat diakses kapan saja dan dimana saja untuk mengoptimalkan proses pembelajaran dan pengelolaan komunitas di Komunitas IT Smansa Parepare.

B. Rumusan Masalah

Berdasarkan latar belakang sebelumnya, rumusan masalah penelitian ini adalah sebagai berikut:

1. Bagaimana cara mendesain dan membuat aplikasi pembelajaran mobile (*M-Learning*) dengan antarmuka responsif yang dapat digunakan di berbagai perangkat seluler?

2. Bagaimana cara mengoptimalkan kinerja *Progressive Web Apps (PWA)* dengan menerapkan strategi *caching* dan *Service Worker js* untuk meningkatkan kecepatan loading aplikasi *M-Learning* Komunitas IT Smansa?

C. Tujuan Penelitian

Berdasarkan apa yang telah dirumuskan sebelumnya, tujuan dari penelitian ini adalah sebagai berikut:

1. Mendesain dan membangun sebuah aplikasi *Mobile learning* (*M-Learning*) untuk Komunitas IT Smansa. Aplikasi ini akan memiliki desain antarmuka yang responsif, sehingga bisa berjalan optimal di berbagai perangkat mobile dengan ukuran layar yang berbeda.
2. Mengimplementasikan dan mengoptimalkan *Progressive Web App (PWA)* pada aplikasi tersebut. Optimalisasi ini akan dilakukan melalui strategi caching yang efisien dan penggunaan *Service Worker*, dengan tujuan utama untuk meningkatkan kecepatan loading aplikasi.

D. Batasan Masalah

Batasan masalah dalam perancangan dan pembangunan aplikasi *Mobile learning* (*M-Learning*) ini adalah sebagai berikut:

1. Ruang Lingkup Pengguna : Penelitian ini terbatas pada siswa SMAN 1 Parepare yang tergabung dalam Komunitas IT dan tidak mencakup siswa dari sekolah lain atau pengguna eksternal.

2. Cakupan Fitur Pembelajaran : Penelitian ini dibatasi pada pengembangan fitur pembelajaran dasar dan tidak mencakup fitur lanjutan seperti live streaming, video conference, atau AI recommendation system.
3. Implementasi *Progressive Web Apps* : Penelitian ini dibatasi pada implementasi fitur PWA dasar meliputi *Service Worker* untuk caching offline, *Web App Manifest* untuk instalasi aplikasi, dan push notifications, namun tidak mencakup fitur PWA lanjutan seperti *background sync, web share target API*, atau *payment* request *API*.
4. Penelitian ini fokus pada pengembangan *Progressive Web App (PWA)* yang dapat diinstal melalui browser dan tidak mencakup pengembangan aplikasi native *Android* (APK) menggunakan Java/Kotlin atau hybrid app menggunakan framework seperti *React* Native atau Flutter.

E. Sistematika Penulisan

Untuk membuat tulisan ini lebih mudah dipahami dan dipahami, kerangka atau pedoman yang digunakan harus dijelaskan secara sistematis. Berikut ini adalah skema yang digunakan dalam penulisannya:

BAB I PENDAHULUAN

Bab ini berisikan latar belakang topik penelitian, perumusan masalah, tujuan, dan batasan penelitian, serta keuntungan dan kerugian penelitian, dibahas dalam bab ini.

BAB II TINJAUAN PUSTAKA

Bab ini menelaah teori-teori yang berkaitan beserta temuan penelitian sebelumnya sebagai landasan, sekaligus menguraikan kerangka konseptual yang menjadi dasar penelitian.

BAB III METODE PENELITIAN

Kerangka metodologis penelitian untuk pembangunan sistem yang diajukan dibahas dalam bab ini. Ini mencakup rancangan pendekatan, alur kerja/prosedur, teknik akuisisi, dan analisis data.

BAB IV HASIL DAN PEMBAHASAN

Bab ini membahas hasil penelitian, yang mencakup penjelasan tentang rancangan sistem, teknik yang digunakan, dan prosedur pengujian. Kemudian, diskusi membahas temuan utama dan hubungannya dengan teori dan penelitian sebelumnya.

BAB IV PENUTUP

Bab terakhir berisi kesimpulan dan rekomendasi. Kesimpulan menegaskan masalah yang diangkat dan solusinya, sementara rekomendasi memberi saran untuk mengatasi masalah dan mengarahkan pengembangan berikutnya.

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Rancang bangun

Dalam Kamus Besar Bahasa Indonesia, “rancang” merujuk pada kegiatan menyusun atau menata sesuatu sebelum pelaksanaan; dengan demikian, rancang bangun dipahami sebagai proses perencanaan yang sistematis hingga realisasi suatu karya/sistem.

Menurut Pressman (dalam Ferdiansyah, 2018), rancang bangun perangkat lunak merupakan tahapan yang menerjemahkan hasil analisis kebutuhan menjadi spesifikasi teknis yang dapat diimplementasikan. Pada fase ini dirumuskan arsitektur, komponen, antarmuka, serta mekanisme interaksi dan pengolahan data yang akan diwujudkan dalam kode program. Rancang bangun dapat diarahkan untuk membangun sistem baru maupun memperbarui sistem yang sudah ada, baik secara menyeluruh maupun sebagian.

Sejalan dengan itu, Maulani dkk. (2018) menekankan bahwa rancang bangun mencakup aktivitas pembuatan dan pengembangan aplikasi/sistem guna memenuhi kebutuhan organisasi yang sebelumnya belum terakomodasi. Dengan demikian, rancang bangun dapat dirangkum sebagai rangkaian langkah terstruktur yang mengubah keluaran analisis menjadi artefak desain siap-implementasi, sekaligus menjadi dasar pengembangan, pengujian, dan pemeliharaan sistem..

2. Aplikasi

Menurut Al Fatih (2024), aplikasi adalah perangkat lunak yang dibuat untuk menjalankan fungsi spesifik pada perangkat elektronik seperti komputer, ponsel pintar, atau tablet. Istilah ini berasal dari bahasa Inggris application software. Dalam praktiknya, jenis aplikasi sangat beragam dan mencakup aneka bidang serta kebutuhan.

Hasan Abdurahman dan Asep Ririh Riswaya (2014) mendefinisikan aplikasi sebagai program siap pakai yang mengeksekusi perintah pengguna guna menghasilkan keluaran yang tepat sesuai tujuan perancangannya. Dengan kata lain, aplikasi berperan sebagai sarana pemecahan masalah melalui teknik pemrosesan data dan komputasi tertentu (lihat juga Saed Novendri, 2019).

Dari pemaparan tersebut dapat disimpulkan bahwa aplikasi merupakan program yang dikembangkan untuk menjalankan fungsi tertentu sehingga mempermudah dan mengefisienkan pekerjaan.

3. *Mobile learning*

Secara leksikal, mobile berarti bergerak/berpindah; aplikasi mobile adalah perangkat lunak yang dioperasikan pada mobile device. Aplikasi tersebut memfasilitasi beragam aktivitas belajar, tugas sekolah, penelusuran informasi, hiburan, hingga perdagangan daring. Implikasinya, media pembelajaran berbasis mobile adalah medium penyampaian materi menggunakan perangkat seperti smartphone (Salamun, 2017).

Adapun *Mobile learning* dipahami sebagai pembelajaran yang mengandalkan perangkat bergerak. Surahman & Surjono (2017) menegaskan bahwa *Mobile*

learning merupakan alternatif yang memaksimalkan pemanfaatan media berbasis perangkat bergerak dan memungkinkan kemandirian belajar tanpa batasan ruang maupun waktu (Jurnal Kajian Teknologi Pendidikan; Rifai, 2020).

(O'Malley, 2003, dikutip dalam Rifai et al., 2020, h. 2) menambahkan bahwa *Mobile learning* mencakup pembelajaran di lokasi tidak tetap (*non-fixed*) dengan memanfaatkan teknologi bergerak untuk interaksi edukatif. Definisi ini menegaskan bahwa *Mobile learning*:

1. Menggunakan perangkat bergerak (misal: *smartphone*) sebagai medium utama.
2. Menyediakan bahan ajar yang dapat diakses on-demand dengan antarmuka visual menarik.

Membuat pembelajaran mobile memungkinkan peserta didik untuk terus belajar kapan pun dan di mana pun mereka mau. Ini membuat pembelajaran lebih menarik dan menyenangkan, meningkatkan minat peserta didik. Selain itu, *Mobile learning* memungkinkan peserta didik untuk dapat membuka aplikasi kapan pun dan dimana pun.

4. Komunitas IT Smansa Parepare

Komunitas IT Smansa, yang lebih dikenal dengan akronim KITS, adalah sebuah organisasi siswa di SMA Negeri 1 Parepare yang menjadi wadah bagi para murid yang memiliki minat dan semangat dalam mempelajari Ilmu Komputer dan Informatika. Didorong oleh pesatnya perkembangan teknologi di era Revolusi Industri 4.0 dan kesadaran akan pentingnya penguasaan ilmu komputer untuk

kemajuan bangsa , para siswa SMA Negeri 1 Parepare menghimpun diri untuk mendirikan komunitas ini.

Diresmikan pada 4 Maret 2013, KITS berkedudukan di UPTD SMA Negeri 1 Parepare; sekretariatnya berada di Laboratorium Multimedia.



Gambar 2. 1 Logo Komunitas IT Smansa Parepare
(Sumber: AD ART Komunitas IT Smansa Parepare)

a. Asas, Sifat, dan Status Organisasi

1) Asas: Kegiatan dan interaksi dalam komunitas ini berasaskan pada prinsip kekeluargaan.
2) Sifat: KITS adalah komunitas nirlaba yang bertujuan untuk mendukung pengembangan bakat, potensi, dan minat dalam teknologi informasi (IT).
3) Status: Secara struktural, KITS adalah komunitas IT yang berada di bawah naungan OSIS SMA Negeri 1 Parepare.

b. Visi dan Misi

Visi KITS adalah menjadi wadah bagi murid SMA Negeri 1 Parepare yang gemar akan Teknologi Informasi (IT), menjadi

penampung ide-ide kreatif untuk karya digital, serta turut menjunjung tinggi nama baik sekolah. Untuk mencapai visi tersebut, KITS memiliki misi sebagai berikut:

1) Menghimpun murid SMA Negeri 1 Parepare yang berminat pada IT.
2) Meningkatkan ilmu pengetahuan, terutama di bidang IT.
3) Mengharumkan nama SMA Negeri 1 Parepare dalam berbagai kegiatan IT di tingkat kota maupun nasional.
4) Menciptakan karakter positif bagi setiap anggota.
5) Membangun kultur komunitas yang patuh pada aturan.

c. Keanggotaan

Keanggotaan KITS terbuka bagi seluruh siswa/i SMA Negeri 1 Parepare yang telah melalui proses Pendidikan Dasar (Diksar). Diksar merupakan kegiatan pelatihan dan pembinaan bagi calon anggota untuk menjadi anggota yang bertanggung jawab. Adapun struktur keanggotaan terbagi menjadi:

1) Calon Anggota: Siswa/i yang akan menjadi anggota aktif.
2) Anggota Aktif: Anggota yang telah mengikuti Diksar, aktif dalam setiap kegiatan, dan belum melaksanakan Laporan Pertanggungjawaban akhir.
3) Anggota Pasif: Anggota yang tidak lagi aktif dalam kegiatan KITS namun masih berstatus sebagai siswa/i SMA Negeri 1 Parepare.
4) Anggota Kehormatan: Anggota yang telah melalui masa pasif dan telah lulus dari SMA Negeri 1 Parepare.

Setiap jenjang keanggotaan memiliki hak dan kewajiban yang telah diatur, seperti hak bicara, hak memilih dan dipilih (untuk anggota aktif), serta kewajiban menjaga nama baik komunitas dan menaati AD/ART .

d. Struktur dan Atribut Organisasi

Musyawarah Besar anggota memiliki otoritas tertinggi dalam komunitas. Ketua Umum, Wakil Ketua Umum, Sekretaris, Wakil Sekretaris, dan Bendahara adalah 5 inti kepengurusan. Masa jabatan ketua umum tidak boleh lebih dari satu tahun.Untuk menunjang pengembangan minat, KITS memiliki lima divisi teknis:

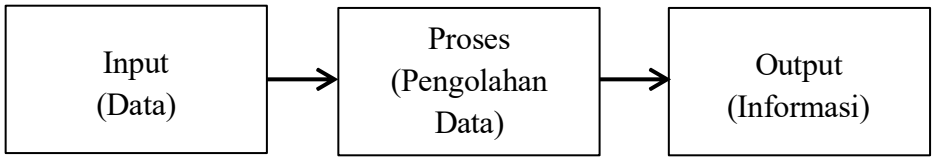
1) Hardware: Mempelajari komponen fisik komputer.
2) Software: Mempelajari perangkat lunak, aplikasi, instalasi konfigurasi dan manajemen perangkat lunak.
3) Video Editing: Mempelajari teknik dasar hingga menengah terkait penyuntingan video.
4) Networking: Mempelajari teknik dasar pengetahuan tentang jaringan..
5) Desain Grafis: Mempelajari keilmuan dasar hingga menegah desain komunikasi visual.

Komunitas ini memiliki atribut resmi yang penggunaannya diatur secara ketat, di antaranya adalah Lambang, Pakaian Dinas Harian (PDH), dan Topi . Lambang KITS memiliki makna filosofis pada setiap elemennya, mulai dari bentuk lima sudut yang melambangkan lima divisi

hingga makna warna seperti hitam (keseriusan), merah (kekuatan dan keberanian), dan putih (kesucian).

5. Pengolahan Data

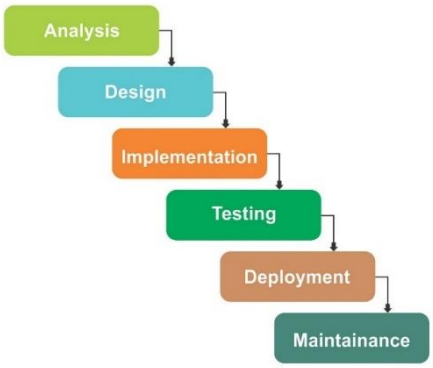
Proses mengubah data mentah menjadi informasi berguna dikenal sebagai pengolahan data. Menurut Kristanto (2018), "waktu yang digunakan untuk menggambarkan perubahan bentuk data menjadi informasi yang memiliki kegunaan" adalah definisi pengolahan data. Definisi ini sejalan dengan penelitian yang dilakukan Sandy (2018), yang menekankan pentingnya efisiensi waktu dalam pengelolaan data berbasis web.



Gambar 2. 2 Siklus Pengolahan Data

Dari gambar di atas dapat disimpulkan bahwa pengolahan data adalah suatu proses dari data menjadi sebuah informasi.

6. Metode Waterfall



Gambar 2. 3 Fase Metode Waterfall
(Sumber: medium.com)

Metode Waterfall diusulkan oleh Winston W. Royce pada tahun 1970 dan dianggap sebagai salah satu model SDLC paling awal dalam rekayasa perangkat lunak adalah pendekatan klasik dalam pengembangan perangkat lunak yang berjalan secara linear dan berurutan, dengan setiap tahap diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya.Metode Waterfall memiliki tahapan pengembangan yang sistematis dan berurutan. Berikut penjelasan detail setiap tahapannya:

a. *Analysis* (Analisis Kebutuhan)

Tahap analisis merupakan tahap awal dalam pengembangan sistem dimana pengembang melakukan pengumpulan informasi secara menyeluruh mengenai kebutuhan pengguna terhadap perangkat lunak yang akan dikembangkan. Pada tahap ini dilakukan identifikasi dan pemahaman mendalam terhadap kebutuhan fungsional dan non-fungsional sistem melalui survei, wawancara, dan diskusi dengan pengguna serta pemangku kepentingan. Semua informasi yang

dikumpulkan kemudian dianalisis dan diolah untuk mendapatkan spesifikasi kebutuhan yang lengkap.

b. *Design* (Perancangan)

Proses menerjemahkan kebutuhan yang telah dianalisis ke dalam desain perangkat lunak yang spesifik dikenal sebagai tahap perancangan. Pada tahap ini, dirancang arsitektur sistem, desain antarmuka pengguna, struktur data, dan desain modul perangkat lunak. Tujuan utama tahap ini adalah memberikan arahan yang jelas bagi tim pengembang tentang cara mengimplementasikan perangkat lunak dan memberikan gambaran menyeluruh tentang sistem yang akan dibangun.

c. *Implementation* (Implementasi)

Tahapan ini merupakan proses pengkodean atau pembuatan program berdasarkan desain yang telah dirancang sebelumnya. Pada tahap ini, tim pengembang menggunakan bahasa pemrograman yang telah ditentukan untuk menghasilkan perangkat lunak sesuai dengan spesifikasi desain. Implementasi dilakukan secara sistematis dengan mengikuti panduan dan dokumentasi yang telah dibuat pada tahap sebelumnya..

d. *Testing* (Pengujian)

Tahapan pengujian ini akan dilakukan setelah implementasi selesai untuk memastikan perangkat lunak berfungsi sesuai dengan persyaratan yang telah ditentukan. Pengujian meliputi berbagai aspek seperti pengujian fungsionalitas, pengujian kesalahan, pengujian

integrasi, dan pengujian kinerja. Tujuan utama tahap tersebut adalah mencari dan memperbaiki masalah-masalah sebelum perangkat lunak digunakan oleh pengguna akhir.

e. *Deployment* (Penerapan)

Tahapan *deployment* merupakan tahapan implementasi *software* ke pengguna/*user*, melakukan tes/pengujian apakah sistem tersebut telah sesuai dengan yang dilaksanakan.

f. *Maintenance* (Pemeliharaan)

Setelah perangkat lunak diluncurkan dan digunakan oleh pengguna, tim pengembang melakukan pemeliharaan rutin yang mencakup perbaikan bug, pembaruan sistem, dan peningkatan fitur sesuai dengan kebutuhan yang berkembang. Pemeliharaan dilakukan secara berkelanjutan untuk memastikan sistem yang telah dibuat tetap berjalan optimal dan sesuai dengan perubahan kebutuhan.

Beberapa keuntungan utama dari metode Waterfall dalam pengembangan sistem adalah sebagai berikut: memiliki alur kerja (*workflow*) yang jelas dan terstruktur sehingga setiap tim dapat bekerja sesuai tahapan yang telah ditentukan; menghasilkan dokumentasi yang terorganisir dengan baik karena setiap tahap harus diselesaikan secara lengkap sebelum melanjutkan ke tahap berikutnya; dapat menghemat biaya karena klien tidak dapat terlibat dalam proses pengembangan; dandan cocok untuk proyek berskala besar dengan kebutuhan yang jelas dan stabil.

Di sisi lain, metode Waterfall juga memiliki beberapa kelemahan yang perlu diperhatikan, yaitu: kurang fleksibel dalam mengakomodasi perubahan karena

setiap tahap harus diselesaikan sebelum melanjutkan ke tahap berikutnya, membutuhkan waktu pengembangan yang relatif lama karena proses dilakukan secara sekuensial, sulit untuk melakukan perbaikan jika ditemukan kesalahan di tahap akhir, dan memerlukan tim yang solid dengan koordinasi yang baik karena kegagalan satu tim dapat mempengaruhi keseluruhan proses pengembangan.

Batasan Cakupan Metode Waterfall

Dalam proposal penelitian Rancang Bangun Aplikasi *Mobile learning (M-Learning)* Komunitas IT Smansa Parepare, model *Waterfall* adalah metode pengembangan perangkat lunak yang digunakan, tetapi metode ini memiliki batasan, seperti berikut:

1) Metode Waterfall menggunakan tahapan sistematis dan berurutan. Tahapan ini terdiri dari analisis kebutuhan (requirements analysis), perancangan sistem (system design), implementasi (implementation), integrasi dan pengujian (integration & *testing*), penyebaran, dan pemeliharaan (maintenance). Sebelum melanjutkan ke tahap berikutnya, setiap tahap harus diselesaikan secara lengkap.
2) Pengumpulan Kebutuhan Terbatas pada Awal Proses Pengumpulan data kebutuhan dilakukan di awal melalui wawancara langsung atau dokumen dari lembaga terkait. Perubahan kebutuhan setelah tahap ini tidak menjadi fokus utama karena sifat linear metode.

3) Fokus pada Pengembangan Aplikasi *Mobile* dengan Fitur Tertentu. Pengembangan aplikasi dibatasi untuk platform *mobile* dengan fitur-fitur spesifik seperti pengelolaan master data, kurikulum pembelajaran, manajemen database anggota, evaluasi penilaian, forum diskusi, presensi digital serta jadwal dan laporan kegiatan.
4) Akses Terbatas pada Pengguna Terdaftar. Aplikasi hanya dapat diakses oleh pengguna/anggota komunitas yang sudah terdaftar dan memiliki akun valid sesuai tiga tingkat akses: pembina komunitas, pengurus komunitas, dan anggota komunitas.
5) Tidak Mengakomodasi Perubahan Dinamis Selama Proses. Karena model Waterfall bersifat sekuensial tanpa iterasi antar tahapan selama proses berjalan maka perubahan signifikan selama pengembangan sulit diakomodir secara fleksibel.
6) Pemeliharaan Dilakukan Setelah Deployment.

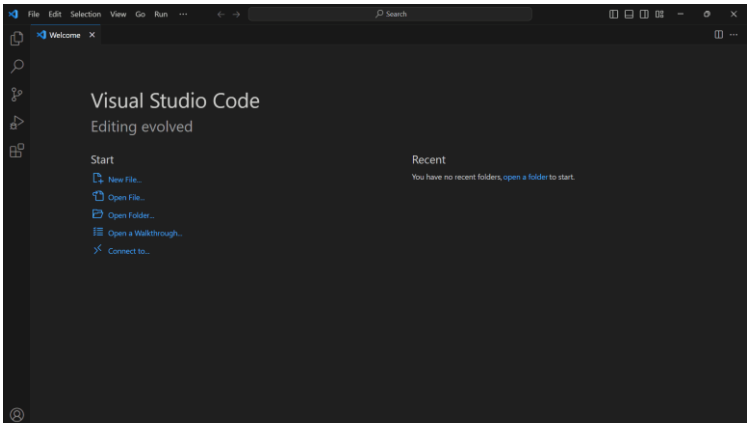
   Pemeliharaan aplikasi dilakukan secara berkelanjutan setelah peluncuran untuk memperbaiki bug serta menyesuaikan fitur sesuai kebutuhan pengguna yang berkembang.

7. Microsoft Visual Studio Code



Gambar 2. 4 Logo Aplikasi *Microsoft Visual Studio Code*

VS Code adalah editor kode sumber yang dikembangkan oleh Microsoft dan dapat diakses melalui browser web. *VS Cod*e adalah lingkungan pengembangan terintegrasi (*IDE*) yang ringan namun kuat yang mendukung berbagai bahasa pemrograman seperti *C, C#, C++, Java, JavaScript, dan Python*, antara lain..



Gambar 2. 5 Tampilan Aplikasi *Microsoft Visual Studio Code*

8. *Website*

Website, juga disebut sebagai situs web, adalah kumpulan halaman web yang saling terhubung dan dapat diakses melalui browser web. *Website* ini menyimpan informasi dalam bentuk teks, gambar, video, audio, dan elemen multimedia lainnya di server web, yang dapat diakses melalui protokol HTTP/HTTPS.

Berdasarkan sifat konten dan interaksinya, *Website* dapat dikategorikan setidaknya menjadi tiga jenis utama, adapun sebagai berikut:

a. Statis

Web statis merupakan jenis *Website* yang kontennya bersifat tetap dan jarang mengalami perubahan. *Website* ini dibangun menggunakan HTML dan CSS murni tanpa menggunakan database. Interaksi pengunjung terbatas hanya pada membaca informasi yang ditampilkan. *Website* statis cocok digunakan untuk company profile atau *Website* informasi sederhana yang tidak memerlukan pembaruan konten secara rutin. Kelebihan utama *Website* statis adalah proses loading yang lebih cepat karena strukturnya yang sederhana.

b. Dinamis

Web dinamis adalah jenis *Website* yang memungkinkan kontennya untuk diperbarui secara berkala. *Website* ini menggunakan database untuk menyimpan dan mengelola konten, serta dibangun menggunakan bahasa pemrograman server-side seperti PHP atau Python. Administrator dapat dengan mudah mengelola konten melalui panel administrasi tanpa perlu mengubah kode program. *Website* dinamis umumnya digunakan untuk portal berita, blog, atau e-commerce yang membutuhkan pembaruan konten secara teratur dan dapat menyajikan konten yang dipersonalisasi sesuai kebutuhan pengguna.

c. Interaktif

Web interaktif adalah situs web yang memungkinkan pengguna berinteraksi satu sama lain. Biasanya terdiri dari forum diskusi dan *blog*, dengan moderator yang mengatur jalan diskusi..

9. *PHP (Hypertext Processor)*

PHP adalah bahasa pemrograman untuk pengembangan web yang memungkinkan pembuatan situs yang dinamis, sehingga proses pemeliharaan menjadi lebih mudah dan efisien. Kodenya dapat dimasukkan langsung ke dalam berkas *HTML*. *PHP* mudah diakses dan digunakan secara luas karena bersifat sumber terbuka, atau *open source*, dan dapat diunduh secara gratis melalui situs resminya (Suhartanto, 2017).

PHP memiliki banyak keuntungan, seperti: sangat mudah untuk dipelajari karena sintaksnya relatif akrab bagi pemula; ekosistem yang matang dengan banyak pustaka, *framework* (seperti *Laravel*, *CodeIgniter*), dan dokumentasi yang luas; kompatibilitas lintas platform (*Windows*, *Linux*, *macOS*), serta integrasi baik dengan server web (*Apache*, *Nginx*) dan berbagai basis data (*MySQL*/*MariaDB*, *PostgreSQL*, *SQLite*); kinerja yang baik untuk aplikasi kontemporer dibantu oleh mekanisme *caching*/*OPcache*.

```
<html>
  <head>
      <title> Contoh Sederhana </title>
  </head>
  <body>
      <?php
          echo("Hallo apa kabar? Nama saya PHP script");
      ?>
  </body>
</html>
```

Gambar 2. 6 Contoh Sintaks PHP

10. MySQL

Salah satu jenis sistem manajemen basis data relasional (RDBMS) yang paling umum digunakan saat mengembangkan aplikasi berbasis web adalah MySQL.Menurut Madcoms (2020), MySQL menggunakan *Structured Query Language* (SQL) yang memungkinkan pengguna mengelola basis data dengan operasi seperti membuat, membaca, memperbarui, dan menghapus data. MySQL dikenal sebagai perangkat lunak sumber terbuka yang mampu beroperasi pada berbagai platform, termasuk Windows, Linux, dan macOS, serta memiliki performa tinggi dalam menangani basis data dengan skala besar (Nugroho, 2021).

Keunggulan utama dari MySQL antara lain kecepatan, kestabilan, dan kemudahan penggunaannya, serta dukungan komunitas yang luas sehingga terus berkembang dalam hal fitur dan keamanannya (Saputra & Dewi, 2019). Oleh sebab itu, MySQL sering menjadi pilihan utama dalam pembangunan aplikasi web yang membutuhkan pengelolaan data secara dinamis dan efektif.

11. Kamus Data

Menurut Kendall & Kendall (2013), kamus data adalah sebuah referensi terpusat yang berfungsi untuk mengumpulkan, mengoordinasikan, dan mendefinisikan setiap istilah data dalam sebuah sistem. Mirip seperti kamus biasa, kamus data menjadi acuan untuk memahami makna setiap elemen data.

Alat ini sangat penting dalam dua tahap utama pengembangan sistem: analisis dan perancangan. Pada tahap analisis, kamus data menjadi jembatan komunikasi antara analis sistem dan pengguna untuk memastikan pemahaman yang sama mengenai data yang diproses. Sementara itu, pada tahap perancangan, kamus data

digunakan sebagai panduan untuk membangun input, output, laporan, dan basis data (database). Fungsi lainnya adalah untuk mencegah kebingungan akibat penggunaan istilah yang sama untuk data yang berbeda, karena disusun secara alfabetis.

Tabel 2. 1 Contoh Tabel Kamus Data

Nama Kolom	Tipe data	Ukuran	Deskripsi
id	Int	11	ID unik untuk setiap pengguna.
nba	Varchar	10	Nomor Baku Anggota.
email	Varchar	100	Alamat email pengguna.
nama	Varchar	100	Nama lengkap pengguna.
password	Varchar	255	Hash dari password pengguna.
role	Enum	-	Peran pengguna: 'anggota', 'pengurus', 'pembina'.
member_type	Enum	-	Jenis keanggotaan: 'aktif' (akses penuh) atau 'pasif' (akses terbatas).

Agar informatif, sebuah kamus data harus mencakup elemen-elemen berikut:

1. Nama Arus Data: Nama yang sama persis seperti yang tertera di DFD agar mudah dicari dan dirujuk.
2. Alias: Nama lain atau sinonim untuk arus data yang mungkin digunakan oleh departemen atau pengguna yang berbeda.
3. Bentuk Data: Format data, apakah berupa dokumen fisik, laporan tercetak, atau kumpulan field digital.
4. Arus Data: Informasi mengenai asal (sumber) dan tujuan (destinasi) data.
5. Penjelasan: Deskripsi singkat untuk memperjelas fungsi dan konteks dari arus data tersebut.
6. Periode: Waktu atau frekuensi kapan data tersebut mengalir dalam sistem.
7. Volume: Jumlah data yang ditransmisikan, biasanya mencakup volume rata-rata dan puncak.

8. Struktur Data: Rincian komponen atau elemen-elemen data yang membentuk arus data tersebut.

12. Laragon

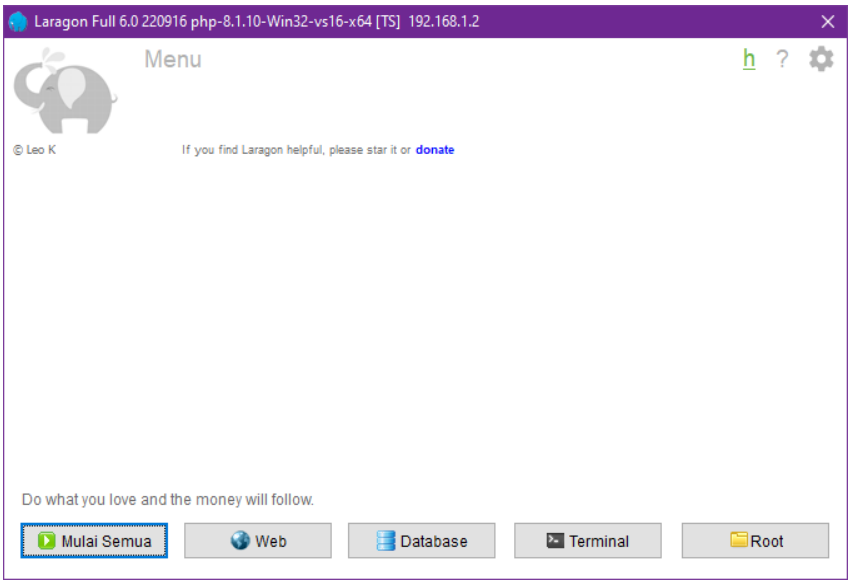


Gambar 2. 7 Logo Aplikasi *Laragon*

Laragon adalah lingkungan pengembangan serbaguna yang dapat digunakan untuk berbagai ekosistem bahasa pemrograman, seperti *PHP*, *Node.js*, *Python*, *Java*, *Go*, dan *Ruby*. Itu portabel, terisolasi, berperforma tinggi, dan andal. Sangat cocok untuk pengembangan dan pengelolaan aplikasi web kontemporer karena *platform* ini sederhana, mudah digunakan, dan mudah diekstensikan.

Dalam proses pengembangan, *Laragon* menekankan stabilitas, kesederhanaan, fleksibilitas, dan kebebasan. *Laragon* biasanya lebih ringan dan dapat diakses dibandingkan *XAMPP* karena bersifat *self-contained* dan tidak membutuhkan konfigurasi sistem yang signifikan. Ini menawarkan orkestrasi layanan yang cepat dengan satu klik start/stop, kemampuan untuk membuat virtual *host* dan *domain.test* secara otomatis, dan opsi *HTTPS* lokal yang tidak memerlukan konfigurasi yang signifikan untuk mempercepat siklus pengembangan. Selain itu, *Laragon* memudahkan perubahan versi komponen, seperti *PHP*, basis data, dan opsi web server *Apache*/*Nginx*, serta integrasi alat kontemporer seperti *Composer*

dan npm secara bawaan, yang seringkali diperlukan untuk penyesuaian manual pada *XAMPP*.



Gambar 2. 8 Tampilan Control Panel Aplikasi *Laragon*

13. *Progressive Web Apps (PWA)*

PWA adalah aplikasi berbasis web yang menggabungkan keunggulan dari aplikasi web dan native mobile. Seperti yang dinyatakan oleh Syahputra dan Yudianto (2018), *Progressive Web Apps* menawarkan pengalaman pengguna yang serupa dengan aplikasi mobile, seperti bekerja secara offline, mendukung *push notification*, memiliki tampilan yang responsif, serta mampu diinstal pada perangkat pengguna tanpa melalui toko aplikasi resmi. Dengan memanfaatkan teknologi *Service Worker* dan *Web App Manifest*, PWA mampu menyimpan data aplikasi sehingga dapat diakses dalam kondisi tanpa koneksi internet (Utomo, 2020).

Penggunaan teknologi PWA memberikan beberapa manfaat, seperti efisiensi dalam pengembangan karena satu kode sumber dapat digunakan pada berbagai

platform, serta mengurangi biaya pemeliharaan aplikasi dibandingkan dengan aplikasi native (Setiawan & Pratama, 2021). Dengan demikian, PWA menjadi alternatif ideal dalam menjawab kebutuhan aplikasi modern yang ringan, cepat, dan mudah diakses dari berbagai perangkat dan platform.

a. Karakteristik Progressive Web App

Sebuah aplikasi web dapat diklasifikasikan sebagai PWA jika memenuhi serangkaian karakteristik kunci yang membedakannya dari situs web tradisional. Karakteristik ini menjadi tolok ukur dalam pengembangan dan evaluasi PWA.

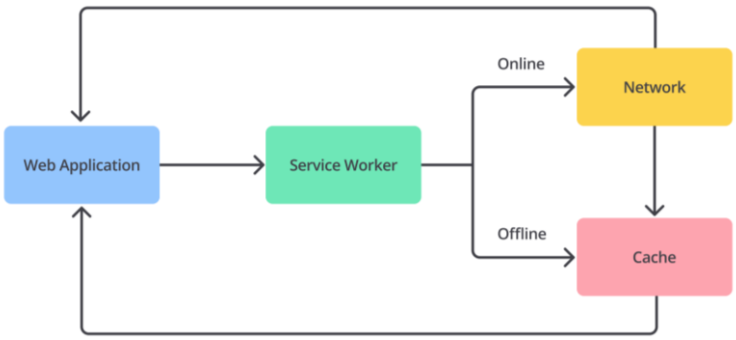
1) Progresif (*Progressive*): Seperti namanya, PWA dibangun dengan prinsip progressive enhancement. Aplikasi harus dapat bekerja untuk setiap pengguna, terlepas dari peramban yang digunakan, karena ia dibangun dengan basis yang solid dan fungsionalitasnya ditingkatkan secara berlapis.
2) Independen dari Konektivitas (*Connectivity Independent*): Melalui penggunaan *Service Worker*, PWA dapat berfungsi dalam kondisi jaringan berkualitas rendah atau bahkan saat luring (offline).
3) Serupa Aplikasi (*App-like*): PWA harus mengadopsi model navigasi dan interaksi yang umum ditemukan pada aplikasi native (dikenal sebagai *App Shell Model*), tanpa menyertakan elemen antarmuka peramban seperti address bar.
4) Selalu Terbaru (*Fresh*): Konten dan fungsionalitas aplikasi selalu dalam versi termutakhir berkat proses pembaruan di latar belakang yang dikelola oleh Service Worker.

5) Aman (*Safe*): Aplikasi harus disajikan melalui protokol HTTPS untuk mencegah penyadapan (*snooping*) dan memastikan konten tidak dimanipulasi oleh pihak ketiga. Ini adalah prasyarat mutlak.
6) Dapat Ditemukan (*Discoverable*): Konten di dalam PWA dapat diindeks oleh mesin pencari seperti Google, menjadikannya dapat ditemukan layaknya situs web biasa.
7) Dapat Diinstal (*Installable*): Pengguna dapat menambahkan PWA ke layar utama perangkat mereka tanpa melalui proses pengunduhan dari toko aplikasi (*app store*), memberikan akses yang lebih cepat dan mudah.

b. Komponen Teknologi Inti PWA

Untuk mewujudkan karakteristik di atas, PWA bergantung pada tiga pilar teknologi utama yang bekerja secara sinergis.

1) *Service Worker*



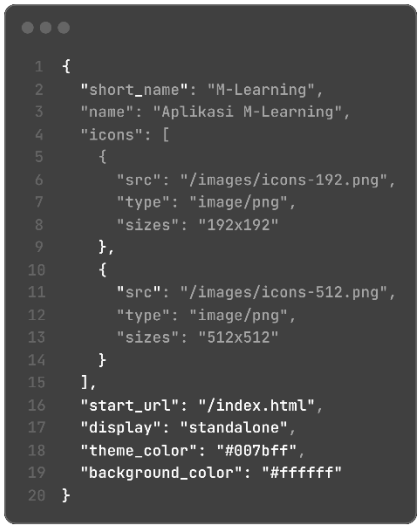
Gambar 2. 9 Konsep *Service Worker*
(Sumber: dicoding.com)

Service Worker adalah komponen teknologi paling fundamental dalam sebuah PWA. Ia merupakan sebuah skrip JavaScript yang dijalankan oleh peramban di latar belakang, terpisah dari halaman web utama. Service

Worker bertindak sebagai proksi jaringan (client-side proxy) yang dapat di-program, yang berada di antara aplikasi web, peramban, dan jaringan.

2) *Web App Manifest*

Web App Manifest adalah sebuah berkas *JSON (JavaScript Object Notation)* sederhana yang memberikan informasi dan metadata tentang aplikasi kepada peramban. Berkas ini memungkinkan aplikasi web untuk diintegrasikan lebih dalam dengan sistem operasi perangkat, sehingga memberikan pengalaman yang lebih menyerupai aplikasi native saat "diinstal".



```
{
  "short_name": "M-Learning",
  "name": "Aplikasi M-Learning",
  "icons": [
    {
      "src": "/images/icons-192.png",
      "type": "image/png",
      "sizes": "192x192"
    },
    {
      "src": "/images/icons-512.png",
      "type": "image/png",
      "sizes": "512x512"
    }
  ],
  "start_url": "/index.html",
  "display": "standalone",
  "theme_color": "#007bff",
  "background_color": "#ffffff"
}
```

Gambar 2. 10 Contoh file *manifest.json*
(Sumber: Kode Program skripsi)

3) *HTTPS (Hypertext Transfer Protocol Secure)*

HTTPS adalah prasyarat keamanan yang tidak dapat ditawar untuk PWA. Karena Service Worker memiliki kemampuan untuk mencegat dan memodifikasi permintaan jaringan, menjalankannya di atas koneksi HTTP

yang tidak aman akan membuka celah kerentanan serius terhadap serangan *man-in-the-middle.*

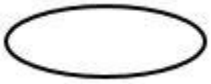
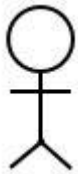
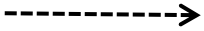
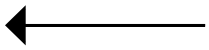
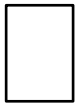
14. UML (Unified Modelling Language)

Untuk membangun sistem berorientasi objek, *Unified Modeling Language* (*UML*) adalah metode pemodelan visual. Standar ini pertama kali dibuat oleh *OMG* dengan merilis versi 1.0 pada Januari 1997. Sering dianggap sebagai "*blueprint*" pengembangan perangkat lunak, *UML* dianggap sebagai bahasa standar untuk memvisualisasikan, merancang, dan mendokumentasikan sistem perangkat lunak (Dicoding intern, 2021). Proses perancangan dan pemodelan yang matang merupakan dasar dari kualitas sistem yang andal. Dalam hal ini, *UML* menyediakan notasi dan konvensi yang terstruktur untuk membantu pemangku kepentingan memahami lebih baik analisis kebutuhan dan perancangan arsitektur, dan meningkatkan keterlacakan dokumentasi sepanjang siklus hidup perangkat lunak.:

a. *Use case diagram*

Dalam UML, *Use case diagram* adalah representasi visual yang menggambarkan himpunan interaksi antara aktor (pengguna atau sistem eksternal) dan sistem untuk mencapai tujuan tertentu. Diagram ini menampilkan aktor, *Use case*, dan hubungan di antara mereka dengan cara-cara seperti menggabungkan, menambah, memperluas, dan generalisasi. Ini membantu menetapkan batas sistem, memperjelas cakupan layanan, dan memudahkan komunikasi dan validasi.

Tabel 2. 2 Simbol Use case diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*Use case*	Menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktif, yang dinyatakan dengan menggunakan kata kerja.
2		*Actor*	*Actor* atau Aktor adalah *abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Perlu dicatat bahwa aktor berinteraksi dengan *use case*, tetapi tidak memiliki control terhadap *use case*
3		*Association*	Apa yang menghubungkan antara objek satu dengan objek yang lainnya.
4		*Generalization*	Hubungan dimana objek anak berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk.
5		*Include*	Pemanggilan *use case* oleh *use case* lain, contohnya adalah pemanggilan sebuah fungsi program.
6		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemanyang bergantung padanya elemen yang tidak mandiri.
7		*Extend*	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan
8		*Extend*	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan

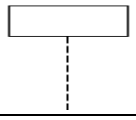
Lanjutan Tabel 2. 2

NO	GAMBAR	NAMA	KETERANGAN
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya.
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

b. *Sequence Diagram*

Sequence diagram adalah artefak UML yang memungkinkan untuk memodelkan perilaku dinamis sistem dengan menampilkan urutan interaksi antar aktor dan objek yang didasarkan pada dimensi waktu. Sementara aliran waktu bergerak ke bawah pada sumbu vertikal, setiap partisipan (aktor atau objek) digambarkan sebagai garis vertikal yang disebut lifeline, yang merepresentasikan keberadaannya selama periode waktu tertentu. Pada lifeline ini, akan muncul kotak persegi panjang yang disebut kotak aktivasi (*activation box*) untuk menandakan periode saat objek tersebut aktif bekerja atau memproses sebuah tugas. Interaksi antar objek divisualisasikan melalui pesan sinkron (menunggu balasan) dan asinkron (tidak menunggu balasan) yang digambarkan sebagai panah. Panah ini tidak hanya menunjukkan pemanggilan operasi, tetapi juga pengembalian nilai (reply message), serta proses pembuatan atau penghancuran objek.

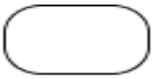
Tabel 2. 3 Simbol *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Suatu objek berfungsi untuk melakukan interaksi antarmuka.
2		*Message*	Berfungsi untuk berkomunikasi antara objek yang memiliki in*form*asi tentang perilaku yang terjadi
3		*Message*	Berfungsi untuk berkomunikasi antara objek yang memiliki in*form*asi tentang perilaku yang terjadi

c. *Activity diagram*

Diagram Aktivitas *UML* memodelkan alur kontrol dan/atau data proses dari awal hingga akhir. *Action*, *control*/*object flow*, *decision-merge*, *fork-join*, *guard*, dan *swimlane* untuk mengelompokkan tanggung jawab adalah elemen utama. Sebagai dasar untuk komunikasi lintas pemangku kepentingan dan penyusunan skenario uji, diagram ini menunjukkan urutan, percabangan, dan kompetisi. Ini membantu memvalidasi kebutuhan fungsional, mengurangi ambiguitas alur kerja, dan menemukan *bottleneck*.

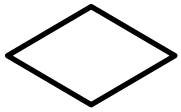
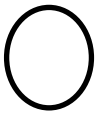
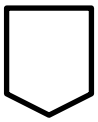
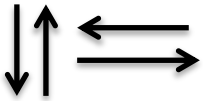
Tabel 2. 4 Simbol *Activity diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Actifity*	Menampilkan suatu objek bagaimna cara saling berinteraksi antara satu dan lainnya.
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		*Initial Node*	Berfungsi untuk mengawali suatu objek
4		*Actifity Final Node*	Berfungsi untuk mengakhiri suatu objek
5		*Fork Node*	Suatu alur yang akan berubah menjadi beberapa aliran pada kondisi tertentu.

d. *Flowchart*

Flowchart adalah representasi grafis dari langkah-langkah suatu proses atau algoritme. Ini menggunakan simbol standar seperti terminator (mulai/selesai), proses, input/output, keputusan (berbentuk belah ketupat), konektor, dan panah alir untuk menunjukkan urutan, percabangan, dan pengulangan. Flowchart mempermudah pemahaman logika, mengurangi ambiguitas, dan membantu menemukan bottleneck dan kesalahan desain sebelum implementasi dengan menyajikan alur kerja secara visual dan terstruktur.

Tabel 2. 5 Simbol *Flowchart*

NO	GAMBAR	NAMA	KETERANGAN
1.		*Terminal*	Berfungsi untuk mengawali suatu program dan mengakhirinya.
2.		*Input / Output*	Berfungsi untuk melakukan suatu proses input atau output.
3.		*Process*	Berfungsi untuk tempat melakukan suatu proses yang terjadi pada sistem.
4		*Decision*	Berfungsi untuk menyelesaikan suatu kondisi yang akan menghasilkan dua jawaban berbeda iya/tidak
5.		*Connector*	Berfungsi sebagai sambungan proses yang berbeda tetapi tetap halaman yang sama.
6.		*Offline Connector*	Berfungsi sebagai sambungan proses yang berbeda tetapi pada halaman yang berbeda.
7.		*Predefined Process*	Berfungsi sebagai tempat pengolahan yang akan diberikan suatu harga.
8.		*Punched Card*	Berfungsi sebagi penyampaian *input* bermula dari kartu atau *output* ditulis pada kartu.
9.		*Document*	Berfungsi sebagai percetakan dalam bentuk suatu dokumen.
10.		*Flow*	Menghubungkan antara simbol yang satu dengan simbol yang lain.

Lanjutan Tabel 2.5

NO	GAMBAR	NAMA	KETERANGAN
11.		*Manual Operation*	Menunjukkan pengolahan yang tidak dilakukan oleh *computer*.
12.		*Manual Input*	Dilakukan untuk memasukkan data secara manual *on-line keyboard.*
13.		*Predefine Process*	Digunakan untuk pelaksanaan suatu bagian (*sub-program*) / prosedur.

15. *White box testing*

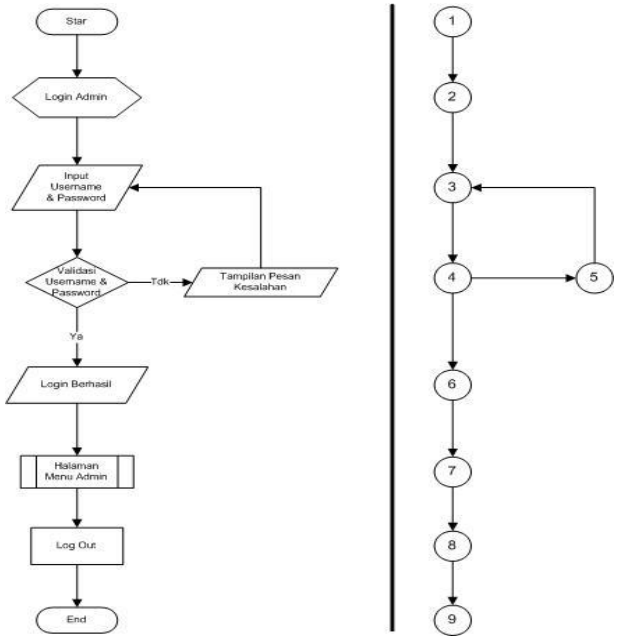
White-box testing adalah pengujian yang memeriksa struktur dan logika internal program. Penguji harus memahami arsitektur dan kode untuk merancang kasus uji *(statement, branch, path, condition*). Tujuannya memastikan alur berjalan benar, data diproses tepat, dan kesalahan terdeteksi sejak awal.

Pada *white box testing* ada beberapa teknik yang digunakan, antara lain sebagai berikut:

a. Basis path *testing* adalah metode yang memungkinkan perancang tast case untuk membuat pengukuran kompleksitas logika dari rancangan prosedural dan menggunakan pengukuran untuk mendefinisikan himpunan basis dari jalur eksekusi.
b. *Flow graph* merupakan notasi sederhana guna mempresentasi control flow.
c. *Cyclomatic complexity* yang digunakan guna mengetahui jumlah path yang perlu dicari. Di mana *cyclomatic complexity* merupakan metric perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logika program.

d. *Branch coverage* merupakan pengujian yang didesain agar masing-masing branch code yang paling tidak diuji satu kali.
e. *Loop testing* merupakan *testing* pada berbagai looping/perulangan yang terdapat pada program, seperti for, while, dan do-while. Dengan pengujian ini Anda bisa mengecek kondisi perulangan, sudahkah berjalan dengan benar ataupun tidak.
f. *Condition* coverage merupakan teknik pengujian yang bertujuan guna melakukan uji semua kode untuk menghasilkan value FALSE atau TRUE. Tester bisa memastikan software dalam beroperasi dan menghasilkan output sesuai input user.
g. *Multiple condition coverage* merupakan teknik yang digunakan guna menguji semua kombinasi beragam kode yang digunakan dalam berbagi kondisi.
h. *Statement coverage* merupakan teknik yang minimal 1 kali dilakukan guna menguji masing-masing statement pada software.

