BAB I

PENDAHULUAN

A. Latar Belakang

Penasehat Akademik (PA) merupakan salah satu elemen penting dalam proses perkuliahan di perguruan tinggi yang berperan dalam memberikan arahan, bimbingan, dan pendampingan kepada mahasiswa terkait kegiatan akademik. Fungsi utama PA adalah membantu mahasiswa dalam merencanakan studi, memilih mata kuliah, menyelesaikan permasalahan akademik, serta memberikan solusi terhadap kendala yang dihadapi mahasiswa selama masa perkuliahan. Dengan adanya peran PA, mahasiswa diharapkan dapat menyelesaikan studi tepat waktu dengan hasil yang optimal.

Namun, dalam praktiknya, proses bimbingan akademik sering menghadapi beberapa permasalahan. Salah satu kendala yang sering muncul adalah keterbatasan waktu dan ruang untuk melakukan pertemuan tatap muka antara mahasiswa dan PA. Padatnya jadwal perkuliahan baik dari sisi mahasiswa maupun dosen sering menyebabkan konsultasi tidak berjalan efektif. Selain itu, kurangnya media komunikasi yang terintegrasi juga menjadi hambatan, di mana sebagian besar komunikasi hanya dilakukan melalui media sosial yang tidak terstruktur dan tidak terdokumentasi dengan baik.

Selain itu, proses administrasi seperti pencatatan hasil bimbingan, pengajuan Kartu Rencana Studi (KRS), atau pengecekan status akademik mahasiswa umumnya masih dilakukan secara manual. Hal ini tidak hanya memakan

waktu tetapi juga rentan terhadap kesalahan, baik dalam pencatatan maupun pengarsipan data. Kondisi ini bertolak belakang dengan tuntutan era digital saat ini yang menuntut kecepatan, kemudahan, dan akurasi dalam penyampaian informasi.

Seiring perkembangan teknologi informasi, khususnya teknologi berbasis mobile, pemanfaatan aplikasi *Android* dalam menunjang aktivitas akademik menjadi salah satu solusi yang potensial. Aplikasi mobile memungkinkan interaksi yang lebih fleksibel dan cepat antara mahasiswa dan PA tanpa dibatasi ruang dan waktu. Dengan memanfaatkan teknologi ini, proses bimbingan akademik dapat dilakukan secara *online*, terjadwal, terdokumentasi, dan terintegrasi dengan data akademik mahasiswa.

Berdasarkan uraian tersebut, diperlukan sebuah Aplikasi Penasehat Akademik (PA) Fakultas Teknik berbasis *Android* yang dapat memberikan layanan bimbingan akademik secara digital. Aplikasi ini dirancang untuk mempermudah komunikasi antara mahasiswa dan PA, mengelola jadwal konsultasi, menampilkan informasi akademik mahasiswa, serta menyimpan riwayat bimbingan dalam satu platform. Dengan adanya aplikasi ini, diharapkan proses bimbingan menjadi lebih efektif, efisien, dan dapat meningkatkan kualitas layanan akademik di Fakultas Teknik.

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka penelitian ini akan mengambil sebuah topik berjudul “Aplikasi Penasehat Akademik (PA) Fakultas Teknik Berbasis *Android*”.

B. Rumusan Masalah

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

1. Bagaimana merancang aplikasi Penasehat Akademik berbasis *Android* untuk Fakultas Teknik?
2. Bagaimana aplikasi ini dapat membantu proses bimbingan akademik secara efektif?
3. Bagaimana tingkat kepuasan pengguna terhadap aplikasi yang dikembangkan?

C. Batasan Masalah

Agar penelitian ini lebih terarah dan sesuai dengan tujuan yang ingin dicapai, maka ditetapkan beberapa batasan masalah sebagai berikut:

1. Aplikasi yang dikembangkan berbasis *Android*, sehingga hanya dapat dijalankan pada perangkat dengan sistem operasi *Android*.
2. Pengguna aplikasi dibatasi hanya untuk dosen Penasehat Akademik dan mahasiswa Fakultas Teknik.
3. Fitur yang tersedia pada aplikasi meliputi:
    a. *Login* pengguna (dosen PA dan mahasiswa).
    b. Jadwal konsultasi.
    c. Komunikasi berupa chat sederhana.
    d. Riwayat bimbingan.
    e. Informasi data akademik (jumlah SKS dan IPK sementara).

4. Aplikasi memerlukan koneksi internet untuk beroperasi.
5. Pengujian dilakukan dengan metode Black Box Testing dan pengukuran usability dengan System Usability Scale (SUS).
6. Keamanan data hanya menggunakan autentikasi *login* (*user*name dan password).

D. Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Merancang dan membangun aplikasi Penasehat Akademik (PA) berbasis *Android* untuk Fakultas Teknik.
2. Mempermudah komunikasi dan konsultasi antara mahasiswa dengan PA secara daring.
3. Menguji tingkat usability dan kepuasan pengguna terhadap aplikasi yang dikembangkan.

E. Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah:

1. Bagi Mahasiswa: Mempermudah akses bimbingan akademik, menghemat waktu, dan memantau perkembangan studi.
2. Bagi Dosen PA: Memudahkan pemantauan mahasiswa, jadwal konsultasi, dan dokumentasi bimbingan.
3. Bagi Fakultas: Menjadi inovasi digital dalam pelayanan akademik yang mendukung transformasi digital pendidikan tinggi.

BAB II

TINJAUAN PUSTAKA

A. Penelitian Terdahulu

Penelitian relevan berisikan penelitian yang terdahulu kemudian dicari bahan penelitian yang ada dan relevan dengan penelitian ini karena sangat berguna untuk mendukung kajian teori yang dikemukakan, sehingga dapat digunakan sebagai landasan pada penyusunan kerangka pikir. Beberapa hasil penelitian yang relevan dengan penelitian ini adalah penelitian yang dilakukan oleh:

1. Choirul Anna Nur Afifah dan Salsabila Romadona (2022) dalam penelitianya yang berjudul "Pengembangan Aplikasi Sehatgram Berbasis *Smartphone* Dalam Mengatasi Dan Mencegah Obesitas Pada Remaja" Masa remaja menjadi salah satu periode penting dalam siklus perkembangan individu. Permasalahan gizi yang dihadapi pada masa remaja salah satunya adalah obesitas. Data Riskesdas tahun 2018 menunjukkan, remaja gemuk berusia 13-15 tahun prevalensinya sebesar 11,2%. Remaja gemuk yang berusia 16-18 tahun prevalensinya yaitu 9,5%. Sejalan dengan perkembangan teknologi yang mengarah pada kehidupan yang lebih mudah, hal tersebut berpotensi dalam meningkatkan kasus obesitas khususnya pada remaja. Penelitian ini bertujuan mengembangkan aplikasi *Sehatgram* sebagai solusi untuk mengatasi dan mencegah obesitas pada remaja dan mengevaluasi kelayakan aplikasi *Sehatgram*. Desain penelitian ini menggunakan metode Rapid Application Development (RAD) yang terdiri dari fase requirements planning, RAD design workshop, dan implementation. Penilaian kelayakan

aplikasi *Sehatgram* dilakukan oleh ahli materi dan ahli media. Penilaian kelayakan materi mendapat nilai rata-rata sebesar 4,33 tergolong "sangat layak". Sedangkan penilaian kelayakan media mendapat nilai rata-rata sebesar 4,08 tergolong "layak". Perlu adanya penyempurnaan lebih lanjut terhadap fitur pada aplikasi Sehatgram yang belum aktif serta implementasi Sehatgram pada remaja.

2. Rita Astari (2018) dalam penelitianya yang berjudul "Rancang Bangun Aplikasi Pengelolaan Hidup Sehat Berbasis *Android*" Pola hidup sehat adalah gaya hidup yang lebih fokus pada kesehatan, baik itu dalam perilaku, makanan dan sebagainya yang mengarah pada hidup lebih sehat baik jasmani maupun rohani. Manfaat dari menerapkan pola hidup sehat di antaranya adalah membuat pikiran lebih positif, membuat tubuh lebih semangat dalam bekerja, mencegah berbagai macam penyakit dan masih banyak lagi manfaat yang dapat diperoleh dari menjaga pola hidup sehat secara benar. Namun masyarakat di Indonesia masih jauh dari penerapan pola hidup sehat, ini berdasarkan berbagai survei yang ada. Kondisi tubuh masyarakat di Indonesia juga sangat beragam, sehingga pola hidup sehat setiap masyarakatpun berbeda sesuai dengan riwayatnya. Patokan dalam penerapan pola hidup sehat utamanya dalam urusan makanan dan olahraga adalah sesuai dengan kadar kolesterol, asam urat dan gula darah dalam tubuh. Tujuan dari penelitian ini adalah untuk merancang suatu aplikasi pengelolaan hidup sehat untuk perangkat mobile khususnya *Smartphone* yang berbasis *Android*. Metode penelitian yang digunakan adalah penelitian deskriptif kualitatif yang bertujuan untuk memahami fenomena-fenomena sosial.

Penelitian ini menggunakan metode pengumpulan data observasi, studi pustaka, kuesioner dan wawancara. Metode perancangan yang digunakan adalah *waterfall* dan Teknik pengujian yang digunakan adalah Black Box. Pada aplikasi ini pengguna harus memasukkan riwayatnya seperti kadar kolesterol, asam urat dan gula darah untuk mendapatkan hasil pola hidup sehat yang sesuai dengan pengguna yaitu jenis makanan yang sebaiknya dikonsumsi dan sebaiknya dihindari, sekaligus jenis olahraga yang sesuai dengan pengguna. Berdasarkan hasil uji coba terbukti aplikasi pengelolaan hidup sehat dapat memberikan informasi dan kemudahan pada beberapa orang dalam mengatur pola hidup sehat.

B. Tinjauan pustaka

1. Penasehat Akademik (PA)

Penasehat Akademik (PA) adalah dosen yang ditugaskan oleh perguruan tinggi untuk memberikan bimbingan dan arahan kepada mahasiswa terkait pelaksanaan kegiatan akademik selama masa studi. PA berperan penting dalam membantu mahasiswa memahami sistem perkuliahan, memilih mata kuliah, menyusun Kartu Rencana Studi (KRS), memantau perkembangan akademik, serta memberikan solusi terhadap masalah akademik yang dihadapi mahasiswa.

Menurut Peraturan Menteri Pendidikan dan Kebudayaan RI tentang penyelenggaraan pendidikan tinggi, Penasehat Akademik bertugas memberikan layanan akademik kepada mahasiswa secara berkesinambungan agar mereka dapat menyelesaikan studinya tepat waktu dan sesuai dengan ketentuan kurikulum

2. Fungsi dan Peran Penasehat Akademik

Penasehat Akademik memiliki beberapa fungsi dan peran, antara lain:

a. **Pembimbing Akademik**

1) Membantu mahasiswa merencanakan studi sesuai dengan kurikulum dan beban SKS.
2) Memberikan arahan dalam pemilihan mata kuliah agar sesuai dengan kemampuan dan minat mahasiswa.

b. **Pemantau Perkembangan Akademik**

1) Memantau Indeks Prestasi (IP) mahasiswa setiap semester.
2) Memberikan solusi jika terjadi masalah seperti penurunan prestasi, cuti kuliah, atau pengulangan mata kuliah.

c. **Konsultan Akademik**

1) Menjadi tempat konsultasi terkait masalah akademik, seperti proses perkuliahan, skripsi, dan jadwal kuliah.

d. **Mediator dan Motivator**

1) Memberikan motivasi kepada mahasiswa agar dapat menyelesaikan studinya tepat waktu.
2) Menjadi penghubung antara mahasiswa dengan pihak fakultas jika ada kendala administrasi atau akademik.

3. Tugas dan Tanggung Jawab Penasehat Akademik

Tugas utama PA antara lain:

a. Memberikan arahan dalam penyusunan Kartu Rencana Studi (KRS) setiap semester.

b. Menyetujui pengisian KRS mahasiswa secara *online* maupun manual.
c. Melakukan bimbingan jika mahasiswa mengalami kendala akademik.
d. Memberikan pertimbangan dan rekomendasi jika mahasiswa ingin mengajukan cuti kuliah, pengunduran diri, atau pindah program studi.
e. Mendokumentasikan hasil bimbingan akademik.

4. Mekanisme Bimbingan Penasehat Akademik

Proses bimbingan biasanya dilakukan melalui:

a. Tatap muka langsung saat pengisian KRS atau konsultasi rutin.
b. Melalui media komunikasi seperti email, WhatsApp, atau platform daring lainnya.
c. Pengisian form konsultasi yang biasanya disediakan oleh fakultas.

Namun, pada praktiknya, mekanisme ini sering terkendala oleh:

a. Padatnya jadwal dosen dan mahasiswa.
b. Proses dokumentasi yang masih manual.
c. Tidak adanya platform terintegrasi untuk komunikasi dan arsip bimbingan.

5. Kendala dalam Pelaksanaan Bimbingan PA

Beberapa kendala yang sering dihadapi dalam pelaksanaan peran PA antara lain:

a. Keterbatasan waktu sehingga konsultasi tidak berjalan efektif.

b. Tidak ada media yang terdokumentasi sehingga riwayat bimbingan sulit dilacak.
c. Kurangnya interaksi antara mahasiswa dan PA karena kesibukan masing-masing pihak.

6. Urgensi Digitalisasi Peran PA

Dengan berkembangnya teknologi, khususnya aplikasi mobile berbasis *Android*, peran PA dapat dioptimalkan melalui platform digital. Digitalisasi ini bermanfaat untuk:

a. Mempermudah komunikasi antara mahasiswa dan PA kapan saja dan di mana saja.
b. Menyediakan fitur jadwal konsultasi, chat, dan monitoring akademik.
c. Menyimpan riwayat bimbingan secara terstruktur dan aman.

Oleh karena itu, pengembangan Aplikasi Penasehat Akademik (PA) berbasis *Android* merupakan solusi inovatif untuk mendukung transformasi digital di bidang pendidikan, khususnya pada Fakultas Teknik.

7. *Smartphone*

Andi widi (2021) Ponsel cerdas (bahasa inggris: *Smartphone*) adalah telepon genggam yang mempunyai kemampuan dengan penggunaan dan fungsi yang menyerupai komputer. Belum ada standar pabrik yang menentukan arti ponsel cerdas. Bagi beberapa orang, ponsel cerdas merupakan telepon yang bekerja menggunakan seluruh perangkat lunak sistem operasi yang menyediakan hubungan standar dan mendasar bagi pengembang aplikasi. Bagi yang lainnya, ponsel cerdas

hanyalah merupakan sebuah telepon yang menyajikan fitur canggih seperti surel (surat *elektronik*), internet dan kemampuan membaca buku *electronik* (*e-book*) atau terdapat papan ketik (baik sebagaimana jadi maupun dihubung keluar) dan penyambung VGA. Dengan kata lain, ponsel cerdas merupakan komputer kecil yang mempunyai kemampuan sebuah telepon.

Singkatnya, *Smartphone* adalah perangkat yang memungkinkan anda melakukan panggilan telepon, sekaligus memiliki fitur yang dimasa lalu hanya bisa ditemukan pada personal *digital assistant* (PDA) atau komputer- seperti kemampuan untuk mengirim dan menerima *e-mail* dan editing dokumen.

Ponsel terutama digunakan untuk menelpon, sementara PDA digunakan sebagai semacam asistensi digital pribadi atau digital organizer. PDA dapat menyimpan info kontak dan agenda harian serta bisa disinkronisasi dengan komputer. Perkembangan ini lantas mendorong terciptanya apa yang kini dikenal sebagai *Smartphone*.