Berikut ini adalah contoh pengujian *white box* pada sistem login yang terdapat pada Aplikasi ini :



Gambar 2. 11 Contoh Pengujian *White Box*
(Sumber: Skripsi)

16. *Black box Testing*

Metode pengujian berbasis spesifikasi yang dikenal sebagai *Blackbox testing* menilai fungsi perangkat lunak dari sudut pandang pengguna tanpa memeriksa struktur internal atau kode sumber. *Partitioning equivalence*, analisis nilai batas, *table of decisions*, dan *state-transition* adalah beberapa metode yang digunakan untuk memisahkan kasus uji dari kebutuhan dan antarmuka. Metode ini mengatasi masalah seperti kesalahan validasi input, ketidaksesuaian keluaran, dan masalah integrasi antarmuka. Selain itu, itu juga memastikan kepatuhan terhadap persyaratan. Namun, pendekatan ini kurang efektif untuk menemukan cacat struktural karena tidak melakukan pemeriksaan jalur eksekusi internal.

Ada beberapa teknik-teknik dalam Pengujian *Black box*, antara lain sebagai berikut:

1) *All Pair Testing*

   Teknik ini dikenal juga sebagai *pairwise testing*. Penguji akan menguji semua kemungkinan kombinasi dari pasangan parameter input untuk memastikan perangkat lunak bekerja dengan benar dalam setiap kondisi.

2) *Boundary Value Analysis*

   Teknik ini fokus pada pengujian nilai batas, seperti nilai minimum dan maksimum, untuk menemukan error yang mungkin muncul di area tersebut.

3) *Cause-Effect Graph*

   Teknik ini menggunakan grafik untuk menggambarkan hubungan antara penyebab dan efek dari error. Grafik ini membantu penguji memvisualisasikan skenario pengujian.

4) *Equivalence Partitioning*

   Pada teknik ini, data input dibagi menjadi beberapa partisi atau kelompok yang dianggap setara. Pengujian dilakukan untuk memastikan setiap partisi memberikan hasil yang sesuai.

5) *Fuzzing*

   Teknik fuzzing melibatkan pemberian data input yang tidak sempurna atau acak untuk menemukan bug yang mungkin terlewat.

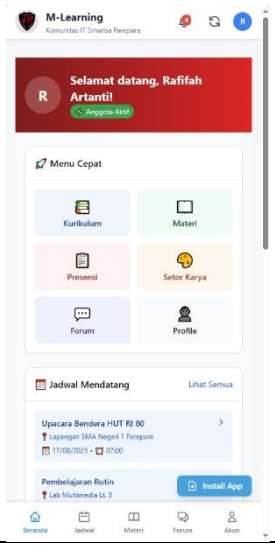
6) *Orthogonal Array Testing*

Teknik ini cocok digunakan jika jumlah input relatif kecil tetapi memiliki tingkat kompleksitas tinggi. Pengujian dilakukan dengan memilih kombinasi input yang optimal.

7) *State Transition Testing*

Teknik ini digunakan untuk menguji perubahan status atau transisi dalam aplikasi, seperti alur navigasi pada antarmuka pengguna (UI).

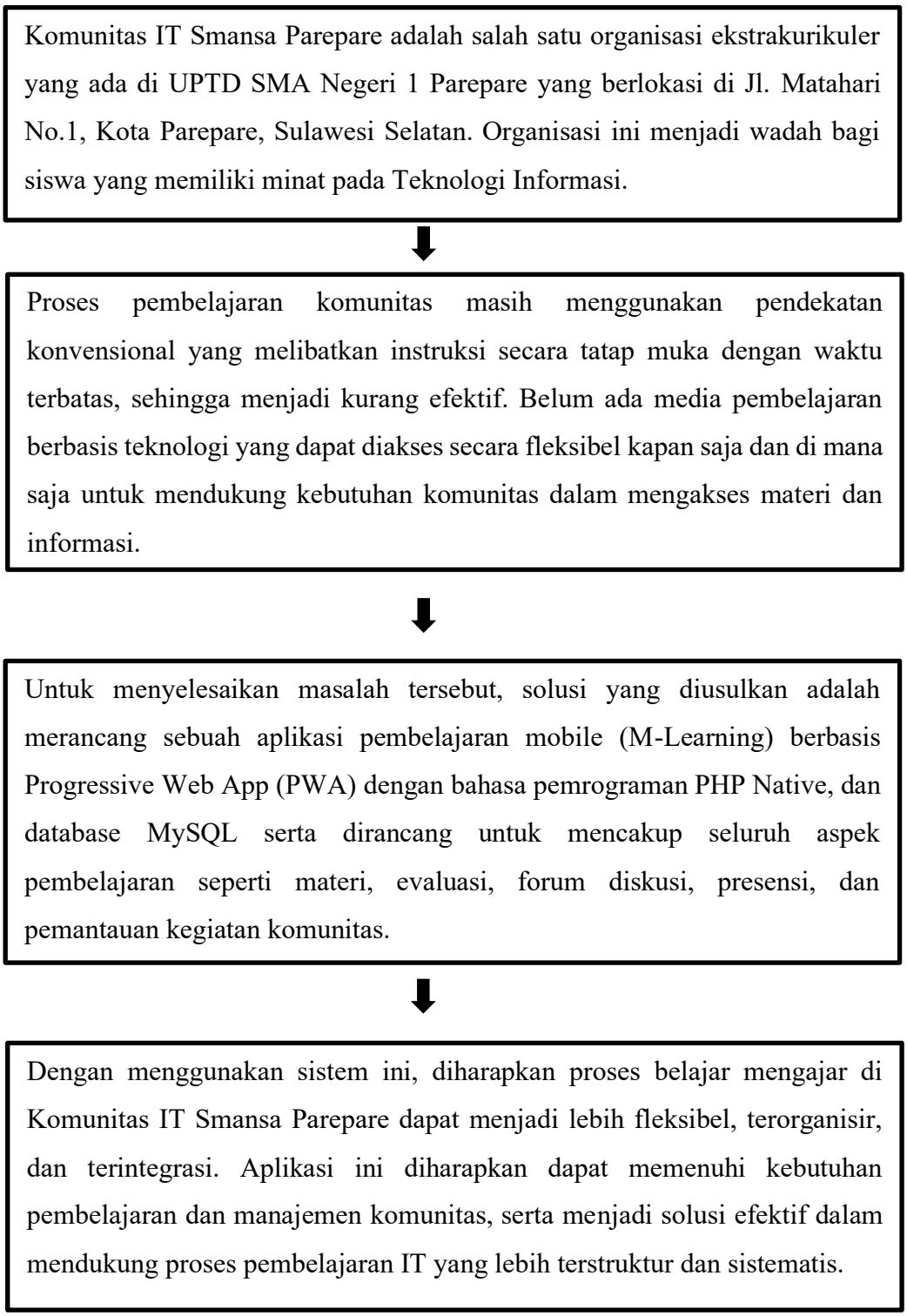
Tabel 2. 6 Contoh Pengujian *Blackbox* fitur *login*

Tes faktor	**Hasil**	**Keterangan**
Memasukkan NBA/Email atau password yang sesuai	✓	Sistem otomatis mengalihkan ke halaman Dashboard sesuai Role/peran pengguna
Tangkapan layar		
		

B. Kajian Hasil Penelitian Terdahulu

1. Harkamsyah Andrianof (2020) – "Perancangan Aplikasi *Mobile learning (M-Learning)* untuk Pembelajaran E-Business bagi Mahasiswa dengan Menggunakan Perangkat Mobile Berbasis *Android*". Penelitian ini menghasilkan prototipe aplikasi *Android Mobile learning (M-Learning)* untuk materi e-business dengan fitur kuis, evaluasi, materi, dan animasi pembelajaran. Tujuan utamanya untuk meningkatkan pemahaman materi e-business kapan saja dan di mana saja.
2. Budi Santoso dkk. (2020) – "Pengembangan Aplikasi *Mobile learning (M-Learning)* Berbasis *Android* Sebagai Sarana Penunjang Pembelajaran dalam Kelas" (SMATIKAJurnal). Aplikasi dikembangkan menggunakan *React* Native/native, berorientasi mobile, menggantikan sistem e-learning web. Menyediakan fitur diskusi, materi, dan ujian yang dapat diakses via smartphone.
3. Hardianty, D. A., Yustiana, I., & Somantri (2022) "Rancang Bangun Aplikasi E-Learning Berbasis *Progressive Web Apps* untuk Menunjang Pembelajaran Online dengan Metode Prototyping" Studi ini menghasilkan aplikasi PWA untuk pembelajaran online (SMA Hayyatan Thayyibah) dengan prototyping. Hasil uji usability mencatat skor 80 %, menunjukkan aplikasi efektif digunakan oleh siswa dan guru serta mendukung absensi, materi, dan interaksi pembelajaran.

C. Kerangka Pikir



Gambar 2.5 Kerangka pikir

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Penelitian ini merupakan Penelitian dan Pengembangan atau *Research and Development* (*R&D*). Metode ini dipilih karena penelitian ini berfokus pada perancangan dan pembuatan sebuah aplikasi baru, yaitu aplikasi *Mobile learning* (*M-Learning*) untuk Komunitas IT Smansa Parepare, serta menguji efektivitas aplikasi tersebut untuk memecahkan masalah yang ada di lapangan.

Penelitian ini tidak sepenuhnya bersifat kuantitatif ataupun kualitatif, melainkan menggabungkan keduanya dalam sebuah siklus pengembangan. Prosesnya dimulai dari analisis kebutuhan di Komunitas IT Smansa Parepare (pendekatan kualitatif melalui observasi dan wawancara), dilanjutkan dengan perancangan dan implementasi aplikasi (pengembangan sistem), dan diakhiri dengan pengujian fungsionalitas sistem menggunakan metode *Black box* dan *white box* untuk memastikan aplikasi berjalan sesuai dengan tujuannya (pendekatan kuantitatif dalam pengujian).

B. Lokasi dan Waktu

Penelitian dilaksanakan di UPTD SMA Negeri 1 Parepare, Jalan. Matahari No. 3 Kota Parepare. Kegiatan penelitian berlangsung selama kurang lebih satu (1) bulan.

C. Alat dan Bahan

Pelaksanaan penelitian memerlukan perangkat dan bahan penunjang untuk pengumpulan, pengolahan, dan pelaporan data. Peralatan dan bahan yang diperlukan antara lain:

1. Alat yang digunakan

a. Perangkat Keras *(Hardware)*

Tabel 3. 1 Perangkat Keras (*Hardware*) yang digunakan

Komponen	Spesifikasi/Detail
Laptop	*Lenovo Thinkpad L380 Yoga*
Prosesor	*Intel Core i5-8250U, 1.7Ghz*
Inti prosesor	*4 core*
RAM	*16 GB*
Layar	*13" FHD IPS*
Penyimpanan (SSD)	*256 GB*

b. Perangkat Lunak *(Software)*

Tabel 3. 2 Perangkat Lunak (*Software*) yang digunakan

Kategori	Detail
Sistem Operasi	*Windows 11*
Desain	*Canva, Photoshop*
Kode Editor	*Visual Studio Code*
Peramban	*Microsoft Edge*
Server Web	*Apache*
Bahasa Pemrograman	*PHP*

2. Bahan penelitian

Bahan penelitian dalam skripsi ini terdiri dari data dan informasi yang berfungsi sebagai landasan analisis dan perancangan sistem aplikasi *M-Learning* untuk Komunitas IT SMANSA Parepare. Data primer

diperoleh melalui wawancara mendalam dengan pengurus komunitas dan observasi langsung di lapangan untuk mengidentifikasi kebutuhan aktual pengguna, sedangkan data sekunder dikumpulkan melalui studi kepustakaan dari jurnal ilmiah, buku, dan penelitian terdahulu yang relevan dengan pengembangan aplikasi pembelajaran berbasis mobile.

D. Teknik Pengumpulan Data

1. Studi kepustakaan

Dilakukan dengan menelusuri dan menelaah sumber tertulis yang relevan buku, literatur ilmiah, artikel, majalah, dan jurnal guna membangun landasan teori, konsep, serta temuan mutakhir yang mendukung penelitian.

2. Wawancara

Pengumpulan data diperoleh melalui tanya jawab langsung dengan pihak terkait, yakni pengurus Komunitas IT SMANSA Parepare, menggunakan pedoman pertanyaan terstruktur/semi-terstruktur untuk menggali informasi yang spesifik.

3. Observasi

Peneliti melakukan pengamatan langsung di lapangan untuk mencatat fakta, perilaku, dan aktivitas yang terkait dengan objek kajian, sehingga memperoleh gambaran empiris mengenai kondisi nyata.

4. Dokumentasi

Peneliti melakukan pengamatan langsung di lapangan untuk mencatat fakta, perilaku, dan aktivitas yang terkait dengan objek kajian, sehingga memperoleh gambaran empiris mengenai kondisi nyata.

E. Teknik Pengembangan Sistem

Metode pengembangan perangkat lunak yang digunakan adalah metode *waterfall.* Langkah-langkah yang dilakukan dalam metode ini adalah:

1. ***Requirements Analysis*** **(Analisis Kebutuhan)**

Tujuan dari langkah ini adalah untuk menentukan kebutuhan fungsional dan nonfungsional, ruang lingkup, dan fitur mana yang akan diprioritaskan. Data dikumpulkan melalui observasi, wawancara, dan penelaahan dokumen dari lembaga yang relevan. Spesifikasi kebutuhan perangkat lunak adalah luaran utama.

2. ***System Design*** **(Perancangan Sistem)**

Perancangan arsitektur, rancangan antarmuka pengguna, model basis data, struktur modul, dan kebutuhan infrastruktur dibuat sesuai dengan SRS. Setelah tahap ini selesai, dokumen desain yang sangat komprehensif dibuat untuk digunakan sebagai acuan implementasi.

3. ***Implementation*** **(Implementasi)**

Standar pengembangan digunakan untuk menerjemahkan desain ke dalam kode program. Aktivitas termasuk mengelola versi, menulis unit, dan menguji unit untuk memastikan bahwa desain sesuai.

4. ***Integration & Testing*** **(Integrasi dan Pengujian)**

Uji integrasi dan sistem menguji fungsi, alur, dan kualitas (seperti kinerja dan keandalan). Cacat modul diperbaiki hingga sistem memenuhi persyaratan penerimaan.

5. *Deployment* (Penerapan/Rilis)

Setelah pengecekan tahap sebelumnya selesai dan aplikasi sesuai dengan permintaan klien, tahap berikutnya adalah penyebaran ke web *server*.

6. *Maintenance* (Pemeliharaan)

Pada tahap ini penulis akan melakukan pemeliharaan terhadap sistem, atau dapat dikatakan pengecekan kembali apakah sistem sudah berjalan tanpa adanya *error*. Dan apabila terdapat kesalahan maka akan dilakukan perbaikan kembali.

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

A. Analisis Aliran Data UML

Dalam bagian ini, akan dibahas *Diagram UML* yang menggambarkan aliran data dan interaksi dalam Aplikasi *Mobile learning (M-Learning)* Komunitas IT Smansa Parepare, yaitu *Use case diagram*, *Activity diagram*, *Sequence diagram* dan rancangan database sebagai berikut:

1. *Use case diagram*

a. Pembina

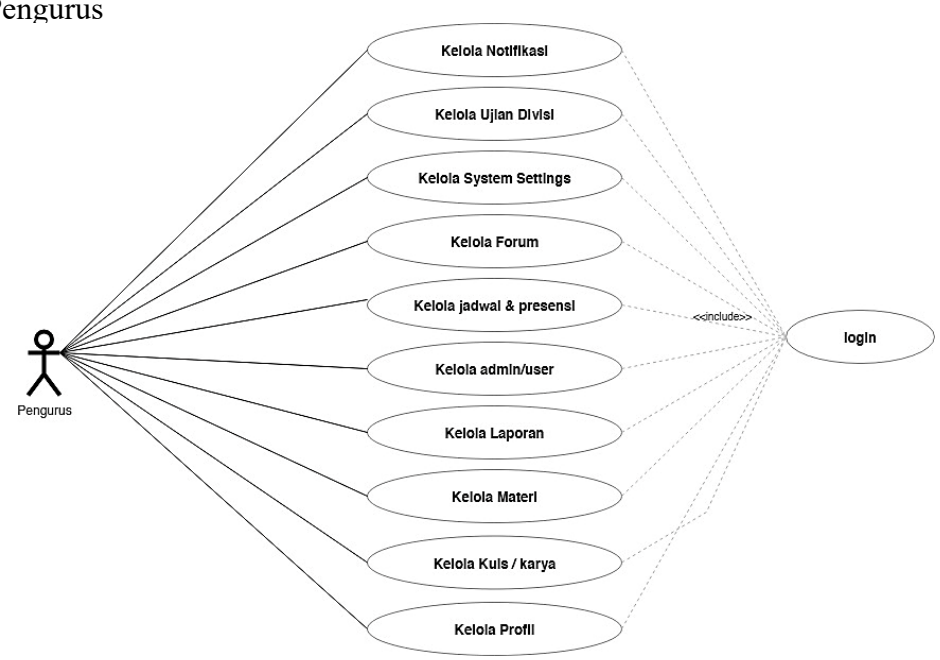


Gambar 4. 1 *Use case diagram* pembina

Tabel 4. 1 Penjelasan *Use case diagram* pembina

Nama *Use case*	Deskripsi *Use case*
Login	Melakukan proses autentikasi Pembina dengan memasukkan kredensial (NBA/Email dan password) sebelum mengakses sistem.
Kelola Laporan	Melihat dan mengelola laporan atau analitik aktivitas pembelajaran.
Monitoring Karya/Tugas	Memantau karya atau tugas yang telah dikumpulkan oleh anggota komunitas.
Kelola Kurikulum	Mengatur dan memperbarui kurikulum pembelajaran yang digunakan oleh komunitas.
Kelola Profil	Mengubah informasi profil Pembina seperti nama, email, dan foto profil.
Kelola Notifikasi	Membuat dan mengatur notifikasi atau pemberitahuan yang akan dikirimkan kepada pengguna.
Monitoring Jadwal Kegiatan	Memantau pelaksanaan kegiatan yang dilakukan oleh pengurus bersama anggota.
Kelola admin/user	Mengelola data pengguna, termasuk menambah, mengubah, menghapus, dan melihat data anggota atau pembina lain.
Monitoring Forum	Memantau dan memoderasi aktivitas forum diskusi antar anggota komunitas.
Kelola System Settings	Mengatur konfigurasi teknis sistem sesuai dengan kebutuhan komunitas.

b. Pengurus

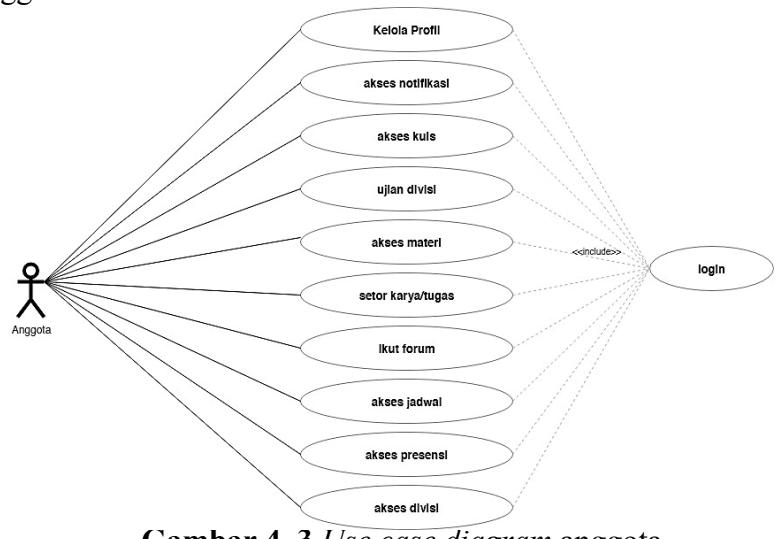


Gambar 4. 2 *Use case diagram* pengurus

Tabel 4. 2 Penjelasan *Use case diagram* Pengurus

Nama Use case	Deskripsi Use case
Login	Melakukan proses autentikasi Pengurus dengan memasukkan kredensial sebelum mengakses sistem.
Kelola Notifikasi	Membuat, mengubah, dan menghapus notifikasi yang dikirimkan ke pengguna.
Kelola Ujian Divisi	Mengelola ujian penentuan divisi, termasuk membuat soal, mengatur jadwal, dan melihat hasil ujian anggota.
Kelola System Settings	Mengatur konfigurasi teknis sistem sesuai dengan kebutuhan komunitas.
Kelola Forum	Mengelola forum diskusi, termasuk membuat, mengedit, dan memoderasi topik atau komentar.
Kelola jadwal & presensi	Mengatur jadwal kegiatan, pelatihan, atau acara komunitas, serta mengelola data kehadiran anggota dalam setiap kegiatan tersebut.
Kelola admin/user	Mengelola data pengguna, termasuk menambah, mengubah, menghapus, dan melihat data anggota.
Kelola Laporan	Melihat dan mengelola laporan atau analitik aktivitas pembelajaran.
Kelola Materi	Mengelola materi pembelajaran, termasuk menambah, mengubah, dan menghapus materi.
Kelola Kuis / karya	Mengelola soal kuis, jadwal, dan hasil kuis peserta, serta mengelola tugas atau karya anggota, termasuk penilaian dan
Kelola Profil	Mengubah informasi profil Pengurus seperti nama, email, dan foto profil.

c. Anggota



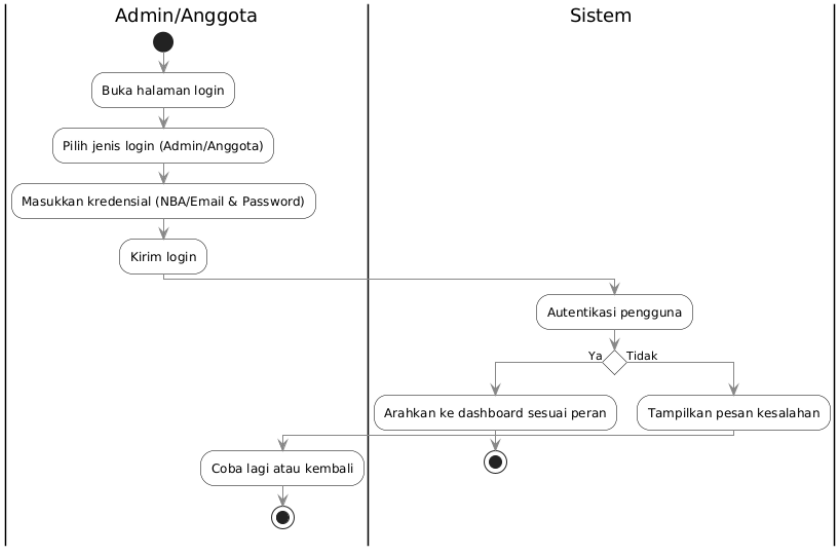
Gambar 4. 3 *Use case diagram* anggota

Tabel 4. 3 Penjelasan *Use case diagram* Anggota

Nama *Use case*	Deskripsi *Use case*
Login	Melakukan proses autentikasi Anggota dengan memasukkan kredensial sebelum mengakses sistem.
Akses Jadwal	Melihat jadwal kegiatan atau acara yang telah diatur oleh pengurus.
Presensi	Melakukan presensi atau absensi pada kegiatan yang diikuti.
Kelola Profil	Mengubah informasi profil Anggota seperti nama, email, dan foto profil.
Lihat Notifikasi	Melihat notifikasi atau pengumuman yang dikirimkan oleh pengurus atau pembina.
Ikut Kuis	Mengikuti kuis yang telah dijadwalkan oleh pengurus atau pembina.
Akses Materi	Mengakses materi pembelajaran yang telah disediakan oleh pengurus atau pembina.
Ikut Forum	Berpartisipasi dalam forum diskusi komunitas.
Submit Karya	Mengunggah karya atau tugas untuk dinilai dan dimoderasi oleh pengurus serta disetujui oleh pembina.

2. *Activity diagram*

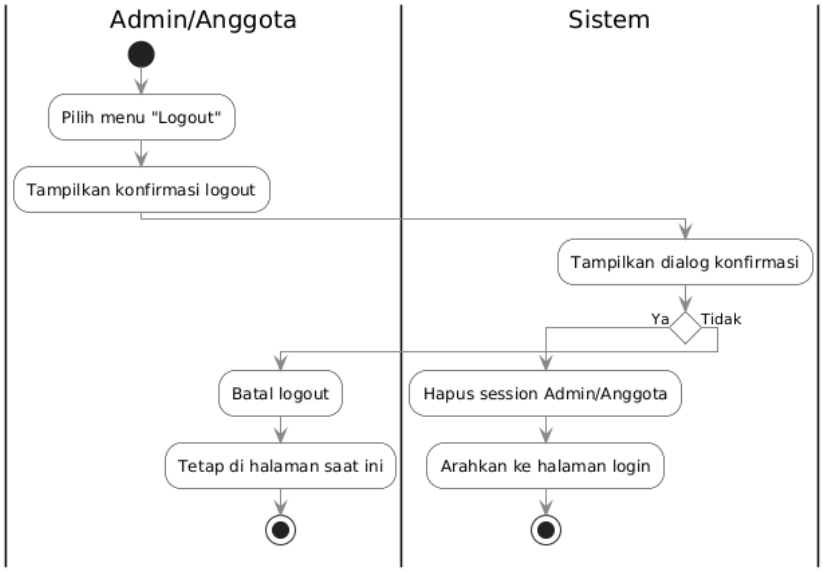
a. *Activity diagram - Login*



Gambar 4. 4 *Activity diagram - Login*

Diagram aktivitas ini menunjukkan bagaimana pengguna masuk ke sistem. Prosesnya dimulai dengan pengguna membuka halaman login, memilih jenis login sebagai anggota atau admin, mengisi informasi NBA/Email dan password, dan kemudian mengirimkan permintaan login. Setelah autentikasi pengguna selesai, sistem mengarahkan kredensial pengguna ke dashboard sesuai perannya. Jika kredensial tidak valid, sistem menampilkan pesan kesalahan dan pengguna dapat mencoba login kembali atau kembali ke langkah sebelumnya.

b. *Activity diagram Logout*

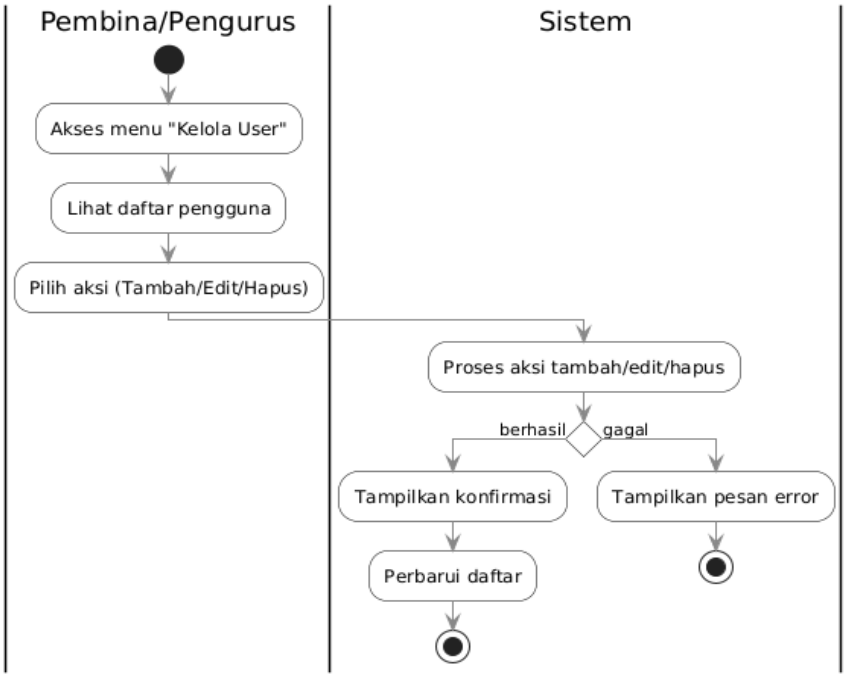


Gambar 4. 5 *Activity diagram* - logout

Diagram aktivitas tersebut menjelaskan proses logout pengguna dari aplikasi. Proses dimulai ketika pengguna sedang menggunakan aplikasi lalu memilih opsi “Logout”. Sistem kemudian menampilkan jendela konfirmasi. Jika pengguna mengonfirmasi logout, sistem akan mengakhiri sesi pengguna dan mengarahkan ke halaman awal atau login. Namun, jika pengguna membatalkan,

sistem menutup jendela konfirmasi dan pengguna tetap berada di halaman yang sedang digunakan.

c. *Activity diagram* Manajemen Pengguna

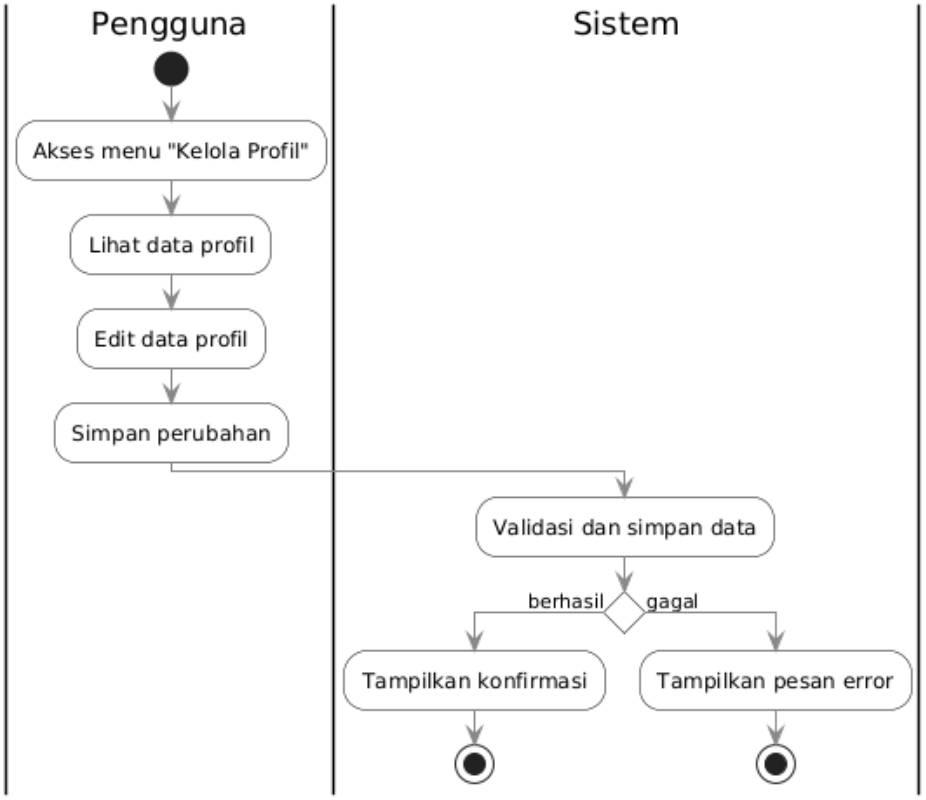


Gambar 4. 3 *Activity diagram* Manajemen Pengguna

Diagram aktivitas tersebut menggambarkan alur manajemen pengguna oleh pembina atau pengurus dalam sistem. Proses dimulai dari dashboard ketika pembina/pengurus memilih menu “Kelola User”. Sistem kemudian menampilkan daftar pengguna. Selanjutnya, pembina/pengurus dapat memilih aksi manajemen seperti menambah, mengedit, mengubah status, mereset password, mengatur peran, atau menghapus pengguna. Sistem akan memproses aksi tersebut, lalu memeriksa apakah aksi berhasil. Jika berhasil, sistem menampilkan konfirmasi dan memperbarui daftar pengguna. Jika gagal, sistem menampilkan peringatan dan pembina/pengurus dapat memperbaiki atau

membatalkan aksi. Proses berakhir dengan melanjutkan pengelolaan atau kembali ke menu sebelumnya.

d. *Activity diagram* Kelola Profil

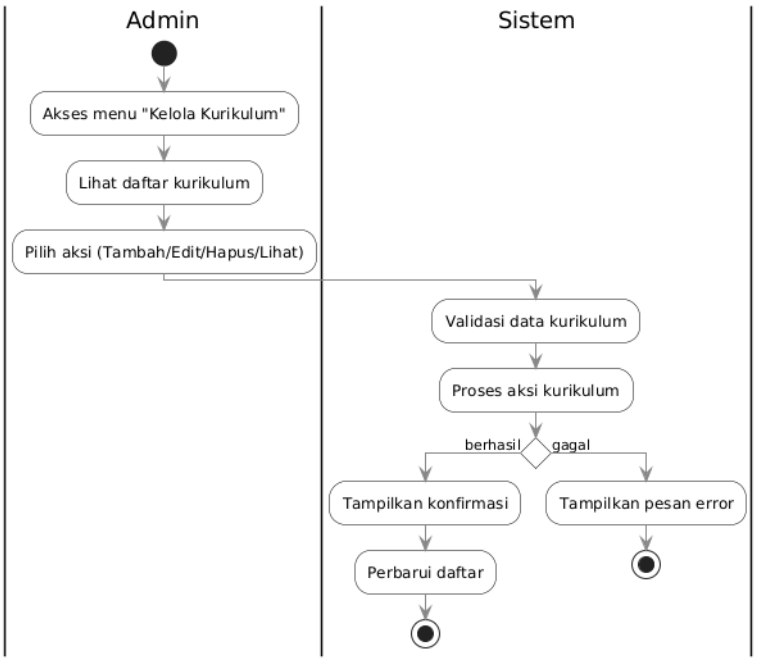


Gambar 4. 6 *Activity diagram* kelola profil

Diagram aktivitas tersebut menggambarkan proses pengelolaan profil oleh pembina, pengurus, atau anggota. Proses dimulai ketika pengguna membuka halaman profil, kemudian sistem menampilkan data profil saat ini. Pengguna dapat mengubah data inti seperti nama, kontak, atau foto, lalu menekan tombol “Simpan”. Sistem memproses penyimpanan perubahan dan memeriksa keberhasilannya. Jika berhasil, sistem menampilkan konfirmasi dan pengguna dapat memilih untuk selesai atau melanjutkan mengubah data lain.

Jika gagal, sistem menampilkan peringatan sehingga pengguna dapat memperbaiki data atau membatalkan perubahan.

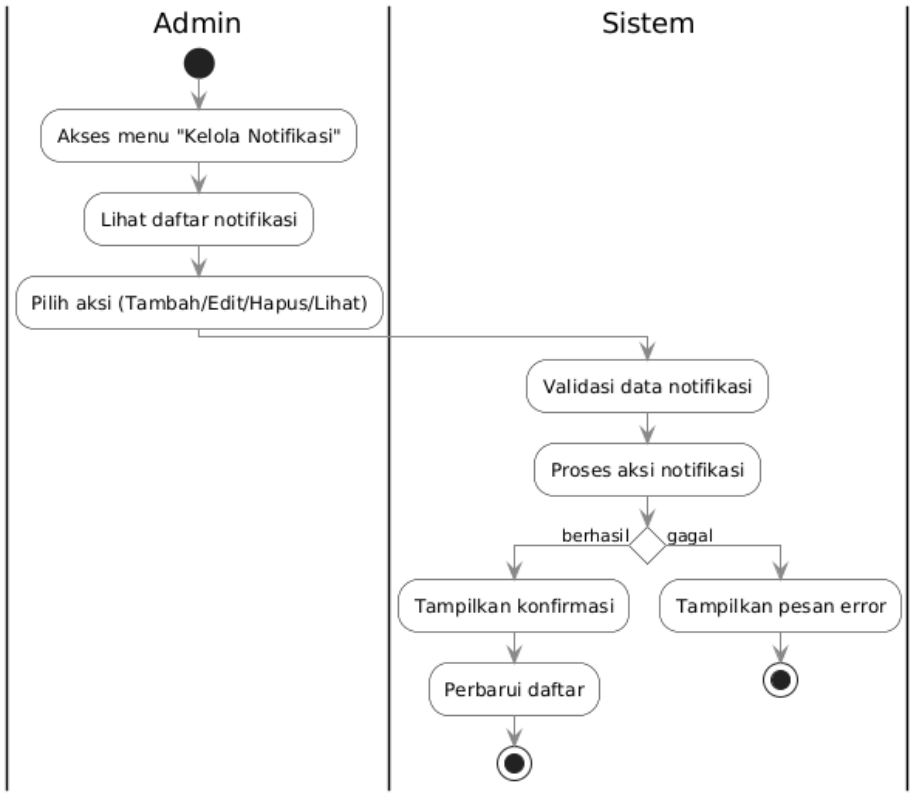
e. *Activity diagram* Manajemen Kurikulum



Gambar 4. 8 *Activity diagram* manajemen kurikulum

Diagram aktivitas tersebut menggambarkan proses manajemen kurikulum oleh pembina atau pengurus. Proses dimulai saat pengguna membuka menu kurikulum, kemudian sistem menampilkan daftar kurikulum yang ada. Pengguna dapat memilih aksi seperti menambah, mengubah, menghapus, atau mengatur struktur kurikulum. Sistem akan memproses aksi tersebut dan memeriksa apakah berhasil. Jika berhasil, sistem menampilkan konfirmasi dan memperbarui daftar kurikulum. Jika tidak berhasil, sistem menampilkan peringatan sehingga pengguna dapat memperbaiki atau membatalkan aksi. Proses diakhiri dengan pilihan untuk melanjutkan pengelolaan atau kembali, atau tetap berada di halaman kurikulum.

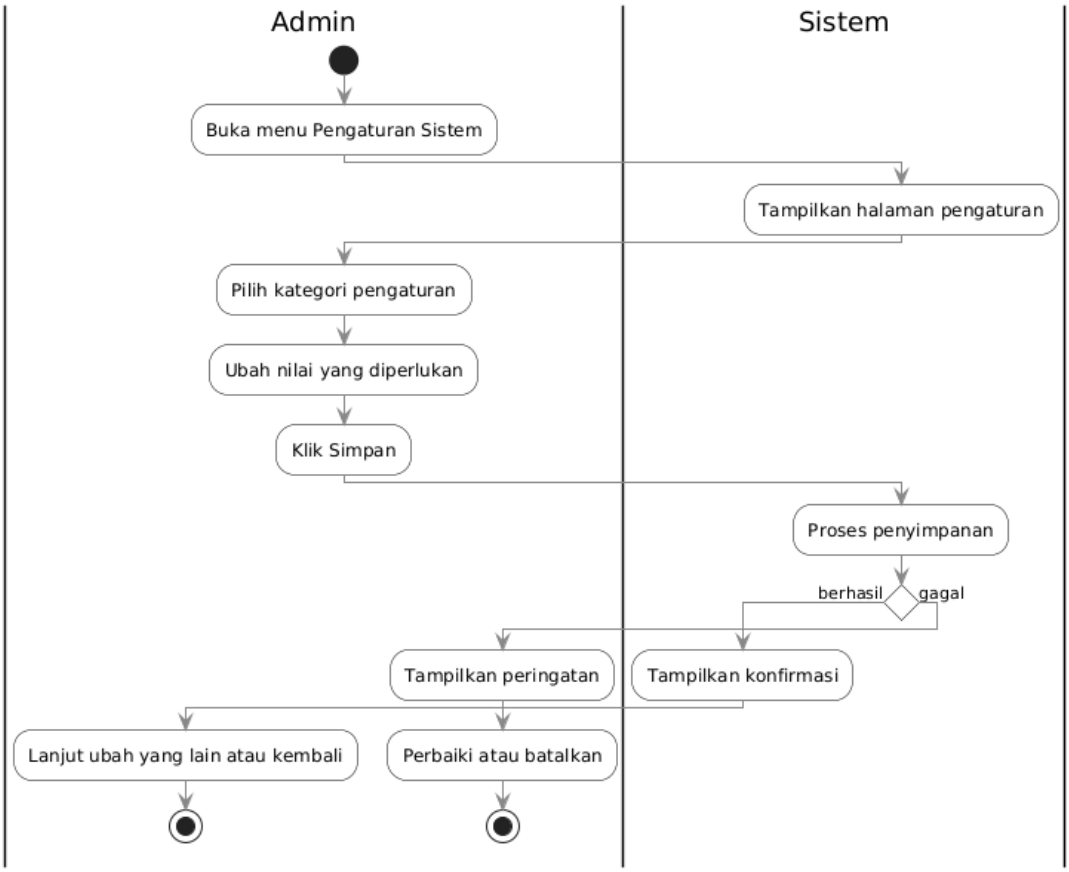
f. *Activity diagram* Kelola Notifikasi



Gambar 4. 9 *Activity diagram* kelola notifikasi

Diagram aktivitas tersebut menjelaskan proses manajemen notifikasi oleh pembina atau pengurus. Proses dimulai ketika pengguna membuka menu notifikasi, lalu sistem menampilkan daftar dan opsi notifikasi yang tersedia. Pengguna kemudian memilih aksi seperti membuat, mengedit, mengatur jadwal, menentukan penerima, atau mengirim notifikasi. Sistem memproses aksi tersebut dan memeriksa keberhasilannya. Jika berhasil, sistem menampilkan konfirmasi dan memperbarui daftar notifikasi. Jika tidak berhasil, sistem menampilkan peringatan sehingga pengguna dapat memperbaiki atau membatalkan aksi. Proses diakhiri dengan pilihan untuk melanjutkan pengelolaan atau kembali ke menu sebelumnya.

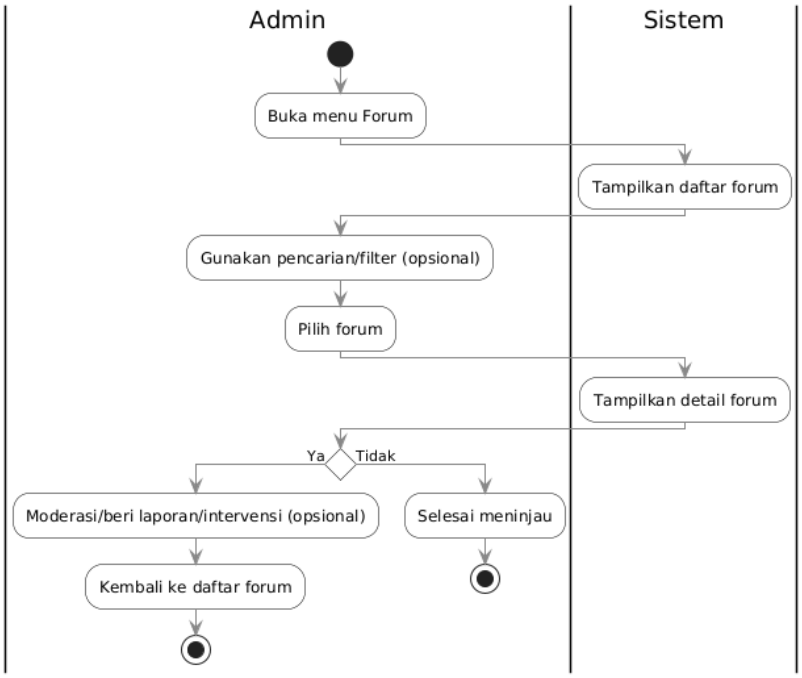
g. *Activity diagram System Settings*



Gambar 4. 10 *Activity diagram system settings*

Diagram aktivitas tersebut menggambarkan proses pengaturan sistem oleh pembina atau pengurus. Proses dimulai ketika pengguna membuka menu pengaturan sistem, lalu sistem menampilkan halaman pengaturan. Pengguna memilih kategori pengaturan, mengubah nilai yang diperlukan, dan menekan tombol “Simpan”. Kemudian sistem memproses penyimpanan perubahan dan mengetahui apakah berhasil. Jika berhasil, sistem menampilkan konfirmasi, sedangkan jika tidak, sistem menampilkan peringatan untuk memberi pengguna kesempatan untuk memperbaiki atau membatalkan perubahan. Proses diakhiri dengan pilihan untuk melanjutkan mengubah pengaturan lain atau kembali ke menu sebelumnya.

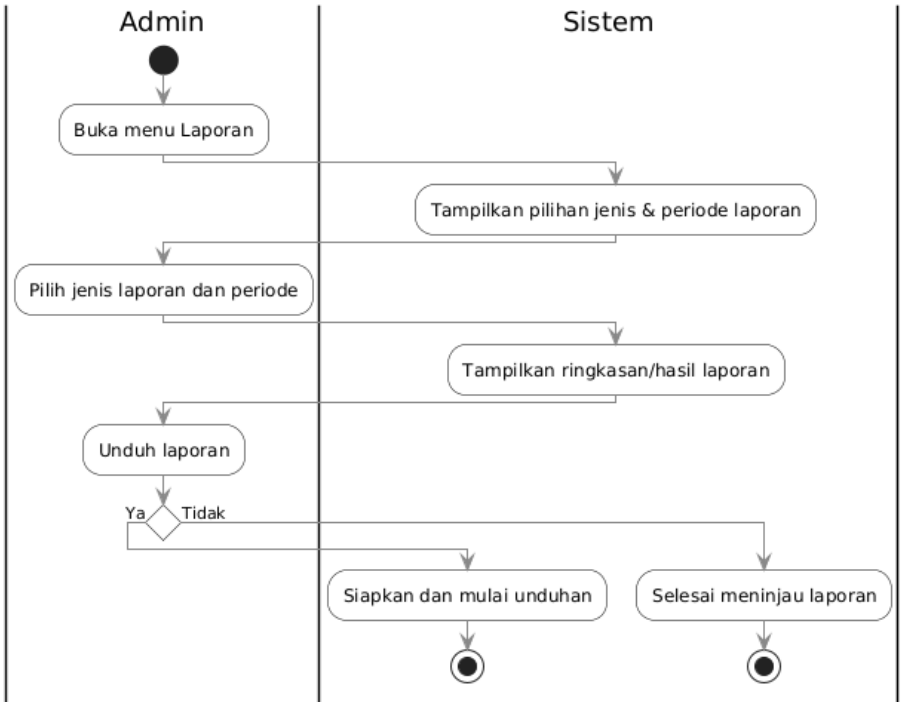
h. *Activity diagram* Pembina Monitoring Forum



Gambar 4. 11 *Activity diagram* pembina *monitoring* forum

Diagram aktivitas tersebut menggambarkan proses penggunaan fitur forum oleh pembina. Proses dimulai ketika pembina membuka fitur forum, lalu sistem menampilkan daftar thread yang tersedia. Pembina kemudian memilih aksi seperti melihat thread, membuat thread baru, memberi komentar, atau melakukan moderasi. Sistem memproses aksi tersebut dan memeriksa keberhasilannya. Jika berhasil, sistem memperbarui tampilan atau menampilkan konfirmasi. Jika gagal, sistem menampilkan peringatan sehingga pembina dapat mencoba lagi atau membatalkan aksi. Proses diakhiri dengan pilihan untuk menyelesaikan atau melanjutkan aksi lain.

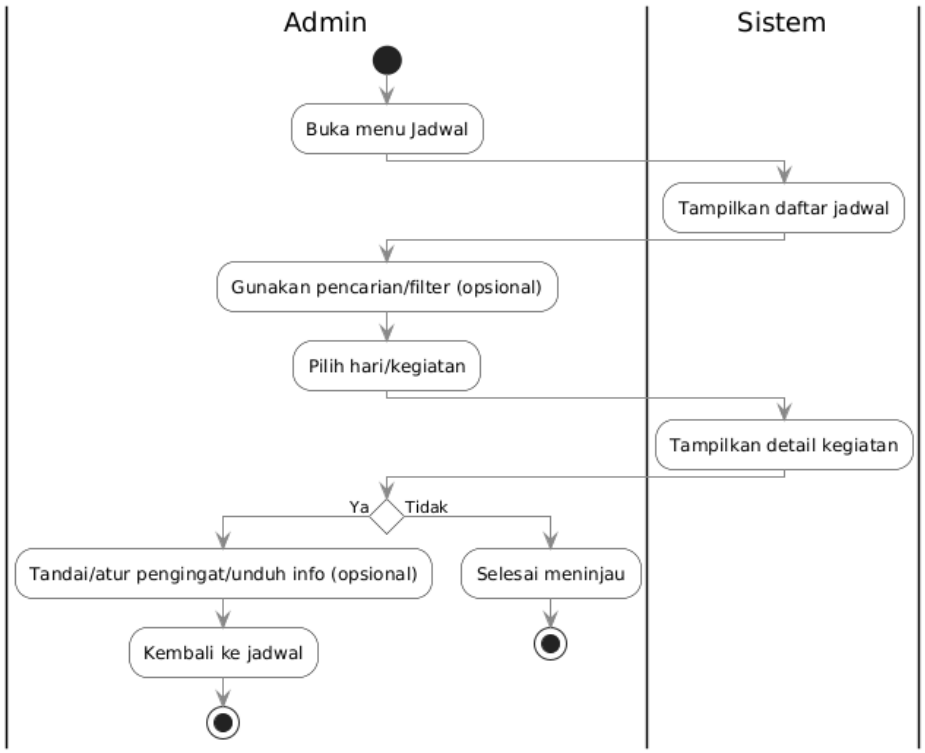
i. *Activity diagram* Pembina Kelola Laporan



Gambar 4. 12 *Activity diagram* Pembina Kelola Laporan

Diagram aktivitas tersebut menggambarkan proses pembina dalam mengakses dan mengunduh laporan dari sistem. Proses dimulai saat pembina membuka menu laporan, kemudian sistem menampilkan pilihan jenis dan periode laporan. Pembina memilih jenis laporan beserta periodenya, lalu sistem menampilkan ringkasan atau hasil laporan. Selanjutnya, pembina dapat memilih apakah ingin mengunduh laporan tersebut. Jika ya, sistem menyiapkan dan memulai proses unduhan. Jika tidak, proses berakhir setelah pembina selesai meninjau laporan.

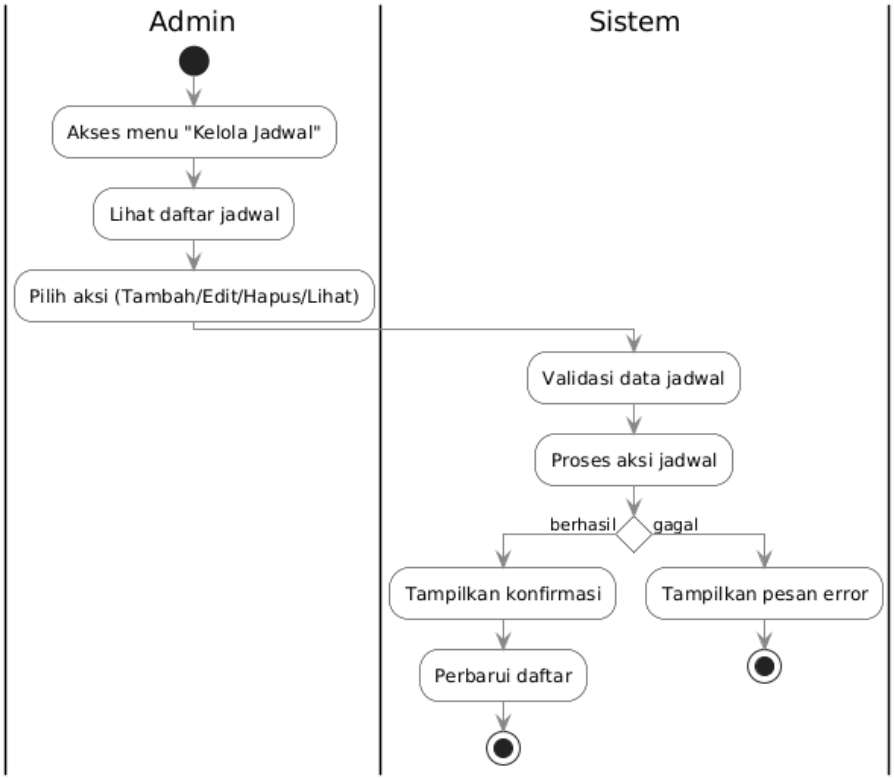
j. *Activity diagram* Pembina Monitoring Jadwal



Gambar 4. 13 *Activity diagram* pembina monitoring kegiatan

Diagram aktivitas tersebut menggambarkan proses pembina dalam melihat dan memantau jadwal kegiatan. Proses dimulai saat pembina membuka menu jadwal, kemudian sistem menampilkan daftar jadwal yang tersedia. Pembina dapat menggunakan pencarian atau filter (opsional) untuk mempermudah pencarian, lalu memilih hari atau kegiatan tertentu. Sistem kemudian menampilkan detail kegiatan tersebut. Setelah itu, pembina memutuskan apakah akan melakukan aksi monitoring. Jika ya, pembina dapat menandai, mengatur pengingat, atau mengunduh informasi (opsional), lalu kembali ke daftar jadwal. Jika tidak, proses berakhir setelah pembina selesai meninjau detail kegiatan.

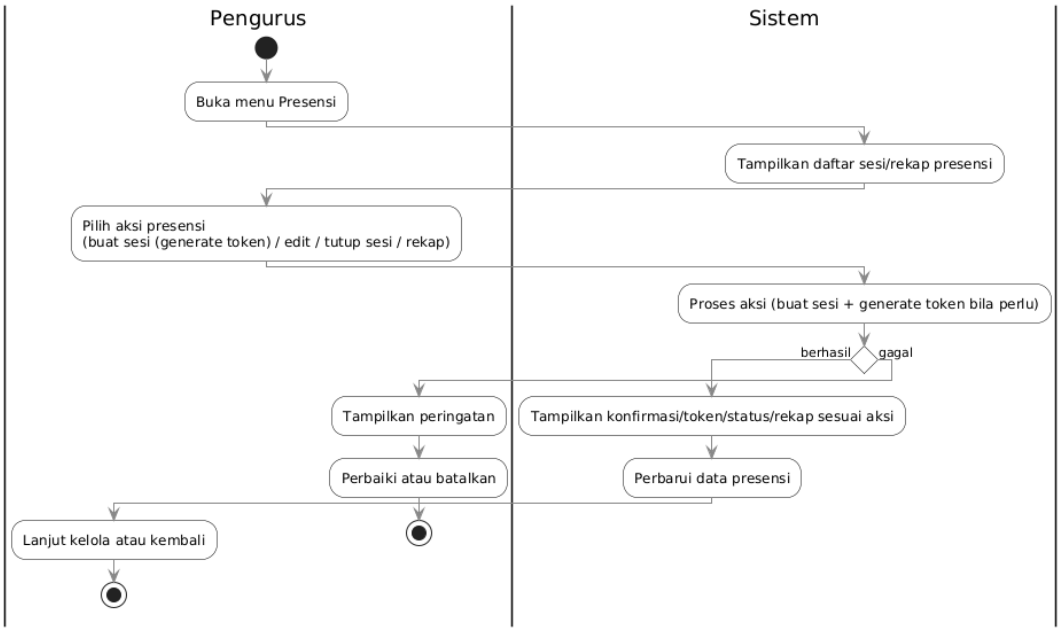
k. *Activity diagram* Pengurus kelola jadwal



Gambar 4. 14 *Activity diagram* Pengurus kelola jadwal

Diagram aktivitas tersebut menggambarkan proses pengelolaan jadwal oleh pengurus. Proses dimulai ketika pengurus membuka menu jadwal, lalu sistem menampilkan daftar jadwal yang ada. Pengurus kemudian memilih aksi seperti membuat, mengedit, menghapus, atau mengatur detail jadwal. Sistem memproses aksi tersebut dan memeriksa keberhasilannya. Jika aksi berhasil, sistem menampilkan konfirmasi dan memperbarui daftar jadwal. Jika gagal, sistem menampilkan peringatan sehingga pengurus dapat memperbaiki atau membatalkan aksi. Proses diakhiri dengan pilihan untuk melanjutkan pengelolaan atau kembali ke menu sebelumnya.

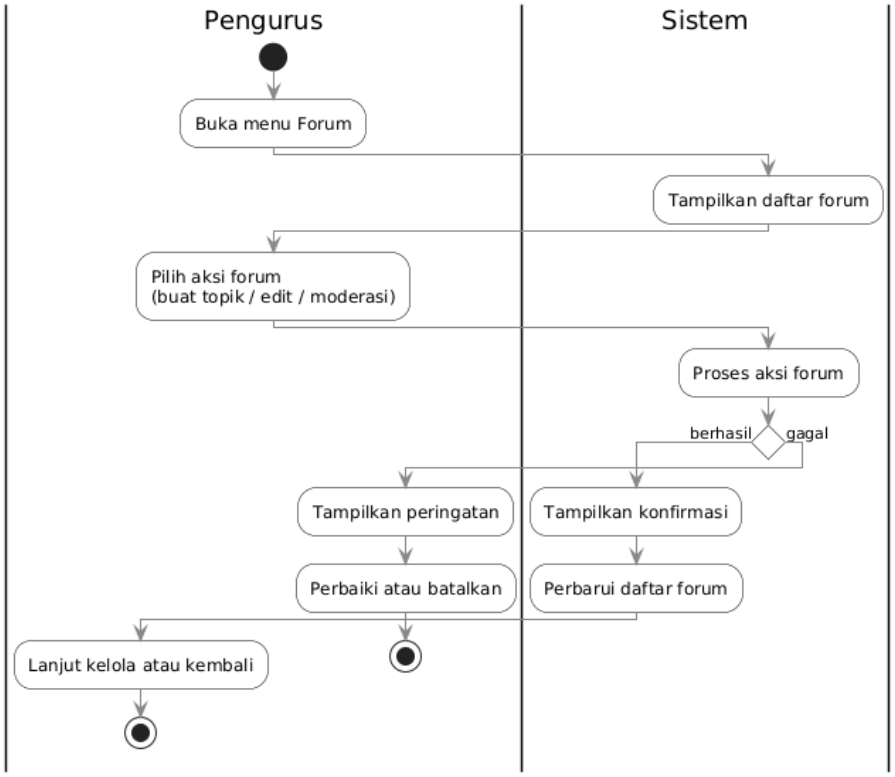
l. *Activity diagram* Pengurus kelola presensi



Gambar 4. 15 *Activity diagram* Pengurus kelola presensi

Diagram aktivitas tersebut menjelaskan proses pengelolaan presensi oleh pengurus. Proses dimulai saat pengurus membuka menu presensi, lalu sistem menampilkan daftar sesi atau rekap presensi. Pengurus memilih aksi seperti membuat sesi (dengan generate token), mengedit, menutup sesi, atau merekap. Sistem memproses aksi tersebut dan memeriksa keberhasilannya. Jika berhasil, sistem menampilkan konfirmasi, token, status, atau rekap sesuai aksi, lalu memperbarui data presensi. Jika gagal, sistem menampilkan peringatan sehingga pengurus dapat memperbaiki atau membatalkan aksi. Proses diakhiri dengan pilihan untuk melanjutkan pengelolaan atau kembali.

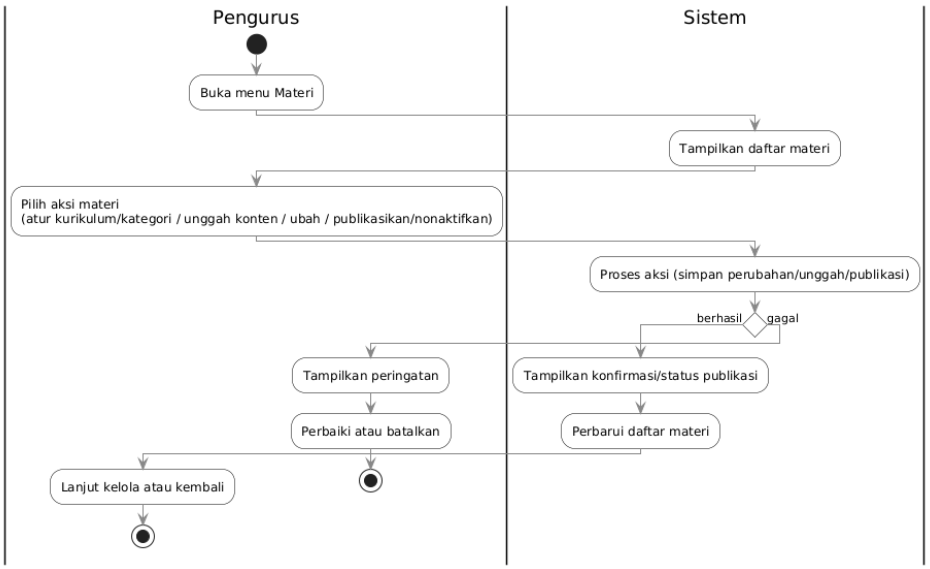
m. *Activity diagram* Pengurus Kelola Forum Diskusi



Gambar 4. 16 *Activity diagram* Pengurus Kelola Forum Diskusi

Diagram aktivitas tersebut menjelaskan proses pengelolaan forum oleh pengurus. Proses dimulai saat pengurus membuka menu forum, lalu sistem menampilkan daftar topik atau thread. Pengurus memilih aksi seperti membuat topik, mengedit, atau moderasi. Sistem memproses aksi tersebut; jika berhasil, menampilkan konfirmasi dan memperbarui daftar forum, sedangkan jika gagal, menampilkan peringatan untuk diperbaiki atau dibatalkan. Proses berakhir dengan pilihan untuk melanjutkan pengelolaan atau kembali.

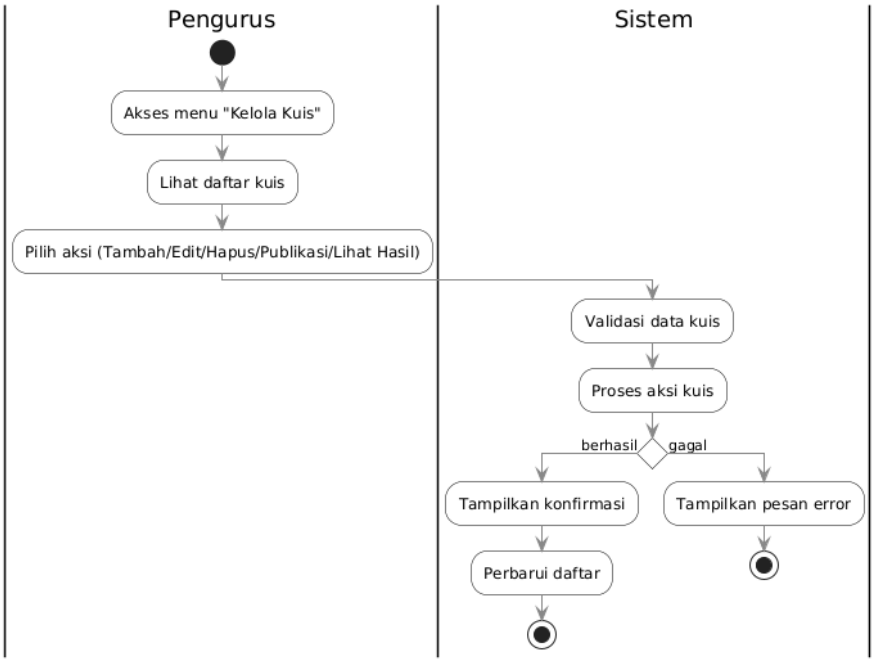
n. *Activity diagram* Pengurus Kelola Materi



Gambar 4. 17 *Activity diagram* pengurus kelola materi

Diagram aktivitas tersebut menggambarkan alur pengelolaan materi oleh pengurus. Proses dimulai ketika pengurus membuka menu Materi, lalu sistem menampilkan daftar materi. Pengurus memilih aksi seperti mengatur kurikulum/kategori, mengunggah konten, mengubah, mempublikasikan, atau menonaktifkan materi. Sistem memproses aksi tersebut (penyimpanan, unggahan, publikasi) dan memeriksa keberhasilannya. Jika berhasil, sistem menampilkan konfirmasi atau status publikasi dan memperbarui daftar materi. Jika gagal, sistem menampilkan peringatan sehingga pengurus dapat memperbaiki atau membatalkan aksi. Proses diakhiri dengan opsi untuk melanjutkan pengelolaan atau kembali.

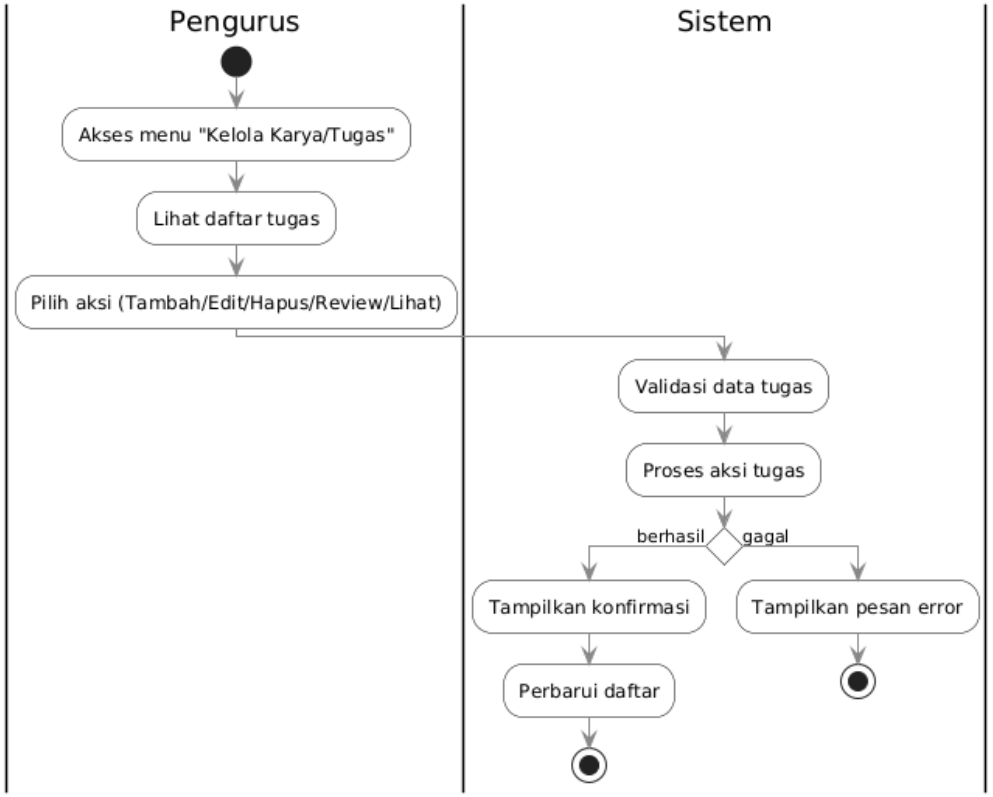
o. *Activity diagram* Pengurus Kelola Kuis Evaluasi



Gambar 4. 18 *Activity diagram* Pengurus Kelola Kuis Evaluasi

Diagram aktivitas tersebut menggambarkan proses pengelolaan kuis oleh pengurus. Dimulai dari membuka menu kuis, sistem menampilkan daftar kuis, lalu pengurus memilih aksi seperti menambah/mengubah soal, mengatur jadwal, atau mempublikasikan. Sistem memproses aksi tersebut; jika berhasil, menampilkan konfirmasi dan memperbarui daftar kuis, jika gagal, menampilkan peringatan untuk diperbaiki atau dibatalkan. Proses diakhiri dengan melanjutkan pengelolaan atau kembali.

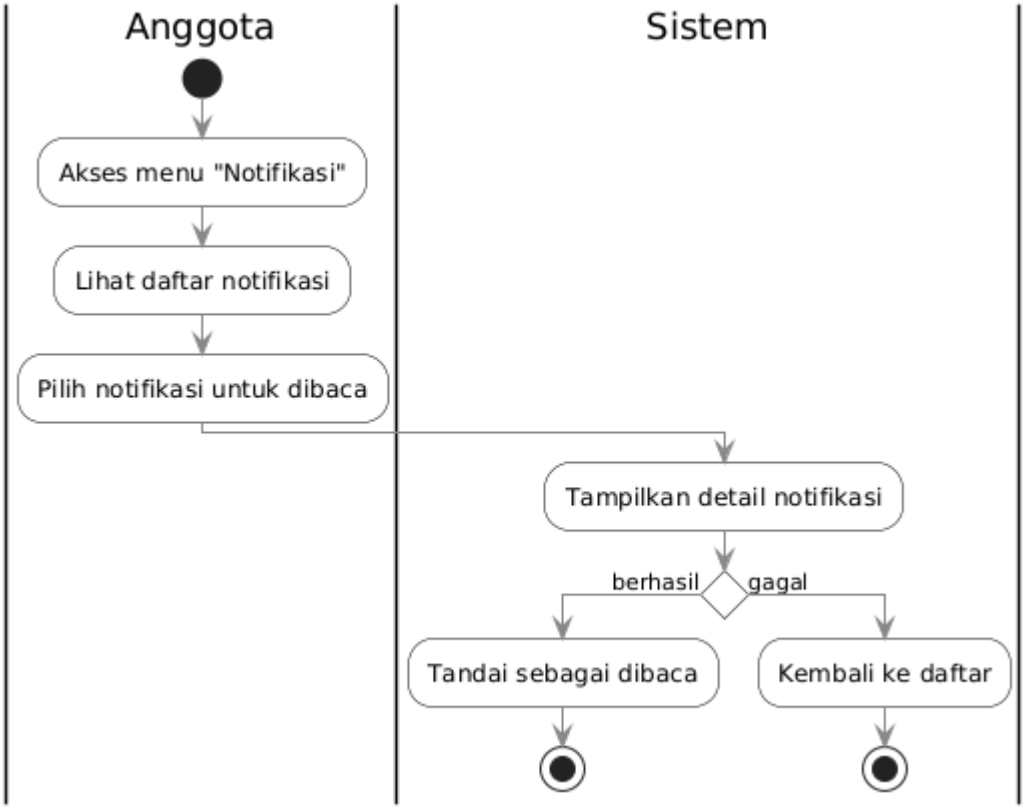
p. *Activity diagram* Pengurus Kelola Karya/Tugas



Gambar 4. 19 *Activity diagram* Pengurus Kelola Karya/Tugas

Diagram aktivitas ini menjelaskan alur kerja untuk pengurus dalam mengelola karya atau tugas dalam suatu sistem. Awalnya, pengurus membuka menu karya/tugas, dan sistem menampilkan daftar yang tersedia. Pengurus kemudian memilih aksi yang ingin dilakukan, seperti memberi nilai, memberikan umpan balik, mengubah status, atau mengelola lampiran. Setelah aksi dipilih, sistem akan memprosesnya. Jika proses berhasil, sistem menampilkan konfirmasi dan memperbarui daftar. Sebaliknya, jika proses gagal, sistem menampilkan peringatan yang meminta pengurus untuk memperbaiki atau membatalkan aksi tersebut. Alur berakhir setelah pengurus memilih untuk melanjutkan pengelolaan atau kembali ke menu utama.

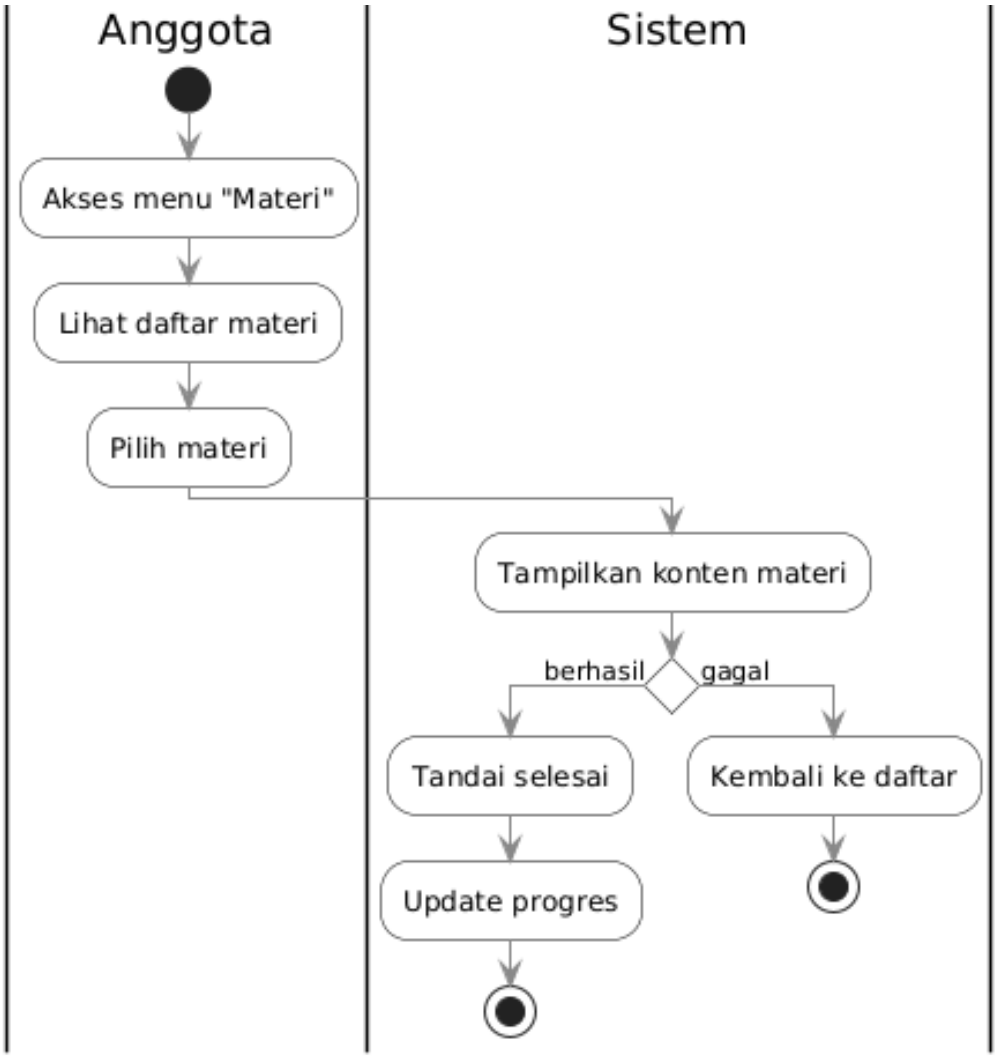
q. *Activity diagram* Anggota lihat notifikasi



Gambar 4. 20 *Activity diagram* Anggota lihat notifikasi

Diagram aktivitas ini menggambarkan alur kerja bagi seorang anggota dalam melihat dan mengelola notifikasi dalam suatu sistem. Proses dimulai ketika anggota membuka menu notifikasi, yang kemudian membuat sistem menampilkan daftar notifikasi yang ada. Anggota dapat secara opsional memilih notifikasi tertentu untuk dibaca, dan sistem akan menampilkan detail dari notifikasi tersebut. Setelah melihat detail, anggota memiliki pilihan untuk melakukan aksi, seperti menandai notifikasi sebagai sudah dibaca atau membuka tautan terkait. Jika tidak ada aksi yang dilakukan, anggota hanya menyelesaikan peninjauan. Setelah aksi selesai atau peninjauan selesai, anggota akan kembali ke daftar notifikasi, menandai akhir dari proses ini.

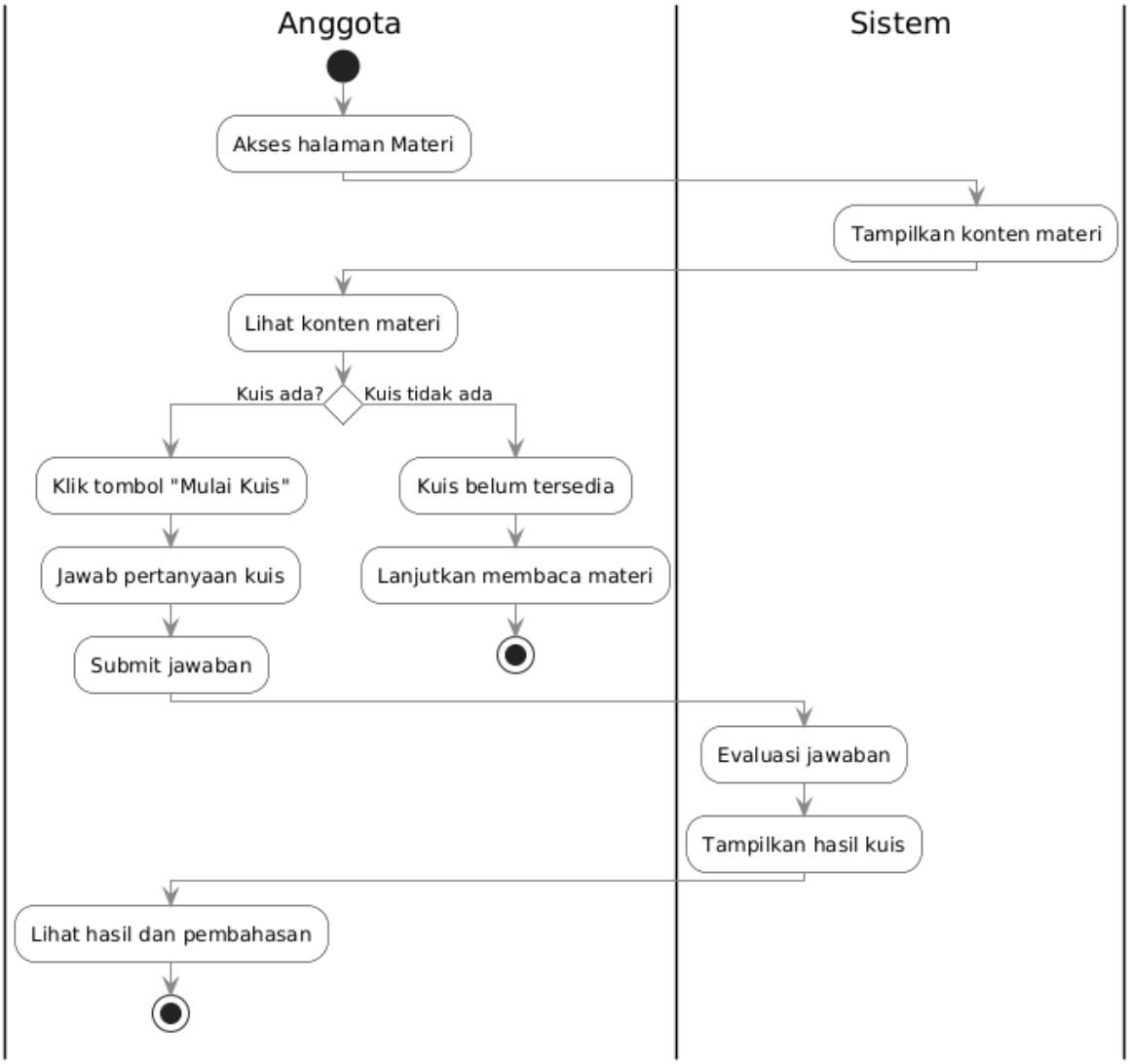
r. *Activity diagram* Anggota Akses Materi



Gambar 4. 21 *Activity diagram* Anggota akses materi

Diagram ini menggambarkan langkah-langkah anggota dalam mengakses materi pembelajaran. Proses dimulai dengan membuka menu materi, lalu sistem menampilkan daftar materi. Anggota memilih salah satu materi, sistem menampilkan kontennya, dan anggota membaca atau menonton. Setelah itu, anggota bisa menandai materi sebagai selesai, sehingga sistem memperbarui progres atau status penyelesaian. Jika tidak ditandai selesai, anggota kembali ke daftar materi tanpa ada perubahan progres.

s. *Activity diagram* Anggota Akses Kuis

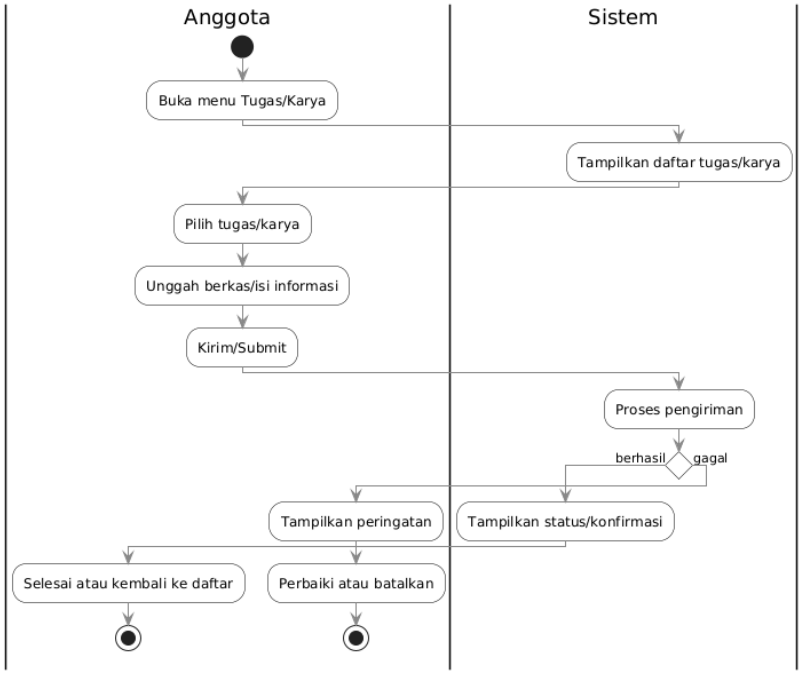


Gambar 4. 22 *Activity diagram* Anggota akses kuis

Diagram ini menunjukkan proses anggota mengikuti kuis. Dimulai dengan membuka menu materi, sistem menampilkan materi dan diakhir halaman materi terdapat kuis yang tersedia, lalu anggota memilih kuis dan memulai. Sistem akan menampilkan soal kuis, anggota menjawab pertanyaan, dan mengirimkan jawabannya. Setelah itu, sistem mengevaluasi jawaban dan menampilkan hasil. Anggota dapat memilih untuk melihat pembahasan ringkas atau langsung

mengakhiri proses. Alur ini menggambarkan interaksi sederhana namun jelas antara anggota dan sistem selama pengerjaan kuis.

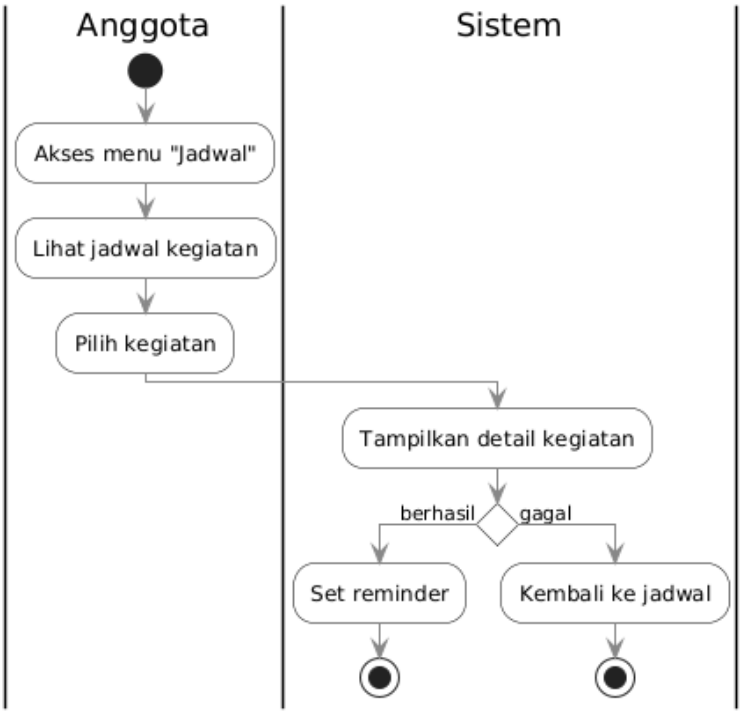
t. *Activity diagram* Anggota Unggah Karya/Tugas



Gambar 4. 23 *Activity diagram* Anggota Unggah Karya/Tugas

Pada diagram ini, anggota membuka menu tugas/karya, kemudian sistem menampilkan daftar tugas atau karya yang ada. Anggota memilih tugas atau karya tertentu, mengunggah berkas atau mengisi informasi yang diminta, lalu mengirimkannya. Sistem memproses pengiriman tersebut; jika berhasil, akan menampilkan status atau konfirmasi, sedangkan jika gagal, akan menampilkan peringatan untuk diperbaiki atau dibatalkan. Proses diakhiri dengan anggota menyelesaikan atau kembali ke daftar tugas/karya.

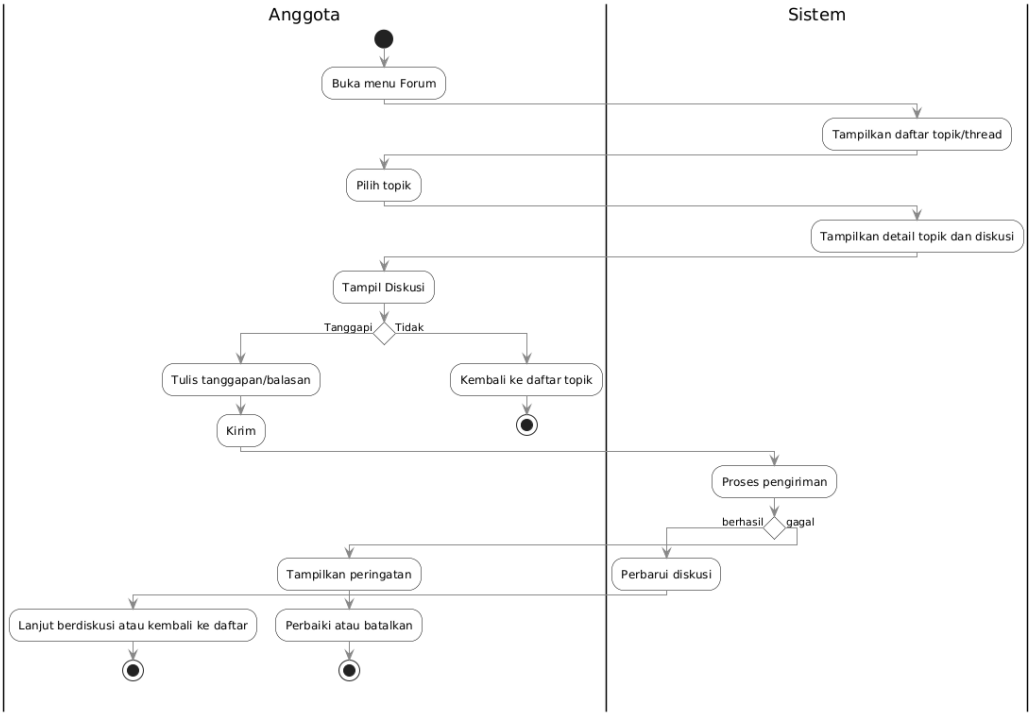
u. *Activity diagram* Anggota Akses Jadwal



Gambar 4. 24 *Activity diagram* anggota akses jadwal

Diagram ini memperlihatkan langkah anggota dalam mengakses jadwal kegiatan. Setelah membuka menu jadwal, sistem menampilkan daftar kegiatan. Anggota memilih kegiatan tertentu, lalu sistem menampilkan detailnya. Anggota kemudian memutuskan apakah akan melakukan aksi seperti mendaftar, mengatur pengingat, atau melihat lokasi. Jika ya, anggota melakukan aksi tersebut dan kembali ke jadwal, sedangkan jika tidak, proses berakhir setelah meninjau detail.

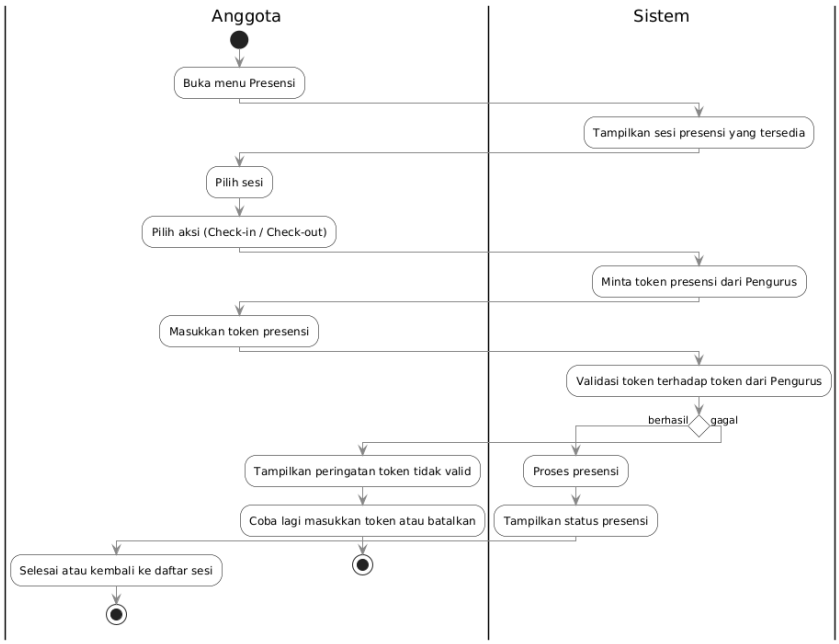
v. *Activity diagram* Anggota Akses Forum



Gambar 4. 25 *Activity diagram* Anggota Akses Forum

Diagram ini menunjukkan alur anggota saat mengakses forum. Anggota membuka menu forum, sistem menampilkan daftar topik atau thread, dan anggota dapat menggunakan fitur pencarian atau filter untuk menemukan topik yang diinginkan. Setelah memilih topik, anggota melihat detail diskusi. Jika ingin berpartisipasi, anggota menulis tanggapan atau balasan dan mengirimkannya, lalu sistem memproses dan memperbarui diskusi. Jika tidak ingin berpartisipasi, anggota kembali ke daftar topik. Jika pengiriman gagal, sistem menampilkan peringatan agar dapat diperbaiki.

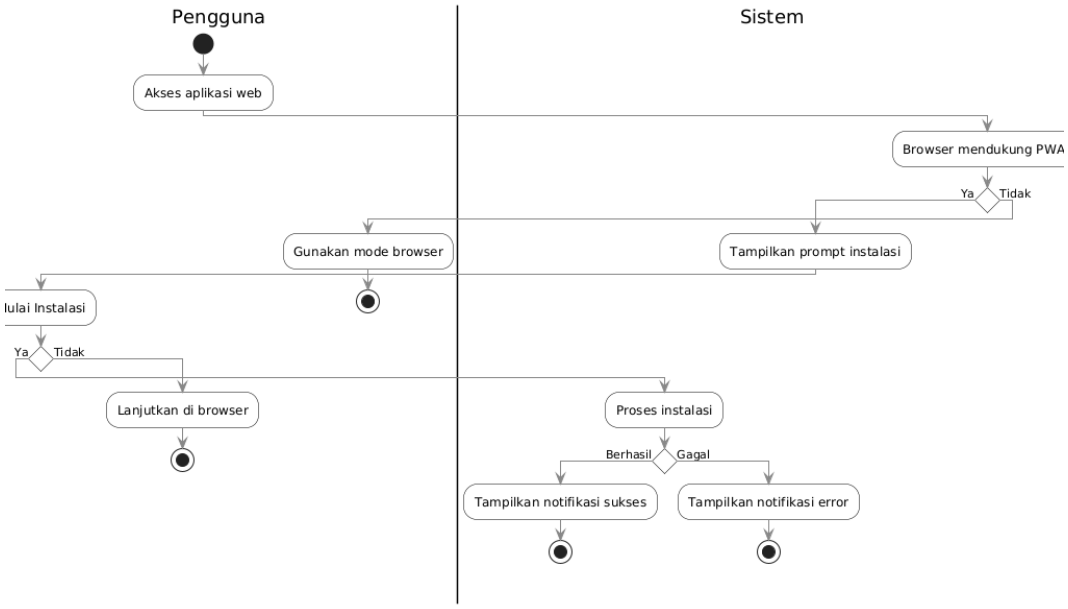
w. *Activity diagram* Anggota Presensi



Gambar 4. 26 *Activity diagram* Anggota Presensi

Diagram ini menjelaskan alur anggota saat melakukan presensi. Anggota membuka menu presensi, sistem menampilkan sesi presensi yang tersedia, lalu anggota memilih sesi dan menentukan aksi check-in atau check-out. Sistem memproses presensi; jika berhasil, status presensi ditampilkan, jika gagal, sistem memberikan peringatan dan anggota dapat mencoba lagi atau membatalkan. Proses diakhiri dengan anggota menyelesaikan atau kembali ke daftar sesi.

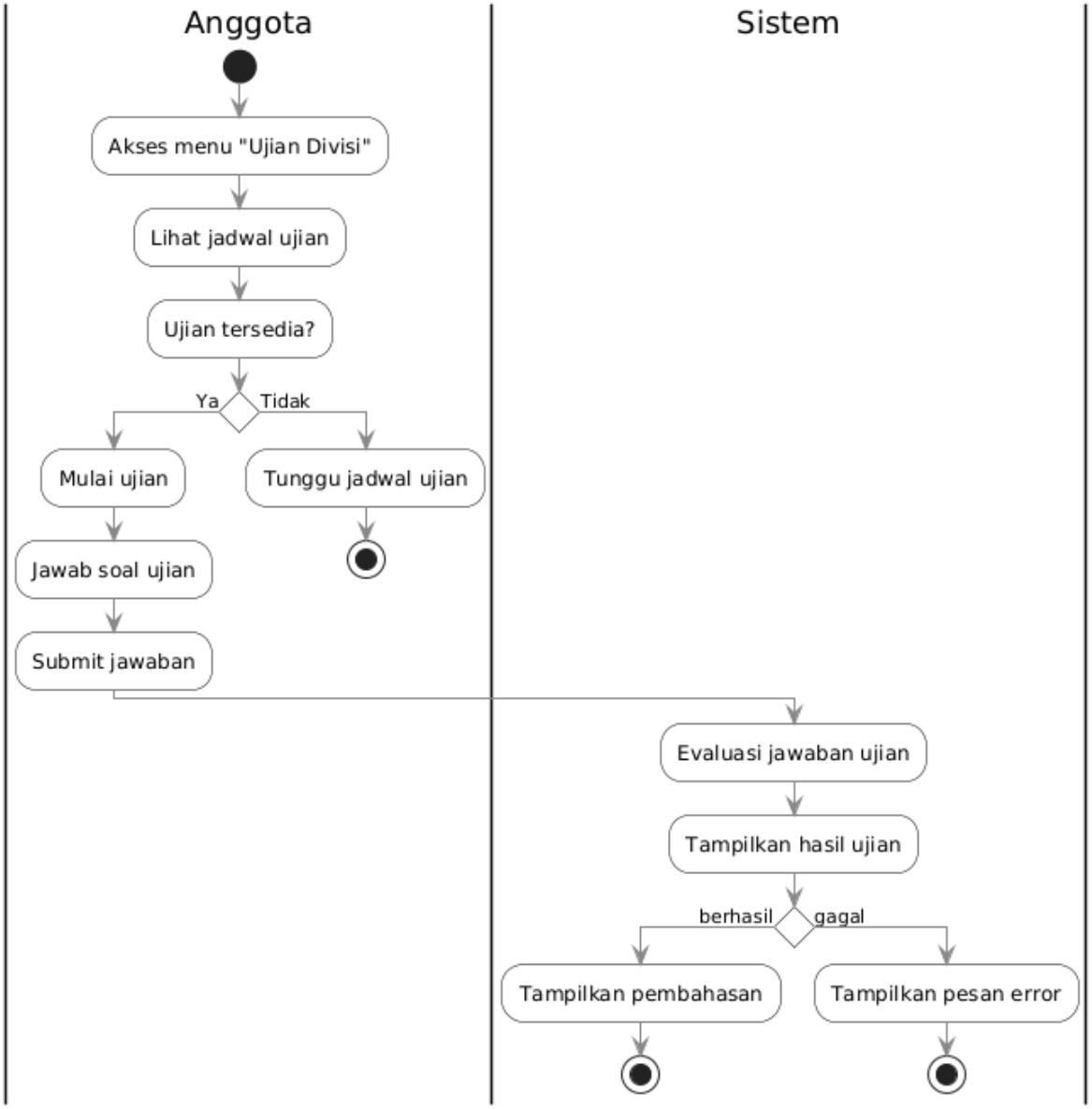
x. *Activity diagram* Instalasi *Progressive Web Apps*



Gambar 4. 27 *Activity diagram Instalasi Progressive Web Apps*

Diagram ini menggambarkan alur pemasangan aplikasi web berbasis PWA (Progressive Web App). Proses dimulai ketika pengguna mengakses aplikasi web, lalu sistem memverifikasi kelayakan PWA. Jika layak diinstal, sistem menampilkan prompt instalasi; jika tidak, aplikasi dijalankan sebagai web biasa. Saat prompt muncul, pengguna dapat memilih “Install” atau “Tolak”. Jika memilih instal, sistem memproses instalasi, membuat shortcut, serta menyediakan fungsionalitas offline dan notifikasi, sehingga PWA terpasang. Jika pengguna menolak, sistem mencatat preferensi dan melanjutkan penggunaan sebagai web biasa.

y. *Activity diagram* Anggota Ujian Divisi



Gambar 4. 28 *Activity diagram* Ujian Divisi

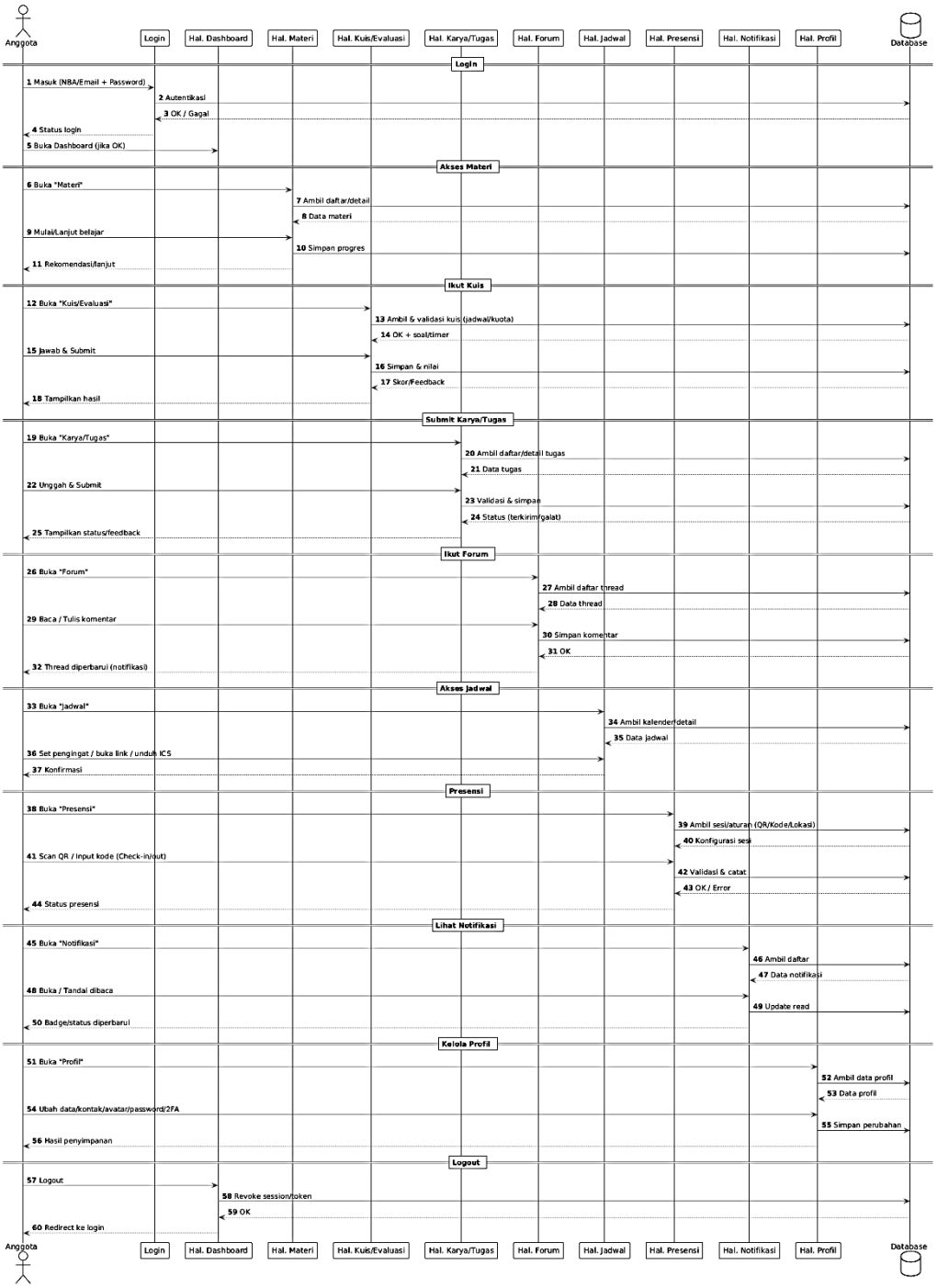
Diagram ini menggambarkan alur ujian divisi. Setelah ujian dipublikasikan, anggota dapat membuka menu ujian divisi dan memilih ujian yang tersedia. Sistem akan memeriksa status ujian; apabila tidak tersedia, anggota mendapat pesan peringatan. Jika tersedia, sistem menampilkan

informasi ujian, dan anggota dapat memulai ujian. Pada tahap ini, sistem menginisialisasi sesi dan menampilkan soal pertama.