8. *Bahasa Pemrograman C#*

C# (baca: C *sharp* atau *see sharp*) sering dianggap sebagai bahasa penerus C++ atau versi canggih dari C++, karena ada anggapan bahwa tanda # adalah perpaduan dari 4 buah tanda tambah yang disusun sedemikian rupa sehingga membentuk tanda pagar. Akan tetapi, terlepas dari benar tidaknya anggapan tersebut, C# adalah sebuah bahasa pemrograman yang sangat menjanjikan. C# adalah sebuah bahasa pemrograman yang beriorentasi pada objek yang

dikembangkan oleh *Microsoft* dan menjadi salah satu bahasa pemrograman yang mendukung. *Net programming* melalui *Visual Studio*. Andi widi (2021)

C# didasarkan pada bahasa pemrograman C++, C# juga memiliki kemiripan dengan beberapa bahasa pemrograman seperti *Visual Basic, Java, Delphi*, dan tentu saja C++. C# memiliki kemudahan *syntax* (cara penulisan) seperti *Visual Basic*, dan tentu saja ketangguhan seperti *Java* dan C++. Kemiripan-kemiripan ini tentunya memudahkan programmer dari berbagai latar belakang bahasa pemrogrman tidak perlu waktu yang lama untuk menguasainya, karena bagaimanapun juga C# lebih sederhana dibandingkan bahasa-bahasa pemrograman seperti C++ dan *Java*.

C# didesain oleh program *designer dari microsoft, Anders Hajlsberg*. Sebelum bekerja pada *Microsoft*, Anders bekerja di *Borland*, tempat dia menulis *Pascal compiler*. Sebelum mengembangkan C# *Anders* pernah mengembangkan J++ untuk *Microsoft*. Setelah itu Anders mengembangkan C# dan *Common Language Runtime* yang merupakan mesin *Virtual dan Runtime library* yang merupakan salah satu tiang utama teknologi. *Net. Anders* mengetahui berbagai macam kekurangan pada bahasa C++, *Delphi, Java, dan Smaltalk*, karena itu Anders menciptakan bahasa C# yang lebih tangguh. Hal ini juga menjelaskan mengapa C# memiliki kemiripan dengan beberapa bahasa tersebut.

C# bergantung pada CLR yang juga merupakan sumber library bagi program. Semua program C# memerlukan CLR (berarti juga memerlukan .*Net Framework*) untuk dapat dijalankan. Sama halnya dengan Visual Basic 6 yang memerlukan *runtime library* tertentu untuk dapat dijalankan. Bahasa C# dapat

digunakan untuk menciptakan aplikasi *windows, console, class* yang dapat digunakan kembali, dan aplikasi *web*. Andi widi (2021)

Ada beberapa alasan kuat yang mendasari pemilihan bahasa C# untuk mengembangkan aplikasi-aplikasi, yaitu:

a. C# benar-benar beriorantasi objek

   C# adalah bahasa yang benar-benar objek *oriented*. Ini dapat dilihat dari kemampuan C# dalam pembentukan objek, *class*, melakukan *encapsulation, inheritance, dan polymorphsm* dengan mudah.

b. C# sangat sederhana

   Bahasa C# bersifat sederhana karena didasarkan pada bahasa C dan C++ bahkan bahasa Java. Tetapi C# lebih sederhana dari bahasa-bahasa tersebut karena C# dibuat dengan menghilangkan kelemahan-kelemahan dari bahasa-bahasa yang mendasarinya.

c. Mampu membuat berbagai aplikasi

   Dengan C# kita dapat membuat berbagai macam aplikasi, mulai dari aplikasi *console*, pengolahan kata, *form web*, dan lain-lain.

d. Efisien

   Bahasa C# merupakan bahasa yang mengandalkan library yang sangat lengkap, karena itu bahasa C# hanya memiliki sedikit *keywords*. Jadi para pengembang dapat mengingatnya dan memahami kegunaanya dengan baik.

e. C# bersifat modular

   Bahasa C# sangat modular, tiap *class* disimpan dalam *namespace* yang dapat dimanfaatkan kembali oleh program lain yang membutuhkannya.

9. *Android* (SDK)

Android SDK (*Software Development Kit*) adalah tools API (*Aplication Programming Interface*) yang diperlukan untuk memulai pengembangan suatu aplikasi pada *platform android* menggunakan bahasa pemrograman java. Andi widi (2021)

Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang direkase oleh *google*. Saat ini disediakan *Android* SDK sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada *platfrom* aplikasi netral, *Android* memberi anda kesempatan untuk membuat aplikasi yang kita butuhkan yang bukan aplikasi bawaan *Handphone/Smartphone*. Beberapa fitur *Android* adalah :

a. *Framewor*k aplikasi yang mendukung penggantian komponen dan *reusable*
b. *Dalvik Virtual Machine* dioptimalkan untuk perangkat *mobile*
c. *Integrated browser* berdasarkan *engine open source Webkit*
d. *Grafis* yang dioptimalkan dan didukung oleh *libraries grafis* 2D, *grafis* 3D berdasarkan spsifikasi *opengl ES* 1,0
e. SQLite untuk penyimpanan data/*database*
f. Media *suport* yang mendukung audio, video, dan gambar
g. *Bluetooth, EDGE,3G,WiFi (tergantung hardware*)
h. Kamera, GPS,kompas dan *accelerometer* (tergantung hardware)

Lingkungan *Development* yang lengkap dan kaya termasuk perangkat *emulator, tools* untuk *debugging*, profil dan kinerja memori dan *plugin* untuk IDE *eclipse*.

10. JDK (*Java Development Kit*)

Java Development Kit merupakan perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode java menjadi *bytecode* yang dapat dimengerti dan dapat dijalankan oleh *Java Runtime Environment*.

Java Development Kit wajib terinstall pada komputer yang akan melakukan proses pembuatan aplikasi berbasis java. Namun *Java Development Kit* tidak wajib terinstall dikomputer yang akan menjalankan aplikasi yang dibangun menggunakan Java.

Seperti namanya, JDK adalah semacam kotak peralatan (kit) yang digunakan untuk *Development.* JDK ini berguna saat anda menulis *code* program. Seperti halnya JRE, JDK juga memiliki JVM di dalamnya.

a. Java yaitu program untuk meng-*compile* kode sumber
b. Java, yaitu program untuk meluncurkan aplikasi.

11. UML (*Unified Modelling Language*)

UML (*Unified Modelling Language*) menurut Adi nugroho (2011:6), "*Unified Modelling Language* adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berpradigma beriorentasi objek",

Metode *Unified Modelling Language* (UML) menggunakan tiga bangunan dasar untuk mendeskripsikan sistem atau perangkat lunak yang akan dikembangkan, yaitu :

a. Sesuatu (*things*)

Ada empat tings dalam *Unified Modelling Language* (UML):

1) *Structural things,* bagian yang relatif statis dapat berupa elemen yang bersifat fisik maupun konseptual.
2) *Behavorial things,* bagian dinamis biasanya merupakan kata kerja dari model UML yang mencerminkan perilaku sepanjang waktu
3) *Grouping things,* bagian yang pengorganisasian dalam UML. Dalam penggambaran model UML yang rumit diperlukan penggambaran paket yang menyederhanakan model. Paket-paket ini kemudian dapat didekomposisi lebih lanjut. Paket berguna bagi pengelompokan sesuatu, misalnya model-model serta subsistem.
4) *An notational things*, merupakan bagian yang memperjelas model UML. Dapat berisi komentar yang menjelaskan fungsi serta ciri-ciri tiap elemen dalam model UML.

b. Relasi (*relationship*)

Ada empat *relationship* (hubungan) dalam *Unified Modelling Language* (UML):

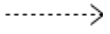
1) Ketergantungan (*dependency*) adalah hubungan dimana perubahan yang terjadi pada suatu elemen independent akan mempengeruhi elemen yang bergantung padanya.

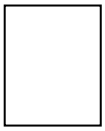
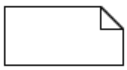
2) Asosiasi adalah apa dan bagaimana yang menghubungkan antara objek satu dengan yang lainnya. Suatu bentuk asosiasi adalah agregasi yang menampilkan hubungan suatu objek dengan bagian-bagiannya.
3) Generalisasi adalah hubungan dimana objek anak berbagai perilaku dan struktur data dari objek yang ada diatasnya (objek induk). Arah dari objek induk ke objek anak dinamakan spesialisai sedangkan arah sebaliknya dinamakan generalisasi.
4) Realisasi adalah operasi yang benar-benar dilakukan oleh suatu objek.

Diagram *Use Case* menyajikan intraksi antara *Use Case* dan aktor, dimana aktor dapat berupa orang, peralatan, atau sistem lain yang berintraksi dengan sistem yang sedang di bangun.

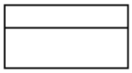
Adapun simbol-simbol *Use Case* diagram antara lain :

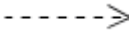
Gambar 2. 1 Simbol Use Case Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
3		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
4		*Include*	Menspesifikasikan bahwa *Use Case* sumber secara *eksplisit*.

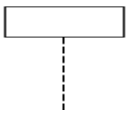
NO	GAMBAR	NAMA	KETERANGAN
5		*Extend*	Menspesifikasikan bahwa *Use Case* target memperluas perilaku dari *Use Case* sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Gambar 2. 2 Simbol Class Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*Generalization*	Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
2		*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.

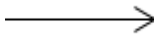
4		*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
5		*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.
6		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya

Gambar 2. 3 Simbol *Sequence diagram*

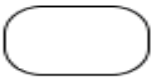
NO	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.
2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

Gambar 2. 4 Simbol StateChart Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*State*	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.

NO	GAMBAR	NAMA	KETERANGAN
2		*Initial Pseudo State*	Bagaimana objek dibentuk atau diawali
3		*Final State*	Bagaimana objek dibentuk dan dihancurkan
4		*Transition*	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		*Node*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Gambar 2. 5 Simbol Actifity Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*Actifity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Actifity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

BAB III

METODOLOGI PENELITIAN

A. Waktu Dan Lokasi Penelitian

Pelaksanaan penelitian dilakukan pada bulan Mei 2025 dan lokasi Fakultas Teknik Universitas Muhammadiyah Parepare di Kota Parepare, Sulawesi Selatan.

B. Jenis Penelitian

Penulis dalam hal ini menggunakan metode penelitian Deskriptif yaitu kegiatan yang dilakukan dengan cara mengumpulkan data secara langsung dari objek penelitian melalui wawancara, pengamatan langsung dan pengumpulan dokumen.

C. Metode Pengumpulan Data

Di penelitian ini, penulis memperoleh data dengan metode sebagai berikut:

1. Observasi

Dilakukan dengan mengadakan penelitian langsung dengan kantor yang terkait untuk mengumpulkan informasi dan data yang dibutuhkan dalam menunjang permasalahan.

2. Wawancara

Wawancara adalah bentuk komunikasi langsung antara peneliti dengan responden. Komunikasi ini berlangsung dalam bentuk tanya jawab dalam hubungan tata muka guna memperoleh data langsung

3. Studi Pustaka

Studi pustaka adalah suatu metode pengumpulan data dengan menggunakan buku-buku, laporan-laporan atau bacaan lain sebagai bahan referensi dalam penulisan laporan dan pembuatan sistem.

D. Alat Dan Bahan Penelitian

Dalam melakukan penelitian, penulis memerlukan alat dan bahan penelitian yang mendukung kegiatan tersebut. Alat dan bahan yang diperlukan antara lain:

1. Perangkat keras

   Perangkat keras yang digunakan dalam pembuatan Aplikasi Penasehat Akademik (PA) Fakultas Teknik Berbasis *Android* adalah sebuah laptop dengan spesifikasi minimal yang dapat dilihat pada tabel 4 berikut :

Gambar 3. 1 Spesifikasi Minimal Perangkat Keras

Jenis	Spesifikasi
Laptop	Asus
Processor	Core-i3-3217U 1,8 GHz
Memory	4 GB

Hardisk	500 GB

2. Perangkat Lunak

   Perangkat lunak yang digunakan untuk membangun aplikasi ini dapat dilihat pada Tabel 5 berikut :

Gambar 3. 2 Spesifikasi Minimal Perangkat Lunak

Jenis	**Spesifikasi**
Sistem Operasi	Windows 11
Bahasa Pemrograman	*Android* Studio
Database	MySql
Tools	Notepad++, Netbeans, XAMPP

E. Tahapan penelitian

Tahapan penelitian yang dimaksud dalam penelitian ini ada beberapa tahapan diantaraya yaitu persiapan penelitian, pengumpulan data, analisis data, perancangan , pengujian dan implementasi. Adapun uraian dari tahapan tersebut adalah sebagai berikut :

1. Persiapan Penelitian

   Tahap persiapan penelitian merupakan fondasi bagi seluruh proses penelitian. Pada tahap ini, peneliti akan merumuskan masalah yang akan diteliti secara jelas dan spesifik. Kemudian, peneliti akan melakukan studi literatur untuk menggali pengetahuan yang sudah ada terkait dengan permasalahan yang sedang di teliti

2. Studi Literatur

   Pada tahapan ini peneliti melakukan apa yang disebut dengan kajian pustaka, yaitu mempelajari buku-buku referensi dan hasil penelitian sejenis sebelumnya yang pernah dilakukan oleh orang lain. Tujuannya adalah untuk mendapatkan landasan teori mengenai masalah yang akan diteliti. Teori merupakan pijakan bagi peneliti untuk memahami persoalan terkait persoalan yang diteliti dengan benar sesuai dengan kerangka berpikir ilmiah.

3. Analisis

   Pada tahap analisis, peneliti melakukan analisa terhadap sistem yang diterapkan sekarang kemudian merumuskan masalah yang menjadi pokok penelitian sehingga dapat dibuat alternatif pemecahan masalah.

4. Perancangan

   Peneliti kemudian merancang *aplikasi* yang ingin dibuat berdasarkan alternatif pemecahan masalah.

5. Pengujian

   Setelah melakukan perancangan, peneliti kemudian menguji hasil perancangan yang telah dibuat. Jika hasil perancangan terdapat kekurangan atau kelemahan maka kembali ke tahap analasis.

6. Implementasi

   Setelah pada pengujian tidak terdapat kekurangan maka *aplikasi* siap digunakan.

BAB IV

PEMBAHASAN DAN HASIL

A. Perancangan sistem

Perancangan sistem aplikasi manajemen akademik terpadu mencakup beberapa tahapan utama, mulai dari analisis kebutuhan, perancangan sistem, implementasi, pengujian, hingga pemeliharaan.

Pada tahap analisis kebutuhan, ditemukan bahwa proses manajemen akademik, seperti pengelolaan pengguna, agenda kuliah, mata kuliah, kegiatan, konsultasi, serta komunikasi mahasiswa-dosen, masih dilakukan secara manual atau menggunakan aplikasi yang terpisah. Hal ini mengakibatkan duplikasi data, keterlambatan informasi, serta kurangnya efektivitas dalam komunikasi akademik. Oleh karena itu, sistem ini dirancang untuk mengintegrasikan seluruh layanan tersebut dalam satu platform berbasis web, dengan peran utama admin, dosen, dan mahasiswa.

Tahap perancangan sistem difokuskan pada penyusunan alur proses aplikasi sesuai kebutuhan setiap aktor. *Flowchart* menunjukkan bahwa pengguna memulai dengan proses *login*, kemudian diarahkan ke halaman *dashboard*, dan selanjutnya dapat memilih menu utama sesuai dengan perannya.

Admin memiliki akses untuk mengelola pengguna, mata kuliah, agenda kuliah, kegiatan, dan konsultasi. Setiap aksi yang dilakukan melalui validasi agar data konsisten dan aman.

Mahasiswa dan dosen dapat memanfaatkan fitur konsultasi *offline* serta chat untuk berkomunikasi.

Struktur database dirancang untuk menyimpan data pengguna, data mata kuliah, agenda kuliah, jadwal kegiatan, jadwal konsultasi, serta riwayat percakapan chat. Database ini juga mendukung autentikasi dan otorisasi pengguna dengan peran yang berbeda (admin, dosen, mahasiswa).

Pada tahap implementasi, sistem akan dikembangkan menggunakan framework modern. Bagian backend dapat dibangun menggunakan Laravel atau Node.js untuk menangani logika aplikasi, validasi, serta interaksi dengan database. Bagian frontend dapat dikembangkan dengan React atau Vue.js agar antarmuka pengguna responsif dan mudah digunakan baik di desktop maupun perangkat mobile. Untuk database, MySQL atau PostgreSQL dipilih karena mendukung pengolahan data relasional yang kompleks, sementara autentikasi dapat diperkuat dengan JWT (JSON Web Token) agar keamanan lebih terjamin.