Selanjutnya, anggota membaca soal, memilih atau menginput jawaban, lalu menekan tombol “Selanjutnya.” Sistem menyimpan jawaban sementara dan menampilkan soal berikutnya hingga semua soal selesai. Setelah semua soal dijawab, anggota dapat melakukan review jawaban secara opsional sebelum mengklik “Submit Ujian.” Sistem menampilkan konfirmasi submit; jika anggota belum yakin, ia bisa kembali melakukan review. Jika submit dilanjutkan, sistem memproses jawaban, menghitung skor, menyimpan hasil ujian, dan mengirim notifikasi kepada pengurus. Anggota kemudian dapat melihat hasil ujian dan skor, sementara pengurus menerima notifikasi hasil untuk keperluan monitoring dan evaluasi.

3. *Sequence Diagram*

a. *Sequence diagram* Anggota



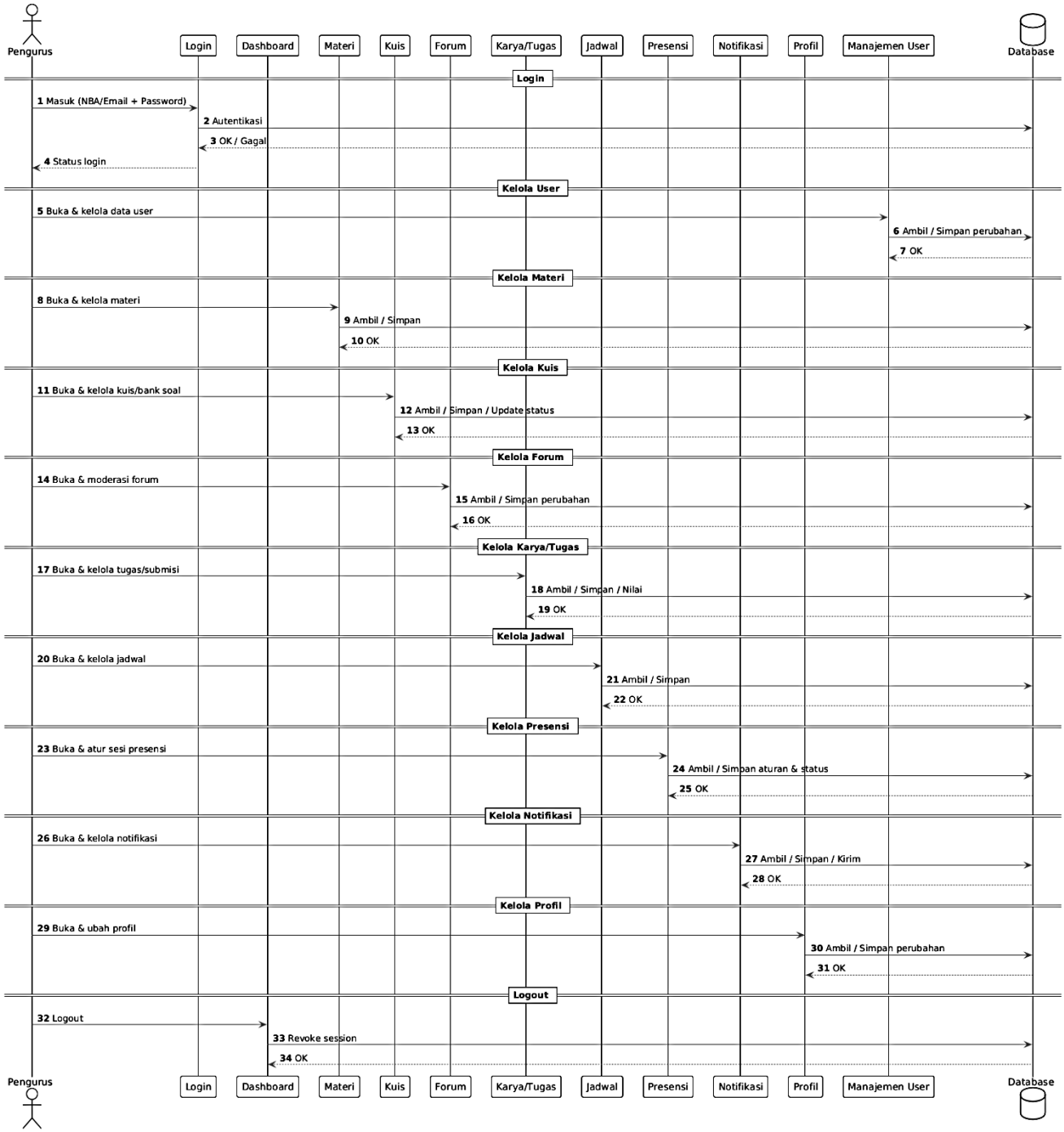
Gambar 4. 29 *Sequence diagram* Anggota

Penjelasan *Sequence diagram* Anggota:

Pada diagram ini menggambarkan interaksi antara Anggota dan berbagai halaman serta database dalam sistem. Awalnya, anggota melakukan Login dengan memasukkan email/password dan sistem melakukan otentikasi. Setelah berhasil, anggota dapat mengakses berbagai fitur. Untuk materi, anggota dapat mengambil detail materi, menyimpan progres, dan menyelesaikan kuis yang melibatkan validasi kunci dan skor. Untuk karya/tugas, anggota dapat mengunggah dan mengirimkan tugas, yang divalidasi dan disimpan oleh sistem. Forum memungkinkan anggota membaca, menulis, dan menyimpan komentar.

Diagram ini juga menampilkan alur untuk fitur lain seperti jadwal, presensi, dan notifikasi. Anggota dapat mengambil detail jadwal, dan untuk presensi, mereka dapat melakukan input kode yang divalidasi oleh sistem. Fitur notifikasi memungkinkan anggota melihat daftar notifikasi dan menandai statusnya sebagai dibaca. Terakhir, diagram mencakup interaksi untuk mengelola profil, di mana anggota dapat mengubah data dan menyimpan perubahan. Proses diakhiri dengan logout, di mana sistem mengakhiri sesi pengguna dan mengalihkan mereka kembali ke halaman login.

b. *Sequence diagram* Pengurus



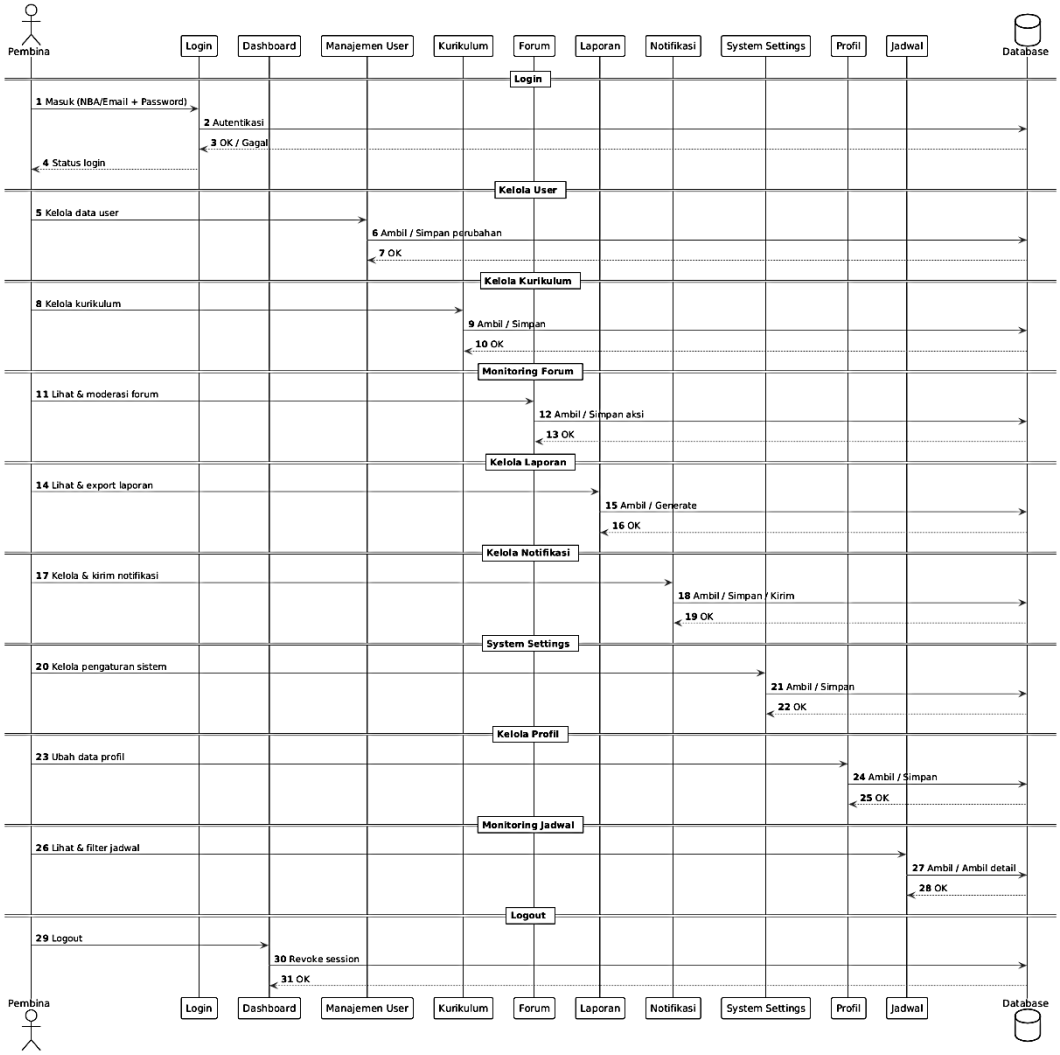
Gambar 4. 30 *Sequence diagram* Pengurus

Penjelasan *Sequence diagram* Pengurus:

Pada diagram ini mengilustrasikan interaksi antara seorang pengurus dan berbagai komponen dalam sistem, termasuk halaman-halaman fungsional dan database. Alur dimulai dengan proses Login, di mana pengurus memasukkan kredensial dan sistem melakukan otentikasi. Setelah berhasil masuk, pengurus dapat mengelola berbagai aspek sistem. Diagram ini menunjukkan bagaimana pengurus dapat mengelola pengguna, materi, kuis, forum, karya/tugas, jadwal, presensi, notifikasi, dan profil mereka sendiri. Setiap interaksi "kelola" melibatkan pengurus yang mengambil atau menyimpan data ke database, yang ditunjukkan dengan panah bolak-balik.

Secara lebih rinci, setiap fitur pengelolaan memiliki alurnya sendiri. Misalnya, dalam Kelola Materi dan Kelola Kuis, pengurus mengambil, mengubah, dan menyimpan data ke database. Untuk Kelola Karya/Tugas, pengurus dapat mengambil tugas/submisi dan memberikan nilai. Demikian pula, dalam Kelola Jadwal dan Kelola Presensi, pengurus dapat mengambil, menyimpan, dan memperbarui aturan atau status. Proses ini diakhiri dengan Logout, di mana sistem membatalkan sesi pengguna, dan pengurus kembali ke halaman login. Interaksi ini menunjukkan peran sentral pengurus dalam menjaga dan memperbarui konten serta data dalam sistem.

c. *Sequence diagram* Pembina



Gambar 4. 31 *Sequence diagram* Pembina

Pada diagram ini menggambarkan interaksi seorang Pembina dengan berbagai komponen dalam sistem, dengan fokus pada peran sebagai monitoring. Setelah berhasil Login, Pembina dapat melakukan berbagai fungsi pemantauan dan pengelolaan. Ia dapat mengelola data pengguna, kurikulum, dan forum, yang melibatkan pengambilan dan penyimpanan data ke database. Pembina juga dapat melihat dan mengekspor laporan, serta mengelola notifikasi untuk dikirimkan kepada pengguna lain. Selain itu, diagram ini menunjukkan kemampuan Pembina

untuk mengubah pengaturan sistem, mengelola profilnya sendiri, dan memantau jadwal yang ada. Semua interaksi ini menunjukkan peran Pembina sebagai administrator yang memiliki visibilitas dan kontrol yang luas terhadap berbagai aspek operasional sistem. Alur diakhiri ketika Pembina melakukan Logout, mengakhiri sesinya dan kembali ke halaman login.

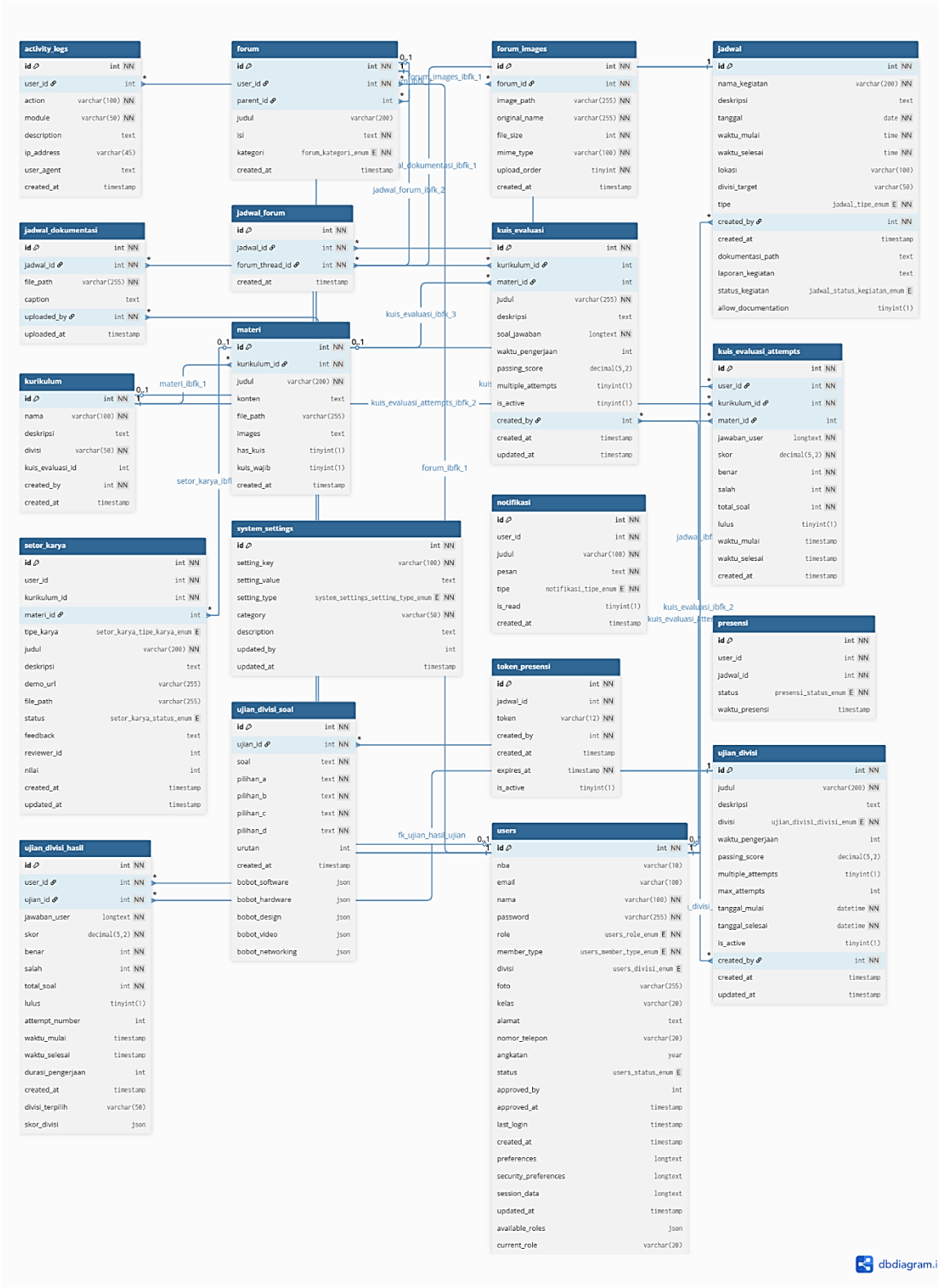
B. Rancangan Database

Dalam penelitian ini, rancangan basis data dibuat untuk mendukung sistem *Mobile learning (M-Learning)* komunitas IT Smansa Parepare yang mengelola berbagai entitas seperti pengguna, materi pembelajaran, kurikulum, evaluasi, forum diskusi, jadwal kegiatan, presensi digital, dan laporan kegiatan.

Database yang digunakan adalah MySQL dengan engine InnoDB. InnoDB mendukung transaksi ACID, kunci asing (foreign key), dan mendukung pengelolaan relasi antar tabel dengan cara yang efisien, baik untuk relasi One-to-Many maupun many-to-many.

Relasi one-to-many (1:N) adalah hubungan di mana satu entitas dari sebuah tabel dapat terhubung dengan banyak entitas di tabel lain, namun setiap entitas di tabel kedua hanya dapat terhubung dengan satu entitas di tabel pertama, contohnya satu guru dapat mengajar banyak mahasiswa. Sebaliknya, relasi many-to-many (M:N) adalah hubungan yang memungkinkan banyak entitas dari satu tabel terhubung dengan banyak entitas di tabel lain, misalnya banyak mahasiswa dapat mengambil banyak mata kuliah. Untuk merepresentasikan relasi many-to-many, dibutuhkan tabel perantara yang berisi kunci penghubung dari kedua tabel utama, yang memecah relasi tersebut menjadi dua relasi one-to-many.

1. *Relasi tabel database*



Gambar 4. 32 Relasi tabel database

Penjelasan Relasi tabel database:

a. **Tabel *user*s → activity_logs (*One-to-Many*)**

1) Relasi: Setiap entri di tabel activity_logs terkait dengan satu pengguna di tabel *user*s (melalui *user*_id). Satu pengguna dapat memiliki banyak aktivitas yang tercatat.

2) Jenis Relasi: *One-to-Many*. Satu pengguna dapat memiliki banyak catatan aktivitas, tetapi satu catatan aktivitas hanya terkait dengan satu pengguna.

3) Penjelasan: Tabel activity_logs menyimpan berbagai aktivitas yang dilakukan oleh pengguna, seperti login atau perubahan data.

b. **Tabel *user*s → forum (*One-to-Many*)**

1) Relasi: Setiap thread di forum dibuat oleh seorang pengguna yang terdapat di tabel *user*s (melalui *user*_id).

2) Jenis Relasi: *One-to-Many*. Satu pengguna dapat membuat banyak thread/forum, tetapi satu thread/forum hanya memiliki satu pembuat (pengguna).

3) Penjelasan: Tabel forum mencatat thread atau diskusi yang dibuat oleh pengguna di platform ini.

c. **Tabel forum → forum_images (*One-to-Many*)**

1) Relasi: Setiap forum atau thread dapat memiliki beberapa gambar yang terkait, yang disimpan di tabel forum_images (melalui forum_id).

2) Jenis Relasi: *One-to-Many*. Satu thread forum dapat memiliki banyak gambar, tetapi setiap gambar hanya terkait dengan satu thread/forum.
3) Penjelasan: Tabel forum_images menyimpan gambar yang diunggah oleh pengguna sebagai bagian dari diskusi di forum.

d. **Tabel jadwal → jadwal_dokumentasi (*One-to-Many*)**
1) Relasi: Setiap jadwal kegiatan dapat memiliki beberapa dokumentasi terkait (seperti file atau gambar), yang disimpan di tabel jadwal_dokumentasi (melalui jadwal_id).
2) Jenis Relasi: *One-to-Many*. Satu jadwal dapat memiliki banyak dokumentasi, tetapi setiap dokumentasi hanya terkait dengan satu jadwal.
3) Penjelasan: Tabel jadwal_dokumentasi menyimpan file atau gambar yang terkait dengan kegiatan yang dijadwalkan.

e. **Tabel jadwal → jadwal_forum (*One-to-Many*)**
1) Relasi: Setiap jadwal kegiatan dapat memiliki satu atau lebih forum diskusi terkait, yang disimpan di tabel jadwal_forum (melalui jadwal_id).
2) Jenis Relasi: *One-to-Many*. Satu jadwal kegiatan dapat memiliki banyak forum diskusi terkait, tetapi satu forum diskusi hanya terkait dengan satu jadwal kegiatan.
3) Penjelasan: Tabel jadwal_forum menyimpan informasi tentang forum yang terkait dengan jadwal kegiatan tertentu.

f. **Tabel jadwal_forum → forum (*One-to-Many*)**

1) Relasi: Setiap forum yang tercatat di jadwal_forum adalah referensi ke sebuah forum yang ada di tabel forum (melalui forum_thread_id).
2) Jenis Relasi: *One-to-Many*. Satu forum diskusi dapat terkait dengan banyak jadwal forum, tetapi satu jadwal forum hanya terkait dengan satu thread/forum.
3) Penjelasan: Tabel jadwal_forum menghubungkan forum diskusi dengan jadwal kegiatan yang terkait.

g. **Tabel kurikulum → materi (*One-to-Many*)**

1) Relasi: Setiap kurikulum dapat memiliki banyak materi yang terkait, yang disimpan di tabel materi (melalui kurikulum_id).
2) Jenis Relasi: *One-to-Many*. Satu kurikulum dapat memiliki banyak materi, tetapi setiap materi hanya terkait dengan satu kurikulum.
3) Penjelasan: Tabel materi mencatat materi atau pelajaran yang merupakan bagian dari suatu kurikulum.

h. **Tabel kuis_evaluasi → kuis_evaluasi_*Attempts* (*One-to-Many*)**

1) Relasi: Setiap kuis evaluasi dapat memiliki banyak percakapan atau usaha pengerjaan yang dilakukan oleh pengguna, yang tercatat di tabel kuis_evaluasi_*Attempts* (melalui kurikulum_id dan *user*_id).

2) Jenis Relasi: *One-to-Many*. Satu kuis evaluasi dapat memiliki banyak upaya pengerjaan, tetapi setiap upaya pengerjaan hanya terkait dengan satu kuis evaluasi.
3) Penjelasan: Tabel kuis_evaluasi_*Attempts* menyimpan catatan mengenai upaya yang dilakukan oleh pengguna untuk menyelesaikan kuis evaluasi.

i. **Tabel *user*s → presensi (*One-to-Many*)**

1) Relasi: Setiap pengguna dapat memiliki banyak catatan kehadiran yang tercatat di tabel presensi (melalui *user*_id).
2) Jenis Relasi: *One-to-Many*. Satu pengguna dapat memiliki banyak catatan kehadiran dalam jadwal kegiatan yang berbeda, tetapi satu catatan kehadiran hanya terkait dengan satu pengguna.
3) Penjelasan: Tabel presensi mencatat status kehadiran pengguna dalam kegiatan yang dijadwalkan.

j. **Tabel jadwal → presensi (*One-to-Many*)**

1) Relasi: Setiap jadwal kegiatan dapat memiliki banyak catatan kehadiran pengguna yang tercatat di tabel presensi (melalui jadwal_id).
2) Jenis Relasi: *One-to-Many*. Satu jadwal kegiatan dapat memiliki banyak catatan kehadiran, tetapi setiap catatan kehadiran hanya terkait dengan satu jadwal.
3) Penjelasan: Tabel presensi mencatat kehadiran pengguna dalam setiap jadwal kegiatan.

k. **Tabel *user*s → setor_karya (*One-to-Many*)**

1) Relasi: Setiap pengguna dapat mengirimkan banyak karya yang tercatat dalam tabel setor_karya (melalui *user*_id).
2) Jenis Relasi: *One-to-Many*. Satu pengguna dapat mengirim banyak karya, tetapi satu karya hanya terkait dengan satu pengguna.
3) Penjelasan: Tabel setor_karya menyimpan karya yang dikumpulkan oleh pengguna dalam bentuk file atau *URL* yang dikaitkan dengan kurikulum.

l. **Tabel ujian_divisi → ujian_divisi_soal (One-to-Many)**

1) Relasi: Setiap ujian yang dibuat di ujian_divisi dapat memiliki banyak butir soal yang disimpan di ujian_divisi_soal melalui ujian_id.
2) Penjelasan: Tabel ujian_divisi_soal berisi daftar pertanyaan untuk setiap paket ujian penentuan divisi.

m. **Tabel ujian_divisi → ujian_divisi_hasil (One-to-Many)**

1) Relasi: Setiap paket ujian di ujian_divisi dapat memiliki banyak hasil pengerjaan dari pengguna yang berbeda, dihubungkan melalui ujian_id.
2) Penjelasan: Tabel ujian_divisi_hasil merekam semua percobaan pengerjaan ujian oleh seluruh pengguna.

2. Kamus Data

Kamus data adalah suatu dokumen atau tabel yang berisi penjelasan terstruktur mengenai seluruh elemen data yang digunakan dalam sebuah sistem, termasuk nama atribut, tipe data, panjang, batasan (constraint), nilai default, hingga keterangannya. Berikut ini Kamus data yang digunakan dalam Rancang Bangun Aplikasi *M-Learning* Komunitas IT Smansa Parepare.

a. Kamus Data Tabel *Activity logs*

Nama *database*: *db_mlearning*

Nama Tabel: *activity_logs*

Primary Key: id

Panjang *record:* 206

Tabel 4. 4 Kamus Data Tabel *Activity logs*

Nama Kolom	Tipe data	Ukuran	Deskripsi
id	Int	11	ID unik untuk setiap log aktivitas.
user_id	Int	11	ID pengguna yang melakukan aksi (merujuk ke tabel users).
action	Varchar	100	Jenis aksi yang dilakukan (contoh: 'login', 'create_materi').
module	Varchar	50	Modul atau bagian sistem tempat aksi terjadi.
description	Text	-	Deskripsi detail dari aktivitas yang dilakukan.
ip_address	Varchar	45	Alamat IP pengguna saat melakukan aksi.
user_agent	Text	-	Informasi browser dan sistem operasi pengguna.
created_at	Timestamp	-	Waktu dan tanggal kapan log aktivitas tercatat.

b. Kamus Data Tabel Kurikulum

Nama *database*: *db_mlearning*

Nama Tabel: *kurikulum*

Primary Key: id

Panjang *record:* 172

Tabel 4. 5 Kamus Data Tabel Kurikulum

Nama Kolom	Tipe data	Ukuran	Deskripsi
id	Int	11	ID unik untuk setiap kurikulum.
nama	Varchar	100	Nama dari kurikulum.
deskripsi	Text	-	Deskripsi detail mengenai kurikulum.
divisi	Varchar	50	Divisi yang dituju oleh kurikulum ini.
kuis_evaluasi_id	Int	11	ID kuis evaluasi akhir untuk kurikulum ini (jika ada).
created_by	Int	11	ID pengguna yang membuat kurikulum (merujuk ke tabel users).
created_at	Timestamp	-	Waktu dan tanggal kapan kurikulum dibuat.

c. Kamus Data Tabel Materi

Nama *database*: *db_mlearning*

Nama Tabel: *kurikulum*

Primary Key: id

Panjang *record:* 468

Tabel 4. 6 Kamus Data Tabel Materi

Nama Kolom	Tipe data	Ukuran	Deskripsi
id	Int	11	ID unik untuk setiap materi pembelajaran.
kurikulum_id	Int	11	ID kurikulum tempat materi ini berada (merujuk ke tabel kurikulum).
judul	Varchar	200	Judul materi.
konten	Text	-	Isi konten utama dari materi.
file_path	Varchar	255	Path atau lokasi penyimpanan file pendukung materi.
images	Text	-	Path atau daftar path gambar yang digunakan dalam materi.
has_kuis	Tinyint	1	Penanda apakah materi ini memiliki kuis (1=Ya, 0=Tidak).
kuis_wajib	Tinyint	1	Penanda apakah kuis untuk materi ini wajib dikerjakan.
created_at	Timestamp	-	Waktu dan tanggal kapan materi dibuat.

d. Kamus Data Tabel Kuis Evaluasi

Nama *database*: *db_mlearning*

Nama Tabel: kuis_evaluasi

Primary Key: id

Panjang *record:* 285

Tabel 4. 7 Kamus Data Tabel Kuis Evaluasi

Nama Kolom	Tipe data	Ukuran	Deskripsi
id	Int	11	ID unik untuk setiap kuis.
kurikulum_id	Int	11	ID kurikulum terkait (merujuk ke tabel kurikulum).
materi_id	Int	11	ID materi terkait (merujuk ke tabel materi).
judul	Varchar	255	Judul dari kuis evaluasi.
deskripsi	Text	-	Deskripsi atau instruksi pengerjaan kuis.
soal_jawaban	Longtext	-	Data JSON yang berisi daftar soal dan pilihan jawaban.
waktu_pengerjaan	Int	11	Durasi pengerjaan kuis dalam detik (default: 900).
passing_score	Decimal	5,2	Nilai minimum untuk lulus kuis (default: 80.00).
multiple_attempts	Tinyint	1	Izin untuk mengerjakan kuis lebih dari sekali (1=Ya, 0=Tidak).
is_active	Tinyint	1	Status aktif kuis (1=Aktif, 0=Tidak Aktif).
created_by	Int	11	ID pengguna yang membuat kuis (merujuk ke tabel users).
created_at	Timestamp	-	Waktu dan tanggal kapan kuis dibuat.
updated_at	Timestamp	-	Waktu dan tanggal kapan kuis terakhir diperbarui.

e. Kamus Data Tabel Kuis Evaluasi Attempts

Nama *database*: *db_mlearning*

Nama Tabel: kuis_evaluasi_attempts

Primary Key: id

Panjang *record:* 60

Tabel 4. 8 Kamus Data Tabel Kuis Evaluasi Attempts

Nama Kolom	Tipe data	Ukuran	Deskripsi
id	Int	11	ID unik untuk setiap percobaan pengerjaan kuis.
user_id	Int	11	ID pengguna yang mengerjakan kuis (merujuk ke tabel users).
kurikulum_id	Int	11	ID kurikulum dari kuis yang dikerjakan.
materi_id	Int	11	ID materi dari kuis yang dikerjakan.
jawaban_user	Longtext	-	Data JSON yang berisi jawaban yang dipilih oleh pengguna.
skor	Decimal	5,2	Skor akhir yang diperoleh pengguna.
benar	Int	11	Jumlah jawaban yang benar.
salah	Int	11	Jumlah jawaban yang salah.
total_soal	Int	11	Total jumlah soal dalam kuis.

Lanjutan Tabel 4. 8

Nama Kolom	Tipe data	Ukuran	Deskripsi
lulus	Tinyint	1	Status kelulusan (1=Lulus, 0=Tidak Lulus).
waktu_mulai	Timestamp	-	Waktu saat pengguna memulai pengerjaan kuis.
waktu_selesai	Timestamp	-	Waktu saat pengguna menyelesaikan pengerjaan kuis.
created_at	Timestamp	-	Waktu dan tanggal kapan data percobaan dibuat.

f. Kamus Data Tabel Setor Karya

Nama *database*: *db_mlearning*

Nama Tabel: setor_karya

Primary Key: id

Panjang *record:* 515

Tabel 4. 9 Kamus Data Tabel Setor Karya

Nama Kolom	Tipe data	Ukuran	Deskripsi
id	Int	11	ID unik untuk setiap karya yang disetor.
user_id	Int	11	ID pengguna yang menyetor karya (merujuk ke tabel users).
kurikulum_id	Int	11	ID kurikulum terkait.
materi_id	Int	11	ID materi terkait (jika ada).
tipe_karya	Enum	-	Tipe karya: 'Tugas', 'Latihan', 'Project', 'Lainnya'.
judul	Varchar	200	Judul dari karya yang disetor.
deskripsi	Text	-	Deskripsi detail mengenai karya.
demo_url	Varchar	255	URL demo atau pratinjau karya.
file_path	Varchar	255	Path penyimpanan file karya.
status	Enum	-	Status penilaian: 'pending', 'review', 'approved', 'rejected'.
feedback	Text	-	Umpan balik dari penilai (reviewer).
reviewer_id	Int	11	ID pengguna yang menilai karya (merujuk ke tabel users).
nilai	Int	11	Nilai atau skor yang diberikan untuk karya.
created_at	Timestamp	-	Waktu dan tanggal kapan karya disetor.
updated_at	Timestamp	-	Waktu dan tanggal kapan data terakhir diperbarui.

g. Kamus Data Tabel Presensi

Nama *database*: *db_mlearning*

Nama Tabel: presensi

Primary Key: id

Panjang *record:* 33

Tabel 4. 10 Kamus Data Tabel Presensi

Nama Kolom	Tipe data	Ukuran	Deskripsi
id	Int	11	ID unik untuk setiap data presensi.
user_id	Int	11	ID pengguna yang melakukan presensi (merujuk ke tabel users).
jadwal_id	Int	11	ID jadwal kegiatan yang dihadiri (merujuk ke tabel jadwal).
status	Enum	-	Status kehadiran: 'hadir', 'tidak_hadir', 'izin'.
waktu_presensi	Timestamp	-	Waktu dan tanggal kapan presensi dilakukan.

h. Kamus Data Tabel Token Presensi

Nama *database*: *db_mlearning*

Nama Tabel: token_presensi

Primary Key: id

Panjang *record:* 35

Tabel 4. 11 Kamus Data Token Presensi

Nama Kolom	Tipe data	Ukuran	Deskripsi
id	Int	11	ID unik untuk setiap token.
jadwal_id	Int	11	ID jadwal terkait (merujuk ke tabel jadwal).
token	Varchar	12	Kode token unik untuk presensi.
created_by	Int	11	ID pengguna yang membuat token (merujuk ke tabel users).
created_at	Timestamp	-	Waktu dan tanggal kapan token dibuat.
expires_at	Timestamp	-	Waktu dan tanggal kedaluwarsa token.
is_active	Tinyint	1	Status aktif token (1=Aktif, 0=Tidak Aktif).

i. Kamus Data Tabel Forum

Nama *database*: *db_mlearning*

Nama Tabel: forum

Primary Key: id

Panjang *record:* 222

Tabel 4. 12 Kamus Data Tabel Forum

Nama Kolom	Tipe data	Ukuran	Deskripsi
id	Int	11	ID unik untuk setiap postingan atau balasan forum.
user_id	Int	11	ID pengguna yang membuat postingan (merujuk ke tabel users).
parent_id	Int	11	ID dari postingan induk (jika ini adalah balasan). NULL jika ini adalah thread baru.
judul	Varchar	200	Judul dari thread forum.
isi	Text	-	Isi konten dari postingan atau balasan.
kategori	Enum	-	Kategori forum: 'diskusi', 'tugas', 'informasi', 'sekolah', 'lainnya'.
created_at	Timestamp	-	Waktu dan tanggal kapan postingan dibuat.

j. Kamus Data Tabel Forum *Images*

Nama *database*: *db_mlearning*

Nama Tabel: forum_images

Primary Key: id

Panjang *record:* 431

Tabel 4. 13 Kamus Data Tabel Forum *Images*

Nama Kolom	Tipe data	Ukuran	Deskripsi
id	Int	11	ID unik untuk setiap gambar yang diunggah ke forum.
forum_id	Int	11	ID postingan forum terkait (merujuk ke tabel forum).
image_path	Varchar	255	Path atau lokasi penyimpanan file gambar.
original_name	Varchar	255	Nama asli file gambar saat diunggah.
file_size	Int	11	Ukuran file gambar dalam bytes.
mime_type	Varchar	100	Tipe MIME dari file gambar (contoh: 'image/jpeg').
upload_order	Tinyint	1	Urutan gambar jika ada lebih dari satu dalam satu post.
created_at	Timestamp	-	Waktu dan tanggal kapan gambar diunggah.

k. Kamus Data Tabel Jadwal Dokumentasi

Nama *database*: *db_mlearning*

Nama Tabel: jadwal_dokumentasi

Primary Key: id

Panjang *record:* 277

Tabel 4. 14 Kamus Data Tabel Jadwal Dokumentasi

Nama Kolom	Tipe data	Ukuran	Deskripsi
id	Int	11	ID unik untuk file dokumentasi.
jadwal_id	Int	11	ID jadwal terkait (merujuk ke tabel jadwal).
file_path	Varchar	255	Path atau lokasi penyimpanan file dokumentasi.
caption	Text	-	Keterangan atau caption untuk file dokumentasi.
uploaded_by	Int	11	ID pengguna yang mengunggah dokumentasi (merujuk ke tabel users).
uploaded_at	Timestamp	-	Waktu dan tanggal kapan dokumentasi diunggah.

l. Kamus Data Tabel Jadwal Forum

Nama *database*: *db_mlearning*

Nama Tabel: jadwal_forum

Primary Key: id

Panjang *record:* 33

Tabel 4. 15 Kamus Data Tabel Jadwal Forum

Nama Kolom	Tipe data	Ukuran	Deskripsi
id	Int	11	ID unik untuk relasi jadwal dan forum.
jadwal_id	Int	11	ID jadwal yang terhubung (merujuk ke tabel jadwal).
forum_thread_id	Int	11	ID thread forum yang terhubung (merujuk ke tabel forum).
created_at	Timestamp	-	Waktu dan tanggal kapan relasi dibuat.

m. Kamus Data Tabel System_*Settings*

Nama *database*: *db_mlearning*

Nama Tabel: jadwal_forum

Primary Key: id

Panjang *record:* 33

Tabel 4. 16 Kamus Data Tabel *System Settings*

Nama Kolom	Tipe data	Ukuran	Deskripsi
id	Int	11	ID unik untuk setiap pengaturan sistem.
setting_key	Varchar	100	Kunci unik untuk identifikasi pengaturan.
setting_value	Text	-	Nilai dari pengaturan.
setting_type	Enum	-	Tipe data nilai: 'text', 'number', 'boolean', 'json', 'file'.
category	Varchar	50	Kategori untuk pengelompokan pengaturan.
description	Text	-	Deskripsi fungsi dari pengaturan.
updated_by	Int	11	ID pengguna yang terakhir mengubah pengaturan.
updated_at	Timestamp	-	Waktu dan tanggal kapan pengaturan terakhir diubah.

n. Kamus Data Tabel Notifikasi

Nama *database*: *db_mlearning*

Nama Tabel: notifikasi

Primary Key: id

Panjang *record:* 112

Tabel 4. 17 Kamus Data Tabel Notifikasi

Nama Kolom	Tipe data	Ukuran	Deskripsi
id	Int	11	ID unik untuk setiap notifikasi.
user_id	Int	11	ID pengguna penerima notifikasi (merujuk ke tabel users).
judul	Varchar	100	Judul singkat dari notifikasi.
pesan	Text	-	Isi pesan notifikasi.
tipe	Enum	-	Tipe notifikasi: 'system', 'learning', 'forum', 'jadwal', dll.
is_read	Tinyint	1	Status keterbacaan notifikasi (1=Sudah dibaca, 0=Belum).
created_at	Timestamp	-	Waktu dan tanggal kapan notifikasi dibuat.

o. Kamus Data Tabel *Users*

Nama *database*: *db_mlearning*

Nama Tabel: *users*

Primary Key: id

Panjang *record: 478*

Tabel 4. 18 Kamus Data Tabel *Users*

Nama Kolom	Tipe data	Ukuran	Deskripsi
id	Int	11	ID unik untuk setiap pengguna.
nba	Varchar	10	Nomor Baku Anggota.
email	Varchar	100	Alamat email pengguna.
nama	Varchar	100	Nama lengkap pengguna.
password	Varchar	255	Hash dari password pengguna.
role	Enum	-	Peran pengguna: 'anggota', 'pengurus', 'pembina'.
member_type	Enum	-	Jenis keanggotaan: 'aktif' (akses penuh) atau 'pasif' (akses terbatas).
divisi	Enum	-	Divisi pengguna: 'hardware', 'software', 'video_editing', dll.
foto	Varchar	255	Path atau nama file foto profil pengguna.
kelas	Varchar	20	Kelas pengguna (jika berlaku).
alamat	Text	-	Alamat tempat tinggal pengguna.
nomor_telepon	Varchar	20	Nomor telepon aktif pengguna.
angkatan	Year	4	Tahun angkatan pengguna.
status	Enum	-	Status akun: 'aktif', 'non_aktif', 'pending'.
approved_by	Int	11	ID pengguna yang menyetujui pendaftaran.
approved_at	Timestamp	-	Waktu dan tanggal pendaftaran disetujui.
last_login	Timestamp	-	Waktu dan tanggal terakhir pengguna login.
created_at	Timestamp	-	Waktu dan tanggal kapan akun dibuat.
preferences	Longtext	-	Preferensi pengguna dalam format JSON.
security_preferences	Longtext	-	Preferensi keamanan dalam format JSON.
session_data	Longtext	-	Data sesi pengguna dalam format JSON.
updated_at	Timestamp	-	Waktu dan tanggal kapan profil terakhir diubah.

p. Kamus Data Tabel Jadwal

Nama *database*: *db_mlearning*

Nama Tabel: jadwal

Primary Key: id

Panjang *record: 362*

Tabel 4. 19 Kamus Data Tabel jadwal

Nama Kolom	Tipe data	Ukuran	Deskripsi
id	Int	11	ID unik untuk setiap jadwal kegiatan.
nama_kegiatan	Varchar	200	Nama dari kegiatan atau acara.
deskripsi	Text	-	Deskripsi detail mengenai kegiatan.
tanggal	Date	-	Tanggal pelaksanaan kegiatan.
waktu_mulai	Time	-	Waktu dimulainya kegiatan.
waktu_selesai	Time	-	Waktu selesainya kegiatan.
lokasi	Varchar	100	Lokasi atau tempat kegiatan dilaksanakan.
divisi_target	Varchar	50	Divisi yang menjadi target peserta kegiatan.
tipe	Enum	-	Tipe kegiatan: 'workshop', 'meeting', 'project', 'exam'.
created_by	Int	11	ID pengguna yang membuat jadwal (merujuk ke tabel users).
created_at	Timestamp	-	Waktu dan tanggal kapan jadwal dibuat.
dokumentasi_path	Text	-	Path penyimpanan untuk file dokumentasi.
laporan_kegiatan	Text	-	Laporan hasil akhir dari kegiatan yang telah dilaksanakan.
status_kegiatan	Enum	-	Status kegiatan: 'planned', 'ongoing', 'completed', 'cancelled'.
allow_documentation	Tinyint	1	Izin untuk mengunggah dokumentasi (1=Ya, 0=Tidak).

q. Kamus Data Tabel Ujian Divisi

Nama *database*: *db_mlearning*

Nama Tabel: ujian_divisi

Primary Key: id

Panjang *record: 251*

Tabel 4. 20 Kamus Data Tabel ujian_divisi

Nama Kolom	Tipe data	Ukuran	Deskripsi
id	Int	11	ID unik untuk setiap ujian.
judul	Varchar	200	Judul dari ujian divisi.
deskripsi	Text	-	Deskripsi atau instruksi pengerjaan ujian.
divisi	Enum	-	Divisi yang diuji: 'hardware', 'software', 'video_editing', 'graphic_design', 'networking', 'all'.
waktu_pengerjaan	Int	11	Durasi pengerjaan ujian dalam detik (default: 3600).
passing_score	Decimal	5,2	Nilai minimum untuk lulus ujian (default: 70.00).
multiple_attempts	Tinyint	1	Izin untuk mengerjakan ujian lebih dari sekali (1=Ya, 0=Tidak).
max_attempts	Int	11	Jumlah maksimal percobaan yang diizinkan (default: 3).
tanggal_mulai	Datetime	-	Waktu dan tanggal ujian mulai dapat diakses.
tanggal_selesai	Datetime	-	Waktu dan tanggal ujian berakhir.
is_active	Tinyint	1	Status aktif ujian (1=Aktif, 0=Tidak Aktif).
created_by	Int	11	ID pengguna yang membuat ujian (merujuk ke tabel users).
created_at	Timestamp	-	Waktu dan tanggal kapan ujian dibuat.
updated_at	Timestamp	-	Waktu dan tanggal kapan ujian terakhir diperbarui.

r. Kamus Data Tabel Ujian Divisi Hasil

Nama *database*: *db_mlearning*

Nama Tabel: ujian_divisi_hasil

Primary Key: id

Panjang *record: 144*

Tabel 4. 21 Kamus Data Tabel ujian_divisi_hasil

Nama Kolom	Tipe data	Ukuran	Deskripsi
id	Int	11	ID unik untuk setiap hasil ujian.
user_id	Int	11	ID pengguna yang mengerjakan ujian (merujuk ke tabel users).
ujian_id	Int	11	ID ujian yang dikerjakan (merujuk ke tabel ujian_divisi).
jawaban_user	Longtext	-	Data JSON yang berisi jawaban yang dipilih oleh pengguna.
skor	Decimal	5,2	Skor akhir yang diperoleh pengguna.
benar	Int	11	Jumlah jawaban yang benar.
salah	Int	11	Jumlah jawaban yang salah.
total_soal	Int	11	Total jumlah soal dalam ujian.
lulus	Tinyint	1	Status kelulusan (1=Lulus, 0=Tidak Lulus).
attempt_number	Int	11	Percobaan pengerjaan ke-berapa (default: 1).
waktu_mulai	Timestamp	-	Waktu saat pengguna memulai pengerjaan ujian.
waktu_selesai	Timestamp	-	Waktu saat pengguna menyelesaikan pengerjaan ujian.
durasi_pengerjaan	Int	11	Durasi pengerjaan pengguna dalam detik.
created_at	Timestamp	-	Waktu dan tanggal kapan data hasil dibuat.
divisi_terpilih	Varchar	50	Divisi yang direkomendasikan berdasarkan skor tertinggi.
skor_divisi	Json	-	Skor untuk setiap divisi dalam format JSON.

s. Kamus Data Tabel Ujian Divisi Soal

Nama *database*: *db_mlearning*

Nama Tabel: ujian_divisi_soal

Primary Key: id

Panjang *record: 33*

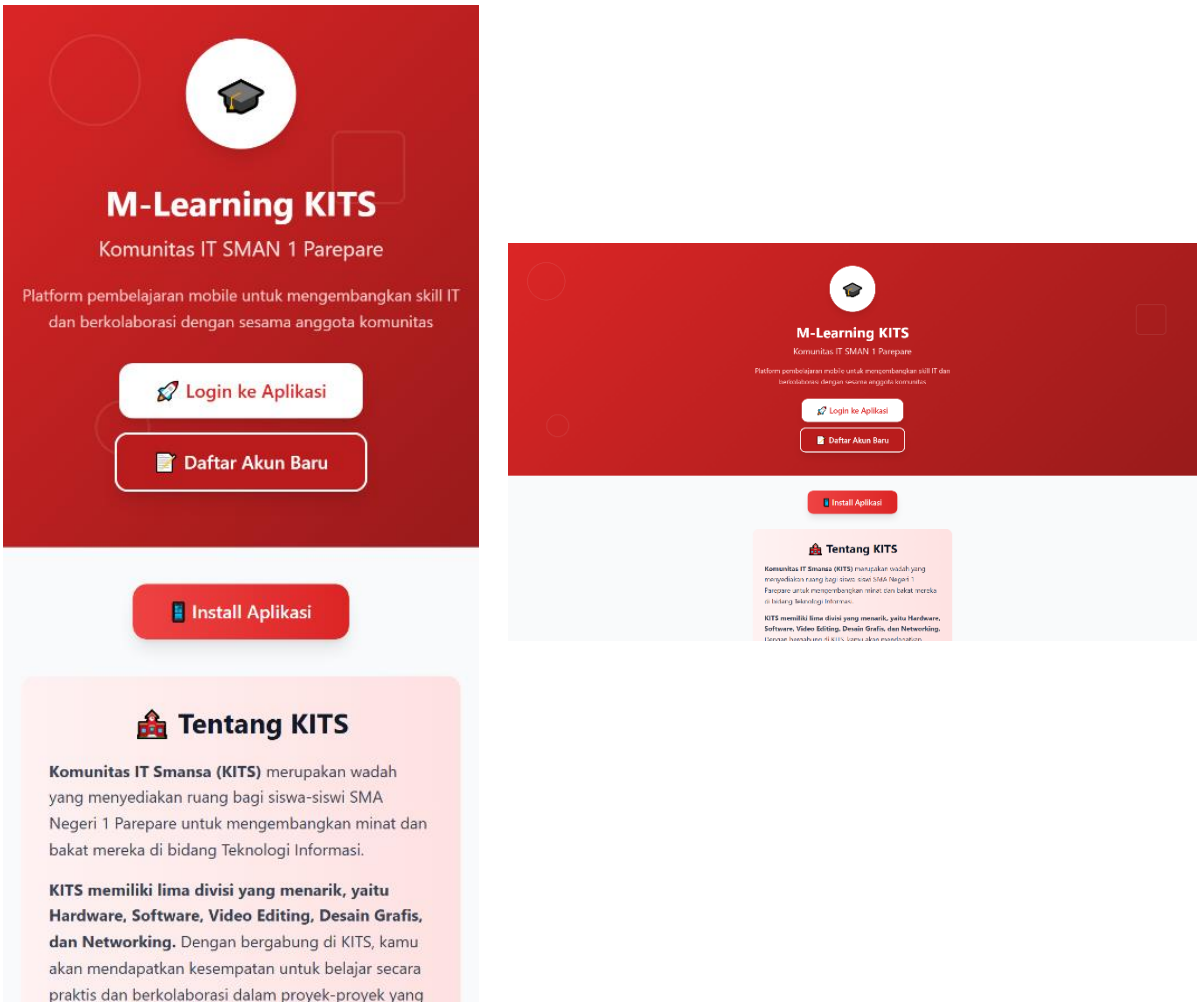
Tabel 4. 22 Kamus Data Tabel ujian_divisi_soal

Nama Kolom	Tipe data	Ukuran	Deskripsi
id	Int	11	ID unik untuk setiap soal ujian.
ujian_id	Int	11	ID ujian terkait (merujuk ke tabel ujian_divisi).
soal	Text	-	Isi pertanyaan atau soal.
pilihan_a	Text	-	Teks untuk pilihan jawaban A.
pilihan_b	Text	-	Teks untuk pilihan jawaban B.
pilihan_c	Text	-	Teks untuk pilihan jawaban C.
pilihan_d	Text	-	Teks untuk pilihan jawaban D.
urutan	Int	11	Urutan tampilan soal (default: 1).
created_at	Timestamp	-	Waktu dan tanggal kapan soal dibuat.
bobot_software	Json	-	Bobot jawaban untuk divisi software dalam format JSON.
bobot_hardware	Json	-	Bobot jawaban untuk divisi hardware dalam format JSON.
bobot_design	Json	-	Bobot jawaban untuk divisi design dalam format JSON.
bobot_video	Json	-	Bobot jawaban untuk divisi video editing dalam format JSON.
bobot_networking	Json	-	Bobot jawaban untuk divisi networking dalam format JSON.

C. Detail Sistem

1. Halaman *Landing Page* / Tampilan Awal

Tampilan ini adalah halaman utama aplikasi *Mobile learning (M-Learning)* yang memperkenalkan platform belajar IT dengan profil singkat tentang Komunitas IT Smansa Parepare.



Gambar 4. 33 Tampilan Utama Aplikasi

```
<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>M-Learning KITS</title>
</head>
```

```
<body>
    <div class="header-section">
        <div class="logo-container">
            <div class="logo-icon">🎓</div>
        </div>
        <h1 class="main-title">M-Learning KITS</h1>
        <p class="subtitle">Komunitas IT SMAN 1 Parepare</p>
        <p class="description">Platform pembelajaran mobile untuk
mengembangkan skill IT dan berkolaborasi dengan sesama anggota
komunitas</p>

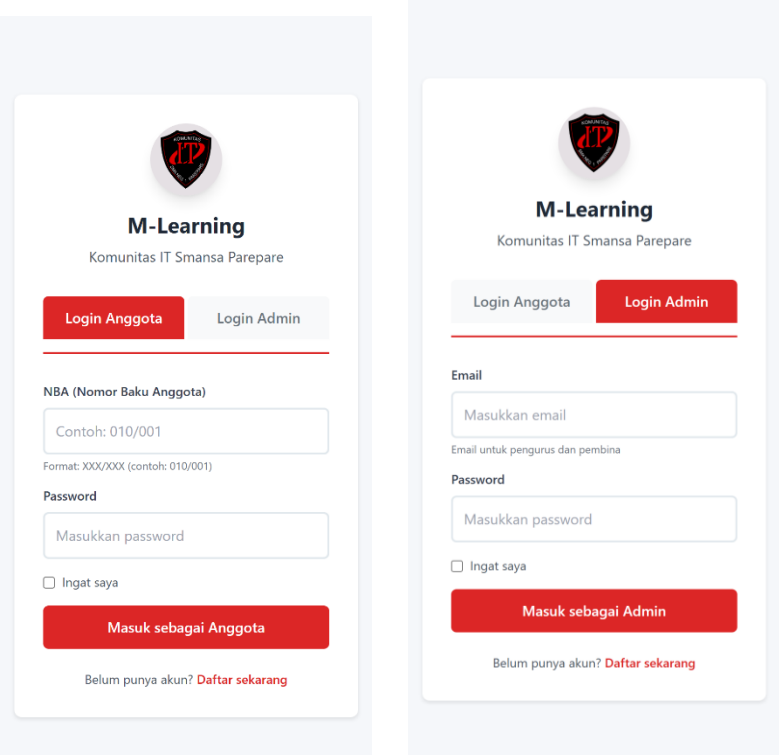
        <div class="button-container">
            <button class="login-btn">
                <span class="btn-icon">🚀</span>
                Login ke Aplikasi
            </button>
            <button class="register-btn">
                <span class="btn-icon">📄</span>
                Daftar Akun Baru
            </button>
        </div>

        <button class="install-btn">
            <span class="btn-icon">□□</span>
            Install Aplikasi
        </button>
    </div>

    <div class="content-section">
        <h2 class="section-title">
            <span class="title-icon">□□</span>
            Tentang KITS
        </h2>
        <p  class="content-text">Komunitas  IT  Smansa  (KITS)
merupakan wadah yang menyediakan ruang bagi siswa-siswi SMA Negeri
1 Parepare untuk mengembangkan minat dan bakat mereka di bidang
Teknologi Informasi.</p>
        <p class="content-text">KITS memiliki lima divisi yang
menarik, yaitu Hardware, Software, Video Editing, Desain Grafis,
dan Networking. Dengan bergabung di KITS, kamu akan mendapatkan
kesempatan untuk belajar secara praktis dan berkolaborasi dalam
proyek-proyek yang</p>
    </div>
</body>
</html>
```

2. Halaman *Login*

Halaman ini menampilkan antarmuka login untuk platform *M-Learning* dari Komunitas IT Smansa Parepare, yang dirancang untuk dua jenis pengguna berbeda: Anggota dan Admin. Tampilan kiri, "Login Anggota," meminta pengguna memasukkan Nomor Baku Anggota (NBA) dan kata sandi, sementara



Gambar 4. 34 Tampilan Login

tampilan kanan, "Login Admin," menggunakan email dan kata sandi, yang secara spesifik diperuntukkan bagi pengurus dan pembina. Kedua antarmuka ini memiliki desain yang konsisten, dilengkapi dengan opsi "Ingat saya" serta tautan untuk pendaftaran bagi pengguna yang belum memiliki akun, memungkinkan pengguna untuk masuk sesuai dengan peran mereka dalam komunitas.

```
<?php
session_start();
include 'config/database.php';
include 'includes/functions.php';
```

```
$error_message = '';
$login_type = isset($_GET['type']) ? $_GET['type'] : 'anggota';

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $password = $_POST['password'];
    $remember = isset($_POST['remember']) ? true : false;

    if ($login_type == 'anggota') {
        $nba = $_POST['nba'];
        $query = "SELECT * FROM anggota WHERE nba = ? AND
password = ?";
        $stmt = $conn->prepare($query);
        $stmt->bind_param("ss", $nba, md5($password));
        $stmt->execute();
        $result = $stmt->get_result();

        if ($result->num_rows > 0) {
            $user = $result->fetch_assoc();
            $_SESSION['user_id'] = $user['id'];
            $_SESSION['user_type'] = 'anggota';
            $_SESSION['nba'] = $user['nba'];
            if ($remember) {
                setcookie('remember_token',
generateToken($user['id']), time() + (86400 * 30), '/');
            }
            header('Location: dashboard/anggota');
            exit();
        } else {
            $error_message = 'NBA atau password salah';
        }
    } else {
        $email = $_POST['email'];
        $query = "SELECT * FROM admin WHERE email = ? AND
password = ?";
        $stmt = $conn->prepare($query);
        $stmt->bind_param("ss", $email, md5($password));
        $stmt->execute();
        $result = $stmt->get_result();

        if ($result->num_rows > 0) {
            $user = $result->fetch_assoc();
            $_SESSION['user_id'] = $user['id'];
            $_SESSION['user_type'] = $user['role'];
            $_SESSION['email'] = $user['email'];
            if ($remember) {
                setcookie('remember_token',
generateToken($user['id']), time() + (86400 * 30), '/');
            }
            header('Location: dashboard/admin');
            exit();
        } else {
            $error_message = 'Email atau password salah';
        }
    }
}
?>
```

```
<div class="login-container">
    <div class="login-header">
        <div class="logo">
            <img src="assets/images/logo-shield.png" alt="Logo">
        </div>
        <h1>M-Learning</h1>
        <p>Komunitas IT Smansa Parepare</p>
    </div>

    <div class="login-tabs">
        <a href="?type=anggota" class="tab-btn <?php echo
$login_type == 'anggota' ? 'active' : ''; ?>">Login Anggota</a>
        <a href="?type=admin" class="tab-btn <?php echo
$login_type == 'admin' ? 'active' : ''; ?>">Login Admin</a>
    </div>

    <form method="POST" class="login-form">
        <?php if ($error_message): ?>
            <div class="error-message"><?php echo
$error_message; ?></div>
        <?php endif; ?>

        <?php if ($login_type == 'anggota'): ?>
            <div class="form-group">
                <label>NBA (Nomor Baku Anggota)</label>
                <input type="text" name="nba"
placeholder="Contoh: 010/001" required>
                <small>Format: XXX/XXX (contoh: 010/001)</small>
            </div>
        <?php else: ?>
            <div class="form-group">
                <label>Email</label>
                <input type="email" name="email"
placeholder="Masukkan email" required>
                <small>Email untuk pengurus dan pembina</small>
            </div>
        <?php endif; ?>

        <div class="form-group">
            <label>Password</label>
            <input type="password" name="password"
placeholder="Masukkan password" required>
        </div>

        <div class="form-group">
            <label class="checkbox-label">
                <input type="checkbox" name="remember">
                Ingat saya
            </label>
        </div>

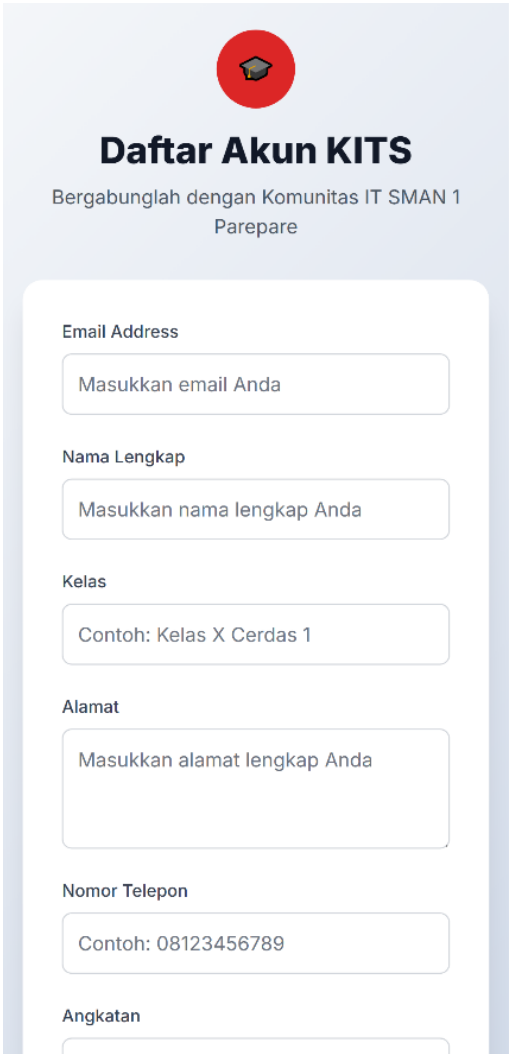
        <button type="submit" class="login-btn">
            <?php echo $login_type == 'anggota' ? 'Masuk sebagai
Anggota' : 'Masuk sebagai Admin'; ?>
        </button>
```

```
    </form>

    <div class="register-link">
        <p>Belum punya akun? <a href="register.php">Daftar
sekarang</a></p>
    </div>
</div>
```

3. Halaman *Register*

Halaman ini adalah halaman pendaftaran akun *Mobile learning (M-Learning)* KITS, di mana calon anggota Komunitas IT SMAN 1 Parepare dapat



Gambar 4. 35 Tampilan Halaman Registrasi Anggota

mengisi informasi pribadi seperti alamat email, nama lengkap, kelas, alamat, nomor telepon, dan angkatan untuk bergabung.

```
<?php
include 'config/database.php';
include 'includes/functions.php';

$error_message = '';
$success_message = '';

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $email = $_POST['email'];
    $nama_lengkap = $_POST['nama_lengkap'];
    $kelas = $_POST['kelas'];
    $alamat = $_POST['alamat'];
    $nomor_telepon = $_POST['nomor_telepon'];
    $angkatan = $_POST['angkatan'];

    // Validasi email sudah terdaftar
    $check_email = "SELECT id FROM anggota WHERE email = ?";
    $stmt = $conn->prepare($check_email);
    $stmt->bind_param("s", $email);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows > 0) {
        $error_message = 'Email sudah terdaftar';
    } else {
        // Generate NBA otomatis
        $nba = generateNBA($angkatan);
        $password = generateRandomPassword();

        // Insert data anggota baru
        $query = "INSERT INTO anggota (email, nama_lengkap,
kelas, alamat, nomor_telepon, angkatan, nba, password, status,
created_at) VALUES (?, ?, ?, ?, ?, ?, ?, ?, 'pending', NOW())";
        $stmt = $conn->prepare($query);
        $hashed_password = md5($password);
        $stmt->bind_param("ssssssss", $email, $nama_lengkap,
$kelas, $alamat, $nomor_telepon, $angkatan, $nba,
$hashed_password);

        if ($stmt->execute()) {
            // Kirim email konfirmasi
            sendRegistrationEmail($email, $nama_lengkap, $nba,
$password);
            $success_message = 'Registrasi berhasil! Silakan cek
email untuk mendapatkan NBA dan password.';
        } else {
            $error_message = 'Terjadi kesalahan saat
registrasi';
        }
    }
}
```

```
?>

<div class="register-container">
    <div class="register-header">
        <div class="logo">
            <img src="assets/images/logo-graduation.png"
alt="Logo">
        </div>
        <h1>Daftar Akun KITS</h1>
        <p>Bergabunglah dengan Komunitas IT SMAN 1 Parepare</p>
    </div>

    <div class="register-form-card">
        <form method="POST" class="register-form">
            <?php if ($error_message): ?>
                <div class="error-message"><?php echo
$error_message; ?></div>
            <?php endif; ?>

            <?php if ($success_message): ?>
                <div class="success-message"><?php echo
$success_message; ?></div>
            <?php endif; ?>

            <div class="form-group">
                <label>Email Address</label>
                <input type="email" name="email"
placeholder="Masukkan email Anda" required>
            </div>

            <div class="form-group">
                <label>Nama Lengkap</label>
                <input type="text" name="nama_lengkap"
placeholder="Masukkan nama lengkap Anda" required>
            </div>

            <div class="form-group">
                <label>Kelas</label>
                <input type="text" name="kelas"
placeholder="Contoh: Kelas X Cerdas 1" required>
            </div>

            <div class="form-group">
                <label>Alamat</label>
                <textarea name="alamat" placeholder="Masukkan
alamat lengkap Anda" required></textarea>
            </div>

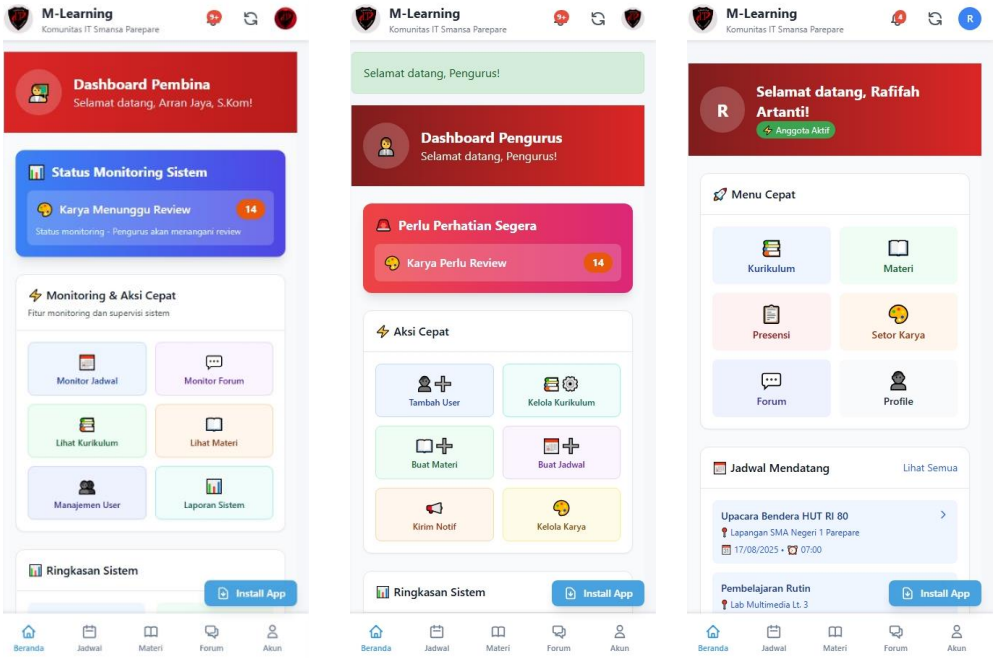
            <div class="form-group">
                <label>Nomor Telepon</label>
                <input type="tel" name="nomor_telepon"
placeholder="Contoh: 08123456789" required>
            </div>

            <div class="form-group">
                <label>Angkatan</label>
```

```
                <input type="text" name="angkatan"
placeholder="Contoh: 2024" required>
            </div>

            <button type="submit" class="register-btn">Daftar
Sekarang</button>
        </form>
    </div>
</div>
```

4. Halaman *Dashboard* Pembina, Pengurus dan Anggota



Gambar 4. 36 Tampilan *Dashboard* Pembina, Pengurus dan Anggota

Menampilkan tampilan dashboard aplikasi *M-Learning* untuk tiga peran pengguna yang berbeda, yaitu Pembina, Pengurus, dan Anggota. Setiap peran memiliki fitur yang disesuaikan dengan kebutuhan dan tanggung jawabnya: pembina difokuskan pada fungsi monitoring sistem, manajemen user, kurikulum, jadwal, hingga laporan; pengurus difasilitasi dengan aksi cepat seperti menambah user, membuat materi dan jadwal, serta mengelola karya; sedangkan anggota diberikan menu sederhana untuk mengakses kurikulum, materi, presensi, forum, dan penyetoran karya, dilengkapi informasi jadwal

kegiatan mendatang, penerapan prinsip *role-based access control*, sehingga akses dan fungsionalitas aplikasi dapat berjalan efektif, terstruktur, serta sesuai hierarki dalam komunitas pembelajaran IT SMAN 1 Parepare..

```
<?php
include 'config/database.php';
include 'includes/functions.php';

// Get user data berdasarkan role
$user_id = $_SESSION['user_id'];
$user_type = $_SESSION['user_type'];

if ($user_type == 'pembina') {
    $query = "SELECT * FROM pembina WHERE id = ?";
    $stmt = $conn->prepare($query);
    $stmt->bind_param("i", $user_id);
    $stmt->execute();
    $user = $stmt->get_result()->fetch_assoc();

    // Get karya menunggu review
    $karya_query = "SELECT COUNT(*) as count FROM karya WHERE
status = 'pending_review'";
    $karya_result = $conn->query($karya_query);
    $karya_count = $karya_result->fetch_assoc()['count'];

} elseif ($user_type == 'pengurus') {
    $query = "SELECT * FROM pengurus WHERE id = ?";
    $stmt = $conn->prepare($query);
    $stmt->bind_param("i", $user_id);
    $stmt->execute();
    $user = $stmt->get_result()->fetch_assoc();

    // Get karya perlu review
    $karya_query = "SELECT COUNT(*) as count FROM karya WHERE
status = 'needs_review'";
    $karya_result = $conn->query($karya_query);
    $karya_count = $karya_result->fetch_assoc()['count'];

} else {
    $query = "SELECT * FROM anggota WHERE id = ?";
    $stmt = $conn->prepare($query);
    $stmt->bind_param("i", $user_id);
    $stmt->execute();
    $user = $stmt->get_result()->fetch_assoc();

    // Get jadwal mendatang
    $jadwal_query = "SELECT * FROM jadwal WHERE tanggal >=
CURDATE() ORDER BY tanggal ASC LIMIT 2";
    $jadwal_result = $conn->query($jadwal_query);
}
?>

<!-- Dashboard Pembina -->
```

```
<?php if ($user_type == 'pembina'): ?>
<div class="dashboard-pembina">
    <div class="welcome-card pembina">
        <h1>Dashboard Pembina</h1>
        <p>Selamat datang, <?php echo $user['nama_lengkap'];
?>!</p>
    </div>

    <div class="status-monitoring">
        <h2>Status Monitoring Sistem</h2>
        <div class="status-item">
            <span class="status-label">Karya Menunggu
Review</span>
            <span class="status-badge"><?php echo
$karya_count; ?></span>
        </div>
    </div>

    <div class="quick-actions">
        <h2>Monitoring & Aksi Cepat</h2>
        <div class="action-grid">
            <a href="monitor/jadwal" class="action-btn">
                <div class="action-icon">📅</div>
                <span>Monitor Jadwal</span>
            </a>
            <a href="monitor/forum" class="action-btn">
                <div class="action-icon">💬</div>
                <span>Monitor Forum</span>
            </a>
            <a href="kurikulum/view" class="action-btn">
                <div class="action-icon">📚</div>
                <span>Lihat Kurikulum</span>
            </a>
            <a href="materi/view" class="action-btn">
                <div class="action-icon">📖</div>
                <span>Lihat Materi</span>
            </a>
            <a href="user/management" class="action-btn">
                <div class="action-icon">👥</div>
                <span>Manajemen User</span>
            </a>
            <a href="laporan/sistem" class="action-btn">
                <div class="action-icon">📊</div>
                <span>Laporan Sistem</span>
            </a>
        </div>
    </div>
</div>

<!-- Dashboard Pengurus -->
<?php elseif ($user_type == 'pengurus'): ?>
<div class="dashboard-pengurus">
    <div class="welcome-banner">
        <h1>Selamat datang, Pengurus!</h1>
    </div>
```

```
    <div class="welcome-card pengurus">
        <h1>Dashboard Pengurus</h1>
        <p>Selamat datang, Pengurus!</p>
    </div>

    <div class="attention-card">
        <h2>Perlu Perhatian Segera</h2>
        <div class="attention-item">
            <span class="attention-label">Karya Perlu
Review</span>
            <span class="attention-badge"><?php echo
$karya_count; ?></span>
        </div>
    </div>

    <div class="quick-actions">
        <h2>Aksi Cepat</h2>
        <div class="action-grid">
            <a href="user/add" class="action-btn">
                <div class="action-icon">➕</div>
                <span>Tambah User</span>
            </a>
            <a href="kurikulum/manage" class="action-btn">
                <div class="action-icon">📋</div>
                <span>Kelola Kurikulum</span>
            </a>
            <a href="materi/create" class="action-btn">
                <div class="action-icon">📝</div>
                <span>Buat Materi</span>
            </a>
            <a href="jadwal/create" class="action-btn">
                <div class="action-icon">⏰</div>
                <span>Buat Jadwal</span>
            </a>
            <a href="notifikasi/send" class="action-btn">
                <div class="action-icon">🔔</div>
                <span>Kirim Notif</span>
            </a>
            <a href="karya/manage" class="action-btn">
                <div class="action-icon">🎨</div>
                <span>Kelola Karya</span>
            </a>
        </div>
    </div>
</div>

<!-- Dashboard Anggota -->
<?php else: ?>
<div class="dashboard-anggota">
    <div class="welcome-card anggota">
        <div class="user-avatar">
            <?php echo
strtoupper(substr($user['nama_lengkap'], 0, 1)); ?>
        </div>
```

```
        <h1>Selamat datang, <?php echo $user['nama_lengkap'];
?>!</h1>
        <p>Anggota Aktif</p>
    </div>

    <div class="quick-menu">
        <h2>Menu Cepat</h2>
        <div class="action-grid">
            <a href="kurikulum" class="action-btn">
                <div class="action-icon">📚</div>
                <span>Kurikulum</span>
            </a>
            <a href="materi" class="action-btn">
                <div class="action-icon">📖</div>
                <span>Materi</span>
            </a>
            <a href="presensi" class="action-btn">
                <div class="action-icon">✅</div>
                <span>Presensi</span>
            </a>
            <a href="karya/submit" class="action-btn">
                <div class="action-icon">🎨</div>
                <span>Setor Karya</span>
            </a>
            <a href="forum" class="action-btn">
                <div class="action-icon">💬</div>
                <span>Forum</span>
            </a>
            <a href="profile" class="action-btn">
                <div class="action-icon">👤</div>
                <span>Profile</span>
            </a>
        </div>
    </div>

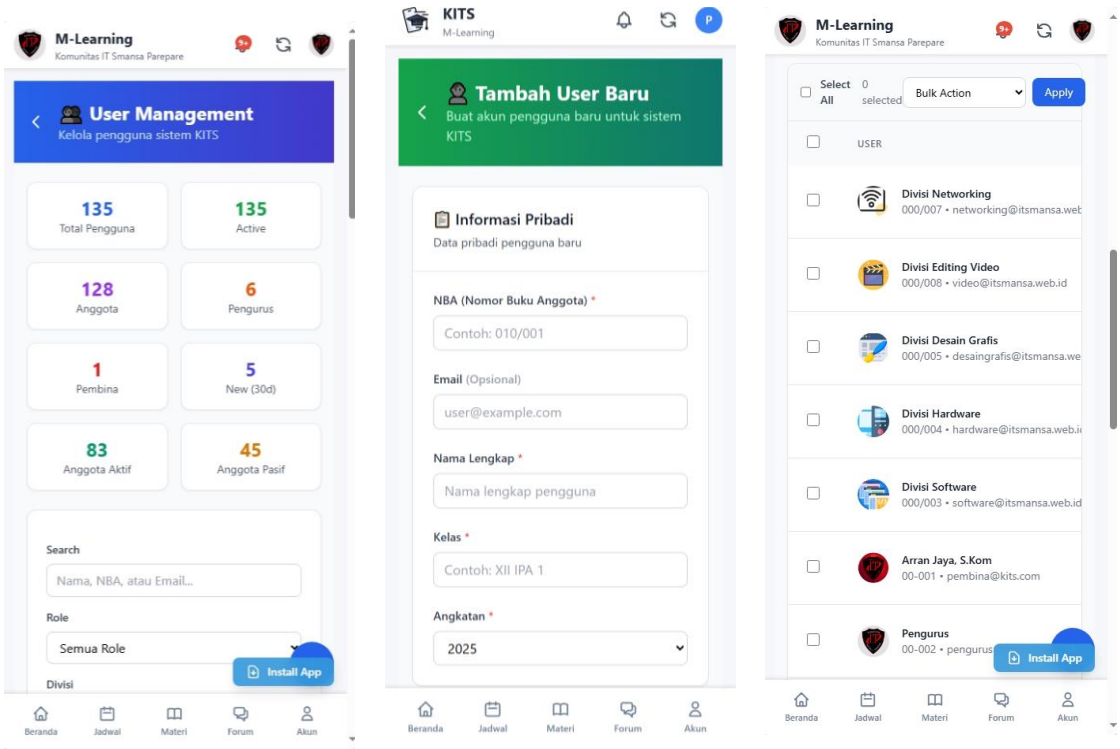
    <div class="upcoming-schedule">
        <div class="schedule-header">
            <h2>Jadwal Mendatang</h2>
            <a href="jadwal" class="view-all">Lihat Semua</a>
        </div>

        <div class="schedule-list">
            <?php while ($jadwal = $jadwal_result-
>fetch_assoc()): ?>
            <div class="schedule-item">
                <div class="schedule-title"><?php echo
$jadwal['judul']; ?></div>
                <div class="schedule-location"><?php echo
$jadwal['lokasi']; ?></div>
                <div class="schedule-date"><?php echo
date('d/m/Y - H:i', strtotime($jadwal['tanggal'])); ?></div>
            </div>
            <?php endwhile; ?>
        </div>
    </div>
```

```
</div>
<?php endif; ?>

<div class="system-summary">
    <h2>Ringkasan Sistem</h2>
    <button class="install-app-btn">Install App</button>
</div>
```

5. Halaman Kelola User



Gambar 4. 37 Tampilan Kelola User

Fitur Kelola User pada aplikasi *M-Learning*, yang berfungsi untuk mengatur seluruh data pengguna dalam sistem KITS. Halaman ini menampilkan ringkasan jumlah total pengguna, pembagian berdasarkan peran (anggota, pengurus, pembina), serta status aktif atau pasif. Selain itu, tersedia form untuk menambahkan pengguna baru dengan mengisi informasi pribadi seperti nomor buku anggota (NBA), email, nama, kelas, dan angkatan. Fitur pencarian dan daftar user yang dapat difilter mempermudah pengelolaan, sedangkan opsi aksi

massal (bulk action) memungkinkan pengurus atau pembina untuk melakukan manajemen data pengguna secara lebih cepat, efektif, dan terorganisir.

```
<?php
include 'config/database.php';
include 'includes/functions.php';

// Handle form tambah user baru
if ($_SERVER['REQUEST_METHOD'] == 'POST' &&
isset($_POST['tambah_user'])) {
    $nba = $_POST['nba'];
    $email = $_POST['email'];
    $nama_lengkap = $_POST['nama_lengkap'];
    $kelas = $_POST['kelas'];
    $angkatan = $_POST['angkatan'];
    $password = generateRandomPassword();

    $query = "INSERT INTO anggota (nba, email, nama_lengkap,
kelas, angkatan, password, status, created_at) VALUES (?, ?,
?, ?, ?, ?, 'active', NOW())";
    $stmt = $conn->prepare($query);
    $hashed_password = md5($password);
    $stmt->bind_param("ssssss", $nba, $email, $nama_lengkap,
$kelas, $angkatan, $hashed_password);
    $stmt->execute();
}

// Get data untuk dashboard
$stats_query = "SELECT
    COUNT(*) as total_users,
    SUM(CASE WHEN status = 'active' THEN 1 ELSE 0 END) as
active_users,
    SUM(CASE WHEN role = 'anggota' THEN 1 ELSE 0 END) as
anggota_count,
    SUM(CASE WHEN role = 'pengurus' THEN 1 ELSE 0 END) as
pengurus_count,
    SUM(CASE WHEN role = 'pembina' THEN 1 ELSE 0 END) as
pembina_count
    FROM anggota";
$result = $conn->query($stats_query);
$stats = $result->fetch_assoc();

// Get user list dengan filter
$search = isset($_GET['search']) ? $_GET['search'] : '';
$where_clause = "";
if (!empty($search)) {
    $where_clause = "WHERE nama_lengkap LIKE '%$search%' OR
nba LIKE '%$search%' OR email LIKE '%$search%'";
}

$users_query = "SELECT * FROM anggota $where_clause ORDER BY
created_at DESC";
$users_result = $conn->query($users_query);
?>
```

```
<!-- Halaman Index User Management -->
<div class="user-management-header">
    <div class="header-content">
        <h1>User Management</h1>
        <p>Kelola pengguna sistem KITS</p>
    </div>
</div>

<div class="stats-grid">
    <div class="stat-card">
        <div class="stat-number"><?php echo
$stats['total_users']; ?></div>
        <div class="stat-label">Total Pengguna</div>
    </div>
    <div class="stat-card">
        <div class="stat-number"><?php echo
$stats['active_users']; ?></div>
        <div class="stat-label">Active</div>
    </div>
    <div class="stat-card">
        <div class="stat-number"><?php echo
$stats['anggota_count']; ?></div>
        <div class="stat-label">Anggota</div>
    </div>
    <div class="stat-card">
        <div class="stat-number"><?php echo
$stats['pengurus_count']; ?></div>
        <div class="stat-label">Pengurus</div>
    </div>
    <div class="stat-card">
        <div class="stat-number"><?php echo
$stats['pembina_count']; ?></div>
        <div class="stat-label">Pembina</div>
    </div>
    <div class="stat-card">
        <div class="stat-number">5</div>
        <div class="stat-label">New (30d)</div>
    </div>
    <div class="stat-card">
        <div class="stat-number">83</div>
        <div class="stat-label">Anggota Aktif</div>
    </div>
    <div class="stat-card">
        <div class="stat-number">45</div>
        <div class="stat-label">Anggota Pasif</div>
    </div>
</div>

<div class="search-filter-section">
    <div class="search-box">
        <input type="text" placeholder="Nama, NBA, atau
Email..." value="<?php echo $search; ?>">
    </div>
    <div class="filter-dropdown">
        <select>
```

```
            <option value="">Semua Role</option>
            <option value="anggota">Anggota</option>
            <option value="pengurus">Pengurus</option>
            <option value="pembina">Pembina</option>
        </select>
    </div>
    <div class="divisi-filter">
        <select>
            <option value="">Semua Divisi</option>
            <option value="hardware">Hardware</option>
            <option value="software">Software</option>
            <option value="networking">Networking</option>
            <option value="desain">Desain Grafis</option>
            <option value="video">Video Editing</option>
        </select>
    </div>
</div>

<!-- Halaman Add User -->
<div class="add-user-header">
    <div class="header-content">
        <h1>Tambah User Baru</h1>
        <p>Buat akun pengguna baru untuk sistem KITS</p>
    </div>
</div>

<div class="add-user-form">
    <div class="form-section">
        <h2>Informasi Pribadi</h2>
        <p>Data pribadi pengguna baru</p>
    </div>

    <form method="POST">
        <div class="form-group">
            <label>NBA (Nomor Baku Anggota) *</label>
            <input type="text" name="nba" placeholder="Contoh:
010/001" required>
        </div>

        <div class="form-group">
            <label>Email (Opsional)</label>
            <input type="email" name="email"
placeholder="user@example.com">
        </div>

        <div class="form-group">
            <label>Nama Lengkap *</label>
            <input type="text" name="nama_lengkap"
placeholder="Nama lengkap pengguna" required>
        </div>

        <div class="form-group">
            <label>Kelas *</label>
            <input type="text" name="kelas"
placeholder="Contoh: XII IPA 1" required>
        </div>
```

```
        <div class="form-group">
            <label>Angkatan *</label>
            <select name="angkatan" required>
                <option value="">Pilih angkatan</option>
                <option value="2023">2023</option>
                <option value="2024">2024</option>
                <option value="2025">2025</option>
            </select>
        </div>

        <button type="submit" name="tambah_user"
class="submit-btn">Simpan User</button>
    </form>
</div>

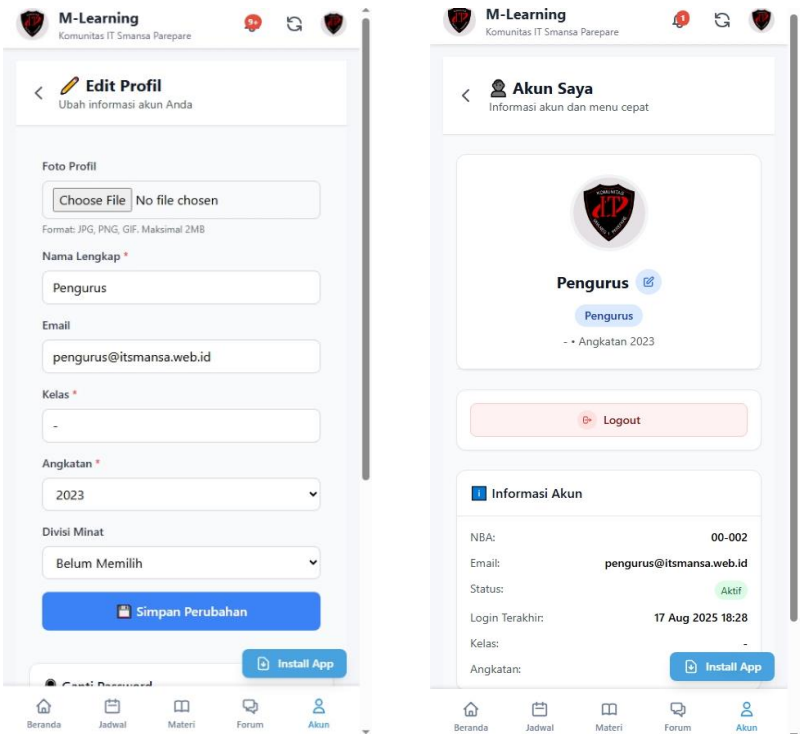
<!-- User List dengan Bulk Actions -->
<div class="user-list-header">
    <div class="bulk-actions">
        <input type="checkbox" id="select-all">
        <label for="select-all">Select All</label>
        <span class="selected-count">0 selected</span>

        <select name="bulk_action">
            <option value="">Bulk Action</option>
            <option value="activate">Aktifkan</option>
            <option value="deactivate">Nonaktifkan</option>
            <option value="delete">Hapus</option>
        </select>

        <button type="button" class="apply-btn">Apply</button>
    </div>
</div>

<div class="user-list">
    <?php while ($user = $users_result->fetch_assoc()): ?>
    <div class="user-item">
        <input type="checkbox" name="selected_users[]"
value="<?php echo $user['id']; ?>">
        <div class="user-avatar">
            <?php echo
strtoupper(substr($user['nama_lengkap'], 0, 1)); ?>
        </div>
        <div class="user-info">
            <div class="user-name"><?php echo
$user['nama_lengkap']; ?></div>
            <div class="user-details"><?php echo $user['nba'];
?> - <?php echo $user['email']; ?></div>
        </div>
    </div>
    <?php endwhile; ?>
</div>
```

6. Halaman Manajemen Profil



Gambar 4. 38 Tampilan Manajemen Profil

Fitur Manajemen Profil pada aplikasi *M-Learning* yang memungkinkan pengguna untuk melihat dan memperbarui informasi akun secara mandiri. Pada halaman edit profil, pengguna dapat mengunggah foto, mengubah nama, kelas, angkatan, serta minat yang sesuai, kemudian menyimpannya melalui tombol Simpan Perubahan. Sementara itu, halaman akun saya menyajikan ringkasan identitas pengguna seperti peran, angkatan, email, serta tanggal pendaftaran, disertai opsi Logout untuk keluar dari sistem. Fitur ini dirancang agar setiap pengguna dapat menjaga akurasi data pribadi sekaligus mendukung keamanan dan personalisasi akun dalam sistem pembelajaran digital.

```
<?php
include 'config/database.php';
include 'includes/functions.php';

$user_id = $_SESSION['user_id'];
$user_type = $_SESSION['user_type'];
```

```
$error_message = '';
$success_message = '';

// Get user data
if ($user_type == 'pengurus') {
    $query = "SELECT * FROM pengurus WHERE id = ?";
} else {
    $query = "SELECT * FROM anggota WHERE id = ?";
}
$stmt = $conn->prepare($query);
$stmt->bind_param("i", $user_id);
$stmt->execute();
$user_data = $stmt->get_result()->fetch_assoc();

// Handle profile update
if ($_SERVER['REQUEST_METHOD'] == 'POST' &&
isset($_POST['simpan_perubahan'])) {
    $nama_lengkap = $_POST['nama_lengkap'];
    $email = $_POST['email'];
    $kelas = $_POST['kelas'];
    $angkatan = $_POST['angkatan'];
    $divisi_minat = $_POST['divisi_minat'];

    // Handle file upload
    $foto_profil_path = $user_data['foto_profil'];
    if (isset($_FILES['foto_profil']) &&
$_FILES['foto_profil']['error'] == 0) {
        $target_dir = "uploads/profile/";
        $file_extension =
pathinfo($_FILES['foto_profil']['name'], PATHINFO_EXTENSION);
        $new_file_name = uniqid('profile_') . '.' .
$file_extension;
        $target_file = $target_dir . $new_file_name;

        if
(move_uploaded_file($_FILES['foto_profil']['tmp_name'],
$target_file)) {
            $foto_profil_path = $target_file;
        }
    }

    // Update database
    if ($user_type == 'pengurus') {
        $update_query = "UPDATE pengurus SET nama_lengkap =
?, email = ?, kelas = ?, angkatan = ?, divisi_minat = ?,
foto_profil = ? WHERE id = ?";
    } else {
        $update_query = "UPDATE anggota SET nama_lengkap = ?,
email = ?, kelas = ?, angkatan = ?, divisi_minat = ?,
foto_profil = ? WHERE id = ?";
    }

    $stmt_update = $conn->prepare($update_query);
    $stmt_update->bind_param("ssssssi", $nama_lengkap,
$email, $kelas, $angkatan, $divisi_minat, $foto_profil_path,
$user_id);
```

```
    if ($stmt_update->execute()) {
        $success_message = "Profil berhasil diperbarui";
        // Refresh user data
        $stmt = $conn->prepare($query);
        $stmt->bind_param("i", $user_id);
        $stmt->execute();
        $user_data = $stmt->get_result()->fetch_assoc();
    } else {
        $error_message = "Gagal memperbarui profil";
    }
}

$view_mode = isset($_GET['mode']) ? $_GET['mode'] : 'view';
?>

<!-- Header Section -->
<div class="header">
    <div class="top-bar">
        <div class="logo">M-Learning</div>
        <div class="community">Komunitas IT Smansa
Parepare</div>
        <div class="header-icons">
            <span class="icon">□□</span>
            <span class="icon">□□</span>
            <span class="icon">👤</span>
        </div>
    </div>

    <div class="nav-bar">
        <a href="dashboard" class="back-arrow">←</a>
        <?php if ($view_mode == 'edit'): ?>
            <div class="page-title">
                <span class="icon">✏️</span> Edit Profil
            </div>
            <div class="subtitle">Ubah informasi akun
Anda</div>
        <?php else: ?>
            <div class="page-title">
                <span class="icon">👤</span> Akun Saya
            </div>
            <div class="subtitle">Informasi akun dan menu
cepat</div>
        <?php endif; ?>
    </div>
</div>

<!-- Main Content -->
<div class="main-content">
    <?php if ($error_message): ?>
        <div class="alert error"><?php echo $error_message;
?></div>
    <?php endif; ?>

    <?php if ($success_message): ?>
```

```
        <div class="alert success"><?php echo
$success_message; ?></div>
    <?php endif; ?>

    <?php if ($view_mode == 'edit'): ?>
        <!-- Edit Profil Form -->
        <form method="POST" enctype="multipart/form-data"
class="edit-form">
            <div class="form-group">
                <label>Foto Profil</label>
                <input type="file" name="foto_profil"
accept="image/*">
                <small>Format: JPG, PNG, GIF. Maksimal
2MB</small>
            </div>

            <div class="form-group">
                <label>Nama Lengkap *</label>
                <input type="text" name="nama_lengkap"
value="<?php echo $user_data['nama_lengkap']; ?>" required>
            </div>

            <div class="form-group">
                <label>Email</label>
                <input type="email" name="email" value="<?php
echo $user_data['email']; ?>" required>
            </div>

            <div class="form-group">
                <label>Kelas *</label>
                <input type="text" name="kelas" value="<?php
echo $user_data['kelas']; ?>" required>
            </div>

            <div class="form-group">
                <label>Angkatan *</label>
                <select name="angkatan" required>
                    <?php for ($year = date('Y'); $year >=
2010; $year--): ?>
                        <option value="<?php echo $year; ?>"
<?php echo ($user_data['angkatan'] == $year) ? 'selected' :
''; ?>><?php echo $year; ?></option>
                    <?php endfor; ?>
                </select>
            </div>

            <div class="form-group">
                <label>Divisi Minat</label>
                <select name="divisi_minat">
                    <option value="">Belum Memilih</option>
                    <option value="Web Development" <?php
echo ($user_data['divisi_minat'] == 'Web Development') ?
'selected' : ''; ?>>Web Development</option>
                    <option value="Mobile Development" <?php
echo ($user_data['divisi_minat'] == 'Mobile Development') ?
'selected' : ''; ?>>Mobile Development</option>
```

```
                    <option value="UI/UX Design" <?php echo
($user_data['divisi_minat'] == 'UI/UX Design') ? 'selected' :
''; ?>>UI/UX Design</option>
                </select>
            </div>

            <button type="submit" name="simpan_perubahan"
class="save-btn">
                <span class="icon">💾</span> Simpan Perubahan
            </button>
        </form>

    <?php else: ?>
        <!-- Akun Saya View -->
        <div class="profile-card">
            <div class="profile-picture">
                <img src="<?php echo
$user_data['foto_profil'] ?:
'assets/images/default_profile.png'; ?>" alt="Profile
Picture">
            </div>
            <div class="profile-info">
                <div class="profile-name">
                    <?php echo $user_data['nama_lengkap']; ?>
                    <a href="?mode=edit" class="edit-
icon">✏️</a>
                </div>
                <div class="profile-role"><?php echo
ucfirst($user_type); ?></div>
                <div class="profile-batch">Angkatan <?php
echo $user_data['angkatan']; ?></div>
            </div>
        </div>

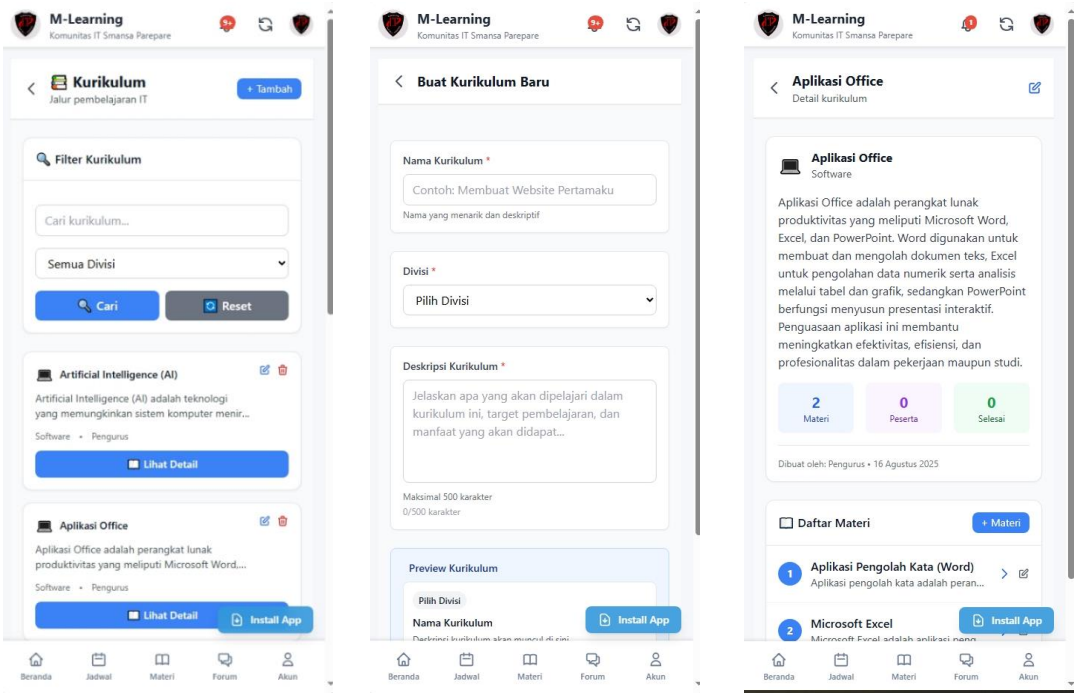
        <button class="logout-btn" onclick="logout()">
            <span class="icon">➡️</span> Logout
        </button>

        <div class="account-info">
            <h3>
                <span class="icon">ℹ️</span> Informasi Akun
            </h3>

            <div class="info-grid">
                <div class="info-item">
                    <span class="label">NBA:</span>
                    <span class="value"><?php echo
$user_data['nba']; ?></span>
                </div>
                <div class="info-item">
                    <span class="label">Email:</span>
                    <span class="value"><?php echo
$user_data['email']; ?></span>
                </div>
                <div class="info-item">
```

```
                    <span class="label">Status:</span>
                    <span class="value status-
active">Aktif</span>
                </div>
                <div class="info-item">
                    <span class="label">Login
Terakhir:</span>
                    <span class="value"><?php echo
$user_data['last_login']; ?></span>
                </div>
                <div class="info-item">
                    <span class="label">Kelas:</span>
                    <span class="value"><?php echo
$user_data['kelas']; ?></span>
                </div>
                <div class="info-item">
                    <span class="label">Angkatan:</span>
                    <span class="value"><?php echo
$user_data['angkatan']; ?></span>
                </div>
            </div>
        </div>
    <?php endif; ?>
</div>
```