Tahap pengujian sistem dilakukan dengan metode black-box testing, yaitu menguji setiap fitur sesuai dengan alur proses yang sudah dirancang. Misalnya, admin diuji dengan skenario menambah pengguna, mengedit data mata kuliah, atau menghapus kegiatan. Dosen diuji dalam mengakses agenda kuliah, mengisi nilai, atau membuat jadwal konsultasi. Mahasiswa diuji dalam melakukan chat, melihat jadwal kuliah, serta mengajukan konsultasi. Hasil pengujian diharapkan

menunjukkan bahwa sistem dapat berjalan sesuai rencana, menampilkan data dengan benar, serta memberikan pengalaman pengguna yang baik.

Tahap terakhir adalah pemeliharaan, yang mencakup perbaikan bug, peningkatan performa sistem, serta pembaruan konten dan data secara berkala. Selain itu, pengembangan fitur tambahan juga dapat dilakukan, seperti notifikasi otomatis melalui email atau aplikasi mobile, integrasi dengan sistem e-learning, atau export data ke format PDF/Excel. Dengan pemeliharaan berkelanjutan, aplikasi ini diharapkan mampu memberikan manfaat jangka panjang dalam mendukung proses manajemen akademik secara efektif, efisien, dan terintegrasi.

1. Sistem yang diusulkan

Sistem yang diusulkan merupakan aplikasi berbasis web yang dirancang untuk mendukung pengelolaan akademik, kegiatan, konsultasi, serta komunikasi antar pengguna (admin, dosen, dan mahasiswa) secara terintegrasi. Sistem ini dibangun agar setiap proses administrasi berjalan lebih efektif, dengan alur kerja yang jelas dan adanya validasi pada setiap tahapan untuk memastikan keamanan serta konsistensi data.

Modul pertama adalah manajemen pengguna, di mana admin dapat melakukan *login*, kemudian diarahkan ke *dashboard* untuk mengakses daftar pengguna. Dari menu ini, admin memiliki wewenang untuk menambahkan pengguna baru, mengedit data pengguna yang sudah ada, menghapus akun yang tidak aktif, maupun melihat detail informasi pengguna. Setiap aksi yang dilakukan melalui tahap validasi agar data yang dimasukkan benar dan sesuai kebutuhan sistem.

Modul kedua adalah agenda kuliah, yang memungkinkan admin untuk mengatur jadwal kuliah. Setelah berhasil *login*, admin dapat menambah agenda baru, mengedit agenda yang sudah ada, melihat detail agenda, menghapus agenda yang tidak relevan, serta mengisi nilai mahasiswa sesuai agenda yang tersedia. Validasi tetap diterapkan pada setiap aksi sehingga jadwal kuliah yang tersimpan selalu akurat.

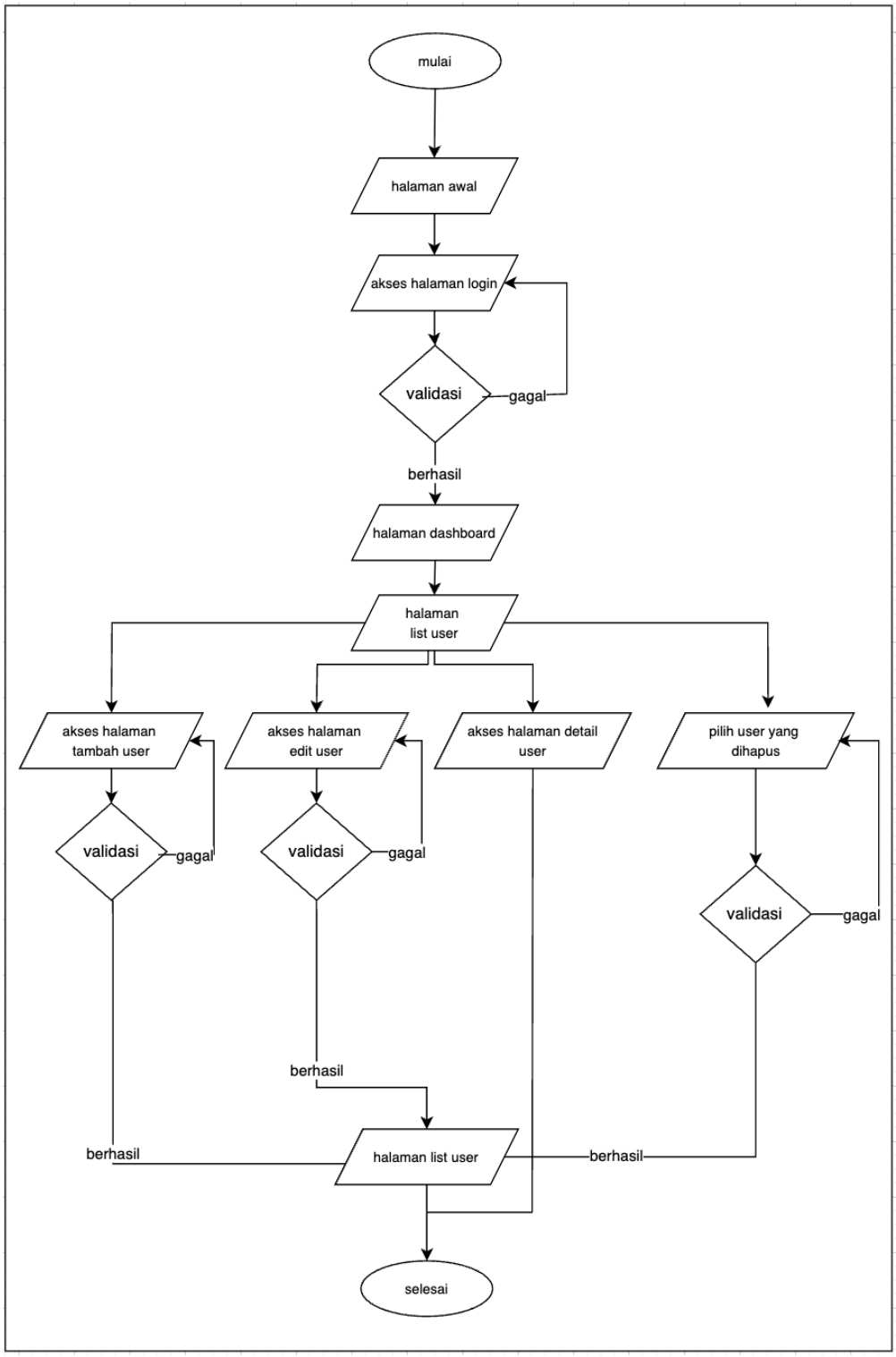
Modul berikutnya adalah manajemen mata kuliah, yang berfungsi untuk mengelola informasi terkait daftar mata kuliah. Dari *dashboard*, admin dapat menambahkan mata kuliah baru, mengedit informasi mata kuliah, melihat detail mata kuliah, maupun menghapus mata kuliah yang sudah tidak digunakan. Semua perubahan melalui tahap validasi untuk memastikan data yang dikelola konsisten.

Selain itu, terdapat modul kegiatan yang berfungsi untuk mendukung pengelolaan event akademik, seperti seminar, pelatihan, atau kegiatan lainnya. Admin dapat menambah kegiatan baru, mengedit kegiatan, melihat detail kegiatan, dan menghapus kegiatan yang tidak relevan. Validasi diterapkan pada tahap ini untuk menjamin kegiatan yang tersimpan benar-benar sesuai.

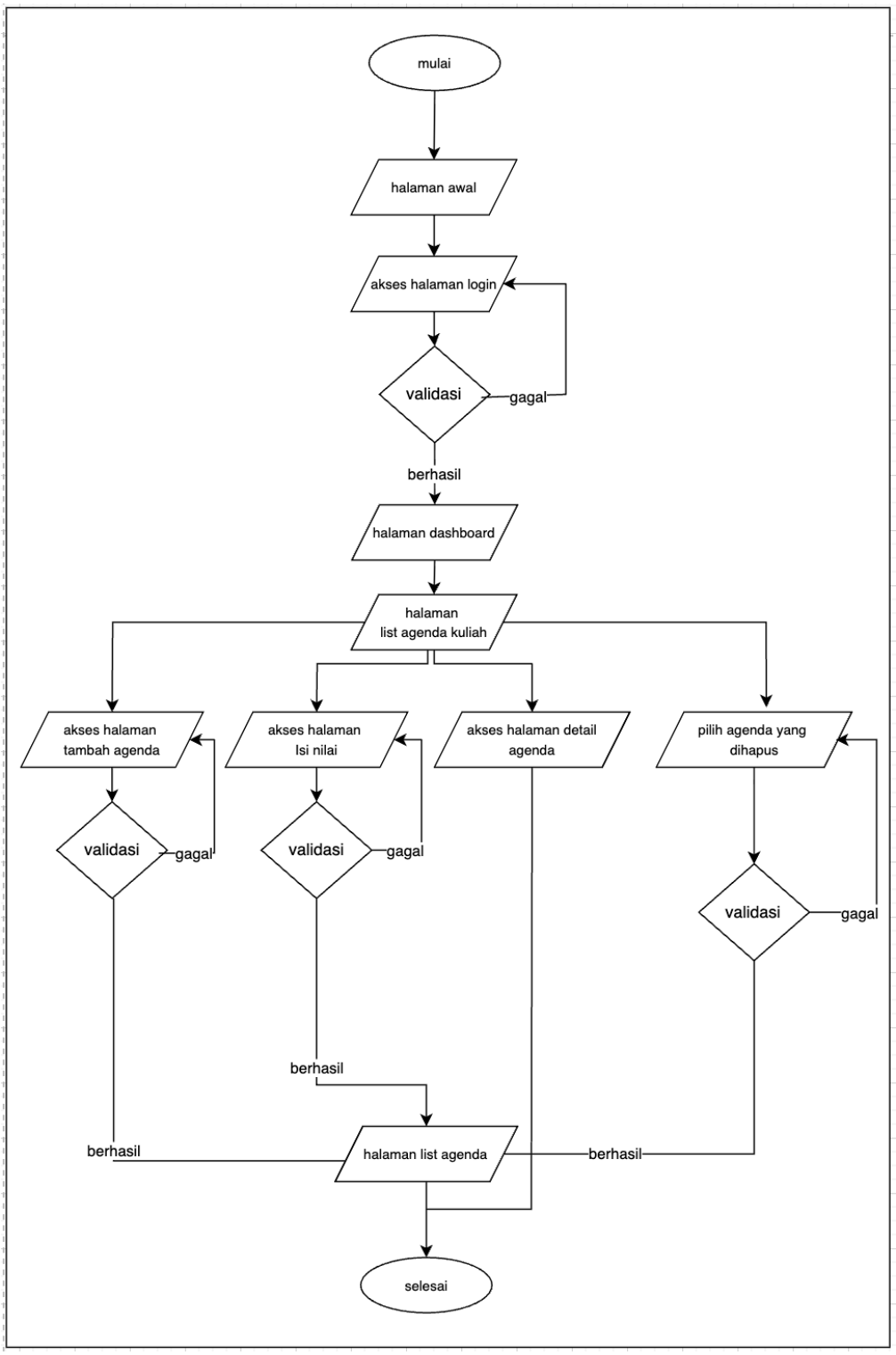
Pada modul konsultasi *offline*, admin atau dosen dapat mengatur jadwal konsultasi dengan mahasiswa. Dari daftar konsultasi, pengguna dapat menambahkan jadwal baru, mengedit data konsultasi, melihat detail konsultasi, maupun menghapus konsultasi tertentu. Proses validasi memastikan hanya data konsultasi yang sah yang tersimpan di sistem.

Terakhir, terdapat modul chat yang memungkinkan komunikasi langsung antar pengguna. Proses dimulai dengan pengisian form chat oleh pengguna, kemudian dilakukan validasi. Jika data yang dimasukkan benar, sistem akan menampilkan halaman chat dan percakapan dapat berlangsung. Jika gagal, pengguna akan diminta untuk mengisi ulang form.

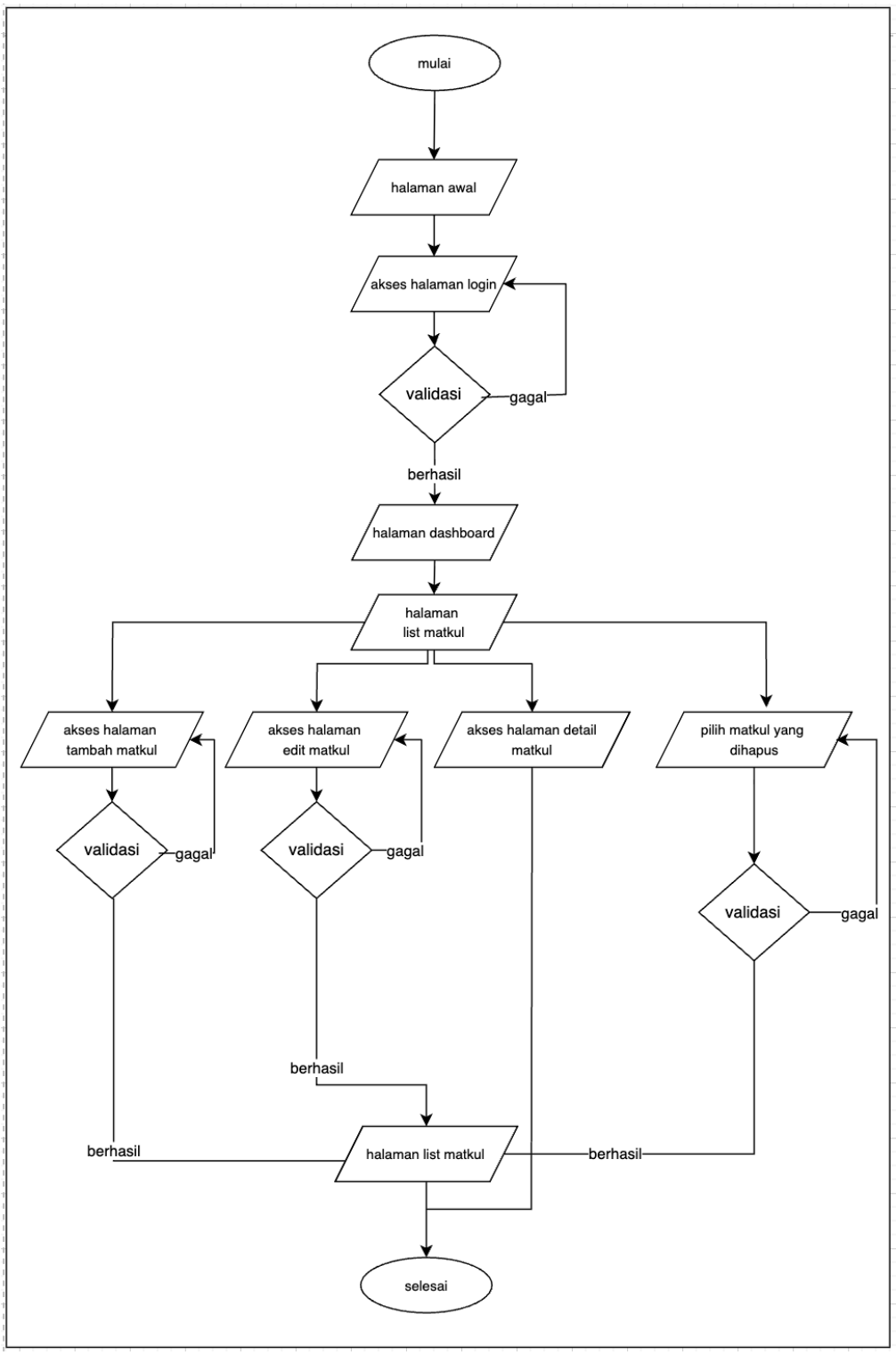
Secara keseluruhan, sistem ini mengintegrasikan seluruh proses manajemen pengguna, mata kuliah, agenda, kegiatan, konsultasi, serta komunikasi dalam satu platform terpadu. Admin memiliki kontrol penuh untuk mengelola data akademik, sementara dosen dan mahasiswa dapat memanfaatkan fitur konsultasi serta chat untuk berkomunikasi secara lebih efisien. Dengan adanya sistem ini, diharapkan pengelolaan informasi akademik menjadi lebih mudah, cepat, dan efektif, serta mampu meningkatkan interaksi antara mahasiswa, dosen, dan admin.