7. Halaman Kurikulum



Gambar 4. 39 Tampilan Halaman Kurikulum

Fitur Halaman Kurikulum pada aplikasi *M-Learning* yang berfungsi sebagai pusat pengelolaan materi pembelajaran. Pengguna dapat memfilter

kurikulum berdasarkan divisi, membuat kurikulum baru dengan mengisi nama, divisi, serta deskripsi, dan melihat detail kurikulum yang berisi uraian, daftar materi, jumlah posting, serta status penyelesaian. Tampilan ini mempermudah pengurus maupun pembina dalam menyusun struktur pembelajaran yang sistematis, sekaligus memberikan akses yang jelas bagi anggota untuk mempelajari materi sesuai bidangnya. Dengan adanya fitur ini, penyampaian kurikulum menjadi lebih terorganisir, transparan, dan mendukung keberlangsungan proses belajar digital.

```
<?php
include 'config/database.php';
include 'includes/functions.php';

// Handle form tambah kurikulum
if ($_SERVER['REQUEST_METHOD'] == 'POST' &&
isset($_POST['tambah_kurikulum'])) {
    $nama_kurikulum = $_POST['nama_kurikulum'];
    $divisi = $_POST['divisi'];
    $deskripsi = $_POST['deskripsi'];

    $query = "INSERT INTO kurikulum (nama_kurikulum, divisi,
deskripsi, created_by, created_at) VALUES (?, ?, ?, ?, NOW())";
    $stmt = $conn->prepare($query);
    $stmt->bind_param("sssi", $nama_kurikulum, $divisi,
$deskripsi, $_SESSION['user_id']);
    $stmt->execute();
}

// Handle delete kurikulum
if (isset($_GET['delete']) && is_numeric($_GET['delete'])) {
    $delete_id = $_GET['delete'];
    $query = "DELETE FROM kurikulum WHERE id = ?";
    $stmt = $conn->prepare($query);
    $stmt->bind_param("i", $delete_id);
    $stmt->execute();
}

// Get kurikulum data dengan filter
$search = isset($_GET['search']) ? $_GET['search'] : '';
$divisi_filter = isset($_GET['divisi']) ? $_GET['divisi'] : '';

$where_conditions = [];
$params = [];

if (!empty($search)) {
    $where_conditions[] = "nama_kurikulum LIKE ?";
```

```
    $params[] = "%$search%";
}

if (!empty($divisi_filter)) {
    $where_conditions[] = "divisi = ?";
    $params[] = $divisi_filter;
}

$where_clause = !empty($where_conditions) ? "WHERE " . implode("
AND ", $where_conditions) : "";

$kurikulum_query = "SELECT k.*, u.nama_lengkap as
created_by_name FROM kurikulum k LEFT JOIN pengurus u ON
k.created_by = u.id $where_clause ORDER BY k.created_at DESC";
$kurikulum_result = $conn->query($kurikulum_query);

// Get detail kurikulum
$kurikulum_id = isset($_GET['id']) ? $_GET['id'] : 0;
$kurikulum_detail = null;
$materi_list = [];

if ($kurikulum_id) {
    $detail_query = "SELECT k.*, u.nama_lengkap as
created_by_name FROM kurikulum k LEFT JOIN pengurus u ON
k.created_by = u.id WHERE k.id = ?";
    $stmt = $conn->prepare($detail_query);
    $stmt->bind_param("i", $kurikulum_id);
    $stmt->execute();
    $kurikulum_detail = $stmt->get_result()->fetch_assoc();

    // Get materi dalam kurikulum
    $materi_query = "SELECT * FROM materi WHERE kurikulum_id = ?
ORDER BY urutan ASC";
    $stmt = $conn->prepare($materi_query);
    $stmt->bind_param("i", $kurikulum_id);
    $stmt->execute();
    $materi_list = $stmt->get_result()->fetch_all(MYSQLI_ASSOC);
}
?>

<!-- Halaman List Kurikulum -->
<?php if (!$kurikulum_id): ?>
<div>
    <h1>Kurikulum</h1>
    <p>Jalur pembelajaran IT</p>
    <button>+ Tambah</button>
</div>

<div>
    <h3>Filter Kurikulum</h3>
    <form method="GET">
        <input type="text" name="search" placeholder="Cari
kurikulum..." value="<?php echo $search; ?>">
        <select name="divisi">
            <option value="">Semua Divisi</option>
```

```
            <option value="Software" <?php echo ($divisi_filter
== 'Software') ? 'selected' : ''; ?>>Software</option>
            <option value="Hardware" <?php echo ($divisi_filter
== 'Hardware') ? 'selected' : ''; ?>>Hardware</option>
            <option value="Networking" <?php echo
($divisi_filter == 'Networking') ? 'selected' : '';
?>>Networking</option>
            <option value="Desain Grafis" <?php echo
($divisi_filter == 'Desain Grafis') ? 'selected' : ''; ?>>Desain
Grafis</option>
            <option value="Video Editing" <?php echo
($divisi_filter == 'Video Editing') ? 'selected' : ''; ?>>Video
Editing</option>
        </select>
        <button type="submit">🔍 Cari</button>
        <button type="button" onclick="resetFilter()">🔄
Reset</button>
    </form>
</div>

<div>
    <?php while ($kurikulum = $kurikulum_result->fetch_assoc()):
?>
    <div>
        <div>
            <a href="?delete=<?php echo $kurikulum['id'];
?>">🗑️</a>
            <a href="?edit=<?php echo $kurikulum['id'];
?>">✏️</a>
        </div>

        <div>💻</div>
        <h3><?php echo $kurikulum['nama_kurikulum']; ?></h3>
        <p><?php echo substr($kurikulum['deskripsi'], 0, 100) .
'...'; ?></p>

        <div>
            <span><?php echo $kurikulum['divisi']; ?></span>
            <span><?php echo $kurikulum['created_by_name'];
?></span>
        </div>

        <a href="?id=<?php echo $kurikulum['id']; ?>">Lihat
Detail</a>
    </div>
    <?php endwhile; ?>
</div>

<!-- Halaman Tambah Kurikulum -->
<?php elseif (isset($_GET['tambah'])): ?>
<div>
    <h1>Buat Kurikulum Baru</h1>
</div>

<form method="POST">
```

```
    <div>
        <label>Nama Kurikulum *</label>
        <input type="text" name="nama_kurikulum"
placeholder="Contoh: Membuat Website Pertamaku" required>
        <small>Nama yang menarik dan deskriptif</small>
    </div>

    <div>
        <label>Divisi *</label>
        <select name="divisi" required>
            <option value="">Pilih Divisi</option>
            <option value="Software">Software</option>
            <option value="Hardware">Hardware</option>
            <option value="Networking">Networking</option>
            <option value="Desain Grafis">Desain Grafis</option>
            <option value="Video Editing">Video Editing</option>
        </select>
    </div>

    <div>
        <label>Deskripsi Kurikulum *</label>
        <textarea name="deskripsi" placeholder="Jelaskan apa
yang akan dipelajari dalam kurikulum ini..." maxlength="500"
required></textarea>
        <small>Maksimal 500 karakter <span id="char-
count">0</span>/500 karakter</small>
    </div>

    <div>
        <h3>Preview Kurikulum</h3>
        <div>
            <div>Pilih Divisi</div>
            <div>Nama Kurikulum</div>
        </div>
    </div>

    <button type="submit" name="tambah_kurikulum">Simpan
Kurikulum</button>
</form>

<!-- Halaman Detail Kurikulum -->
<?php else: ?>
<div>
    <h1><?php echo $kurikulum_detail['nama_kurikulum']; ?></h1>
    <p>Detail kurikulum</p>
    <a href="?edit=<?php echo $kurikulum_detail['id'];
?>">✏️</a>
</div>

<div>
    <div>💻</div>
    <h2><?php echo $kurikulum_detail['nama_kurikulum']; ?></h2>
    <span><?php echo $kurikulum_detail['divisi']; ?></span>

    <p><?php echo $kurikulum_detail['deskripsi']; ?></p>
```

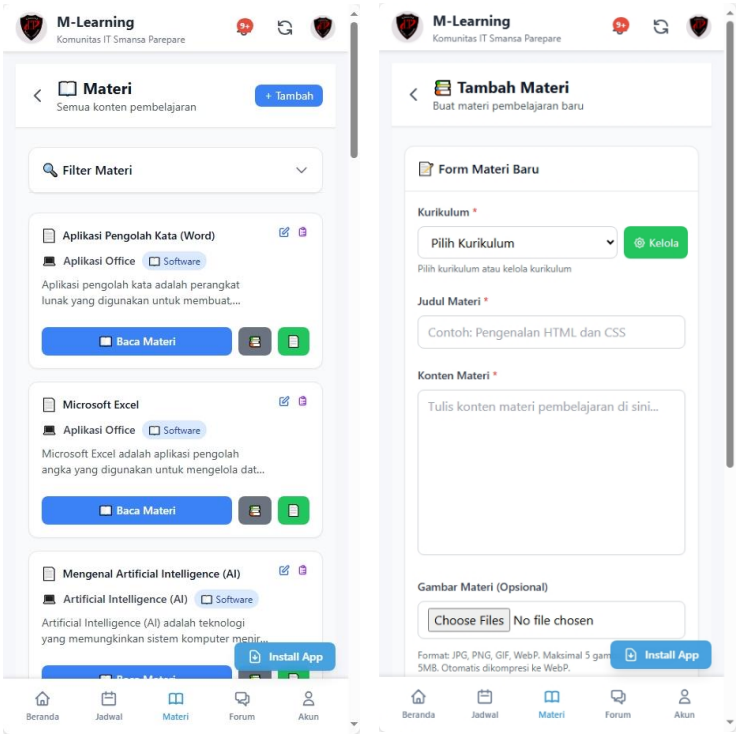
```
    <div>
        <div>
            <span><?php echo count($materi_list); ?></span>
            <span>Materi</span>
        </div>
        <div>
            <span>0</span>
            <span>Peserta</span>
        </div>
        <div>
            <span>0</span>
            <span>Selesai</span>
        </div>
    </div>

    <div>
        <span>Dibuat oleh <?php echo
$kurikulum_detail['created_by_name']; ?> • <?php echo date('d F
Y', strtotime($kurikulum_detail['created_at'])); ?></span>
    </div>
</div>

<div>
    <div>
        <h3>Daftar Materi</h3>
        <button>+ Materi</button>
    </div>

    <div>
        <?php foreach ($materi_list as $index => $materi): ?>
        <div>
            <div><?php echo $index + 1; ?></div>
            <div>
                <h4><?php echo $materi['judul']; ?></h4>
                <p><?php echo substr($materi['deskripsi'], 0,
50) . '...'; ?></p>
            </div>
            <div>
                <a href="materi/detail/<?php echo $materi['id'];
?>">→</a>
                <a href="materi/edit/<?php echo $materi['id'];
?>">✏</a>
            </div>
        </div>
        <?php endforeach; ?>
    </div>
</div>
<?php endif; ?>
```

8. Halaman Materi



Gambar 4. 40 Tampilan Halaman Materi

Halaman Materi pada aplikasi *M-Learning* yang berfungsi sebagai wadah penyimpanan dan pengelolaan konten pembelajaran. Pada halaman utama, materi ditampilkan berdasarkan kategori kurikulum dengan fitur pencarian dan filter untuk mempermudah akses. Setiap materi dilengkapi dengan deskripsi singkat serta opsi untuk membaca secara detail. Selain itu, tersedia menu Tambah Materi yang memungkinkan pengurus atau pembina membuat materi baru dengan mengisi judul, konten, memilih kurikulum terkait, serta menambahkan file pendukung. Fitur ini memudahkan proses distribusi pengetahuan secara terstruktur, sehingga anggota dapat memperoleh materi pembelajaran dengan lebih praktis dan terorganisir.

```
<?php
include 'config/database.php';
include 'includes/functions.php';
```

```
// Handle form tambah materi
if ($_SERVER['REQUEST_METHOD'] == 'POST' &&
isset($_POST['tambah_materi'])) {
    $kurikulum_id = $_POST['kurikulum_id'];
    $judul = $_POST['judul'];
    $konten = $_POST['konten'];

    // Handle file upload
    $gambar_paths = [];
    if (isset($_FILES['gambar_materi']) &&
!empty($_FILES['gambar_materi']['name'][0])) {
        $target_dir = "uploads/materi/";
        foreach ($_FILES['gambar_materi']['name'] as $key =>
$filename) {
            if ($_FILES['gambar_materi']['error'][$key] == 0) {
                $file_extension = pathinfo($filename,
PATHINFO_EXTENSION);
                $new_file_name = uniqid('materi_') . '.' .
$file_extension;
                $target_file = $target_dir . $new_file_name;

                if
(move_uploaded_file($_FILES['gambar_materi']['tmp_name'][$key],
$target_file)) {
                    $gambar_paths[] = $target_file;
                }
            }
        }
    }

    $gambar_json = json_encode($gambar_paths);

    $query = "INSERT INTO materi (kurikulum_id, judul, konten,
gambar, created_by, created_at) VALUES (?, ?, ?, ?, ?, NOW())";
    $stmt = $conn->prepare($query);
    $stmt->bind_param("isssi", $kurikulum_id, $judul, $konten,
$gambar_json, $_SESSION['user_id']);
    $stmt->execute();
}

// Handle delete materi
if (isset($_GET['delete']) && is_numeric($_GET['delete'])) {
    $delete_id = $_GET['delete'];
    $query = "DELETE FROM materi WHERE id = ?";
    $stmt = $conn->prepare($query);
    $stmt->bind_param("i", $delete_id);
    $stmt->execute();
}

// Get materi data dengan filter
$search = isset($_GET['search']) ? $_GET['search'] : '';
$kurikulum_filter = isset($_GET['kurikulum']) ?
$_GET['kurikulum'] : '';

$where_conditions = [];
```

```
$params = [];

if (!empty($search)) {
    $where_conditions[] = "m.judul LIKE ?";
    $params[] = "%$search%";
}

if (!empty($kurikulum_filter)) {
    $where_conditions[] = "m.kurikulum_id = ?";
    $params[] = $kurikulum_filter;
}

$where_clause = !empty($where_conditions) ? "WHERE " . implode("
AND ", $where_conditions) : "";

$materi_query = "SELECT m.*, k.nama_kurikulum, k.divisi FROM
materi m LEFT JOIN kurikulum k ON m.kurikulum_id = k.id
$where_clause ORDER BY m.created_at DESC";
$materi_result = $conn->query($materi_query);

// Get kurikulum list untuk dropdown
$kurikulum_list_query = "SELECT * FROM kurikulum ORDER BY
nama_kurikulum ASC";
$kurikulum_list_result = $conn->query($kurikulum_list_query);
?>

<!-- Halaman List Materi -->
<?php if (!isset($_GET['tambah'])): ?>
<div>
    <h1>Materi</h1>
    <p>Semua konten pembelajaran</p>
    <a href="?tambah=1">+ Tambah</a>
</div>

<div>
    <h3>Filter Materi</h3>
    <form method="GET">
        <input type="text" name="search" placeholder="Cari
materi..." value="<?php echo $search; ?>">
        <select name="kurikulum">
            <option value="">Semua Kurikulum</option>
            <?php while ($kurikulum = $kurikulum_list_result-
>fetch_assoc()): ?>
                <option value="<?php echo $kurikulum['id']; ?>"
<?php echo ($kurikulum_filter == $kurikulum['id']) ? 'selected'
: ''; ?>><?php echo $kurikulum['nama_kurikulum']; ?></option>
            <?php endwhile; ?>
        </select>
        <button type="submit">Cari</button>
    </form>
</div>

<div>
    <?php while ($materi = $materi_result->fetch_assoc()): ?>
    <div>
        <div>
```

```
            <a href="?delete=<?php echo $materi['id'];
?>">□□</a>
            <a href="?edit=<?php echo $materi['id']; ?>">✎</a>
        </div>

        <h3><?php echo $materi['judul']; ?></h3>
        <div>
            <span><?php echo $materi['nama_kurikulum'];
?></span>
            <span><?php echo $materi['divisi']; ?></span>
        </div>
        <p><?php echo substr($materi['konten'], 0, 100) . '...';
?></p>

        <a href="materi/read/<?php echo $materi['id']; ?>">Baca
Materi</a>
    </div>
    <?php endwhile; ?>
</div>

<!-- Halaman Tambah Materi -->
<?php else: ?>
<div>
    <h1>Tambah Materi</h1>
    <p>Buat materi pembelajaran baru</p>
</div>

<form method="POST" enctype="multipart/form-data">
    <div>
        <h3>Form Materi Baru</h3>
    </div>

    <div>
        <label>Kurikulum *</label>
        <select name="kurikulum_id" required>
            <option value="">Pilih Kurikulum</option>
            <?php
            $kurikulum_list_result->data_seek(0);
            while ($kurikulum = $kurikulum_list_result-
>fetch_assoc()):
            ?>
                <option value="<?php echo $kurikulum['id'];
?>"><?php echo $kurikulum['nama_kurikulum']; ?></option>
            <?php endwhile; ?>
        </select>
        <a href="kurikulum">Kelola</a>
        <small>Pilih kurikulum atau kelola kurikulum</small>
    </div>

    <div>
        <label>Judul Materi *</label>
        <input type="text" name="judul" placeholder="Contoh:
Pengenalan HTML dan CSS" required>
    </div>

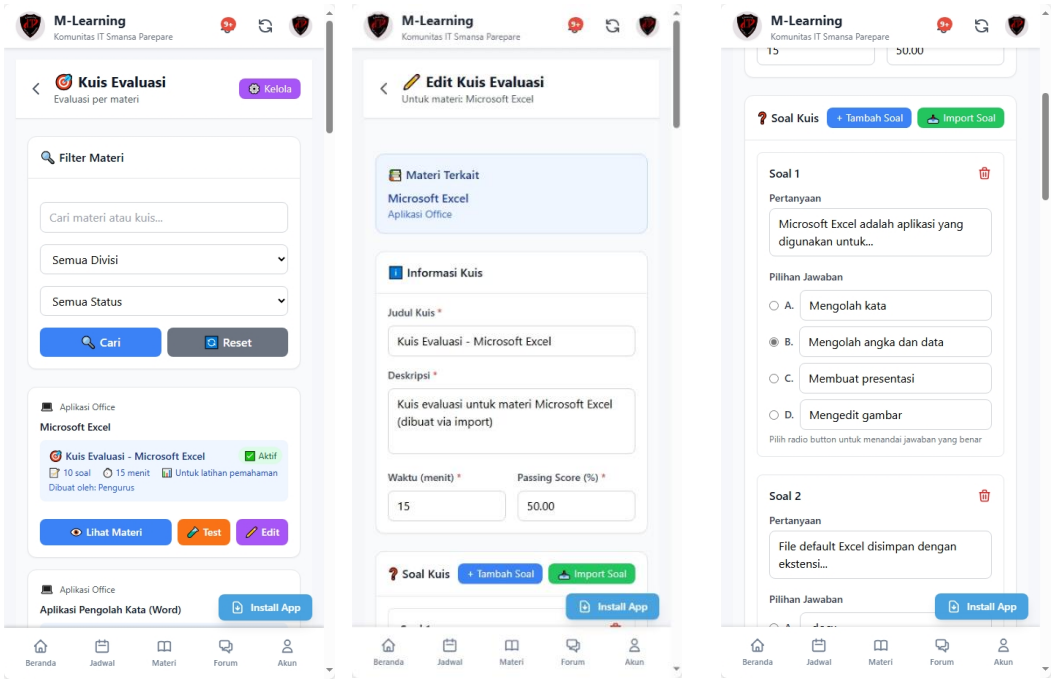
    <div>
```

```
        <label>Konten Materi *</label>
        <textarea name="konten" placeholder="Tulis konten materi
pembelajaran di sini..." required></textarea>
    </div>

    <div>
        <label>Gambar Materi (Opsional)</label>
        <input type="file" name="gambar_materi[]" multiple
accept="image/*">
        <small>Format: JPG, PNG, GIF, WebP. Maksimal 5 gambar,
otomatis dikompresi ke WebP</small>
    </div>

    <button type="submit" name="tambah_materi">Simpan
Materi</button>
</form>
<?php endif; ?>
```

9. Halaman Kuis Evaluasi



Gambar 4. 41 Tampilan Halaman Kuis Evaluasi

Halaman Kuis Evaluasi pada aplikasi *M-Learning* yang berfungsi sebagai sarana penilaian untuk mengukur tingkat pemahaman anggota terhadap materi yang telah dipelajari. Pada tampilan awal, pengguna dapat melakukan pencarian dan filter kuis berdasarkan divisi, status, maupun materi terkait, sehingga mempermudah navigasi dalam menemukan evaluasi yang relevan.

Selanjutnya, halaman detail kuis menampilkan informasi penting seperti judul, materi terkait, alokasi waktu pengerjaan, serta batas nilai kelulusan (passing score) yang harus dicapai oleh peserta. Pada bagian pembuatan soal, pengurus atau pembina dapat menambahkan pertanyaan secara manual maupun mengimpor soal, dengan format pilihan ganda yang dilengkapi opsi jawaban dan indikator kebenaran. Fitur ini mendukung proses evaluasi yang lebih terstruktur, objektif, dan efisien, sekaligus memastikan keberlangsungan pembelajaran digital berjalan sesuai standar penilaian yang ditetapkan..

```
<?php
include 'config/database.php';
include 'includes/functions.php';

// Handle delete kuis
if (isset($_GET['delete']) && is_numeric($_GET['delete'])) {
    $delete_id = $_GET['delete'];
    $query = "DELETE FROM kuis_evaluasi WHERE id = ?";
    $stmt = $conn->prepare($query);
    $stmt->bind_param("i", $delete_id);
    $stmt->execute();
}

// Get kuis data dengan filter
$search = isset($_GET['search']) ? $_GET['search'] : '';
$divisi_filter = isset($_GET['divisi']) ? $_GET['divisi'] : '';
$status_filter = isset($_GET['status']) ? $_GET['status'] : '';

$where_conditions = [];
$params = [];

if (!empty($search)) {
    $where_conditions[] = "(k.judul LIKE ? OR m.judul LIKE ?)";
    $search_param = "%$search%";
    $params[] = $search_param;
    $params[] = $search_param;
}

if (!empty($divisi_filter)) {
    $where_conditions[] = "k.divisi = ?";
    $params[] = $divisi_filter;
}

if (!empty($status_filter)) {
    $where_conditions[] = "k.status = ?";
    $params[] = $status_filter;
}
```

```
$where_clause = !empty($where_conditions) ? "WHERE " . implode("
AND ", $where_conditions) : "";

$kuis_query = "SELECT k.*, m.judul as materi_judul,
u.nama_lengkap as created_by_name FROM kuis_evaluasi k LEFT JOIN
materi m ON k.materi_id = m.id LEFT JOIN pengurus u ON
k.created_by = u.id $where_clause ORDER BY k.created_at DESC";
$kuis_result = $conn->query($kuis_query);

// Group kuis by materi
$kuis_by_materi = [];
while ($kuis = $kuis_result->fetch_assoc()) {
    $materi_key = $kuis['materi_judul'] ?: 'Tanpa Materi';
    if (!isset($kuis_by_materi[$materi_key])) {
        $kuis_by_materi[$materi_key] = [];
    }
    $kuis_by_materi[$materi_key][] = $kuis;
}
?>

<div>
    <h1>Kuis Evaluasi</h1>
    <p>Evaluasi per materi</p>
    <a href="kuis/kelola">Kelola</a>
</div>

<div>
    <h3>Filter Materi</h3>
    <form method="GET">
        <input type="text" name="search" placeholder="Cari
materi atau kuis..." value="<?php echo $search; ?>">
        <select name="divisi">
            <option value="">Semua Divisi</option>
            <option value="Software" <?php echo ($divisi_filter
== 'Software') ? 'selected' : ''; ?>>Software</option>
            <option value="Hardware" <?php echo ($divisi_filter
== 'Hardware') ? 'selected' : ''; ?>>Hardware</option>
            <option value="Networking" <?php echo
($divisi_filter == 'Networking') ? 'selected' : '';
?>>Networking</option>
            <option value="Desain Grafis" <?php echo
($divisi_filter == 'Desain Grafis') ? 'selected' : ''; ?>>Desain
Grafis</option>
            <option value="Video Editing" <?php echo
($divisi_filter == 'Video Editing') ? 'selected' : ''; ?>>Video
Editing</option>
        </select>
        <select name="status">
            <option value="">Semua Status</option>
            <option value="aktif" <?php echo ($status_filter ==
'aktif') ? 'selected' : ''; ?>>Aktif</option>
            <option value="nonaktif" <?php echo ($status_filter
== 'nonaktif') ? 'selected' : ''; ?>>Nonaktif</option>
        </select>
        <button type="submit">🔍 Cari</button>
```

```
        <button type="button" onclick="resetFilter()">□□
Reset</button>
    </form>
</div>

<?php foreach ($kuis_by_materi as $materi_judul => $kuis_list):
?>
<div>
    <h3>Aplikasi Office</h3>
    <p><?php echo $materi_judul; ?></p>

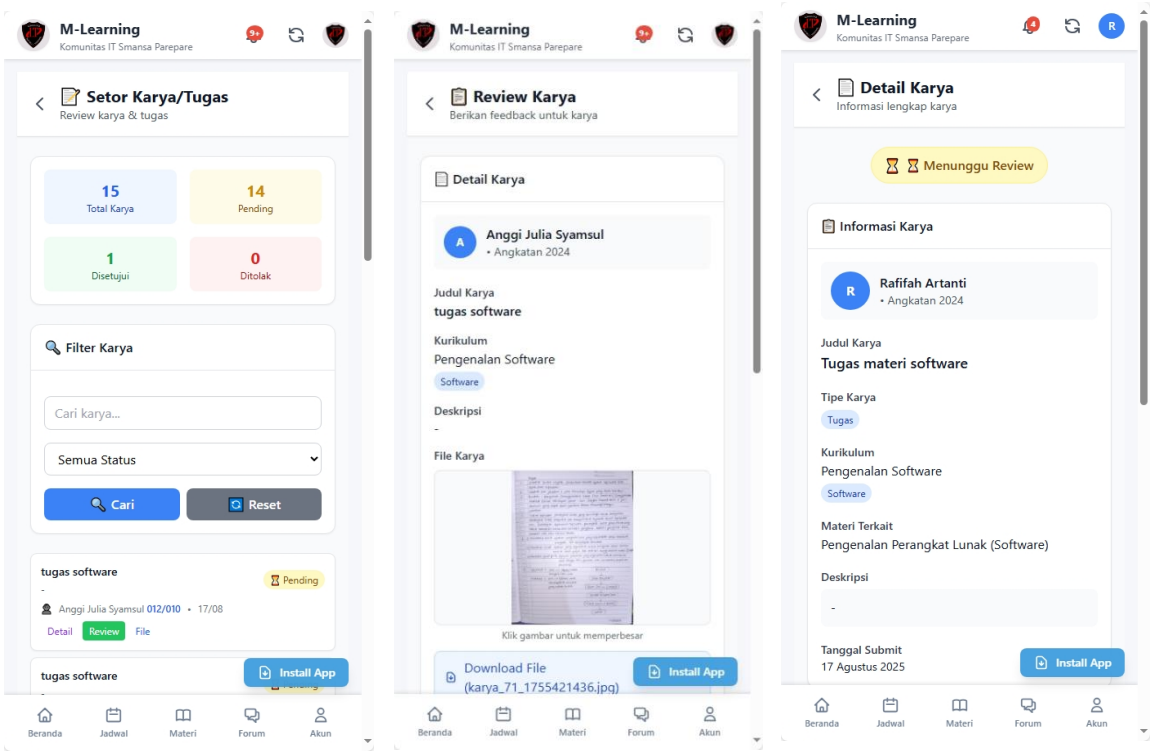
    <?php foreach ($kuis_list as $kuis): ?>
    <div>
        <div>
            <span><?php echo ucfirst($kuis['status']); ?></span>
        </div>

        <h4><?php echo $kuis['judul']; ?></h4>
        <p><?php echo $kuis['jumlah_soal']; ?> soal, <?php echo
$kuis['waktu_menit']; ?> menit</p>
        <p><?php echo $kuis['deskripsi']; ?></p>
        <p>Dibuat oleh <?php echo $kuis['created_by_name'];
?></p>

        <div>
            <a href="materi/detail/<?php echo
$kuis['materi_id']; ?>">Lihat Materi</a>
            <a href="kuis/test/<?php echo $kuis['id'];
?>">Test</a>
            <a href="kuis/edit/<?php echo $kuis['id'];
?>">Edit</a>
        </div>
    </div>
    <?php endforeach; ?>
</div>
<?php endforeach; ?>

<div>
    <button>Install App</button>
</div>
```

10. Halaman Setor Karya



Gambar 4. 42 Tampilan Halaman Setor Karya

Halaman ini berfungsi sebagai media bagi anggota untuk mengunggah tugas atau karya sesuai kurikulum yang diikuti. Pada halaman utama ditampilkan ringkasan jumlah karya yang berstatus diterima, pending, maupun ditolak, serta fitur filter untuk memudahkan pencarian berdasarkan judul atau status. Halaman Review Karya memungkinkan pembina atau pengurus meninjau karya yang diunggah, lengkap dengan identitas pengirim, judul, kurikulum, divisi, serta file karya yang dapat diunduh. Sementara itu, halaman Detail Karya menampilkan informasi lengkap terkait status review, identitas pengunggah, serta deskripsi karya. Fitur ini dirancang untuk memastikan transparansi proses penilaian, mempermudah komunikasi antara anggota dan pembina/pengurus, serta mendukung dokumentasi hasil belajar dalam bentuk karya digital.

```
<?php
include 'config/database.php';
include 'includes/functions.php';

$karya_id = isset($_GET['id']) ? $_GET['id'] : 0;

// Get karya detail
$karya_query = "SELECT k.*, a.nama_lengkap, a.angkatan, m.judul
as materi_judul, kur.nama_kurikulum, kur.divisi FROM karya k
LEFT JOIN anggota a ON k.anggota_id = a.id LEFT JOIN materi m ON
k.materi_id = m.id LEFT JOIN kurikulum kur ON m.kurikulum_id =
kur.id WHERE k.id = ?";
$stmt = $conn->prepare($karya_query);
$stmt->bind_param("i", $karya_id);
$stmt->execute();
$karya = $stmt->get_result()->fetch_assoc();

// Handle review submission
if ($_SERVER['REQUEST_METHOD'] == 'POST' &&
isset($_POST['submit_review'])) {
    $rating = $_POST['rating'];
    $feedback = $_POST['feedback'];
    $status = $_POST['status'];

    $review_query = "INSERT INTO review_karya (karya_id,
reviewer_id, rating, feedback, status, created_at) VALUES (?, ?,
?, ?, ?, NOW())";
    $stmt = $conn->prepare($review_query);
    $stmt->bind_param("iiiss", $karya_id, $_SESSION['user_id'],
$rating, $feedback, $status);
    $stmt->execute();

    // Update status karya
    $update_karya_query = "UPDATE karya SET status = ? WHERE id
= ?";
    $stmt_update = $conn->prepare($update_karya_query);
    $stmt_update->bind_param("si", $status, $karya_id);
    $stmt_update->execute();

    header('Location: dashboard');
    exit();
}
?>

<div>
    <h1>Review Karya</h1>
    <p>Berikan feedback untuk karya</p>
</div>

<div>
    <h3>Detail Karya</h3>

    <div>
        <p><strong><?php echo $karya['nama_lengkap'];
?></strong></p>
        <p>Angkatan <?php echo $karya['angkatan']; ?></p>
```

```
    </div>

    <div>
        <h4><?php echo $karya['judul_karya']; ?></h4>
        <p>Kurikulum: <?php echo $karya['nama_kurikulum'];
?></p>
        <span><?php echo $karya['divisi']; ?></span>
    </div>

    <div>
        <p>Deskripsi: <?php echo $karya['deskripsi'] ?: '-';
?></p>
    </div>
</div>

<div>
    <h3>File Karya</h3>

    <div>
        <img src="<?php echo $karya['file_path']; ?>"
alt="Preview Karya" onclick="openImageModal(this.src)">
        <p>Klik gambar untuk memperbesar</p>
        <a href="<?php echo $karya['file_path']; ?>"
download>Download File (<?php echo
basename($karya['file_path']); ?>)</a>
    </div>
</div>

<form method="POST">
    <div>
        <h3>Review & Feedback</h3>

        <div>
            <label>Rating</label>
            <select name="rating" required>
                <option value="">Pilih Rating</option>
                <option value="1">1 - Sangat Buruk</option>
                <option value="2">2 - Buruk</option>
                <option value="3">3 - Cukup</option>
                <option value="4">4 - Baik</option>
                <option value="5">5 - Sangat Baik</option>
            </select>
        </div>

        <div>
            <label>Feedback</label>
            <textarea name="feedback" placeholder="Berikan
feedback untuk karya ini..." required></textarea>
        </div>

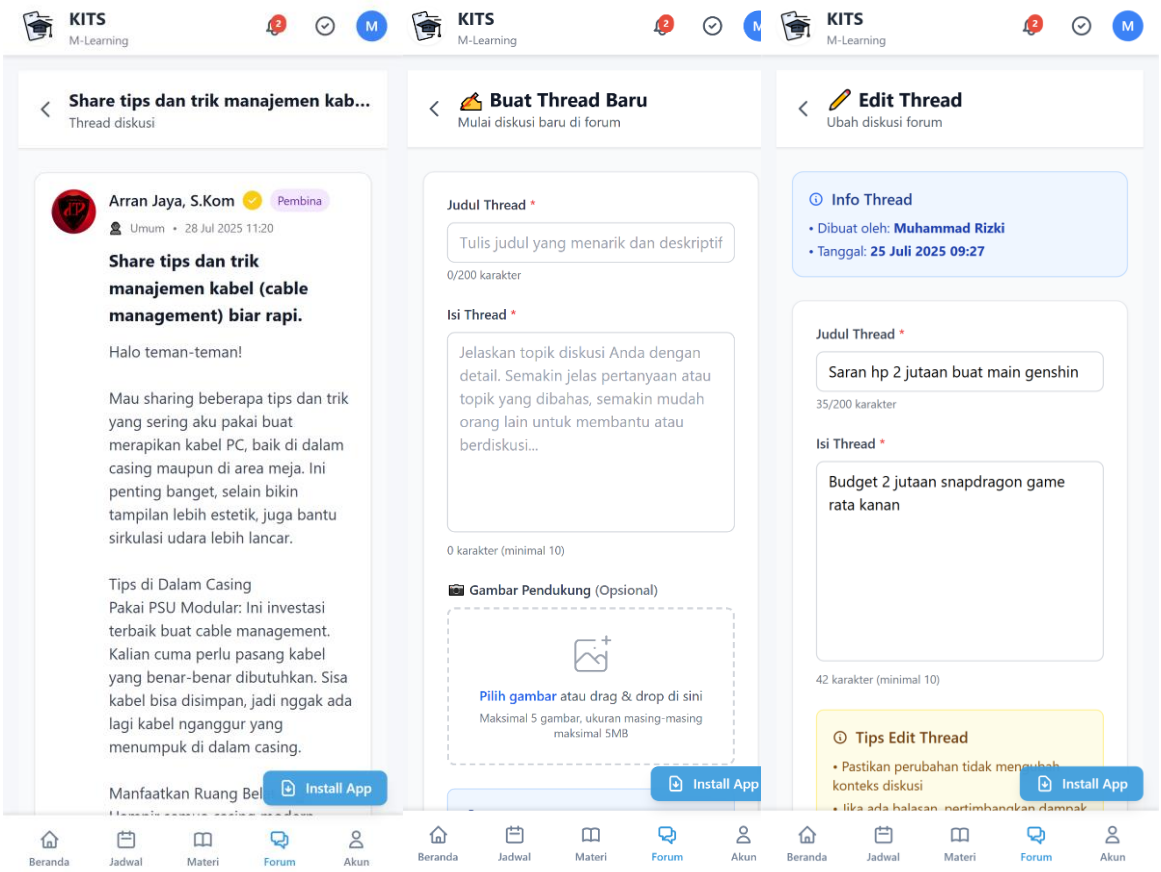
        <div>
            <label>Status</label>
            <select name="status" required>
                <option value="">Pilih Status</option>
                <option value="approved">Disetujui</option>
                <option value="rejected">Ditolak</option>
```

```
                <option value="revision">Perlu Revisi</option>
            </select>
        </div>

        <button type="submit" name="submit_review">Submit
Review</button>
    </div>
</form>
```

11. Halaman Forum Diskusi

Halaman ini adalah "Tampilan Halaman Forum Diskusi" dalam aplikasi *Mobile learning (M-Learning)* KITS, yang menyajikan tiga fungsi utama. Panel paling kiri menampilkan detail sebuah thread diskusi, termasuk judul, penulis,



Gambar 4. 43 Tampilan Halaman Forum Diskusi

tanggal, dan isi diskusi. Panel tengah adalah antarmuka "Buat Thread Baru", memungkinkan pengguna untuk memulai diskusi baru dengan mengisi judul dan

isi thread, serta mengunggah gambar. Panel paling kanan adalah "Edit Thread", berfungsi untuk mengubah informasi atau konten dari thread diskusi yang sudah ada.

```
<?php
include 'config/database.php';
include 'includes/functions.php';

// Handle form submission
if ($_SERVER['REQUEST_METHOD'] == 'POST' &&
isset($_POST['buat_thread'])) {
    $judul = $_POST['judul'];
    $isi = $_POST['isi'];

    // Handle file upload
    $gambar_paths = [];
    if (isset($_FILES['gambar_pendukung']) &&
!empty($_FILES['gambar_pendukung']['name'][0])) {
        $target_dir = "uploads/forum/";
        foreach ($_FILES['gambar_pendukung']['name'] as $key =>
$filename) {
            if ($_FILES['gambar_pendukung']['error'][$key] == 0)
{
                $file_extension = pathinfo($filename,
PATHINFO_EXTENSION);
                $new_file_name = uniqid('forum_') . '.' .
$file_extension;
                $target_file = $target_dir . $new_file_name;

                if
(move_uploaded_file($_FILES['gambar_pendukung']['tmp_name'][$key
], $target_file)) {
                    $gambar_paths[] = $target_file;
                }
            }
        }
    }

    $gambar_json = json_encode($gambar_paths);

    $query = "INSERT INTO forum_thread (judul, isi, gambar,
author_id, created_at) VALUES (?, ?, ?, ?, NOW())";
    $stmt = $conn->prepare($query);
    $stmt->bind_param("sssi", $judul, $isi, $gambar_json,
$_SESSION['user_id']);
    $stmt->execute();

    header('Location: forum');
    exit();
}
?>

<div>
```

```
    <h1>Buat Thread Baru</h1>
    <p>Mulai diskusi baru di forum</p>
</div>

<form method="POST" enctype="multipart/form-data">
    <div>
        <label>Judul Thread *</label>
        <input type="text" name="judul" placeholder="Tulis judul
yang menarik dan deskriptif" maxlength="200" required>
        <small><span id="judul-count">0</span>/200
karakter</small>
    </div>

    <div>
        <label>Isi Thread *</label>
        <textarea name="isi" placeholder="Jelaskan topik diskusi
Anda dengan detail. Semakin jelas pertanyaan atau topik yang
dibahas, semakin mudah orang lain untuk membantu atau
berdiskusi..." minlength="10" required></textarea>
        <small><span id="isi-count">0</span> karakter (minimal
10)</small>
    </div>

    <div>
        <label>Gambar Pendukung (Opsional)</label>
        <div>
            <input type="file" name="gambar_pendukung[]"
multiple accept="image/*" id="gambar-input">
            <div>
                <span>📷</span>
                <p>Pilih gambar atau drag & drop di sini</p>
                <small>Maksimal 5 gambar, ukuran masing-masing
maksimal 5MB</small>
            </div>
        </div>
    </div>

    <button type="submit" name="buat_thread">Buat
Thread</button>
</form>

<script>
document.getElementById('judul').addEventListener('input',
function() {
    document.getElementById('judul-count').textContent =
this.value.length;
});

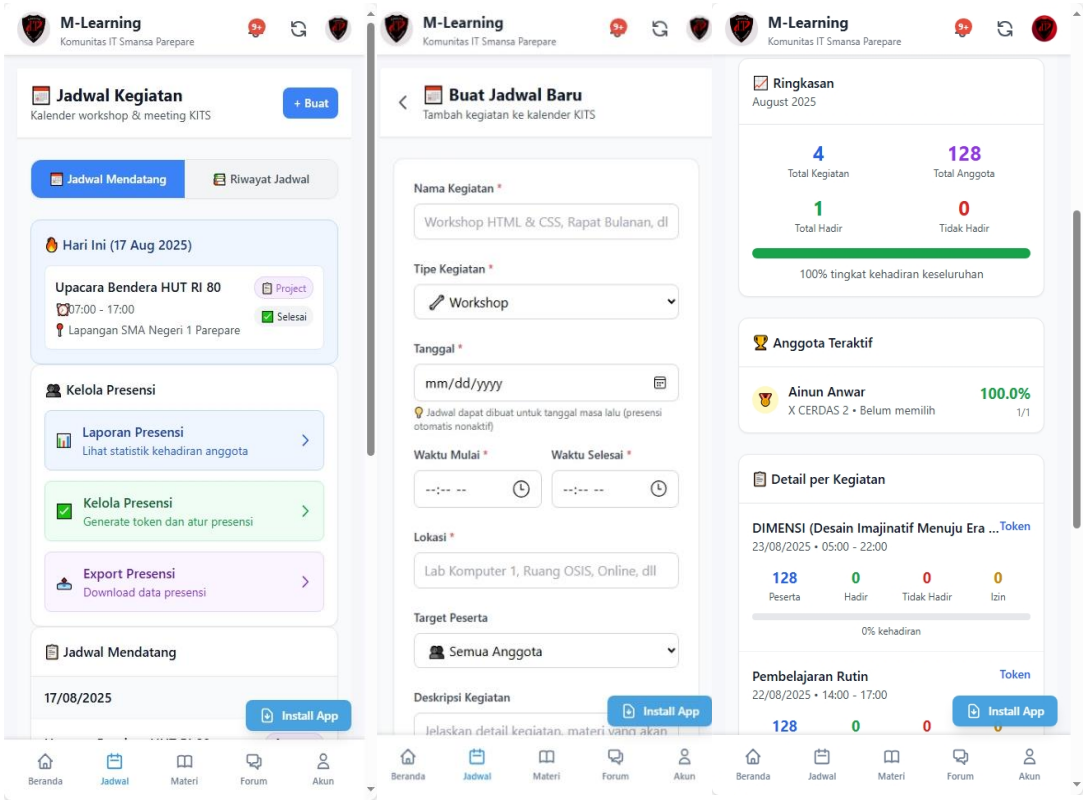
document.getElementById('isi').addEventListener('input',
function() {
    document.getElementById('isi-count').textContent =
this.value.length;
});

document.getElementById('gambar-
input').addEventListener('change', function() {
```

```
    if (this.files.length > 5) {
        alert('Maksimal 5 gambar');
        this.value = '';
    }

    for (let file of this.files) {
        if (file.size > 5 * 1024 * 1024) {
            alert('Ukuran file ' + file.name + ' terlalu besar.
Maksimal 5MB');
            this.value = '';
            break;
        }
    }
});
</script>
```

12. Halaman Jadwal



Gambar 4. 44 Tampilan Halaman Jadwal

Halaman Jadwal pada aplikasi *M-Learning* yang berfungsi sebagai pusat info kegiatan komunitas untuk mengatur dan memantau berbagai agenda pembelajaran maupun acara resmi. Pada tampilan utama, pengguna dapat melihat jadwal kegiatan mendatang, mengelola presensi, hingga mengekspor data kehadiran. Fitur Buat Jadwal Baru memungkinkan pengurus atau pembina menambahkan agenda dengan mengisi informasi nama kegiatan, tipe kegiatan, tanggal, waktu, lokasi, target peserta, serta deskripsi kegiatan. Selain itu, terdapat tampilan ringkasan yang menampilkan jumlah kegiatan, kehadiran anggota, serta detail aktivitas dengan indikator keaktifan peserta.

```
<?php
include 'config/database.php';
include 'includes/functions.php';

// Get jadwal data
$today = date('Y-m-d');
$current_month = date('Y-m');

// Get jadwal hari ini
$jadwal_hari_ini_query = "SELECT * FROM jadwal WHERE
DATE(tanggal) = ? ORDER BY waktu_mulai ASC";
$stmt = $conn->prepare($jadwal_hari_ini_query);
$stmt->bind_param("s", $today);
$stmt->execute();
$jadwal_hari_ini = $stmt->get_result()->fetch_all(MYSQLI_ASSOC);

// Get jadwal mendatang
$jadwal_mendatang_query = "SELECT * FROM jadwal WHERE
DATE(tanggal) > ? ORDER BY tanggal ASC LIMIT 5";
$stmt = $conn->prepare($jadwal_mendatang_query);
$stmt->bind_param("s", $today);
$stmt->execute();
$jadwal_mendatang = $stmt->get_result()-
>fetch_all(MYSQLI_ASSOC);

// Get statistik bulan ini
$stats_query = "SELECT
    COUNT(*) as total_kegiatan,
    (SELECT COUNT(*) FROM anggota WHERE status = 'active') as
total_anggota,
    SUM(CASE WHEN status = 'completed' THEN 1 ELSE 0 END) as
total_hadir,
    SUM(CASE WHEN status = 'pending' THEN 1 ELSE 0 END) as
tidak_hadir
```

```
    FROM jadwal
    WHERE DATE_FORMAT(tanggal, '%Y-%m') = ?";
$stmt = $conn->prepare($stats_query);
$stmt->bind_param("s", $current_month);
$stmt->execute();
$stats = $stmt->get_result()->fetch_assoc();

// Get anggota teraktif
$anggota_teraktif_query = "SELECT a.nama_lengkap, a.kelas,
a.divisi_minat,
    COUNT(p.id) as total_presensi,
    ROUND((COUNT(p.id) / (SELECT COUNT(*) FROM jadwal WHERE
DATE_FORMAT(tanggal, '%Y-%m') = ?)) * 100, 1) as persentase
    FROM anggota a
    LEFT JOIN presensi p ON a.id = p.anggota_id
    LEFT JOIN jadwal j ON p.jadwal_id = j.id AND
DATE_FORMAT(j.tanggal, '%Y-%m') = ?
    WHERE a.status = 'active'
    GROUP BY a.id
    ORDER BY total_presensi DESC
    LIMIT 1";
$stmt = $conn->prepare($anggota_teraktif_query);
$stmt->bind_param("ss", $current_month, $current_month);
$stmt->execute();
$anggota_teraktif = $stmt->get_result()->fetch_assoc();

// Get detail per kegiatan
$detail_kegiatan_query = "SELECT j.*,
    COUNT(p.id) as total_peserta,
    SUM(CASE WHEN p.status = 'hadir' THEN 1 ELSE 0 END) as
hadir,
    SUM(CASE WHEN p.status = 'tidak_hadir' THEN 1 ELSE 0 END) as
tidak_hadir,
    SUM(CASE WHEN p.status = 'izin' THEN 1 ELSE 0 END) as izin
    FROM jadwal j
    LEFT JOIN presensi p ON j.id = p.jadwal_id
    WHERE DATE_FORMAT(j.tanggal, '%Y-%m') = ?
    GROUP BY j.id
    ORDER BY j.tanggal DESC";
$stmt = $conn->prepare($detail_kegiatan_query);
$stmt->bind_param("s", $current_month);
$stmt->execute();
$detail_kegiatan = $stmt->get_result()->fetch_all(MYSQLI_ASSOC);

$active_tab = isset($_GET['tab']) ? $_GET['tab'] : 'mendatang';
?>

<div>
    <h1>Jadwal Kegiatan</h1>
    <p>Kalender workshop & meeting KITS</p>
    <a href="jadwal/buat">+ Buat</a>
</div>

<div>
    <a href="?tab=mendatang" <?php echo ($active_tab ==
'mendatang') ? 'class="active"' : ''; ?>>Jadwal Mendatang</a>
```

```
    <a href="?tab=riwayat" <?php echo ($active_tab == 'riwayat')
? 'class="active"' : ''; ?>>Riwayat Jadwal</a>
</div>

<?php if ($active_tab == 'mendatang'): ?>
<div>
    <h3>Hari Ini (<?php echo date('d M Y'); ?>)</h3>

    <?php foreach ($jadwal_hari_ini as $jadwal): ?>
    <div>
        <h4><?php echo $jadwal['nama_kegiatan']; ?></h4>
        <p><?php echo date('H:i',
strtotime($jadwal['waktu_mulai'])); ?> - <?php echo date('H:i',
strtotime($jadwal['waktu_selesai'])); ?></p>
        <div>
            <span><?php echo $jadwal['tipe_kegiatan']; ?></span>
            <span><?php echo ucfirst($jadwal['status']);
?></span>
        </div>
        <p>□□ <?php echo $jadwal['lokasi']; ?></p>
    </div>
    <?php endforeach; ?>
</div>

<div>
    <h3>Kelola Presensi</h3>

    <div>
        <a href="presensi/laporan">
            <span>□□</span>
            <div>
                <h4>Laporan Presensi</h4>
                <p>Lihat statistik kehadiran anggota</p>
            </div>
            <span>→</span>
        </a>
    </div>

    <div>
        <a href="presensi/kelola">
            <span>✅</span>
            <div>
                <h4>Kelola Presensi</h4>
                <p>Generate token dan atur presensi</p>
            </div>
            <span>→</span>
        </a>
    </div>

    <div>
        <a href="presensi/export">
            <span>□□</span>
            <div>
                <h4>Export Presensi</h4>
                <p>Download data presensi</p>
            </div>
```

```
            <span>→</span>
        </a>
    </div>
</div>

<div>
    <h3>Jadwal Mendatang</h3>
    <?php foreach ($jadwal_mendatang as $jadwal): ?>
    <div>
        <p><?php echo date('d/m/Y',
strtotime($jadwal['tanggal'])); ?></p>
    </div>
    <?php endforeach; ?>
</div>

<?php else: ?>
<div>
    <h3>Riwayat Jadwal</h3>
    <!-- Riwayat jadwal content -->
</div>
<?php endif; ?>

<div>
    <h3>Ringkasan</h3>
    <p><?php echo date('F Y'); ?></p>

    <div>
        <div>
            <span><?php echo $stats['total_kegiatan']; ?></span>
            <span>Total Kegiatan</span>
        </div>
        <div>
            <span><?php echo $stats['total_anggota']; ?></span>
            <span>Total Anggota</span>
        </div>
        <div>
            <span><?php echo $stats['total_hadir']; ?></span>
            <span>Total Hadir</span>
        </div>
        <div>
            <span><?php echo $stats['tidak_hadir']; ?></span>
            <span>Tidak Hadir</span>
        </div>
    </div>

    <div>
        <div>
            <span>100%</span>
            <span>tingkat kehadiran keseluruhan</span>
        </div>
    </div>
</div>

<div>
    <h3>Anggota Teraktif</h3>
    <div>
```

```
        <div>
            <h4><?php echo $anggota_teraktif['nama_lengkap'];
?></h4>
            <p><?php echo $anggota_teraktif['kelas']; ?> - <?php
echo $anggota_teraktif['divisi_minat'] ?: 'Belum memilih';
?></p>
        </div>
        <div>
            <span><?php echo $anggota_teraktif['persentase'];
?>%</span>
            <span><?php echo
$anggota_teraktif['total_presensi']; ?>/<?php echo
$stats['total_kegiatan']; ?></span>
        </div>
    </div>
</div>

<div>
    <h3>Detail per Kegiatan</h3>

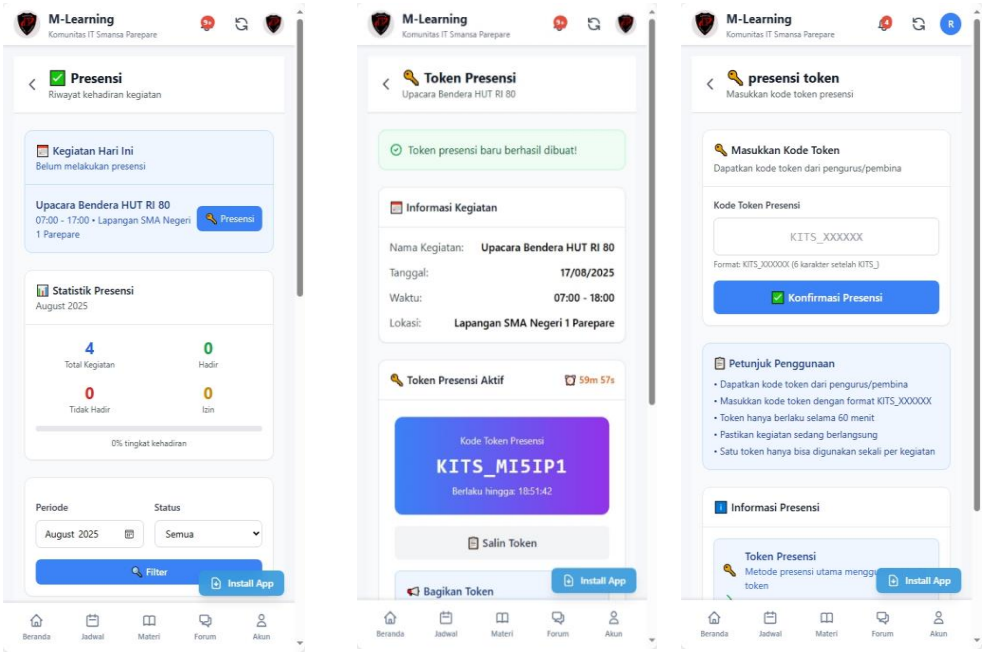
    <?php foreach ($detail_kegiatan as $kegiatan): ?>
    <div>
        <h4><?php echo $kegiatan['nama_kegiatan']; ?></h4>
        <p><?php echo date('d/m/Y',
strtotime($kegiatan['tanggal'])); ?> - <?php echo date('H:i',
strtotime($kegiatan['waktu_mulai'])); ?> - <?php echo
date('H:i', strtotime($kegiatan['waktu_selesai'])); ?></p>

        <div>
            <span><?php echo $kegiatan['total_peserta']; ?>
Peserta</span>
            <span><?php echo $kegiatan['hadir']; ?> Hadir</span>
            <span><?php echo $kegiatan['tidak_hadir']; ?> Tidak
Hadir</span>
            <span><?php echo $kegiatan['izin']; ?> Izin</span>
        </div>

        <p><?php echo round(($kegiatan['hadir'] /
max($kegiatan['total_peserta'], 1)) * 100, 1); ?>% kehadiran</p>

        <div>
            <span>Token</span>
            <button>Install App</button>
        </div>
    </div>
    <?php endforeach; ?>
</div>
```

13. Halaman Presensi



Gambar 4. 45 Tampilan Halaman Presensi

Halaman Presensi pada aplikasi menampilkan fitur utama untuk mengelola kehadiran kegiatan secara digital. Pada tampilan awal, pengguna dapat melihat jadwal kegiatan hari ini, status presensi, serta ringkasan statistik kehadiran bulanan. Sistem juga menyediakan fitur Token Presensi, di mana pembina atau pengurus dapat membuat token khusus yang berlaku pada kegiatan tertentu, kemudian token tersebut dapat dibagikan ke anggota. Anggota cukup memasukkan kode token tersebut untuk melakukan konfirmasi presensi secara cepat dan akurat. Selain itu, terdapat informasi detail kegiatan, petunjuk penggunaan, dan indikator waktu aktif token, sehingga proses presensi menjadi lebih transparan, terkontrol, dan efisien.

```
<?php
include 'config/database.php';
include 'includes/functions.php';

// Handle generate token
```

```
if ($_SERVER['REQUEST_METHOD'] == 'POST' &&
isset($_POST['generate_token'])) {
    $jadwal_id = $_POST['jadwal_id'];
    $token = 'KITS_' . strtoupper(substr(md5(uniqid()), 0,
6));
    $expires_at = date('Y-m-d H:i:s', strtotime('+60
minutes'));

    $query = "INSERT INTO token_presensi (jadwal_id, token,
expires_at, created_by, created_at) VALUES (?, ?, ?, ?,
NOW())";
    $stmt = $conn->prepare($query);
    $stmt->bind_param("issi", $jadwal_id, $token,
$expires_at, $_SESSION['user_id']);
    $stmt->execute();

    $token_id = $conn->insert_id;
    header('Location: presensi/token/' . $token_id);
    exit();
}

// Handle absen token
if ($_SERVER['REQUEST_METHOD'] == 'POST' &&
isset($_POST['konfirmasi_presensi'])) {
    $token = $_POST['token'];
    $anggota_id = $_SESSION['user_id'];

    // Check token validity
    $token_query = "SELECT t.*, j.* FROM token_presensi t
LEFT JOIN jadwal j ON t.jadwal_id = j.id WHERE t.token = ?
AND t.expires_at > NOW() AND t.status = 'active'";
    $stmt = $conn->prepare($token_query);
    $stmt->bind_param("s", $token);
    $stmt->execute();
    $token_data = $stmt->get_result()->fetch_assoc();

    if ($token_data) {
        // Check if already presensi
        $check_presensi = "SELECT id FROM presensi WHERE
jadwal_id = ? AND anggota_id = ?";
        $stmt_check = $conn->prepare($check_presensi);
        $stmt_check->bind_param("ii",
$token_data['jadwal_id'], $anggota_id);
        $stmt_check->execute();

        if ($stmt_check->get_result()->num_rows == 0) {
            // Insert presensi
            $presensi_query = "INSERT INTO presensi
(jadwal_id, anggota_id, token_id, status, created_at) VALUES
(?, ?, ?, 'hadir', NOW())";
            $stmt_presensi = $conn->prepare($presensi_query);
            $stmt_presensi->bind_param("iii",
$token_data['jadwal_id'], $anggota_id, $token_data['id']);
            $stmt_presensi->execute();

            $success_message = "Presensi berhasil!";
```

```
        } else {
            $error_message = "Anda sudah melakukan presensi
untuk kegiatan ini";
        }
    } else {
        $error_message = "Token tidak valid atau sudah
expired";
    }
}

// Get token detail
$token_id = isset($_GET['id']) ? $_GET['id'] : 0;
$token_detail = null;

if ($token_id) {
    $token_query = "SELECT t.*, j.* FROM token_presensi t
LEFT JOIN jadwal j ON t.jadwal_id = j.id WHERE t.id = ?";
    $stmt = $conn->prepare($token_query);
    $stmt->bind_param("i", $token_id);
    $stmt->execute();
    $token_detail = $stmt->get_result()->fetch_assoc();
}
?>

<!-- Halaman Generate Token -->
<?php if ($token_detail): ?>
<div>
    <h1>Token Presensi</h1>
    <p><?php echo $token_detail['nama_kegiatan']; ?></p>
</div>

<div>
    <div>
        <span>✅</span>
        <p>Token presensi baru berhasil dibuat!</p>
    </div>
</div>

<div>
    <h3>Informasi Kegiatan</h3>
    <div>
        <p><strong>Nama Kegiatan:</strong> <?php echo
$token_detail['nama_kegiatan']; ?></p>
        <p><strong>Tanggal:</strong> <?php echo date('d/m/Y',
strtotime($token_detail['tanggal'])); ?></p>
        <p><strong>Waktu:</strong> <?php echo date('H:i',
strtotime($token_detail['waktu_mulai'])); ?> - <?php echo
date('H:i', strtotime($token_detail['waktu_selesai']));
?></p>
        <p><strong>Lokasi:</strong> <?php echo
$token_detail['lokasi']; ?></p>
    </div>
</div>

<div>
    <h3>Token Presensi Aktif</h3>
```

```
    <div>
        <span>⏳</span>
        <span>Sisa <span id="countdown"><?php echo
$token_detail['expires_at']; ?></span>s</span>
    </div>

    <div>
        <h4>Kode Token Presensi</h4>
        <div><?php echo $token_detail['token']; ?></div>
        <p>Berlaku hingga <?php echo date('H:i:s',
strtotime($token_detail['expires_at'])); ?></p>
    </div>

    <button onclick="copyToken('<?php echo
$token_detail['token']; ?>')">Salin Token</button>
</div>

<div>
    <h3>Bagikan Token</h3>
    <button>Install App</button>
</div>

<script>
function copyToken(token) {
    navigator.clipboard.writeText(token);
    alert('Token berhasil disalin!');
}

// Countdown timer
function updateCountdown() {
    const expiresAt = new Date('<?php echo
$token_detail['expires_at']; ?>').getTime();
    const now = new Date().getTime();
    const distance = expiresAt - now;

    if (distance < 0) {
        document.getElementById('countdown').innerHTML =
"EXPIRED";
        return;
    }

    const seconds = Math.floor((distance % (1000 * 60)) /
1000);
    document.getElementById('countdown').innerHTML = seconds;
}

setInterval(updateCountdown, 1000);
</script>

<!-- Halaman Absen Token -->
<?php else: ?>
<div>
    <h1>Presensi Token</h1>
    <p>Masukkan kode token presensi</p>
</div>
```

```
<?php if (isset($error_message)): ?>
    <div>
        <p><?php echo $error_message; ?></p>
    </div>
<?php endif; ?>

<?php if (isset($success_message)): ?>
    <div>
        <p><?php echo $success_message; ?></p>
    </div>
<?php endif; ?>

<div>
    <h3>Masukkan Kode Token</h3>
    <p>Dapatkan kode token dari pengurus/pembina</p>

    <form method="POST">
        <div>
            <label>Kode Token Presensi</label>
            <input type="text" name="token"
placeholder="KITS_XXXXXX" pattern="KITS_[A-Z0-9]{6}"
required>
            <small>Format KITS_XXXXXX (8 karakter setelah
KITS_)</small>
        </div>

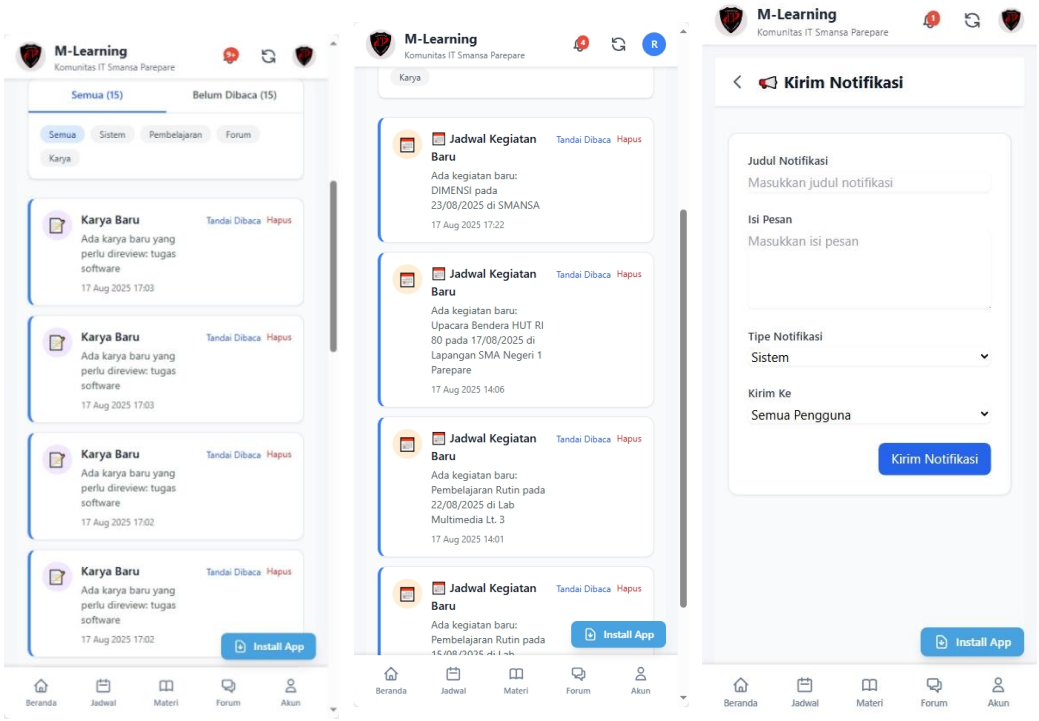
        <button type="submit" name="konfirmasi_presensi">
            <span>✅</span> Konfirmasi Presensi
        </button>
    </form>
</div>

<div>
    <h3>Petunjuk Penggunaan</h3>
    <ul>
        <li>Dapatkan kode token dari pengurus/pembina</li>
        <li>Masukkan kode token dengan format
KITS_XXXXXX</li>
        <li>Token hanya berlaku selama 60 menit</li>
        <li>Pastikan kegiatan sedang berlangsung</li>
        <li>Satu token hanya bisa digunakan sekali per
kegiatan</li>
    </ul>
</div>

<div>
    <h3>Informasi Presensi</h3>
    <div>
        <div>
            <h4>Token Presensi</h4>
            <p>Metode presensi utama menggunakan token</p>
        </div>
        <button>Install App</button>
    </div>
</div>
<?php endif; ?>
```

14. Halaman Notifikasi

Halaman ini menampilkan "Tampilan Halaman Notifikasi" dalam aplikasi *Mobile learning (M-Learning)* KITS. Pengguna dapat melihat daftar notifikasi yang diterima, dengan opsi untuk menandai semua sudah dibaca. Terdapat juga pengaturan untuk mengaktifkan atau menonaktifkan push notification. Notifikasi dikelompokkan berdasarkan kategori (Semua, Sistem, Pembelajaran, Forum, Karya), dan setiap notifikasi menampilkan informasi singkat tentang event, tanggal, dan waktu, dengan opsi untuk menandai sudah dibaca atau menghapus.



Gambar 4. 46 Tampilan Halaman Notifikasi

```
<?php
include 'config/database.php';
include 'includes/functions.php';

$user_id = $_SESSION['user_id'];
$user_type = $_SESSION['user_type'];

// Handle mark as read
```

```
if (isset($_GET['mark_read']) && is_numeric($_GET['mark_read']))
{
    $notif_id = $_GET['mark_read'];
    $query = "UPDATE notifikasi SET status = 'read' WHERE id = ?
AND user_id = ?";
    $stmt = $conn->prepare($query);
    $stmt->bind_param("ii", $notif_id, $user_id);
    $stmt->execute();
    header('Location: notifikasi');
    exit();
}

// Handle delete notification
if (isset($_GET['delete']) && is_numeric($_GET['delete'])) {
    $notif_id = $_GET['delete'];
    $query = "DELETE FROM notifikasi WHERE id = ? AND user_id =
?";
    $stmt = $conn->prepare($query);
    $stmt->bind_param("ii", $notif_id, $user_id);
    $stmt->execute();
    header('Location: notifikasi');
    exit();
}

// Get filter parameters
$tab = isset($_GET['tab']) ? $_GET['tab'] : 'semua';
$kategori = isset($_GET['kategori']) ? $_GET['kategori'] :
'semua';

// Build query conditions
$where_conditions = ["user_id = ?"];
$params = [$user_id];

if ($tab == 'belum_dibaca') {
    $where_conditions[] = "status = 'unread'";
}

if ($kategori != 'semua') {
    $where_conditions[] = "kategori = ?";
    $params[] = $kategori;
}

$where_clause = "WHERE " . implode(" AND ", $where_conditions);

// Get notifications
$notif_query = "SELECT * FROM notifikasi $where_clause ORDER BY
created_at DESC";
$stmt = $conn->prepare($notif_query);
$stmt->bind_param(str_repeat('s', count($params)), ...$params);
$stmt->execute();
$notifications = $stmt->get_result()->fetch_all(MYSQLI_ASSOC);

// Count notifications
$count_semua = count($notifications);
$count_belum_dibaca = 0;
foreach ($notifications as $notif) {
```

```
    if ($notif['status'] == 'unread') {
        $count_belum_dibaca++;
    }
}
?>

<div>
    <h1>Notifikasi</h1>
</div>

<div>
    <a href="?tab=semua&kategori=<?php echo $kategori; ?>" <?php
echo ($tab == 'semua') ? 'class="active"' : ''; ?>>Semua (<?php
echo $count_semua; ?>)</a>
    <a href="?tab=belum_dibaca&kategori=<?php echo $kategori;
?>" <?php echo ($tab == 'belum_dibaca') ? 'class="active"' : '';
?>>Belum Dibaca (<?php echo $count_belum_dibaca; ?>)</a>
</div>

<div>
    <a href="?tab=<?php echo $tab; ?>&kategori=semua" <?php echo
($kategori == 'semua') ? 'class="active"' : ''; ?>>Semua</a>
    <a href="?tab=<?php echo $tab; ?>&kategori=sistem" <?php
echo ($kategori == 'sistem') ? 'class="active"' : '';
?>>Sistem</a>
    <a href="?tab=<?php echo $tab; ?>&kategori=pembelajaran"
<?php echo ($kategori == 'pembelajaran') ? 'class="active"' :
''; ?>>Pembelajaran</a>
    <a href="?tab=<?php echo $tab; ?>&kategori=forum" <?php echo
($kategori == 'forum') ? 'class="active"' : ''; ?>>Forum</a>
    <a href="?tab=<?php echo $tab; ?>&kategori=karya" <?php echo
($kategori == 'karya') ? 'class="active"' : ''; ?>>Karya</a>
</div>

<div>
    <?php foreach ($notifications as $notif): ?>
    <div>
        <div>
            <span></span>
        </div>

        <div>
            <h4><?php echo $notif['judul']; ?></h4>
            <p><?php echo $notif['pesan']; ?></p>
            <p><?php echo date('d M Y H:i',
strtotime($notif['created_at'])); ?></p>
        </div>

        <div>
            <a href="?mark_read=<?php echo $notif['id'];
?>">Tandai Dibaca</a>
            <a href="?delete=<?php echo $notif['id'];
?>">Hapus</a>
        </div>
    </div>
    <?php endforeach; ?>
```

```
</div>

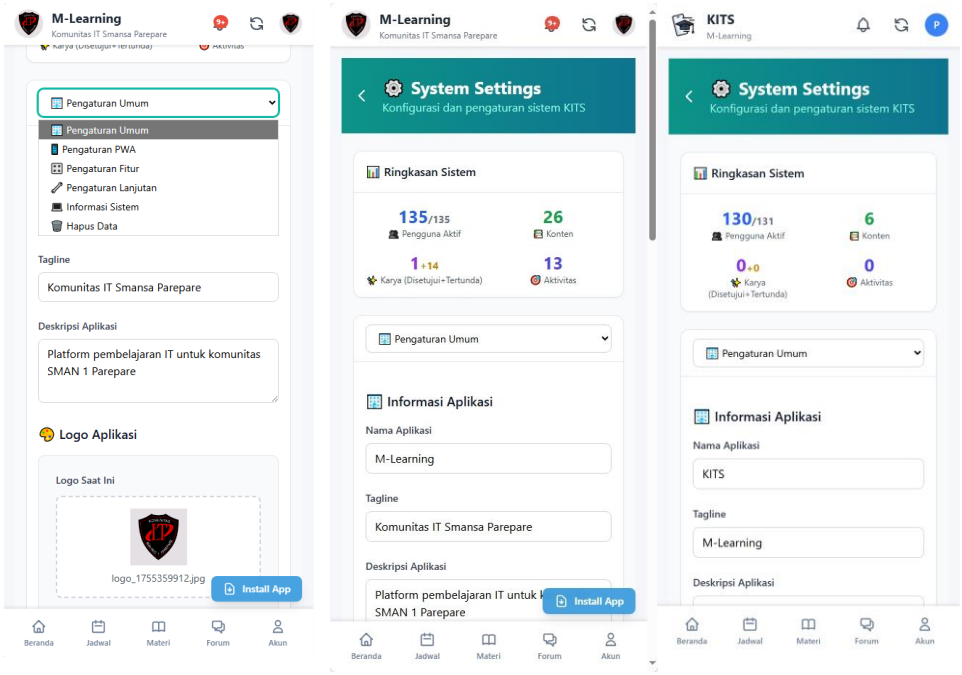
<div>
    <button>Install App</button>
</div>
```

15. Halaman Sistem Setting

Halaman Sistem Setting yang berfungsi sebagai pusat pengaturan aplikasi *M-Learning* Komunitas IT Smansa Parepare. Pada sisi kiri, terdapat ringkasan sistem yang menampilkan jumlah pengguna aktif, total kegiatan, dan aktivitas harian, sehingga admin dapat memantau kondisi sistem secara cepat. Bagian bawahnya memuat menu dropdown Pengaturan Umum yang berisi informasi dasar aplikasi, seperti nama aplikasi, tagline, dan deskripsi singkat.

Sementara itu, tampilan di sisi kanan memperlihatkan form pengaturan yang lebih detail. Admin dapat memperbarui data dasar aplikasi, termasuk tagline dan deskripsi, serta mengunggah atau mengganti logo aplikasi yang ditampilkan. Selain Pengaturan Umum, terdapat pula opsi menu lain seperti Pengaturan PWA, Pengaturan Fitur, Pengaturan Lanjutan, dan Privasi Data, yang memberikan fleksibilitas dalam mengelola tampilan serta fungsi aplikasi

sesuai kebutuhan. Dengan adanya halaman ini, pengelolaan sistem menjadi lebih terstruktur, terpusat, dan mudah diakses oleh administrator.



Gambar 4. 47 Tampilan Sistem Setting

```
<?php
include 'config/database.php';
include 'includes/functions.php';

// Handle form submission
if ($_SERVER['REQUEST_METHOD'] == 'POST' &&
isset($_POST['update_settings'])) {
    $nama_aplikasi = $_POST['nama_aplikasi'];
    $tagline = $_POST['tagline'];
    $deskripsi_aplikasi = $_POST['deskripsi_aplikasi'];

    $query = "UPDATE system_settings SET
        nama_aplikasi = ?,
        tagline = ?,
        deskripsi_aplikasi = ?,
        updated_at = NOW()
        WHERE id = 1";
    $stmt = $conn->prepare($query);
    $stmt->bind_param("sss", $nama_aplikasi, $tagline,
$deskripsi_aplikasi);
    $stmt->execute();

    $success_message = "Pengaturan berhasil diperbarui";
}

// Get system settings
$settings_query = "SELECT * FROM system_settings WHERE id = 1";
```

```
$settings_result = $conn->query($settings_query);
$settings = $settings_result->fetch_assoc();

// Get system statistics
$stats_query = "SELECT
    (SELECT COUNT(*) FROM anggota WHERE status = 'active') as
pengguna_aktif,
    (SELECT COUNT(*) FROM materi) as konten,
    (SELECT COUNT(*) FROM karya WHERE status = 'approved') as
karya_disetujui,
    (SELECT COUNT(*) FROM karya WHERE status = 'pending') as
karya_tertunda,
    (SELECT COUNT(*) FROM jadwal WHERE MONTH(tanggal) =
MONTH(CURDATE())) as aktivitas";
$stats_result = $conn->query($stats_query);
$stats = $stats_result->fetch_assoc();

// Get pengaturan umum options
$pengaturan_options = [
    'maintenance_mode' => 'Mode Maintenance',
    'registration_open' => 'Pendaftaran Terbuka',
    'auto_approve_karya' => 'Auto Approve Karya',
    'email_notifications' => 'Notifikasi Email',
    'push_notifications' => 'Push Notifications'
];
?>

<div>
    <h1>System Settings</h1>
    <p>Konfigurasi dan pengaturan sistem KITS</p>
</div>

<?php if (isset($success_message)): ?>
    <div>
        <p><?php echo $success_message; ?></p>
    </div>
<?php endif; ?>

<div>
    <h3>Ringkasan Sistem</h3>

    <div>
        <div>
            <span><?php echo $stats['pengguna_aktif']; ?>/<?php
echo $stats['pengguna_aktif']; ?></span>
            <span>Pengguna Aktif</span>
        </div>
        <div>
            <span><?php echo $stats['konten']; ?></span>
            <span>Konten</span>
        </div>
        <div>
            <span><?php echo $stats['karya_disetujui'];
?>+<?php echo $stats['karya_tertunda']; ?></span>
            <span>Karya (Disetujui-Tertunda)</span>
        </div>
```

```
        <div>
            <span><?php echo $stats['aktivitas']; ?></span>
            <span>Aktivitas</span>
        </div>
    </div>
</div>

<div>
    <h3>Pengaturan Umum</h3>
    <select>
        <option value="">Pilih Pengaturan</option>
        <?php foreach ($pengaturan_options as $key => $label):
?>
            <option value="<?php echo $key; ?>"><?php echo
$label; ?></option>
        <?php endforeach; ?>
    </select>
</div>

<div>
    <h3>Informasi Aplikasi</h3>

    <form method="POST">
        <div>
            <label>Nama Aplikasi</label>
            <input type="text" name="nama_aplikasi"
value="<?php echo $settings['nama_aplikasi']; ?>" required>
        </div>

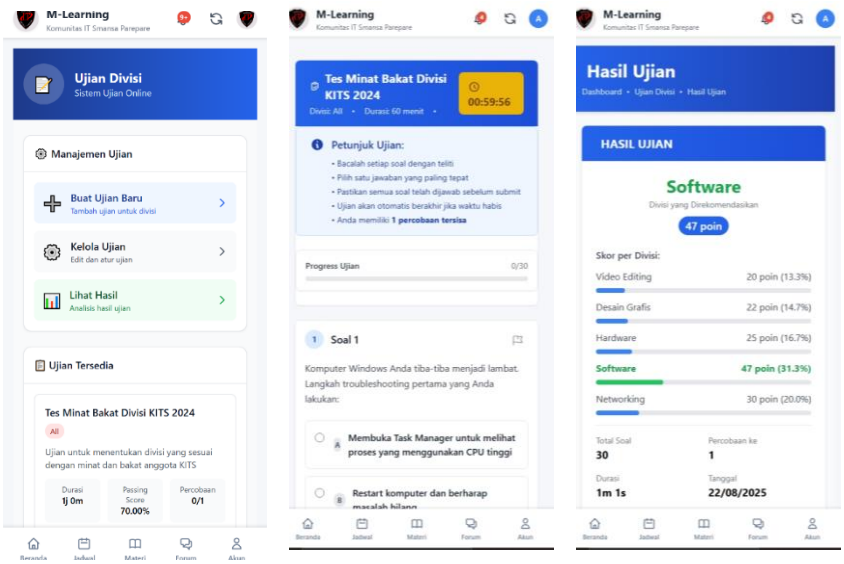
        <div>
            <label>Tagline</label>
            <input type="text" name="tagline" value="<?php echo
$settings['tagline']; ?>" required>
        </div>

        <div>
            <label>Deskripsi Aplikasi</label>
            <textarea name="deskripsi_aplikasi" required><?php
echo $settings['deskripsi_aplikasi']; ?></textarea>
        </div>

        <button type="submit" name="update_settings">Simpan
Perubahan</button>
    </form>
</div>

<div>
    <button>Install App</button>
</div>
```

16. Halaman Ujian Divisi



Gambar 4. 48 Tampilan Ujian Divisi

Gambar tersebut menampilkan antarmuka aplikasi "*M-Learning*" yang dirancang untuk mengelola dan melakukan ujian online. Pada gambar pertama, terlihat halaman utama "Ujian Divisi" yang memiliki opsi untuk Manajemen Ujian, seperti membuat, mengelola, dan melihat hasil ujian, serta daftar ujian yang tersedia. Gambar kedua menunjukkan antarmuka saat memulai ujian, menampilkan detail seperti nama ujian, durasi, petunjuk pengerjaan, dan pertanyaan ujian yang sedang berlangsung. Terakhir, gambar ketiga adalah halaman Hasil Ujian, yang menampilkan skor total, durasi pengerjaan, dan rekomendasi divisi berdasarkan hasil ujian, lengkap dengan rincian skor untuk setiap divisi. Secara keseluruhan, gambar ini mendemonstrasikan alur lengkap penggunaan aplikasi untuk ujian online, mulai dari halaman beranda hingga hasil akhir.

```
<?php
include 'config/database.php';
include 'includes/functions.php';

$ujian_id = isset($_GET['id']) ? $_GET['id'] : 0;
```

```
$user_id = $_SESSION['user_id'];

// Get ujian detail
$ujian_query = "SELECT * FROM ujian WHERE id = ? AND status =
'active'";
$stmt = $conn->prepare($ujian_query);
$stmt->bind_param("i", $ujian_id);
$stmt->execute();
$ujian = $stmt->get_result()->fetch_assoc();

// Get soal ujian
$soal_query = "SELECT * FROM soal_ujian WHERE ujian_id = ? ORDER
BY urutan ASC";
$stmt = $conn->prepare($soal_query);
$stmt->bind_param("i", $ujian_id);
$stmt->execute();
$soal_list = $stmt->get_result()->fetch_all(MYSQLI_ASSOC);

// Handle submit ujian
if ($_SERVER['REQUEST_METHOD'] == 'POST' &&
isset($_POST['submit_ujian'])) {
    $jawaban_data = $_POST['jawaban'];
    $waktu_mulai = $_POST['waktu_mulai'];
    $waktu_selesai = date('Y-m-d H:i:s');

    // Calculate skor per divisi
    $skor_divisi = [];
    foreach ($soal_list as $soal) {
        $jawaban_user = $jawaban_data[$soal['id']] ?? '';
        if ($jawaban_user == $soal['jawaban_benar']) {
            $divisi = $soal['divisi'];
            $skor_divisi[$divisi] = ($skor_divisi[$divisi] ?? 0)
+ $soal['bobot'];
        }
    }

    // Find recommended division
    $divisi_rekomendasi = array_keys($skor_divisi,
max($skor_divisi))[0];
    $skor_tertinggi = max($skor_divisi);

    // Save hasil ujian
    $hasil_query = "INSERT INTO hasil_ujian (ujian_id, user_id,
divisi_rekomendasi, skor_tertinggi, skor_detail, waktu_mulai,
waktu_selesai, created_at) VALUES (?, ?, ?, ?, ?, ?, ?, NOW())";
    $stmt = $conn->prepare($hasil_query);
    $skor_json = json_encode($skor_divisi);
    $stmt->bind_param("iisisss", $ujian_id, $user_id,
$divisi_rekomendasi, $skor_tertinggi, $skor_json, $waktu_mulai,
$waktu_selesai);
    $stmt->execute();

    header('Location: ujian/hasil/' . $ujian_id);
    exit();
}
```

```
// Get hasil ujian
$hasil_id = isset($_GET['hasil']) ? $_GET['hasil'] : 0;
$hasil_ujian = null;

if ($hasil_id) {
    $hasil_query = "SELECT * FROM hasil_ujian WHERE id = ? AND
user_id = ?";
    $stmt = $conn->prepare($hasil_query);
    $stmt->bind_param("ii", $hasil_id, $user_id);
    $stmt->execute();
    $hasil_ujian = $stmt->get_result()->fetch_assoc();
}
?>

<!-- Halaman Ujian -->
<?php if ($ujian && !$hasil_ujian): ?>
<div>
    <h2><?php echo $ujian['judul']; ?></h2>
    <p>Divisi: <?php echo $ujian['target_divisi'] ?: 'All';
?></p>
    <p>Durasi: <?php echo $ujian['durasi_menit']; ?> menit</p>
    <div>
        <span id="countdown">00:<?php echo
$ujian['durasi_menit']; ?>:00</span>
    </div>
</div>

<div>
    <h3>Petunjuk Ujian:</h3>
    <ul>
        <li>Bacalah setiap soal dengan teliti</li>
        <li>Pilih satu jawaban yang paling tepat</li>
        <li>Pastikan semua soal telah dijawab sebelum
submit</li>
        <li>Ujian akan otomatis berakhir jika waktu habis</li>
        <li>Anda memiliki <?php echo $ujian['max_attempts'] - 1;
?> percobaan tersisa</li>
    </ul>
</div>

<div>
    <h3>Progress Ujian</h3>
    <div>
        <span id="progress">0/<?php echo count($soal_list);
?></span>
    </div>
</div>

<form method="POST" id="ujian-form">
    <input type="hidden" name="waktu_mulai" value="<?php echo
date('Y-m-d H:i:s'); ?>">

    <?php foreach ($soal_list as $index => $soal): ?>
    <div class="soal-item" <?php echo ($index > 0) ?
'style="display:none;"' : ''; ?>>
```

```
        <h4><?php echo $index + 1; ?> Soal <?php echo $index +
1; ?></h4>
        <p><?php echo $soal['pertanyaan']; ?></p>

        <div>
            <?php
            $pilihan = json_decode($soal['pilihan'], true);
            foreach ($pilihan as $key => $value):
            ?>
                <label>
                    <input type="radio" name="jawaban[<?php echo
$soal['id']; ?>]" value="<?php echo $key; ?>" required>
                    <?php echo $key; ?>. <?php echo $value; ?>
                </label>
            <?php endforeach; ?>
        </div>
    </div>
    <?php endforeach; ?>

    <div>
        <button type="button" id="prev-btn"
style="display:none;">Sebelumnya</button>
        <button type="button" id="next-btn">Selanjutnya</button>
        <button type="submit" name="submit_ujian" id="submit-
btn" style="display:none;">Submit Ujian</button>
    </div>
</form>

<script>
let currentQuestion = 0;
const totalQuestions = <?php echo count($soal_list); ?>;
let timeLeft = <?php echo $ujian['durasi_menit'] * 60; ?>;

function updateProgress() {
    document.getElementById('progress').textContent =
(currentQuestion + 1) + '/' + totalQuestions;
}

function showQuestion(index) {
    document.querySelectorAll('.soal-item').forEach((item, i) =>
{
        item.style.display = i === index ? 'block' : 'none';
    });

    document.getElementById('prev-btn').style.display = index >
0 ? 'block' : 'none';
    document.getElementById('next-btn').style.display = index <
totalQuestions - 1 ? 'block' : 'none';
    document.getElementById('submit-btn').style.display = index
=== totalQuestions - 1 ? 'block' : 'none';

    updateProgress();
}

document.getElementById('next-btn').addEventListener('click',
function() {
```

```
    if (currentQuestion < totalQuestions - 1) {
        currentQuestion++;
        showQuestion(currentQuestion);
    }
});

document.getElementById('prev-btn').addEventListener('click',
function() {
    if (currentQuestion > 0) {
        currentQuestion--;
        showQuestion(currentQuestion);
    }
});

// Countdown timer
function updateTimer() {
    const hours = Math.floor(timeLeft / 3600);
    const minutes = Math.floor((timeLeft % 3600) / 60);
    const seconds = timeLeft % 60;

    document.getElementById('countdown').textContent =
        String(hours).padStart(2, '0') + ':' +
        String(minutes).padStart(2, '0') + ':' +
        String(seconds).padStart(2, '0');

    if (timeLeft <= 0) {
        document.getElementById('ujian-form').submit();
    }

    timeLeft--;
}

setInterval(updateTimer, 1000);
</script>

<!-- Halaman Hasil Ujian -->
<?php elseif ($hasil_ujian): ?>
<div>
    <h2>HASIL UJIAN</h2>
    <div>
        <h3><?php echo $hasil_ujian['divisi_rekomendasi'];
?></h3>
        <p>Divisi yang Direkomendasikan</p>
        <span><?php echo $hasil_ujian['skor_tertinggi']; ?>
poin</span>
    </div>
</div>

<div>
    <h3>Skor per Divisi:</h3>

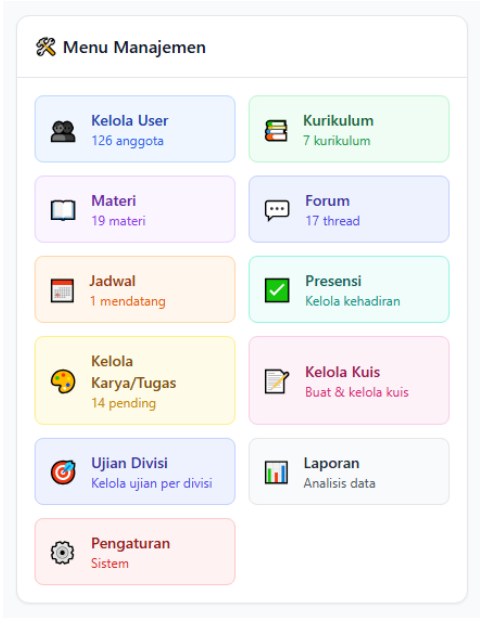
    <?php
    $skor_detail = json_decode($hasil_ujian['skor_detail'],
true);
    $total_skor = array_sum($skor_detail);
```

```
    foreach ($skor_detail as $divisi => $skor):
        $persentase = $total_skor > 0 ? round(($skor /
$total_skor) * 100, 1) : 0;
    ?>
    <div>
        <span><?php echo $divisi; ?></span>
        <span><?php echo $skor; ?> poin (<?php echo $persentase;
?>%)</span>
        <div>
            <div style="width: <?php echo $persentase;
?>%;"></div>
        </div>
    </div>
    <?php endforeach; ?>
</div>

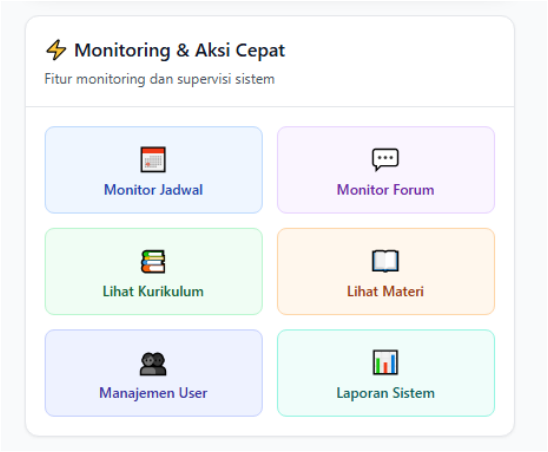
<div>
    <p>Total Soal: <?php echo count($soal_list); ?></p>
    <p>Percobaan ke: <?php echo $hasil_ujian['attempt_number'];
?></p>
    <p>Durasi: <?php echo $hasil_ujian['durasi']; ?></p>
    <p>Tanggal: <?php echo date('d/m/Y',
strtotime($hasil_ujian['created_at'])); ?></p>
</div>
<?php endif; ?>
```

17.

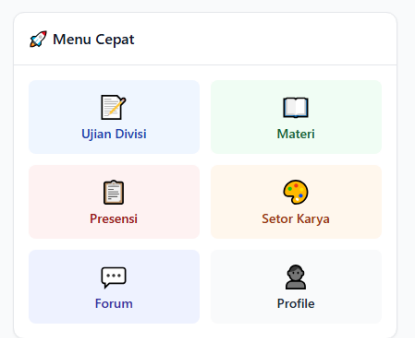
18. Menu Manajemen



Gambar 4. 50 Menu Manajemen Akun Pengurus



Gambar 4. 49 Menu Manajemen Akun Pembina



Gambar 4. 51 Menu Manajemen Akun Anggota

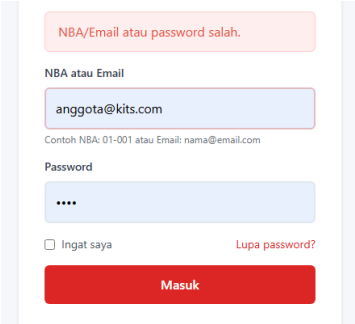
D. Pengujian Sistem

Pengujian sistem bertujuan untuk mengidentifikasi kesalahan, kesenjangan, atau kekurangan dalam sistem sebelum digunakan oleh pengguna akhir. Ada dua pendekatan utama dalam pengujian sistem ini, yaitu pengujian *Black box* dan pengujian *white box.*

1. Black box

a. Black box testing kesalahan NBA/Email dan Password

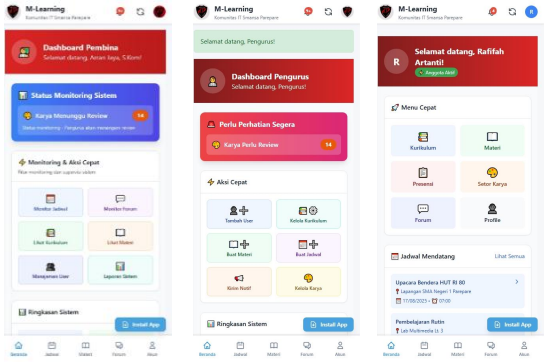
Tabel 4. 23 *Black box testing* kesalahan NBA/Email dan Password

Tes faktor	Hasil	Keterangan
Memasukkan NBA/Email atau password yang tidak sesuai	✓	Berhasil, ketika NBA/Email atau password tidak sesuai tampil alert bahwa NBA/Email atau Password Salah
Tangkapan layar		
		

b. *Black box testing login* berhasil

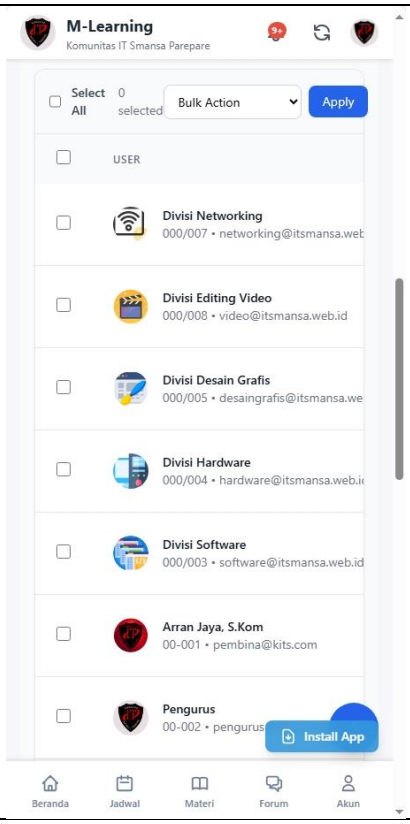
Tabel 4. 24 *Black box testing* login berhasil

Tes faktor	Hasil	Keterangan
Memasukkan NBA/Email atau password yang sesuai	✓	Sistem otomatis mengalihkan ke halaman Dashboard sesuai Role/peran pengguna

Tangkapan layar


c. *Black box testing* tambah data

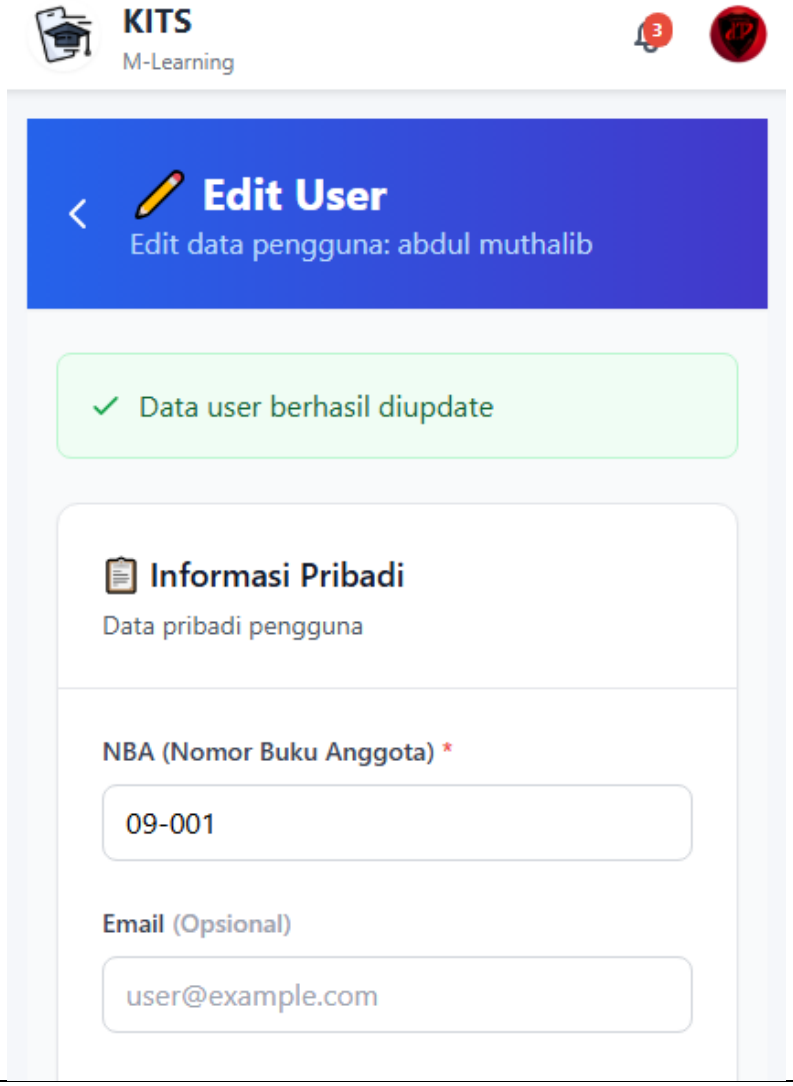
Tabel 4. 25 Tabel *Testing* Tambah Data

Tes faktor	Hasil	Keterangan
Admin mengisi form tambah anggota dan menekan simpan	✓	Sukses, sebuah data baru telah berhasil untuk ditambahkan
Tangkapan layar		
		

d. *Black box testing ubah data*

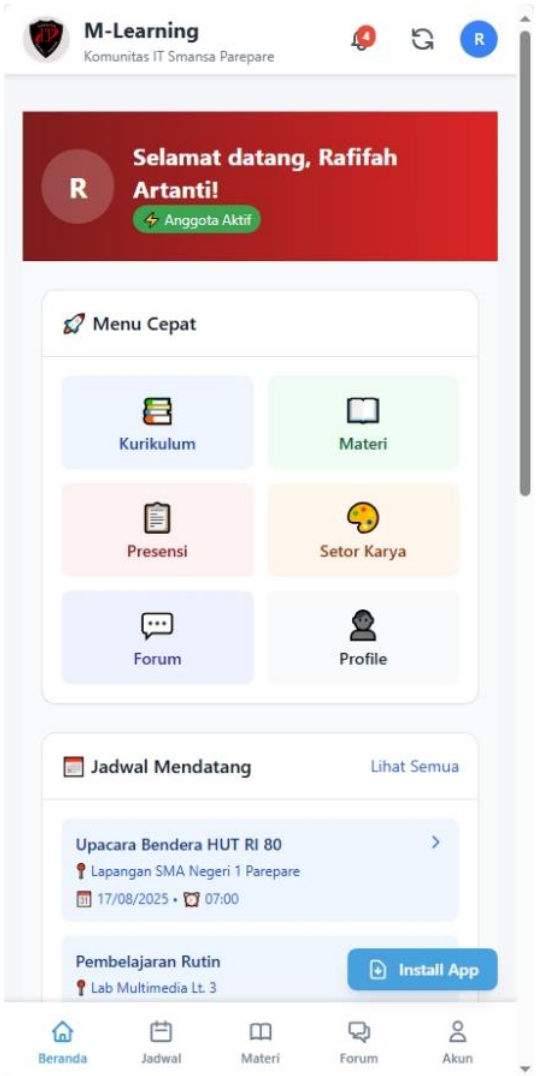
Tabel 4. 26 *Black box testing* ubah data

Tes faktor	Hasil	Keterangan
Admin mengisi form ubah data anggota dan menekan simpan	✓	Sukses, sebuah notifikasi data anggota "abdul" telah berhasil untuk diubah
Tangkapan layar		



e. ***Black box testing*** **halaman** ***dashboard***

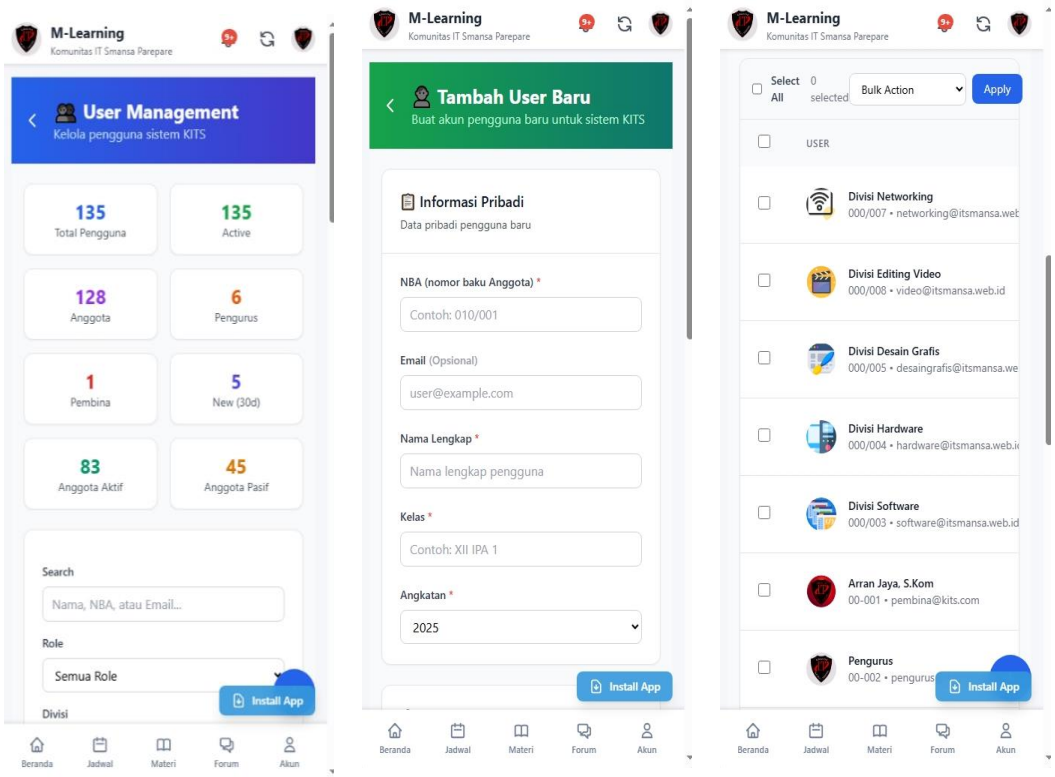
Tabel 4. 27 *Black box testing halaman dashboard*