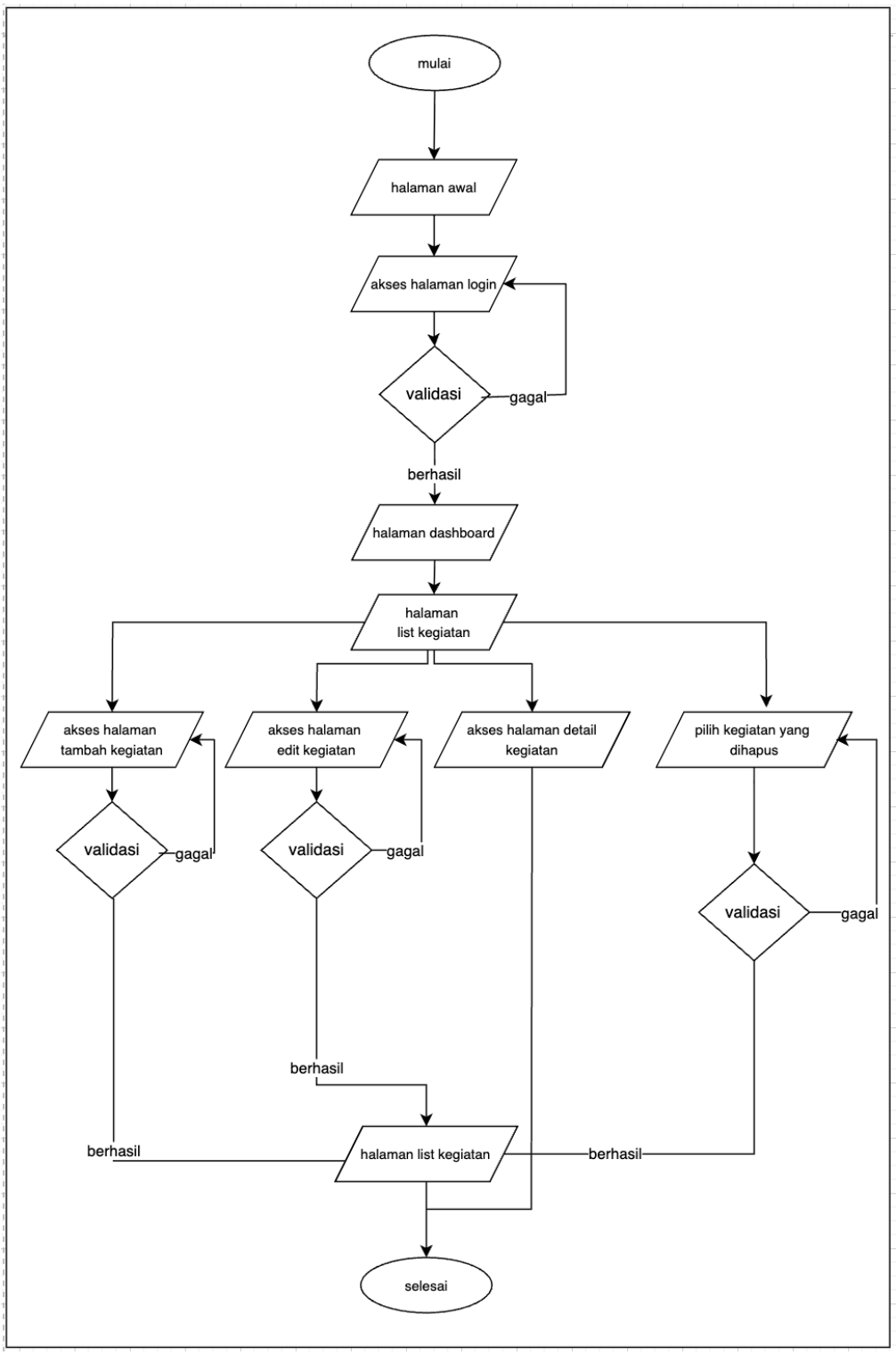
Gambar 4. 1 *Flowchart* fitur atur *user*



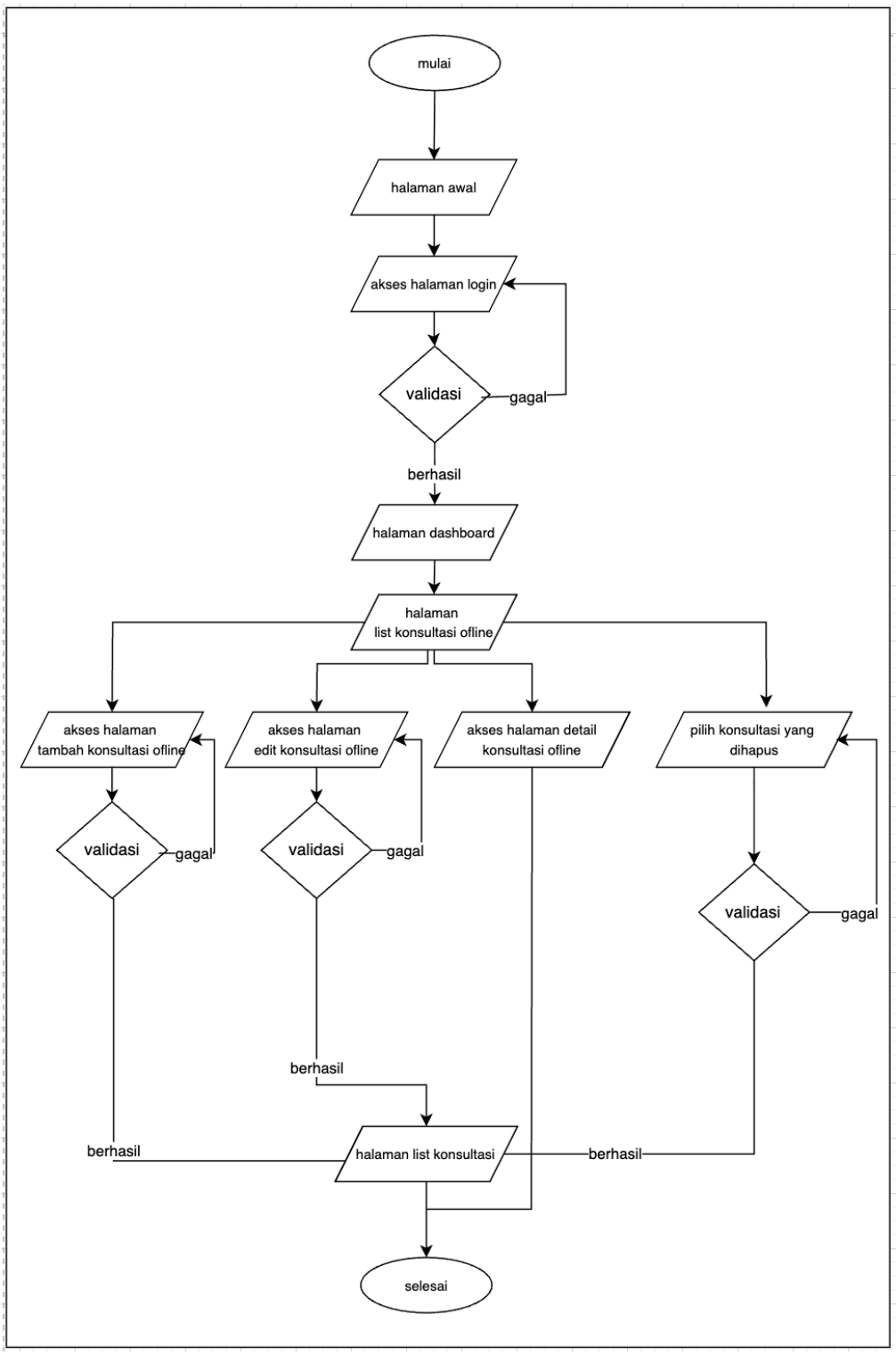
Gambar 4. 2 *Flowchart* fitur agenda kuliah



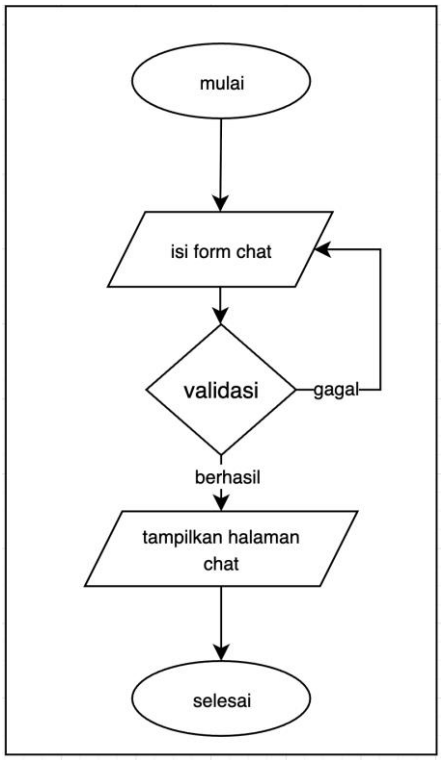
Gambar 4. 3 *Flowchart* fitur matkul



Gambar 4. 4 *Flowchart* fitur kegiatan

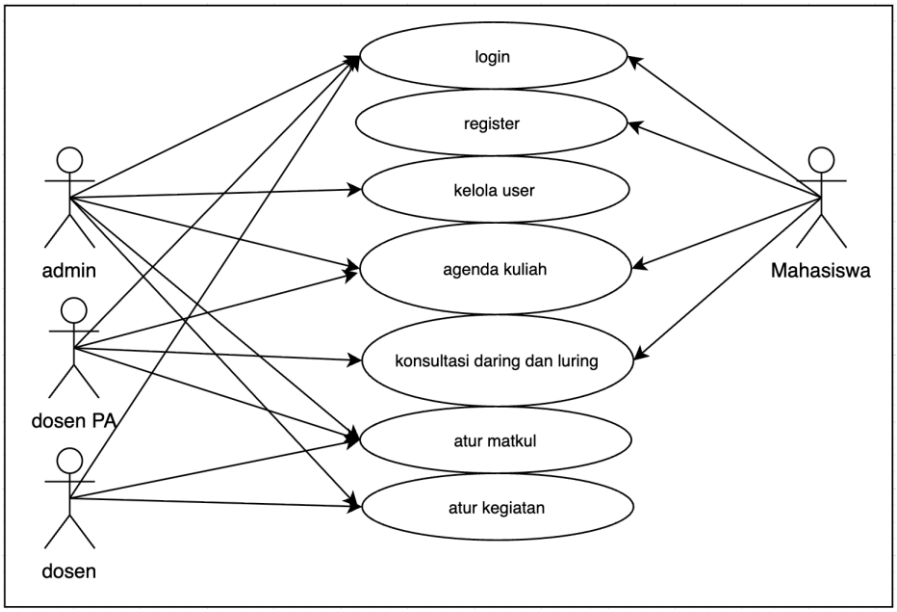


Gambar 4. 5 fitur konsultasi *offline*



Gambar 4. 6 fitur konsultasi *online*

2. *Usecase* diagram



Gambar 4. 7 *Usecase* alur sistem

3. Rancangan basis data



Gambar 4. 8 Rancangan basis data

rancangan ini menggambarkan rancangan basis data untuk sistem informasi akademik yang melibatkan pengguna, kegiatan, mata kuliah, jadwal, pesan, dan pengelolaan peran. Tabel *user*s menjadi pusat utama karena menyimpan informasi dasar mahasiswa atau pengguna lain, termasuk identitas, jurusan, dan kredensial *login*. Relasinya terhubung ke tabel agenda_kuliahs, akademiks, jadwals, serta messages, menunjukkan peran pengguna dalam mengatur agenda kuliah, mencatat nilai akademik, membuat jadwal konsultasi, hingga melakukan komunikasi antar pengguna.

Tabel agenda_kuliahs berfungsi mencatat aktivitas akademik dengan atribut seperti judul, deskripsi, tanggal, serta keterhubungan dengan *user*s sebagai penyelenggara dan kegiatans sebagai jenis aktivitas. Data ini selanjutnya terkait dengan tabel akademiks, yang berperan mencatat hubungan antara agenda kuliah, mata kuliah (matkuls), pengguna, serta nilai yang diperoleh. Tabel matkuls sendiri menyimpan informasi mata kuliah lengkap seperti kode, nama, SKS, semester, jurusan, hingga status aktif.

Selain itu, tabel jadwals digunakan untuk mengatur jadwal pertemuan pengguna dengan atribut tambahan seperti masalah yang dibahas, status, dan antrian. Untuk komunikasi, tabel messages memungkinkan pengiriman pesan antar pengguna, di mana terdapat relasi ganda ke tabel *user*s, baik sebagai pengirim maupun penerima.

Dari sisi manajemen peran, tabel roles menyimpan data hak akses dan nama peran, yang kemudian dihubungkan ke tabel model_has_roles sebagai penghubung

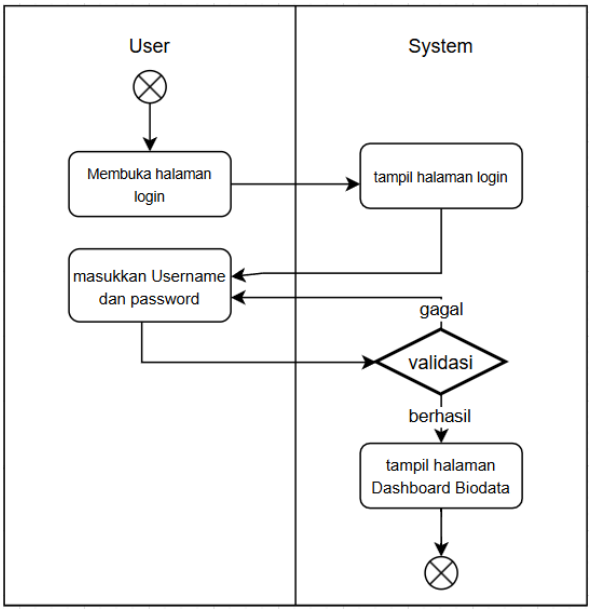
fleksibel antara role dan entitas model. Adapun tabel cache berdiri sendiri sebagai penyimpanan sementara data yang mempercepat akses aplikasi.

Secara keseluruhan, ERD ini menunjukkan sistem terintegrasi yang tidak hanya mendukung kegiatan akademik seperti pencatatan agenda, nilai, dan mata kuliah, tetapi juga mencakup manajemen jadwal, komunikasi, serta kontrol akses berbasis peran, sehingga sistem mampu berjalan efisien dan sesuai kebutuhan pengguna.

4. *Activity diagram*

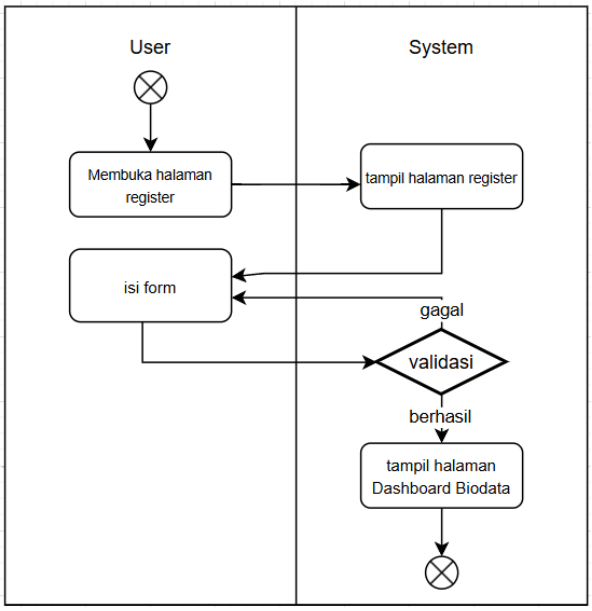
Activity diagram ini menjelaskan tentang aktivitas-aktivitas yang telah terjadi dalam sebuah aliran proses pada sebuah sistem.

a. *Activity diagram login*



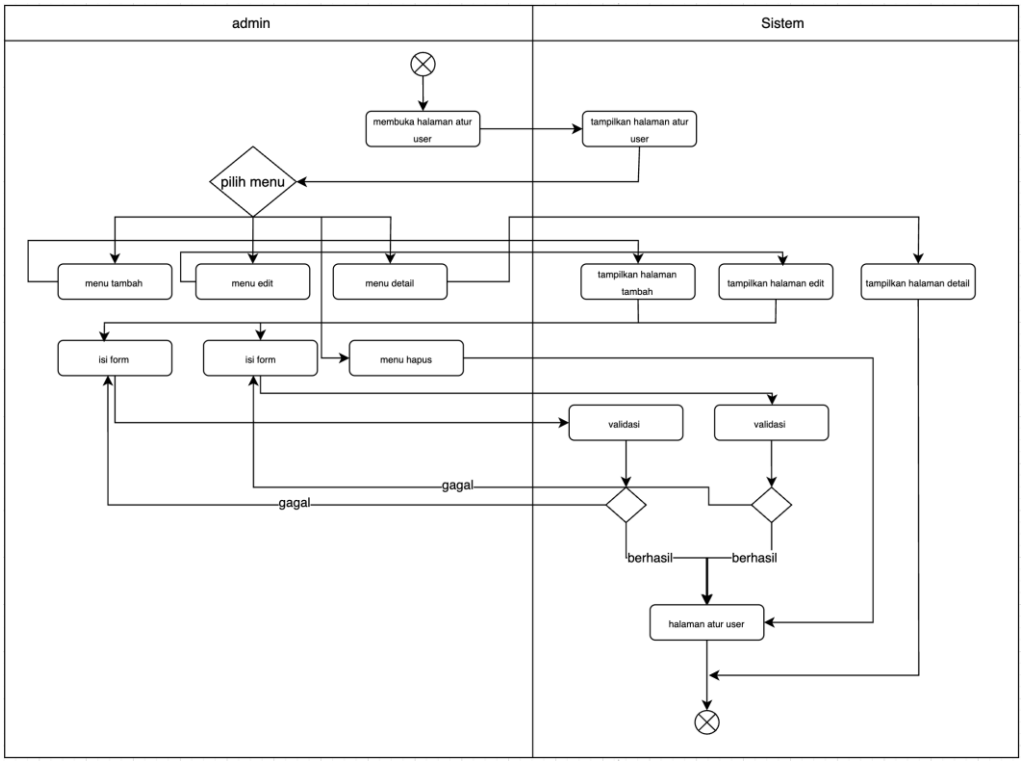
Gambar 4. 9 *Activity diagram login*

b. *Activity diagram register*



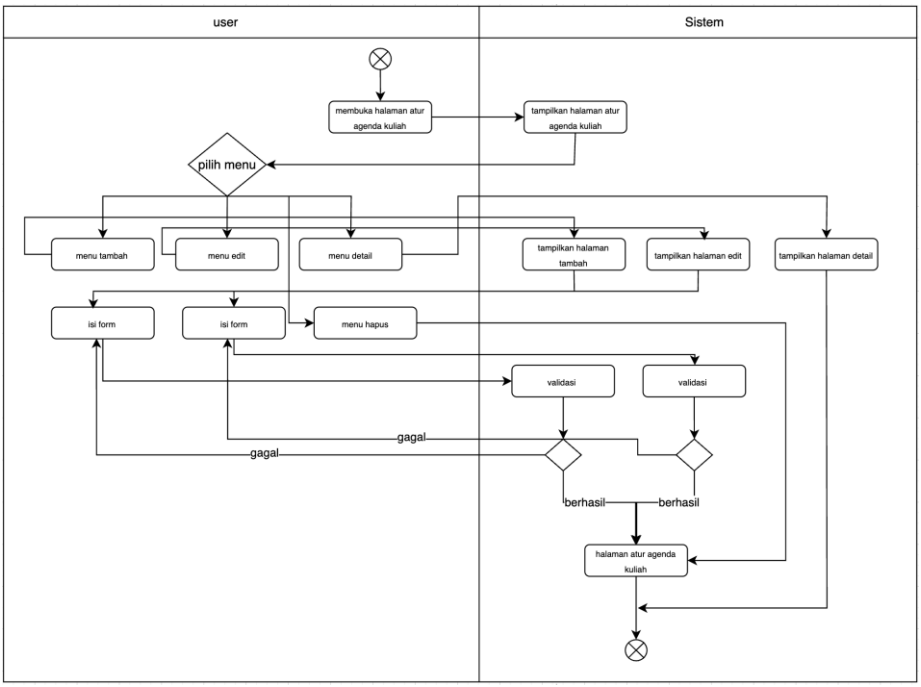
Gambar 4. 10 *Activity diagram register*

c. *Activity diagram* fitur atur *user*



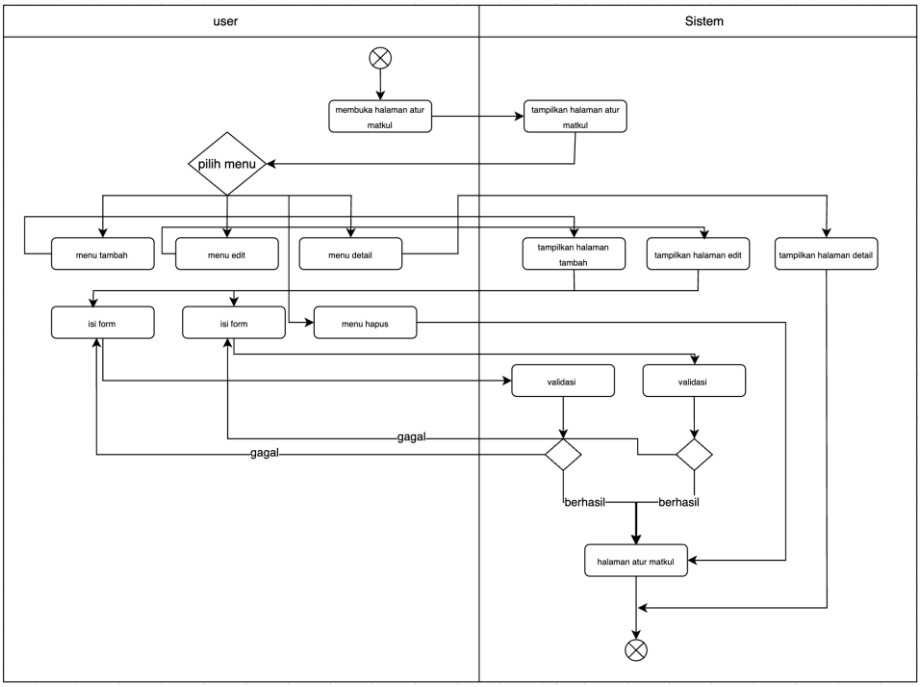
Gambar 4. 11 *Activity diagram* fitur atur *user*

d. *Activity diagram* fitur agenda kuliah



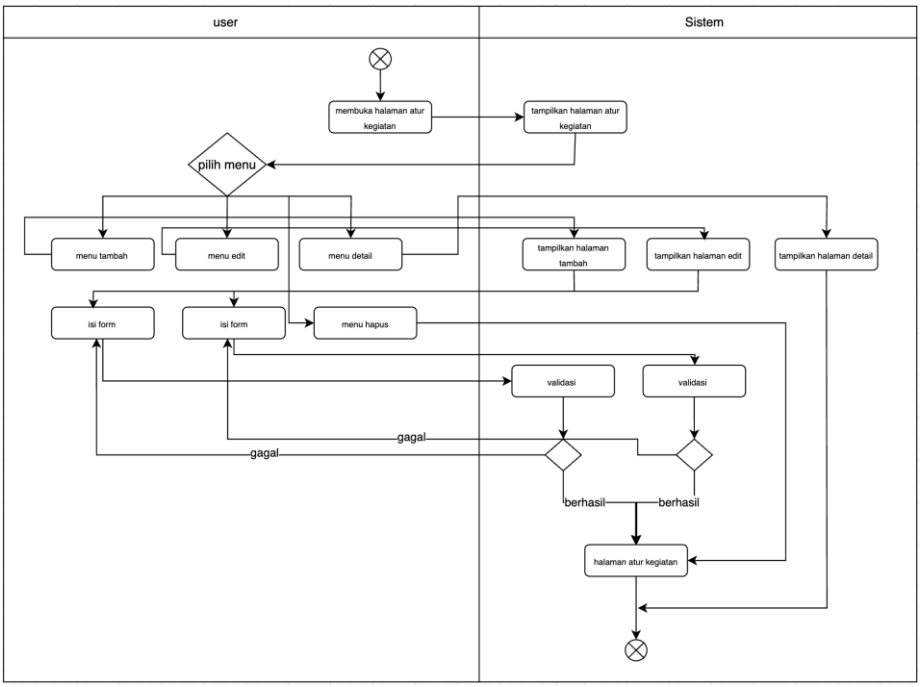
Gambar 4. 12 *Activity diagram* agenda kuliah

e. *Activity diagram* fitur matkul



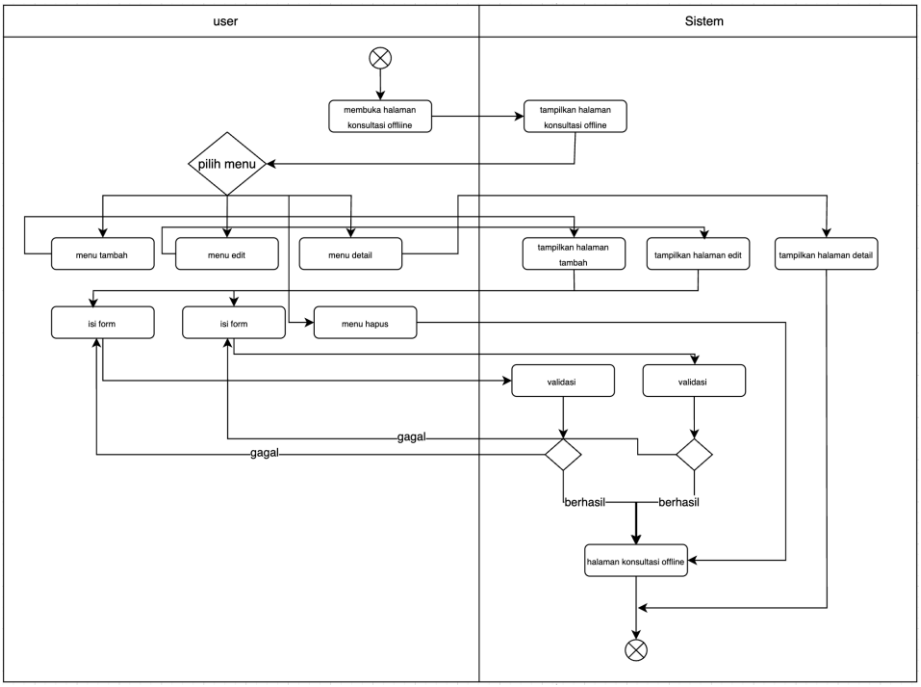
Gambar 4. 13 *Activity diagram* atur matkul

f. *Activity diagram* fitur kegiatan



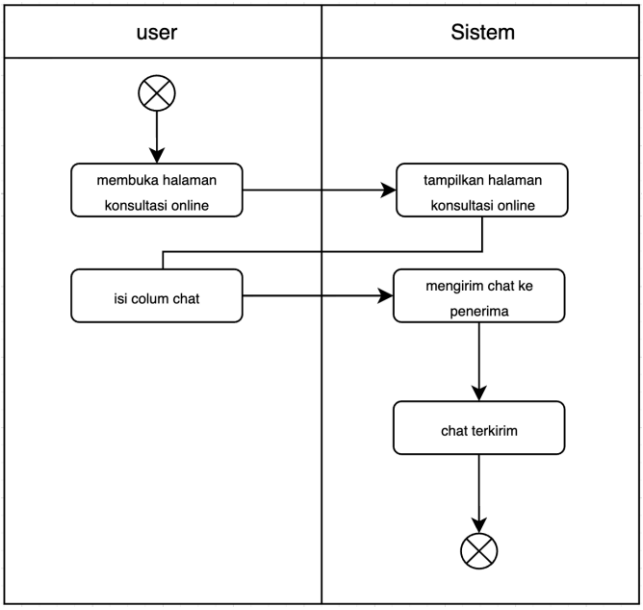
Gambar 4. 14 *Activity diagram* atur kegiatan

g. *Activity diagram* fitur Konsultasi *offline*



Gambar 4. 15 *Activity diagram* konsultasi *offline*

h. *Activity diagram* konsultasi *online*

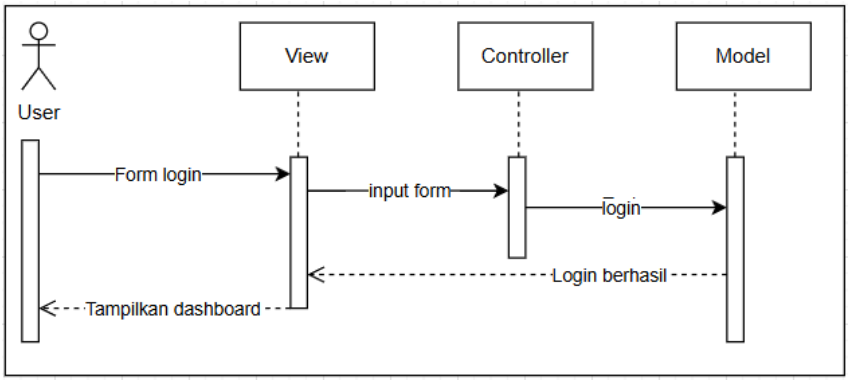


Gambar 4. 16 *Activity diagram* konsultasi *online*

5. Sequence diagram

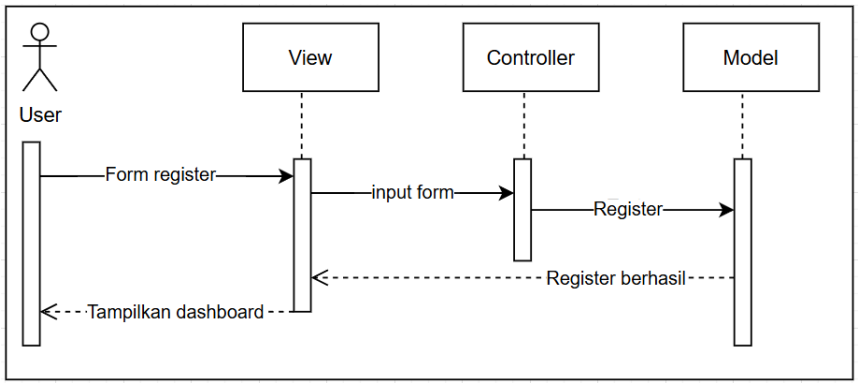
Sequence diagram merupakan *diagram* yang menggambarkan interaksi antara objek-objek dalam sistem secara berurutan berdasarkan waktu. *Diagram* ini menunjukkan bagaimana objek-objek saling bertukar pesan atau data untuk menyelesaikan suatu tugas atau *use case*.