Tes faktor	Hasil	Keterangan
Pengguna melakukan inisiasi penggunaan aplikasi	✓	Sukses, tampil halaman beranda
Tangkapan layar		
		

f. *Black box testing* halaman Manajemen pengguna

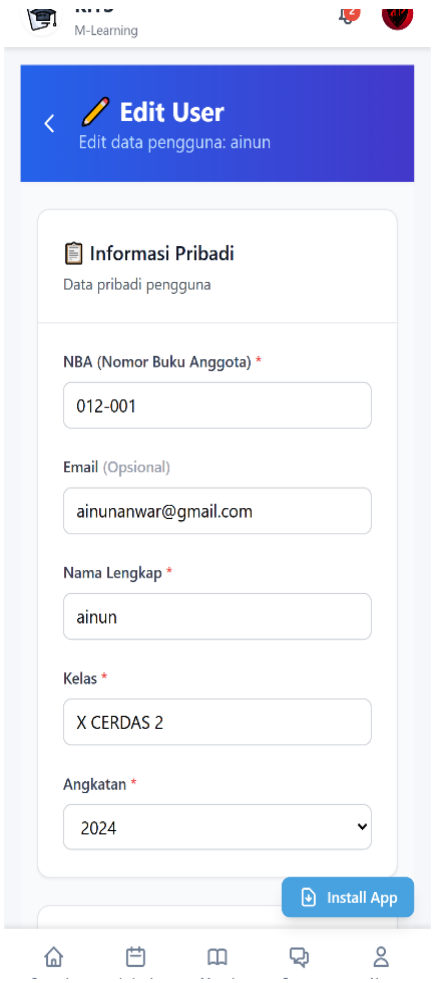
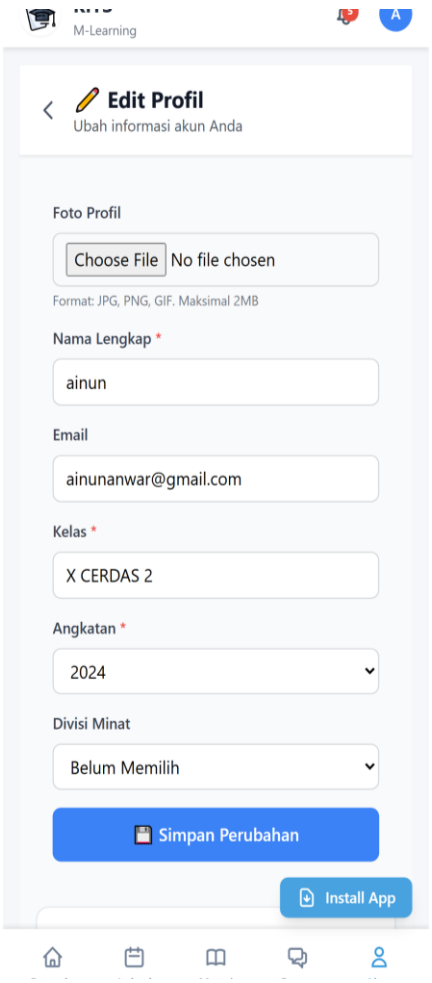
Tabel 4. 28 *Black box testing* halaman Manajemen Pengguna

Tes faktor	Hasil	Keterangan
Admin membuka halaman manajemen Pengguna	✓	Sukses, tampil halaman Manajemen Pengguna
Tangkapan layar		



g. *Black box testing* halaman Manajemen Profil

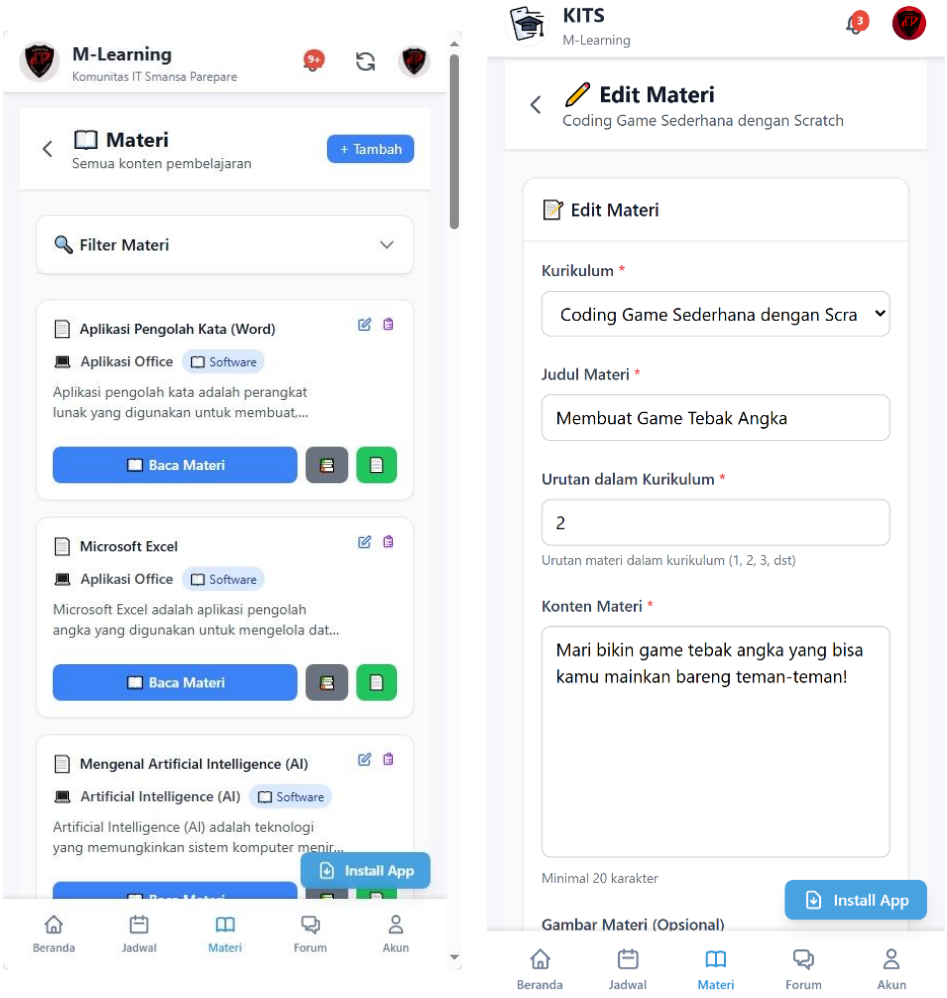
Tabel 4. *29 Black box testing* Manajemen Profil

Tes faktor	Hasil	Keterangan
Pengguna Mencoba mengakses Menu Profil	✓	Sukses, tampil halaman Profil Pengguna
Tangkapan layar		
		

h. *Black box testing* halaman Materi

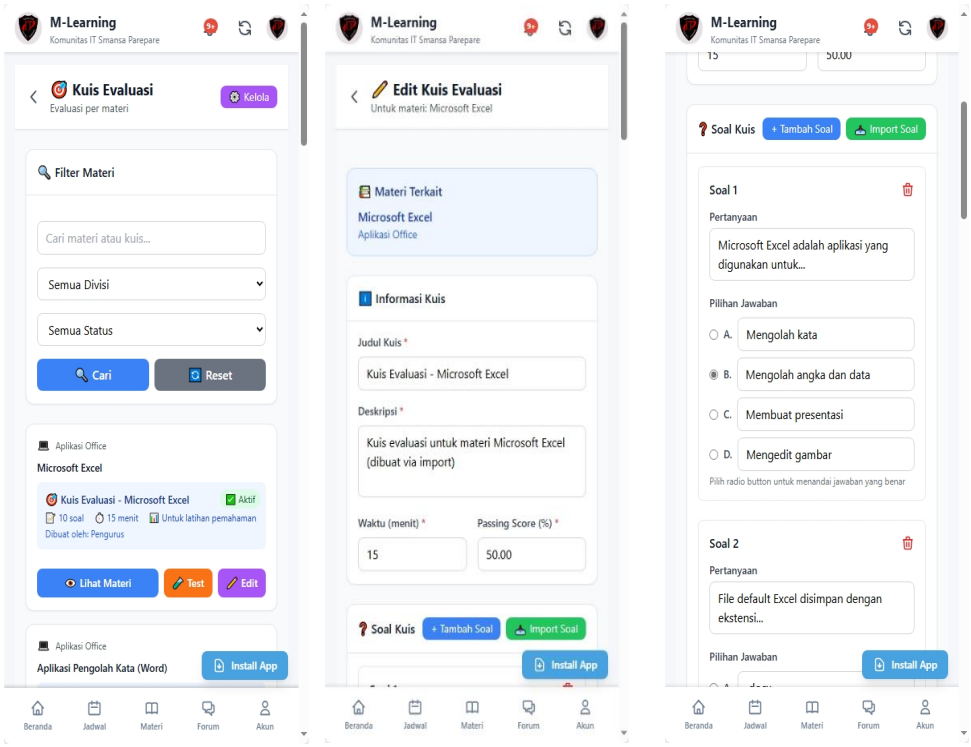
Tabel 4. 30 *Black box testing* Materi

Tes faktor	Hasil	Keterangan
Pengguna Mencoba mengakses Menu Materi	✓	Sukses, tampil halaman Materi
Tangkapan layar		



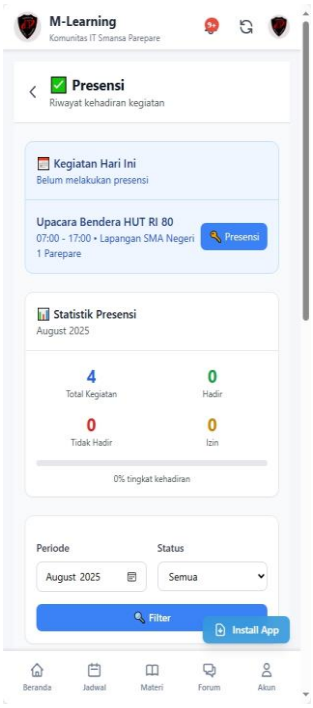
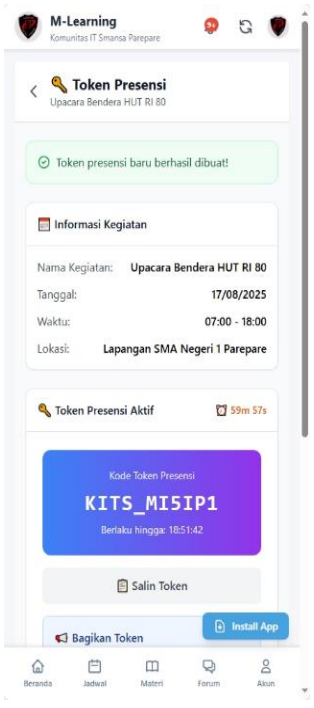
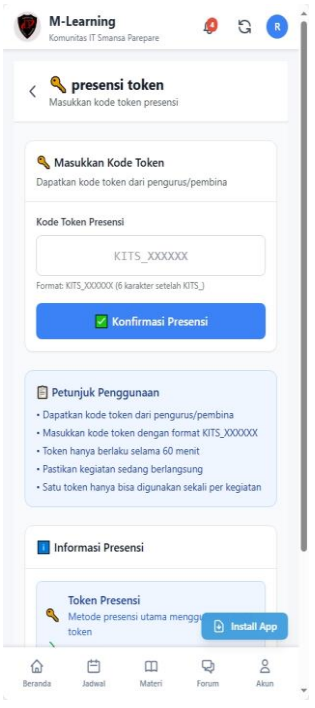
i. *Black box testing* halaman Kuis Evaluasi

Tabel 4. 31 *Black box testing* Kuis Evaluasi

Tes faktor	Hasil	Keterangan
Pengguna Mencoba mengakses Kuis Evaluasi	✓	Sukses, tampil halaman Kuis Evaluasi
Tangkapan layar		
		

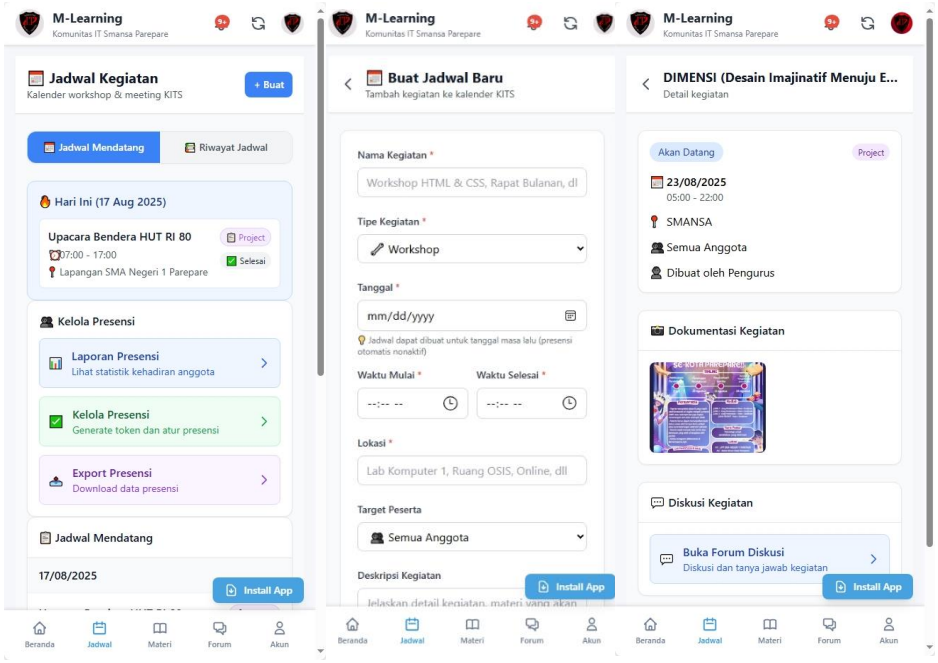
j. *Black box testing* halaman Presensi

Tabel 4. 32 *Black box testing* Presensi

Tes faktor	Hasil	Keterangan
Pengguna Mencoba mengakses Menu Presensi	✓	Sukses, tampil presensi
Tangkapan layar		
  		

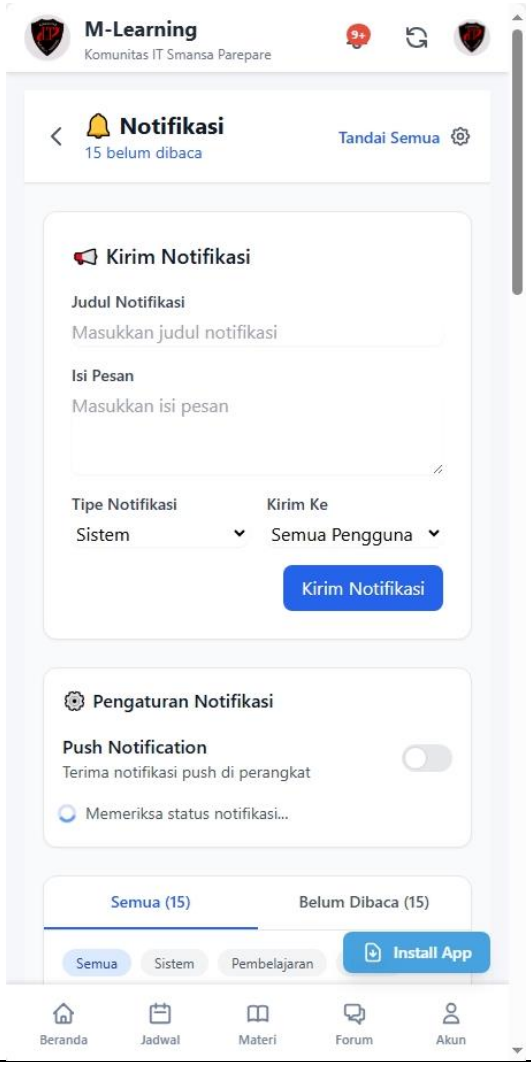
k. *Black box testing* halaman Jadwal

Tabel 4. 33 *Black box testing* Jadwal

Tes faktor	Hasil	Keterangan
Pengguna Mencoba mengakses Menu Jadwal	✓	Sukses, tampil halaman jadwal
Tangkapan layar		
		

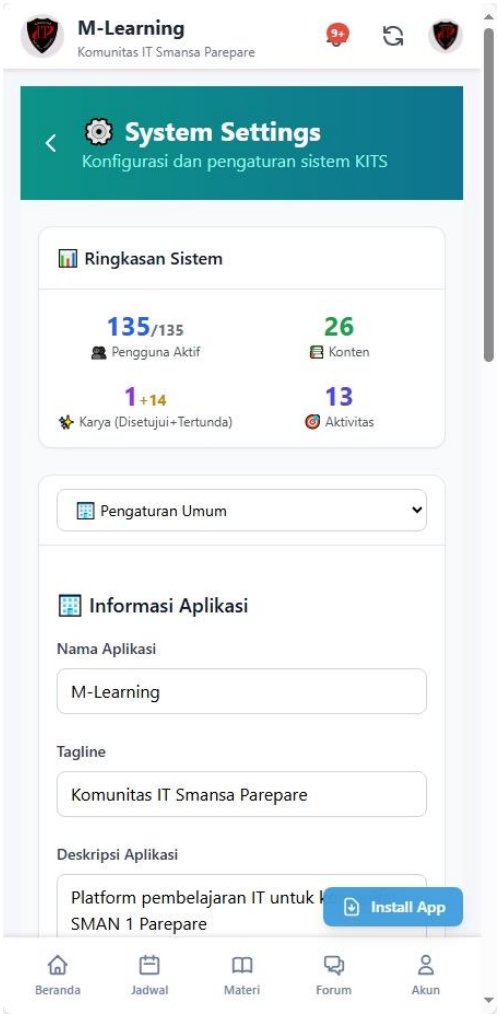
l. *Black box testing* halaman Notifikasi

Tabel 4. 34 *Black box testing* Notifikasi

Tes faktor	Hasil	Keterangan
Pengguna Mencoba mengakses Menu Notifikasi	✓	Sukses, tampil halaman Notifikasi dan settingan push notifikasi
Tangkapan layar		
		

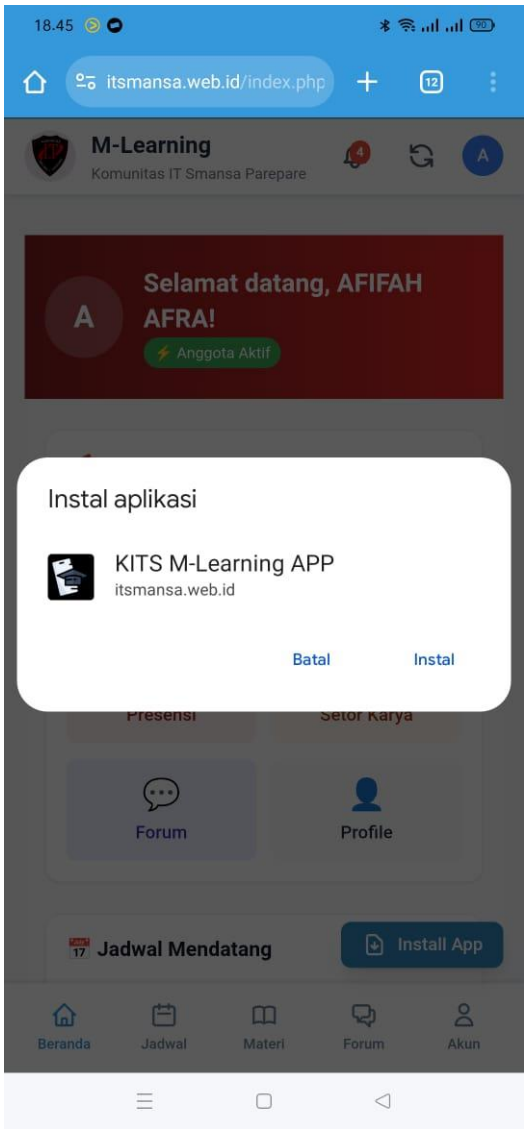
m. *Black box testing* halaman Sistem Setting

Tabel 4. 35 *Black box testing* halaman Sistem Setting

Tes faktor	Hasil	Keterangan
Admin mengakses Menu Sistem setting	✓	Sukses, tampil halaman setting aplikasi
Tangkapan layar		
		

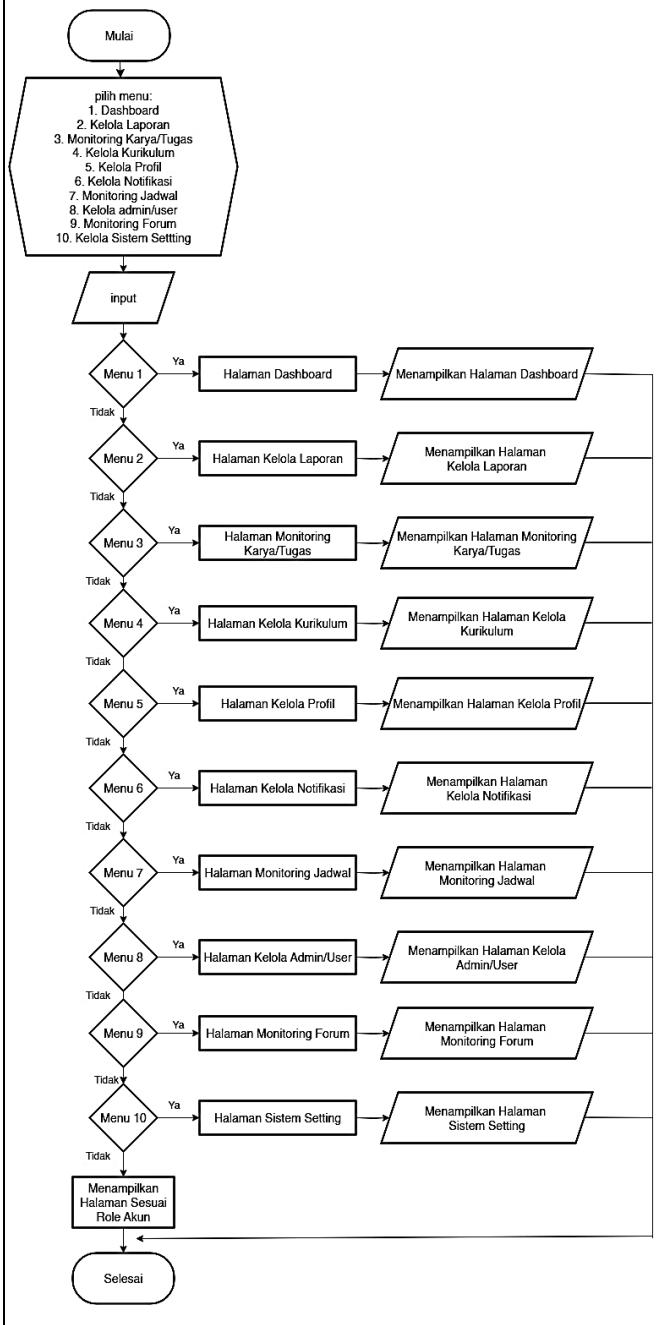
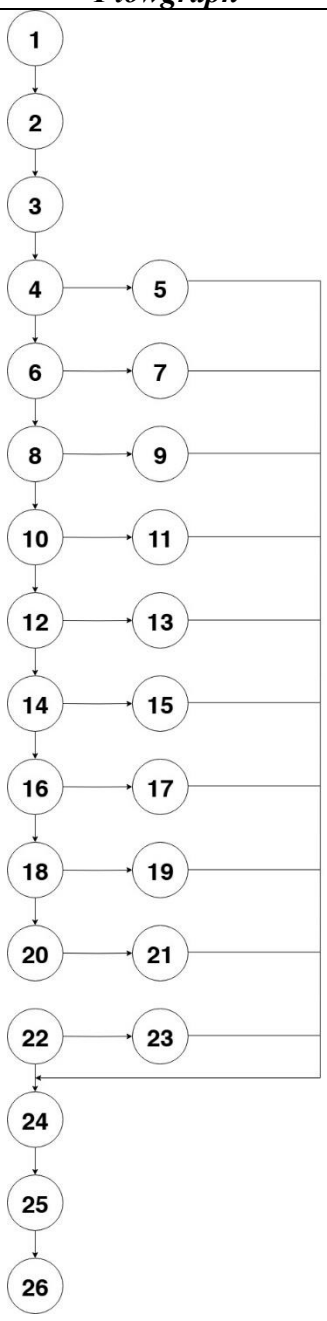
n. *Black box testing* Instalasi *PWA*

Tabel 4. 36 *Black box testing* Instalasi PWA

Tes faktor	Hasil	Keterangan
Pengguna mencoba menekan tombol Instal Aplikasi yang muncul	✓	Sukses, *icon / shortcut* aplikasi muncul di tampilan *homescreen.*
Tangkapan layar		
		

2. *White box*

Tabel 4. 37 *White box* Pembina

Flowchart	***Flowgraph***
	

Independent Path Pembina:

1. *Independent Path 1-2-3-4-5-22-23-24-25-26*
2. *Independent Path 1-2-3-4-6-7-22-23-24-25-26*

3. *Independent Path 1-2-3-4-6-8-9-22-23-24-25-26*
4. *Independent Path 1-2-3-4-6-8-10-11-22-23-24-25-26*
5. *Independent Path 1-2-3-4-6-8-10-12-13-22-23-24-25-26*
6. *Independent Path 1-2-3-4-6-8-10-12-14-15-22-23-24-25-26*
7. *Independent Path 1-2-3-4-6-8-10-12-14-16-17-22-23-24-25-26*
8. *Independent Path 1-2-3-4-6-8-10-12-14-16-18-19-22-23-24-25-26*
9. *Independent Path 1-2-3-4-6-8-10-12-14-16-18-20-21-22-23-24-25-26*
10. *Independent Path 1-2-3-4-6-8-10-12-14-16-18-20-22-23-24-25-26 (jalur langsung tanpa memilih cabang 1–9; "menu 10")*

Grafik Matriks (*Graph Matrix*):

Tabel 4. 38 Grafik Matriks (*Graph Matrix*)

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	E-1
1		1																									1-1=0
2			1																								1-1=0
3				1																							1-1=0
4					1	1																					2-1=1
5																						1					1-1=0
6							1	1																			2-1=1
7																						1					1-1=0
8									1	1																	2-1=1
9																						1					1-1=0
10											1	1															2-1=1
11																						1					1-1=0
12													1	1													2-1=1
13																						1					1-1=0
14															1	1											2-1=1
15																						1					1-1=0
16																	1	1									2-1=1
17																						1					1-1=0
18																			1	1							2-1=1
19																						1					1-1=0
20																					1	1					2-1=1
21																						1					1-1=0
22																							1	1			2-1=1
23																						1					1-1=0
24																										1	1-1=0
25																											0-1=0
26																											0-1=0
SUM (E+1)																											10+1=11

Berdasarkan perhitungan diatas: Region = 11, Independent Path = 11

***Cyclomatic complexity* = 11**, maka modul Pembina dinyatakan bebas dari kesalahan.

1. **Perhitungan Kompleksitas Siklomatis:**

Jumlah Node (N) = 26

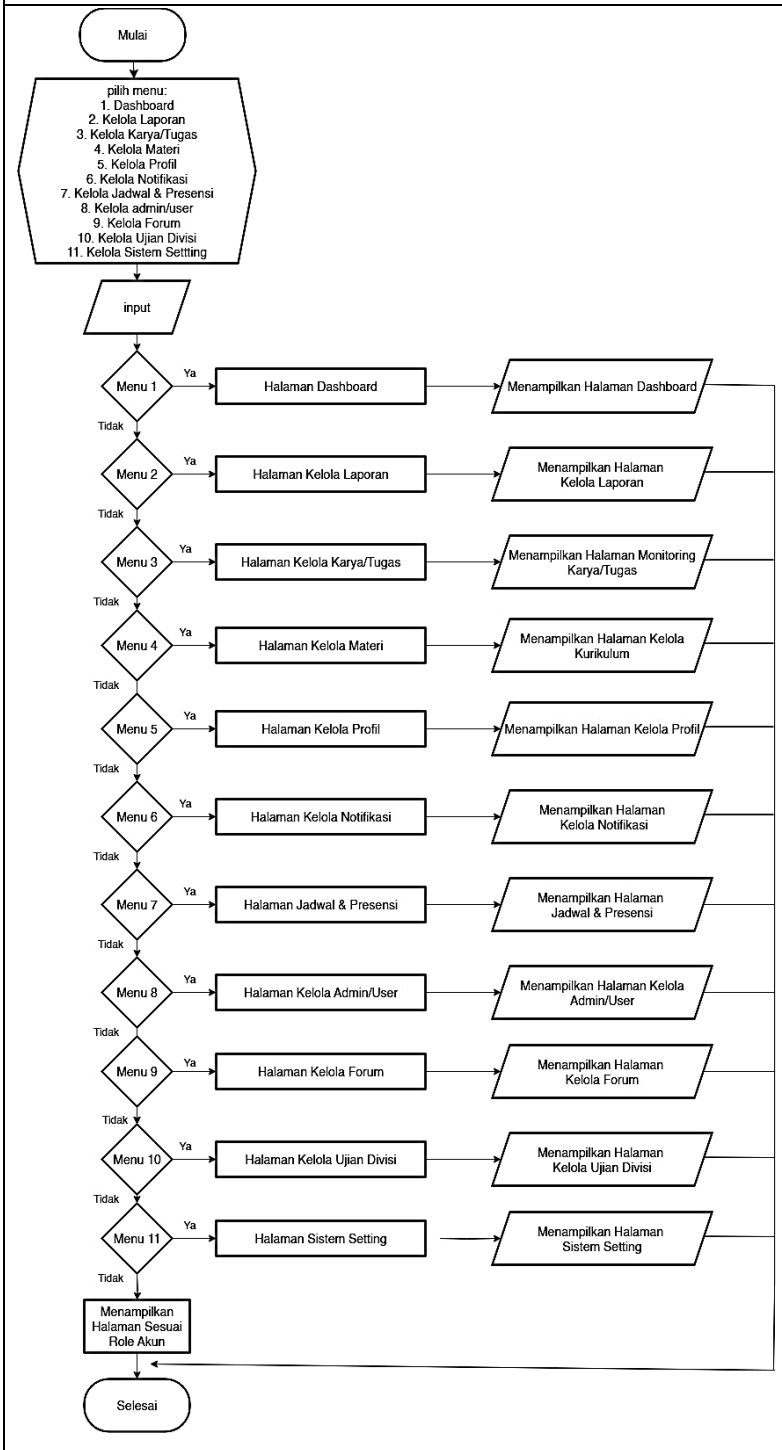
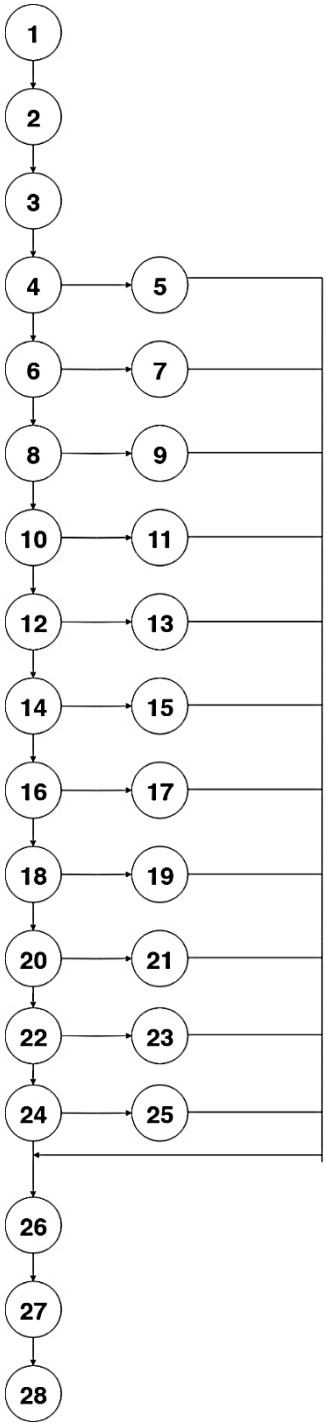
Jumlah Edge (E) = 34

V(G) = E - N + 2 = 34 - 26 + 2 = 11

Region = 11

Independent Path = 11

Tabel 4. 39 *White box* Pengurus

Flowchart	*Flowgraph*
	

Independent Path Pengurus:

1. *Independent path 1* = 1-2-3-4-5-24-26-27-28
2. *Independent path 2* = 1-2-3-4-6-7-24-26-27-28
3. *Independent path 3* = 1-2-3-8-9-24-25-26
4. *Independent path 4* = 1-2-3-4-6-8-9-24-26-27-28
5. *Independent path 5* = 1-2-3-4-6-8-10-12-13-24-26-27-28
6. *Independent path 6* = 1-2-3-4-6-8-10-12-14-15-24-26-27-28
7. *Independent path 7* = 1-2-3-4-6-8-10-12-14-16-17-24-26-27-28
8. *Independent path 8* = 1-2-3-4-6-8-10-12-14-16-18-19-24-26-27-28
9. *Independent path 9* = 1-2-3-4-6-8-10-12-14-16-18-20-21-24-26-27-28
10. *Independent path 10* = 1-2-3-4-6-8-10-12-14-16-18-20-22-23-24-26-27-28
11. *Independent path 11* = 1-2-3-4-6-8-10-12-14-16-18-20-22-24-25-24-26-27-28

Grafik Matriks (*Graph Matrix*):

Tabel 4. 40 *Grafik Matriks white box Pengurus*

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	E-1
1		1																											1-1=0
2			1																										1-1=0
3				1																									1-1=0
4					1	1																							2-1=1
5																								1					1-1=0
6							1	1																					2-1=1
7																								1					1-1=0
8									1	1																			2-1=1
9																								1					1-1=0
10											1	1																	2-1=1
11																								1					1-1=0
12													1	1															2-1=1
13																								1					1-1=0
14															1	1													2-1=1
15																								1					1-1=0
16																	1	1											2-1=1
17																								1					1-1=0
18																			1	1									2-1=1
19																								1					1-1=0
20																					1	1							2-1=1
21																								1					1-1=0
22																							1	1					2-1=1
23																								1					1-1=0
24																									1	1			2-1=1
25																								1					1-1=0
26																												1	1-1=0
27																													0-1=0
28																													0-1=0
SUM (E+1)																													11+1=12

Berdasarkan perhitungan diatas: Region = 12, Independent Path = 12

Jumlah Node (N) = 28

Jumlah Edge (E) = 38

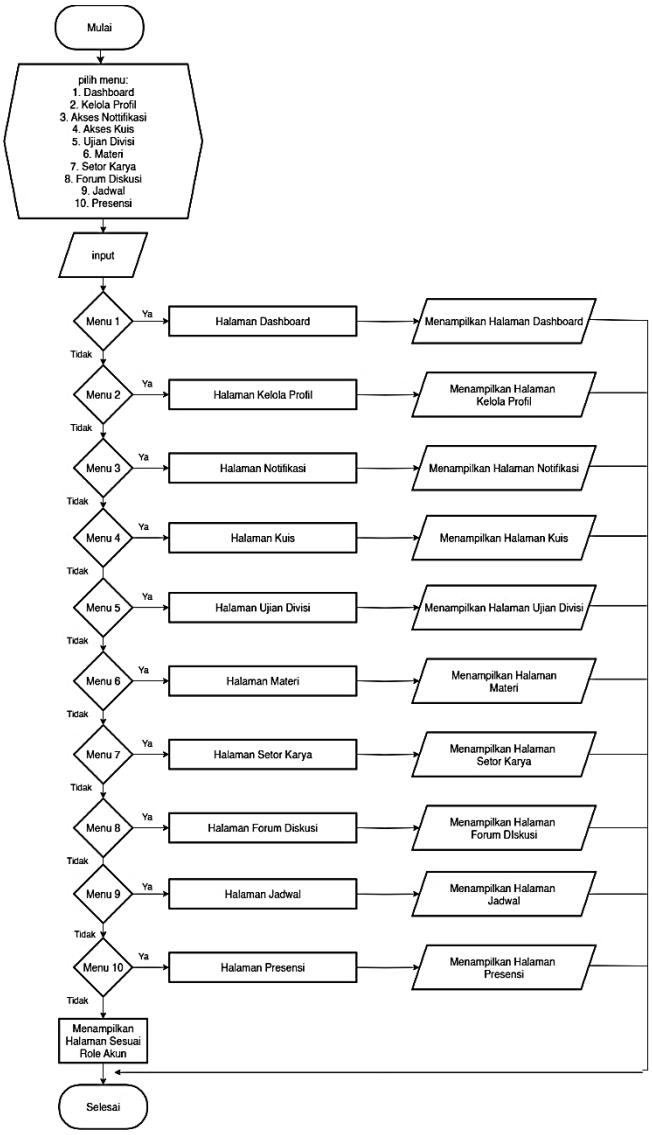
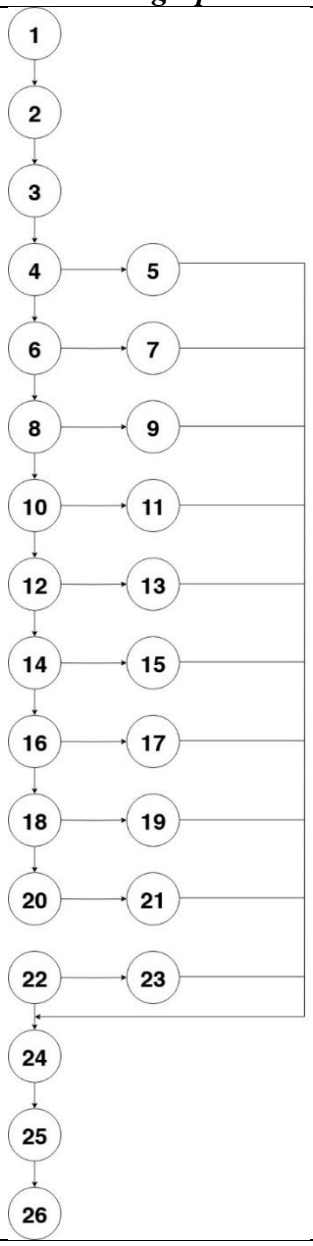
V(G) = E - N + 2 = 38 - 28 + 2 = 12

Region = 12

Independent Path (jumlah minimal) = 12

Jumlah jalur yang dicantumkan di bawah = 12

Tabel 4. 41 *White box* Anggota

Flowchart	***Flowgraph***
	

Independent Path Anggota:

1. *Independent Path 1-2-3-4-5-22-23-24-25-26*
2. *Independent Path 1-2-3-4-6-7-22-23-24-25-26*
3. *Independent Path 1-2-3-4-6-8-9-22-23-24-25-26*
4. *Independent Path 1-2-3-4-6-8-10-11-22-23-24-25-26*

5. *Independent Path 1-2-3-4-6-8-10-12-13-22-23-24-25-26*
6. *Independent Path 1-2-3-4-6-8-10-12-14-15-22-23-24-25-26*
7. *Independent Path 1-2-3-4-6-8-10-12-14-16-17-22-23-24-25-26*
8. *Independent Path 1-2-3-4-6-8-10-12-14-16-18-19-22-23-24-25-26*
9. *Independent Path 1-2-3-4-6-8-10-12-14-16-18-20-21-22-23-24-25-26*
10. *Independent Path 1-2-3-4-6-8-10-12-14-16-18-20-22-23-24-25-26 (jalur langsung tanpa memilih cabang 1–9; "menu 10")*

Grafik Matriks (*Graph Matrix*):

Tabel 4. 42 *Grafik Matriks white box* Anggota

Node	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	E-1
1		1																							1-1=0
2			1																						1-1=0
3				1																					1-1=0
4					1	1																			2-1=1
5																						1			1-1=0
6							1	1																	2-1=1
7																						1			1-1=0
8									1	1															2-1=1
9																						1			1-1=0
10											1	1													2-1=1
11																						1			1-1=0
12													1	1											2-1=1
13																						1			1-1=0
14															1	1									2-1=1
15																						1			1-1=0
16																	1	1							2-1=1
17																						1			1-1=0
18																			1	1					2-1=1
19																						1			1-1=0
20																					1	1			2-1=1
21																						1			1-1=0
22																							1	1	2-1=1
23																						1			1-1=0
24																									0-1=0
SUM (E+1)																									10+1=11

Berdasarkan perhitungan diatas: Region = 10, Independent Path = 10

***Cyclomatic complexity* = 10**, maka modul Anggota dinyatakan bebas dari kesalahan.

2. **Perhitungan Kompleksitas Siklomatis:**

Jumlah Node (N) = 26

Jumlah Edge (E) = 34

V(G) = E - N + 2 = 34 - 26 + 2 = 10

Region = 10

Independent Path = 10

Perhitungan Kompleksitas:

Tabel 4. 43 Tabel perhitungan Kompleksitas

Role	**Jumlah Edge (E)**	**Jumlah Node (N)**	**Kompleksitas Siklomatis (V(G))**	**Jumlah Independent Path**
Pembina	34	20	V(G) = E - N + 2 = 34 - 26 + 2 = 10	10
Pengurus	38	22	V(G) = E - N + 2 = 38 - 28 + 2 = 12	12
Anggota	34	24	V(G) = E - N + 2 = 34 - 26 + 2 = 10	10

Kesimpulan *White box testing*

Analisis Hasil:

Pembina: Memiliki 10 independent path dengan fokus pada manajemen administratif sistem seperti kelola user, monitoring forum, kelola system settings, kelola kurikulum, kelola laporan, kelola notifikasi, kelola profil, dan monitoring jadwal.

Pengurus: Memiliki 12 independent path dengan peran sebagai fasilitator **pembelajaran** yang mengelola materi, kuis, notifikasi, forum, karya/tugas, jadwal & presensi, system setting terbatas, dan profil.

Anggota: Memiliki 10 independent path dengan variasi aktivitas pembelajaran **seperti** akses materi & kurikulum, ikut kuis, submit karya, ikut forum, akses jadwal & presensi, serta lihat notifikasi & kelola profil.

Metrik Kualitas:

Tabel 4. 44 Tabel perhitungan Kompleksitas

Metrik	**Pembina (10 Path)**	**Pengurus (12 Path)**	**Anggota (10 Path)**
Kompleksitas	Sedang-Tinggi (10)	Tinggi (12)	Sedang-tinggi (10)
Maintainability	Baik	Baik	Sangat Baik
Testability	Baik	Baik	Sangat Baik

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian dan pengembangan yang telah dilakukan, penelitian ini telah berhasil merancang dan membangun aplikasi *M-Learning* berbasis *Progressive Web App (PWA)* untuk Komunitas IT Smansa Parepare yang menjawab kedua rumusan masalah penelitian. Aplikasi ini memiliki antarmuka responsif yang dapat berjalan optimal di berbagai perangkat mobile dengan ukuran layar yang berbeda, serta telah mengintegrasikan teknologi *PWA* dengan strategi caching efisien dan *Service Worker* untuk meningkatkan kecepatan loading aplikasi. Aplikasi yang dibangun menggunakan metodologi waterfall dengan teknologi *PHP Native* dan basis data *MySQL* ini memiliki fitur-fitur komprehensif meliputi autentikasi berbasis peran (Anggota, Pengurus, Pembina), manajemen kurikulum berdasarkan divisi IT (*Software, Hardware, Video Editing, Graphic Design, Networking*), sistem kuis, forum diskusi, fitur setor karya, presensi digital, jadwal kegiatan, ujian divisi, dan sistem notifikasi. Hasil pengujian *Black box* dan *White box* menunjukkan bahwa aplikasi berfungsi sesuai dengan spesifikasi yang dirancang, sehingga memberikan solusi digital untuk mengatasi masalah keterbatasan pembelajaran konvensional pada Komunitas IT Smansa Parepare.

B. Saran

Saran pengembangan Platform Pembelajaran KITS mencakup peningkatan interaktivitas melalui gamifikasi (poin, lencana, papan peringkat), kelas virtual dengan video konferensi, papan tulis interaktif, serta pembelajaran adaptif yang menyesuaikan materi sesuai gaya belajar pengguna. Disarankan pula penambahan fitur kolaboratif seperti proyek kelompok dan tinjauan sejawat. Integrasi teknologi kecerdasan buatan dan pembelajaran mesin diharapkan mampu menghadirkan chatbot pembelajaran 24 jam, sistem rekomendasi materi, penilaian otomatis, analitik prediktif untuk deteksi dini kesulitan belajar, serta analisis sentimen forum. Selain itu, perluasan platform meliputi aplikasi mobile native iOS/*Android*, integrasi LMS populer, Single Sign-On, dukungan multi-bahasa, API publik untuk pengembangan pihak ketiga.

DAFTAR PUSTAKA

Al Fatih, M. T. (2024) Memahami aplikasi: Pengertian, fungsi, dan jenis-jenisnya. Telkom University https://dif.telkomuniversity.ac.id/memahami-apa-itu-aplikasi-dan-fungsinya-di-era-digital/

Biznet Gio. (2023). Apa itu PHP? Pengertian, sejarah, dan fungsinya. Diakses 28 April 2025, dari https://www.biznetgio.com/news/apa-itu-php/

Cambridge Dictionary. (2025). Website. Diakses 25 April 2025, dari https://dictionary.cambridge.org/dictionary/english/website

Dicoding. (2021). Apa itu UML? Pengertian dan contohnya. Dicoding Blog. Diakses 28 April 2025, dari https://www.dicoding.com/blog/apa-itu-uml/

Ferdiansyah, M. (2018). Perancangan sistem informasi persediaan barang menggunakan metode FEFO studi kasus: PT Nutrifood Indonesia Bandar Lampung. Techno Preneurship Jurnal Ilmiah Politeknik Piksi Input Serang, 5(1).

Hostinger. (2025). Dynamic website: What it is, how it works + building one. Diakses 25 April 2025, dari https://www.hostinger.com/tutorials/dynamic-website

Humam, F. R., Maulani, I. E., Zuhriyah, N. F., & Marlina, N. S. (2023). Pengembangan aplikasi *Mobile learning* berbasis *Android* untuk meningkatkan keterampilan pemrograman pada mahasiswa teknik informatika. Jurnal Sosial dan Teknologi, 3(2), 108–113. doi:10.36418/jurnalsostech.v3i2.638

Laragon. (2025). Laragon portable, isolated, fast & powerful universal development environment. Diakses 27 April 2025, dari https://laragon.org/index.html

Madcoms. (2020). Kupas tuntas MySQL database. Yogyakarta: Andi Publisher.

Merdeka.com. (2022). Kepanjangan PHP adalah Hypertext Preprocessor dalam bahasa pemrograman. Diakses 28 April 2025, dari https://www.merdeka.com/trending/kepanjangan-php-adalah-hypertext-preprocessor-dalam-bahasa-pemrograman-ini-contohnya-kln.html

Morgan, J. D. (2018). Applying 1970 waterfall lessons learned within today's agile development process. PM World Journal, 7(7), 1–19.

Novendri, M. S., Saputra, A., Firman, C. E., Manajemen Informatika, J., Dumai, A., Informatika, J. T., ... Kode, D. (2019). Aplikasi inventaris barang pada

MTs Nurul Islam Dumai menggunakan PHP dan MySQL. [Detail penerbit/jurnal tidak tersedia].

Nugroho, B. (2021). Pemrograman web dinamis dengan PHP dan MySQL. Bandung: Informatika.

Object Management Group. (1997). Unified Modeling Language specification version 1.1. Diakses 28 April 2025, dari https://www.omg.org/spec/UML/1.1

Parsinta. (2023). Keuntungan memakai Laragon dalam pengembangan web. Diakses 27 April 2025, dari https://parsinta.com/articles/keuntungan-memakai-laragon-73fwcz

Purnamasari, A. W., Wiranatha, A. A. K. A. C., & Mandenni, N. M. I. M. (2014). Aplikasi *Mobile learning* (*M-Learning*) pada platform *Android*. MERPATI: Jurnal Ilmiah Menara Penelitian Akademika Teknologi Informasi, 2(2), 167–178. https://ojs.unud.ac.id/index.php/merpati/article/view/17883

Rifai, A., Sulton, & Sulthoni. (2021). Perbandingan metode pengembangan perangkat lunak (waterfall, prototype, RAD). Jurnal Bangkit Indonesia, 10(1), 6–12. doi:10.21009/jktp.v3i1.15032

Royce, W. W. (1970). Managing the development of large software systems. Proceedings of IEEE WESCON, 26, 1–9.

Salamun, S. (2017). Sistem monitoring nilai siswa berbasis *Android*. Rabit: Jurnal Teknologi dan Sistem Informasi Univrab, 2(2), 210–219. doi:10.36341/rabit.v2i2.221

Sandy, F. (2018). Sistem informasi pengelolaan data siswa di SMK Negeri 3 Makassar berbasis web [Skripsi Sarjana, Universitas Islam Negeri Alauddin Makassar]. Repositori UIN Alauddin Makassar. http://repositori.uin-alauddin.ac.id/view/creators/Sandy=3AFaradhillah=3A=3A.html

Saputra, D., & Dewi, I. K. (2019). Konsep dasar database menggunakan MySQL. Jakarta: Elex Media Komputindo.

Scaler. (2024). How to make dynamic website in PHP? Diakses 28 April 2025, dari https://www.scaler.com/topics/php-tutorial/dynamic-website-in-php/

Sekawan Media. (2025). Apa itu UML? Ini pengertian, fungsi, dan contohnya. Diakses 28 April 2025, dari https://www.sekawanmedia.co.id/blog/apa-itu-uml/

Setiawan, A., & Pratama, R. (2021). Penerapan *Progressive Web Apps* untuk optimasi aplikasi web. Bandung: Informatika.

Shopify. (2024). What is a website? Definition, components, and types. Diakses 25 April 2025, dari https://www.shopify.com/blog/what-is-a-website

SiteGround. (2025). What is a static website? Advantages & disadvantages. Diakses 25 April 2025, dari https://www.siteground.com/kb/what-is-a-static-website/

Specbee. (2025). How to make interactive websites and why you need one? Diakses 25 April 2025, dari https://www.specbee.com/blogs/how-make-interactive-websites-and-why-you-need-one

Staditek.id. (2023). Apa itu PHP: Pengertian, sejarah, fungsi dan sintak dasarnya. Diakses 28 April 2025, dari https://www.staditek.id/insight/php-adalah/

Suhartanto, M. (2017). Pembuatan website Sekolah Menengah Pertama Negeri 3 Delanggu dengan menggunakan PHP dan MySQL. Speed Sentra Penelitian Engineering dan Edukasi, 4(1).

Syahputra, A., & Yudianto, A. (2018). Panduan praktis membangun *Progressive Web Apps*. Jakarta: Elex Media Komputindo.

Uda, S. K., Prasetyo, D., Dopo, E. S. E. B., Uda, S. A. K. A., & Basrowi. (2024). Development of *Mobile learning* application system for environmental science material (SARITHA-Apps). International Journal of Information and Education Technology, 14(3), 452–463. doi:10.18178/ijiet.2024.14.3.2066

Utomo, S. D. (2020). Pemrograman aplikasi web progresif: Mengembangkan aplikasi web berbasis teknologi modern. Yogyakarta: Andi Publisher.