a. *Sequence diagram login*



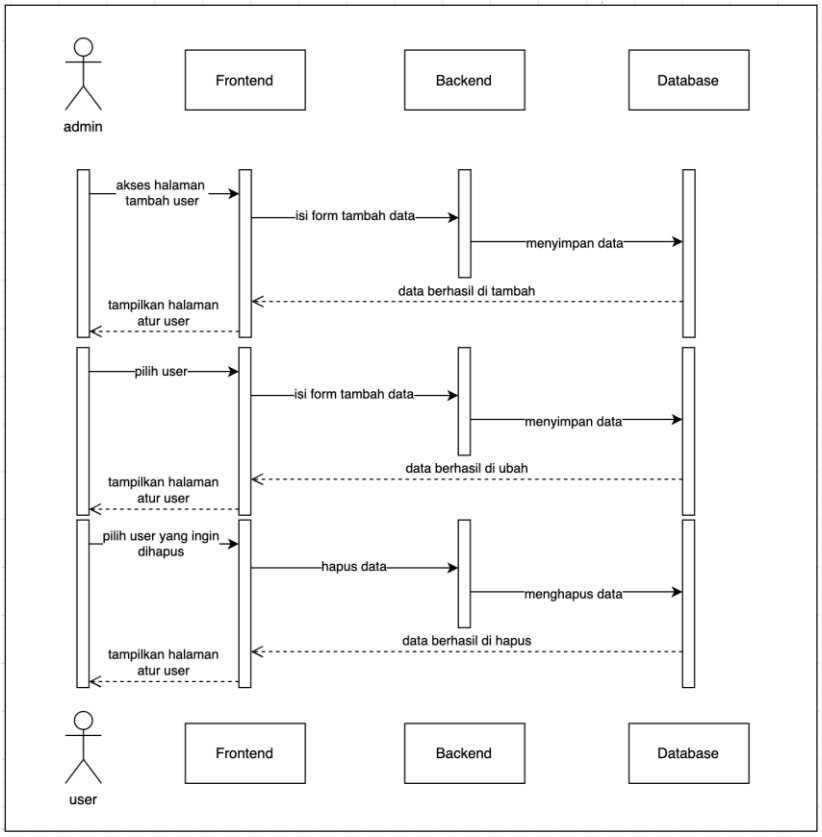
Gambar 4. 17 *Sequence diagram login*

b. *Sequence diagram register*



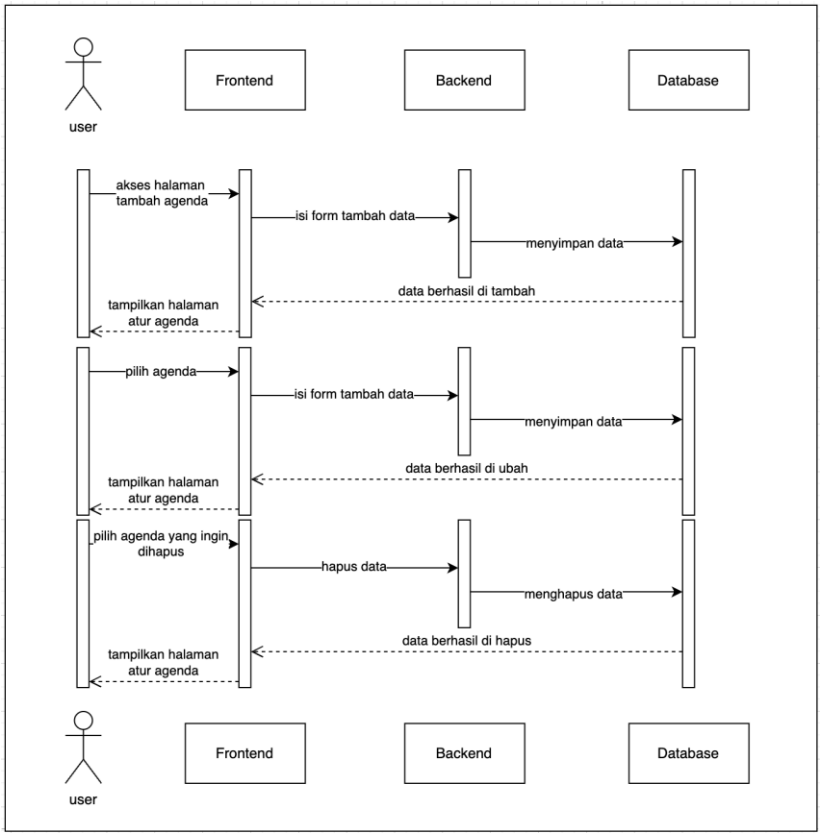
Gambar 4. 18 *Sequence diagram register*

c. *Sequence diagram* fitur atur *user*



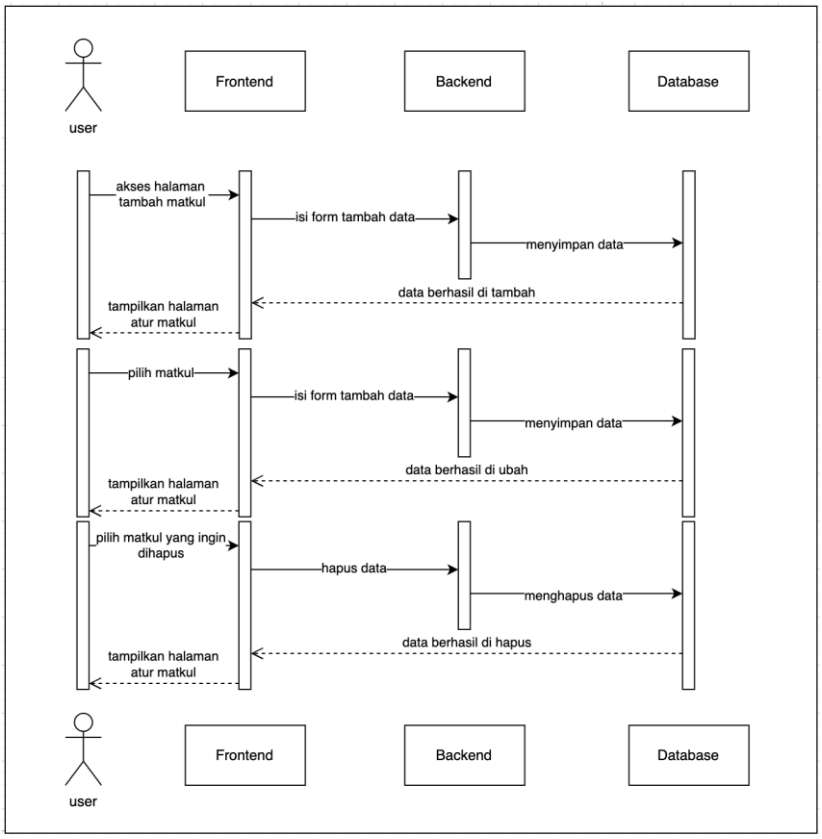
Gambar 4. 19 *Sequence diagram* atur *user*

d. *Sequence diagram* fitur agenda kuliah



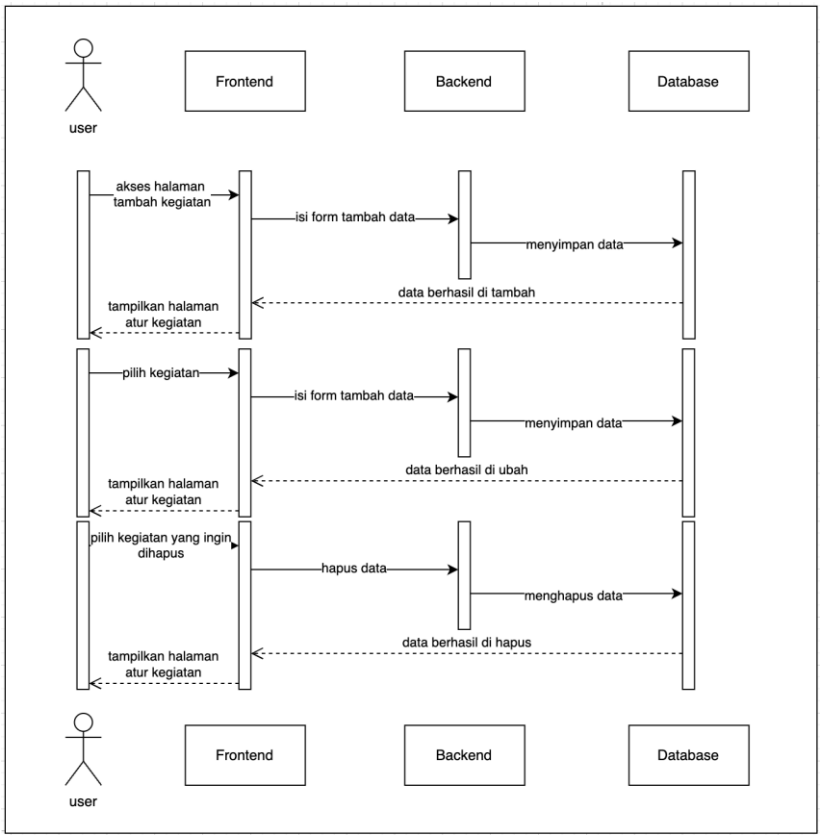
Gambar 4. 20 *Sequence diagram* atur agenda kuliah

e. *Sequence diagram* fitur atur matkul



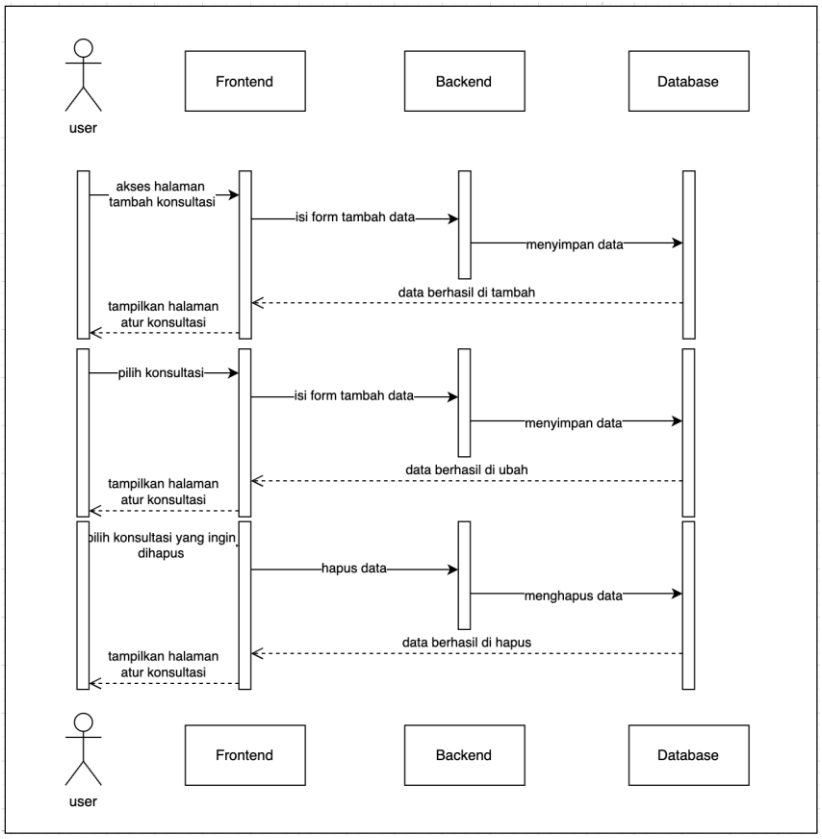
Gambar 4. 21 *Sequence diagram* atur matkul

f. *Sequence diagram* fitur kegiatan



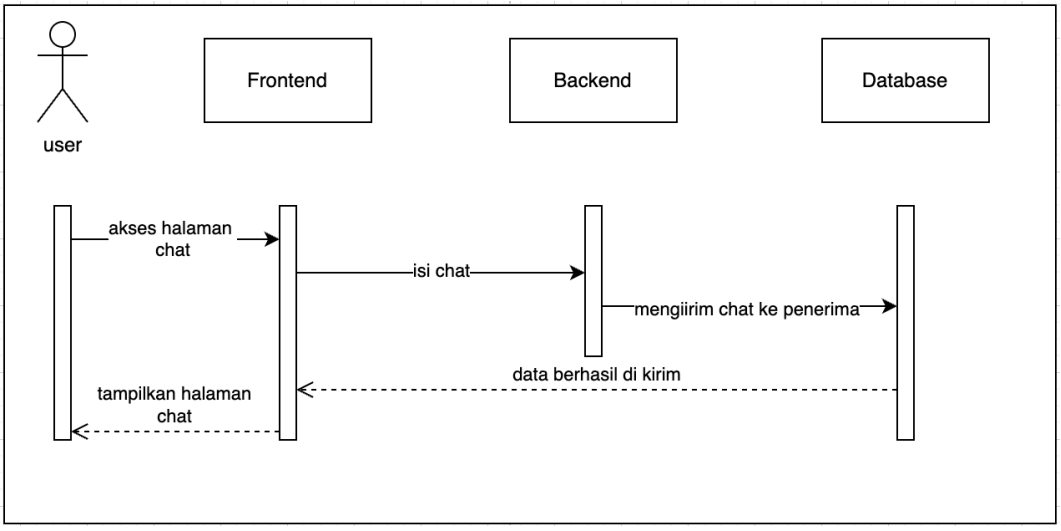
Gambar 4. 22 *Sequence diagram* fitur kegiatan

g. *Sequence diagram* konsultasi *offline*



Gambar 4. 23 *Sequence diagram* konsultasi *offline*

h. *Sequence diagram* konsultasi *online*



Gambar 4. 24 *Sequence diagram* konsultasi *online*

B. Perancangan Input Output

1. Halaman awal

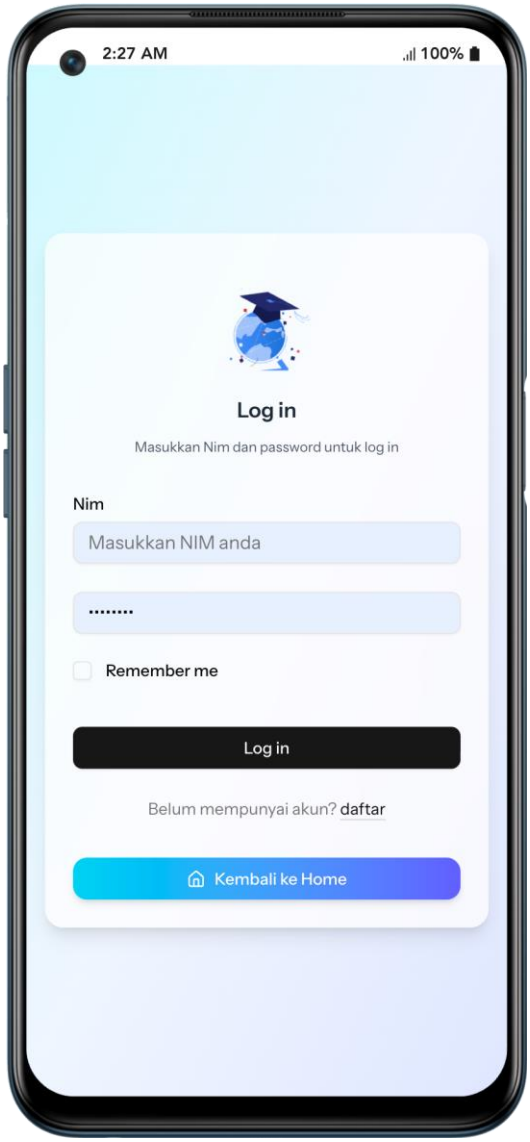
Halaman Awal merupakan halaman yang pertama kali muncul saat *aplikasi* pertama kali di akses.



Gambar 4. 25 Halaman awal

2. Halaman login

Halaman *login* merupakan tampilan pertama sebelum masuk pada menu utama *aplikasi*.



Gambar 4. 26 Halaman *login*

3. Halaman register

Halaman *Register* merupakan Halaman untuk membuat akun sebelum masuk pada menu utama *aplikasi*.



Gambar 4. 27 Halaman *register*

4. Halaman Dashboard

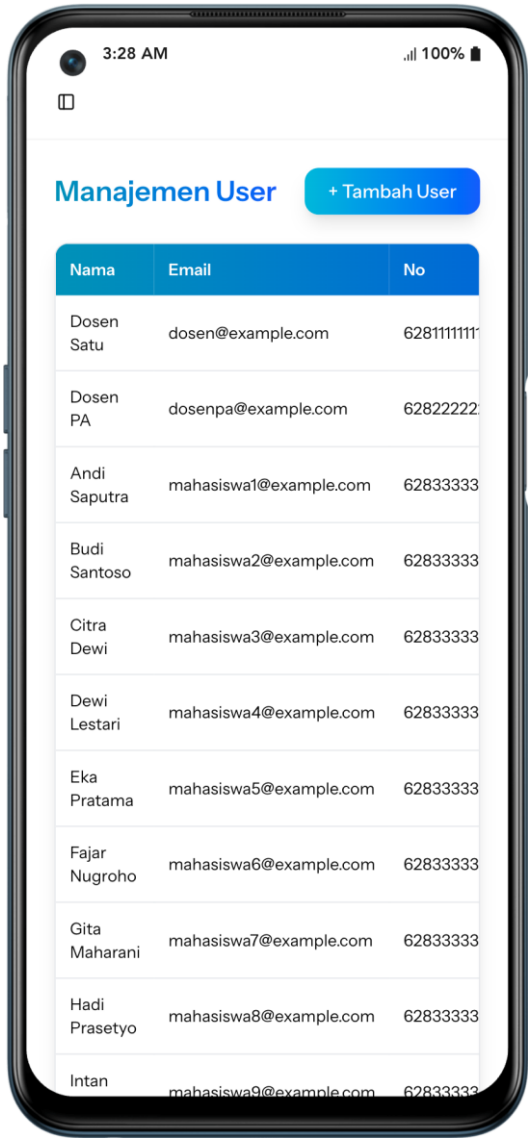
Halaman ini merupakan halaman antar muka aplikasi.



Gambar 4. 28 Halaman *dashboard*

5. Halaman atur user

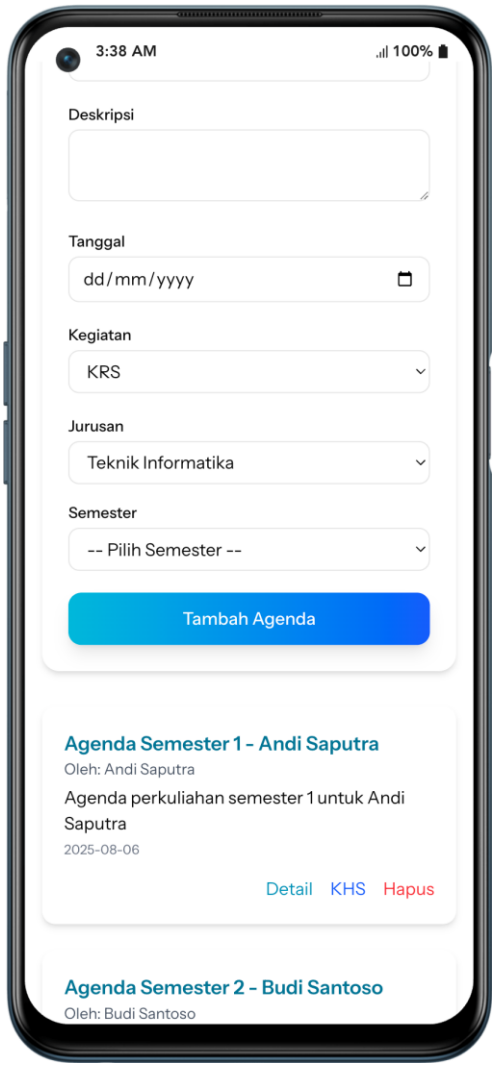
Halaman ini selain menampilkan semua *user*, terdapat 3 tombol utama yaitu tambah *user* yang akan membuka pop up tambah *user* untuk menambah *user*.terdapat juga tombol edit yang akan membuka pop up edit *user* . dan terdapat juga tombol hapus untuk menghapus *user*.



Gambar 4. 29 Halaman Atur *user*

6. Halaman Atur agenda kuliah

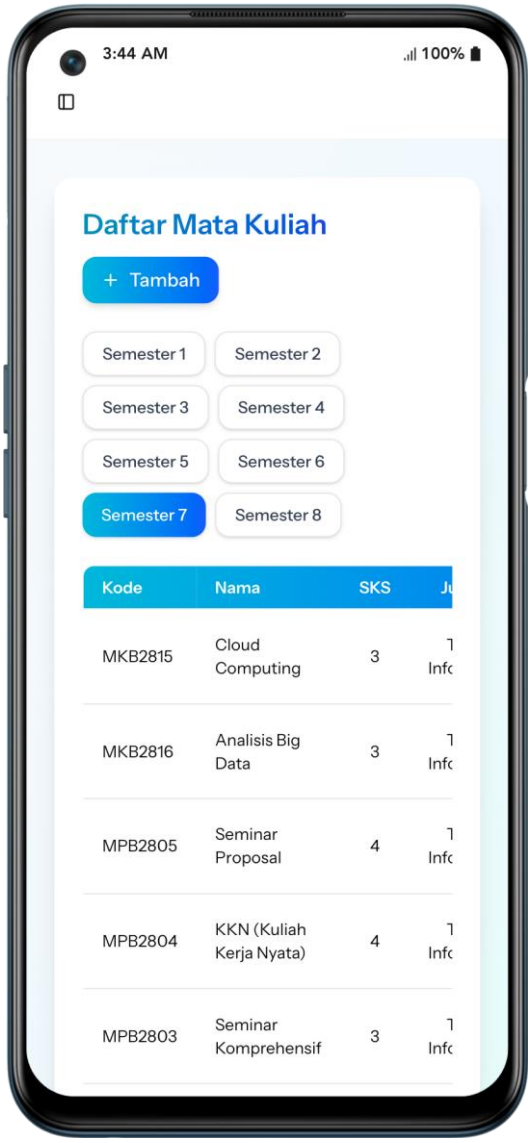
Halaman ini selain menampilkan semua data, terdapat 3 tombol utama yaitu tambah data yang akan membuka pop up tambah data untuk menambah data.terdapat juga tombol Detail yang akan membuka halaman untuk melihat detail agenda yang di pilih. terdapat juga tombol hapus untuk menghapus data.Selain 3 tombol utama, terdapat juga 3 tombol ,yaitu konsultasi *online* atau *offline* yang sesuai Namanya, jika diklik akan menuju ke halama konsultasi yang dipilih.



Gambar 4. 30 Halaman atur agenda kuliah

7. Halaman Atur matkul

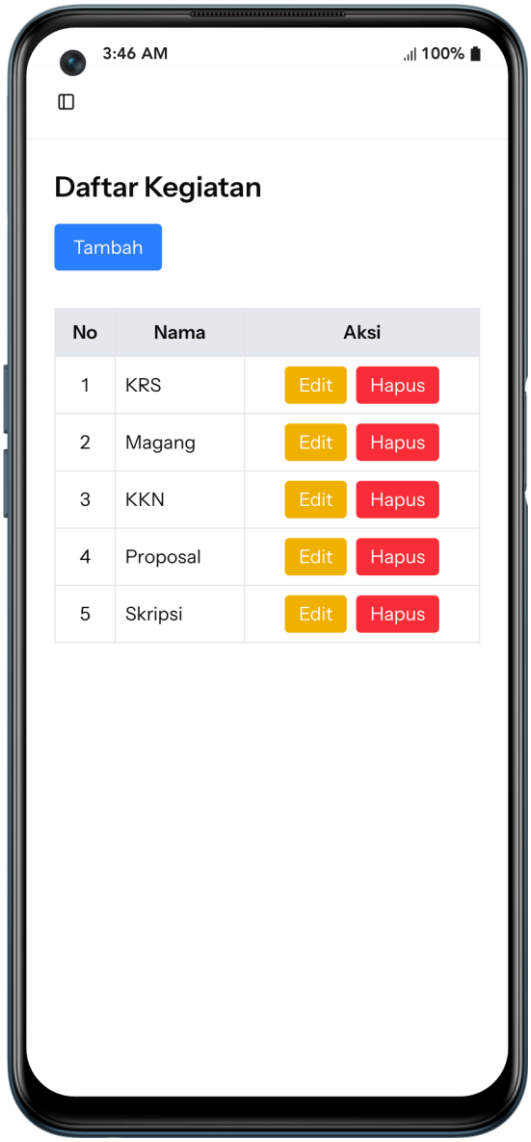
Halaman ini selain menampilkan semua data, terdapat 3 tombol utama yaitu tambah data yang akan membuka pop up tambah data untuk menambah data.terdapat juga tombol edit yang akan membuka pop up edit data . dan terdapat juga tombol hapus untuk menghapus data.



Gambar 4. 31 Halaman atur matkul

8. Halaman atur kegiatan

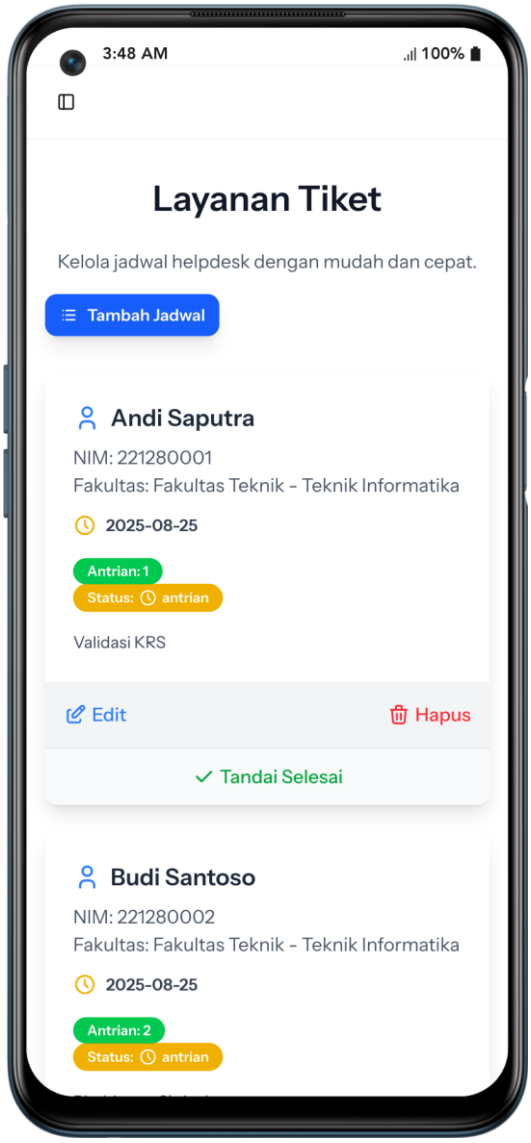
Halaman ini selain menampilkan semua data, terdapat 3 tombol utama yaitu tambah data yang akan membuka pop up tambah data untuk menambah data.terdapat juga tombol edit yang akan membuka pop up edit data . dan terdapat juga tombol hapus untuk menghapus data.



Gambar 4. 32 halaman atur kegiatan

9. Halaman konsultasi offline

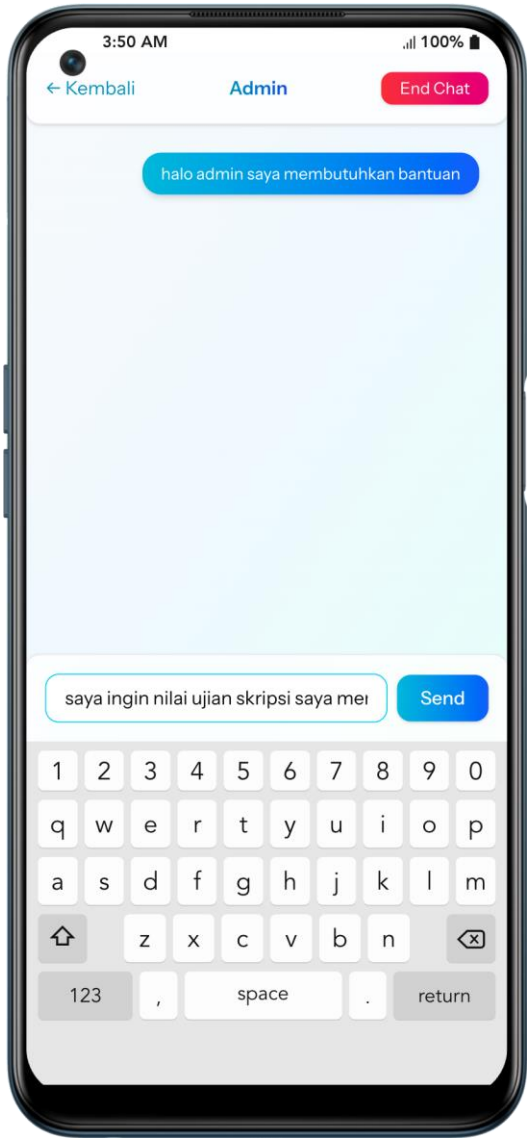
Halaman ini selain menampilkan semua data, terdapat 3 tombol utama yaitu tambah data yang akan membuka pop up tambah data untuk menambah data.terdapat juga tombol edit yang akan membuka pop up edit data . dan terdapat juga tombol hapus untuk menghapus data.



Gambar 4. 33 Halaman konsultasi *offline*

10. Halaman konsultasi online

Halaman ini berfungsi untuk mahasiswa yang ingin konsultasi secara *online* dengan dosen PA .



Gambar 4. 34 Halaman konsultasi *online*

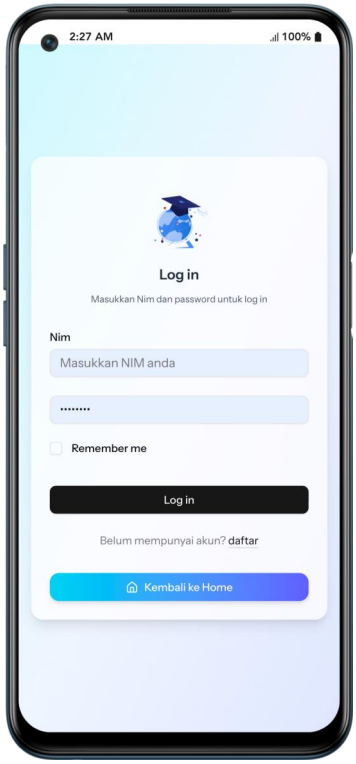
C. Pengujian sistem

Pengujian *Aplikasi* dilakukan dengan menggunakan dua metode pengujian yaitu pengujian *Black Box* dan Pengujian *White Box.*

1. Blackbox

a. *Blackbox* pengujian fitur *login*

Tabel 4. 1 *Blackbox login*

Test Faktor	Hasil	Keterangan
Pengujian *aplikasi* saat *user* membuka *aplikasi* dan melakukan *login*.	☐	Berhasil, karena dapat menampilkan halaman menu *login*.
screenshot		
		

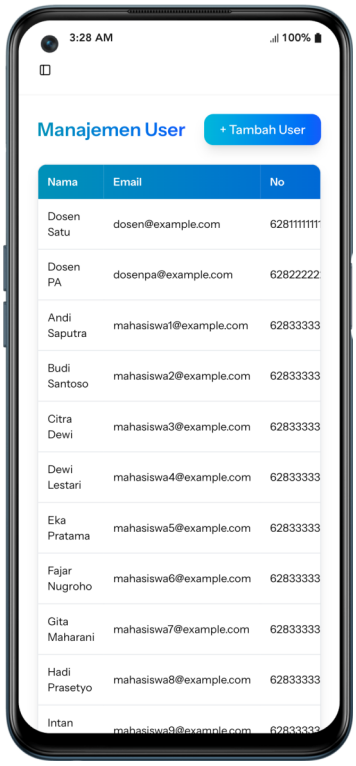
b. *Blackbox* pengujian fitur *register*

Tabel 4. 2 *Blackbox register*

Test Faktor	Hasil	Keterangan
Pengujian aplikasi saat *user* membuka aplikasi dan melakukan *register*.		Berhasil, karena dapat menampilkan halaman menu *login*.
screenshot		
		

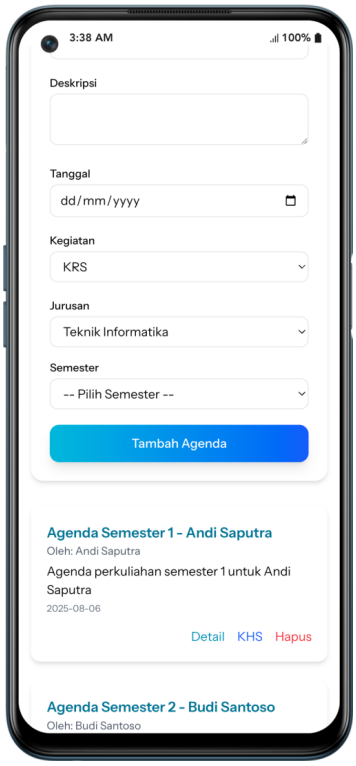
c. *Blackbox* pengujian fitur atur *user*

Tabel 4. 3 *Blackbox* atur *user*

Test Faktor	Hasil	Keterangan
Pengujian *aplikasi* saat *user* membuka *aplikasi* dan mencoba fitur tambah,lihat edit,hapus dan detail		Berhasil, karena dapat menampilkan halaman menu *login*.
screenshot		
		

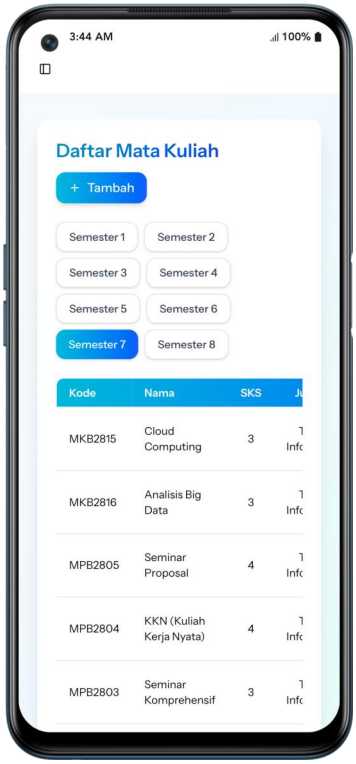
d. *Blackbox* pengujian fitur atur agenda kuliah

Tabel 4. 4 *Blackbox* atur agenda kuliah

Test Faktor	Hasil	Keterangan
Pengujian *aplikasi* saat *user* membuka *aplikasi* dan mencoba fitur tambah,detail,khs,hapus dan detail		Berhasil, karena dapat menampilkan halaman menu *login*.
screenshot		
		

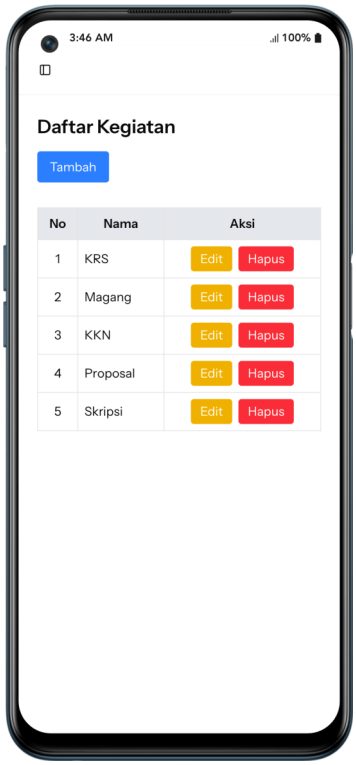
e. *Blackbox* pengujian fitur atur matkul

Tabel 4. 5 *Blackbox* atur matkul

Test Faktor	Hasil	Keterangan
Pengujian *aplikasi* saat *user* membuka *aplikasi* dan mencoba fitur tambah,lihat edit,hapus dan detail	?	Berhasil, karena dapat menampilkan halaman menu *login*.
screenshot		
		

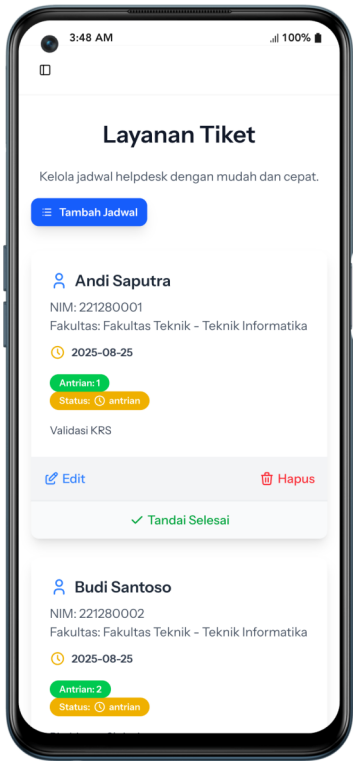
f. *Blackbox* pengujian fitur atur kegiatan

Tabel 4. 6 *Blackbox* atur kegiatan

Test Faktor	Hasil	Keterangan
Pengujian *aplikasi* saat *user* membuka *aplikasi* dan mencoba fitur tambah,lihat edit,hapus dan detail		Berhasil, karena dapat menampilkan halaman menu *login*.
screenshot		
		

g. *Blackbox* pengujian fitur konsultasi *offline*

Tabel 4. 7 *Blackbox* konsultasi *online*

Test Faktor	Hasil	Keterangan
Pengujian *aplikasi* saat *user* membuka *aplikasi* dan mencoba fitur tambah,lihat edit,hapus dan detail		Berhasil, karena dapat menampilkan halaman menu *login*.
screenshot		
		

h. *Blackbox* pengujian fitur konsultasi *offline*

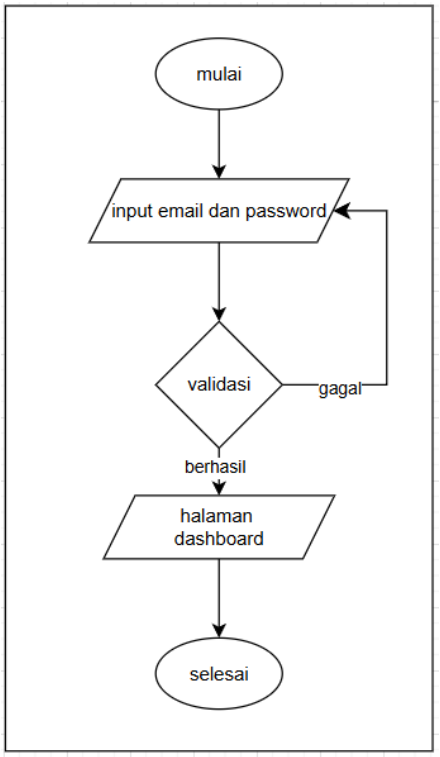
Tabel 4. 8 *Blackbox* konsultasi *offline*

Test Faktor	Hasil	Keterangan
Pengujian *aplikasi* saat *user* membuka *aplikasi* dan mencoba fitur tambah,lihat edit,hapus dan detail		Berhasil, karena dapat menampilkan halaman menu *login*.
screenshot		
		

2. Whitebox

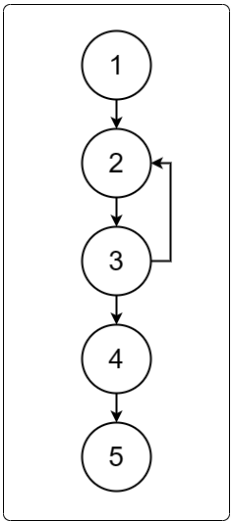
a. *Whitebox* pengujian *login*

1) *Flowchart*



Gambar 4. 35 *Flowchart login*

2) *Flowgraph*



Gambar 4. 36 *Flowgraph login*

Berdasarkan gambar 4. 36 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 5

N (*node*) = 5

P (predikat *node*) = 1

Penyelesaian :

V(G) = E – N + 2

= 5 – 5 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 2*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 2

Path 2 = 1 – 2 – 3 – 4 – 5

(4) Grafik *matriks login* berhasil

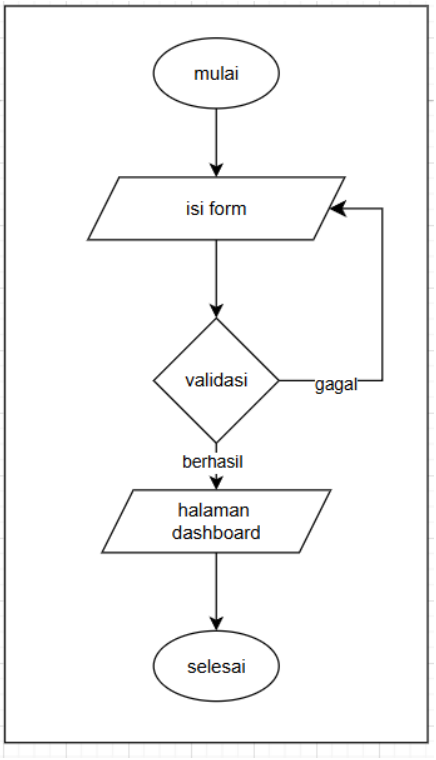
Tabel 4. 9 *Matriks login*

	1	2	3	4	5	E-1
1		1				1 - 1 = 0
2			1			1 – 1 = 0
3		1		1		2 – 1 = 1
4					1	1 – 1 = 0

5						0
	SUM (E + 1)					1 + 1 = 2

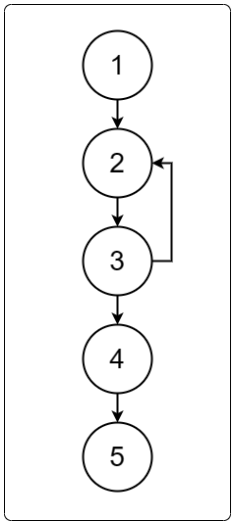
b. *Whitebox* pengujian *register*

1) *Flowchart*



Gambar 4. 37 *Flowchart register*

2) *Flowgraph*



Gambar 4. 38 *Flowgraph register*

Berdasarkan gambar 4. 38 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 5

N (*node*) = 5

P (predikat *node*) = 1

Penyelesaian :

V(G) = E – N + 2

= 5 – 5 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 2*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 2

Path 2 = 1 – 2 – 3 – 4 – 5

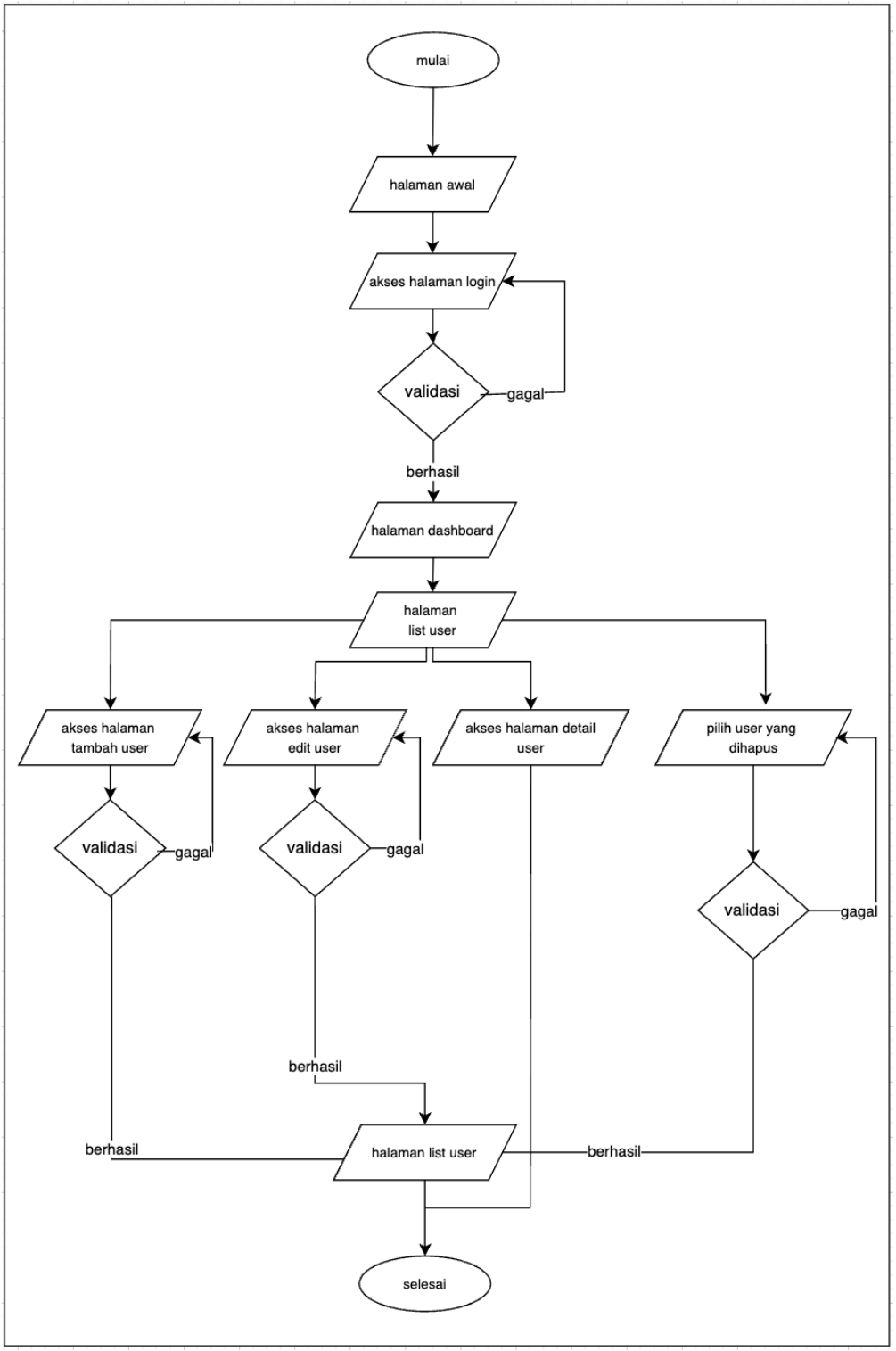
(4) Grafik *matriks login* berhasil

Tabel 4. 10 *Matriks register*

	1	2	3	4	5	E-1
1		1				1 - 1 = 0
2			1			1 – 1 = 0
3		1		1		2 – 1 = 1
4					1	1 – 1 = 0
5						0
	SUM (E + 1)					1 + 1 = 2

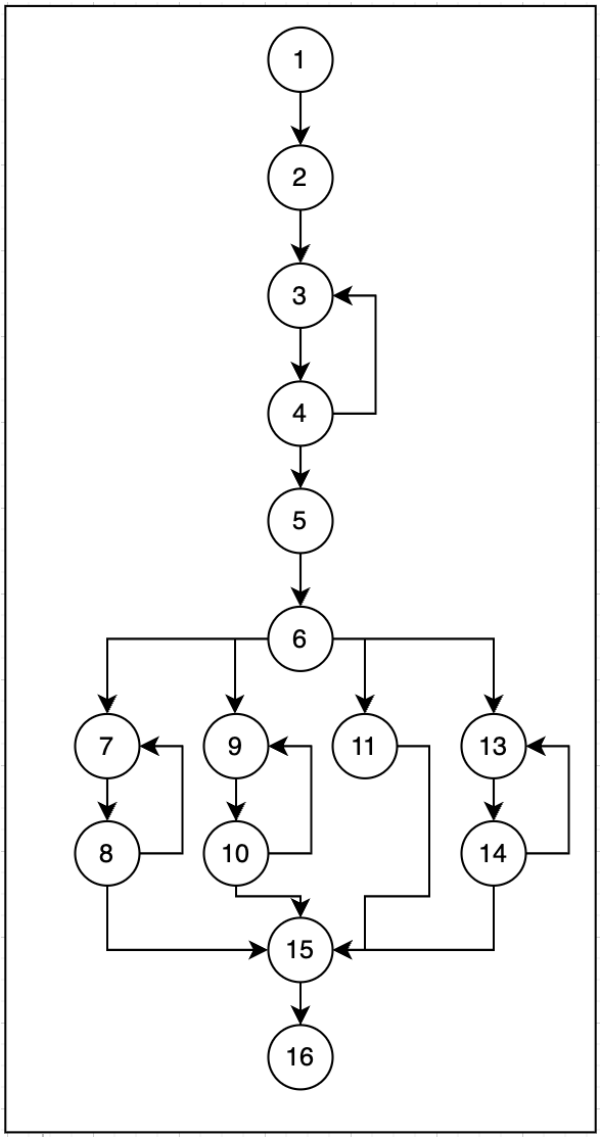
c. *Whitebox* pengujian atur *user*

1) *Flowchart*



Gambar 4. 39 *Flowchart* atur *user*

2) *Flowgraph*



Gambar 4. 40 *Flowgraph* atur *user*

Berdasarkan gambar 4. 40 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 16

N (*node*) = 20

P (predikat *node*) = 4

Penyelesaian :

V(G) = E – N + 2

= 20 – 16 + 2

= 6

Predikat (P) = P + 1

= 4 + 1

= 5

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 6*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1–2–3–4–5–6–7–8–15–16

Path 2 = 1–2–3–4–5–6–9–10–15–16

Path 3 = 1–2–3–4–5–6–11–15–16

Path 4 = 1–2–3–4–5–6–13–14–15–16

Path 5 = 1–2–3–4–3–4–5–6–11–15–16

Path 6 = 1–2–3–4–5–6–9–10–9–10–15–16

(4) Grafik *matriks edukasi* berhasil

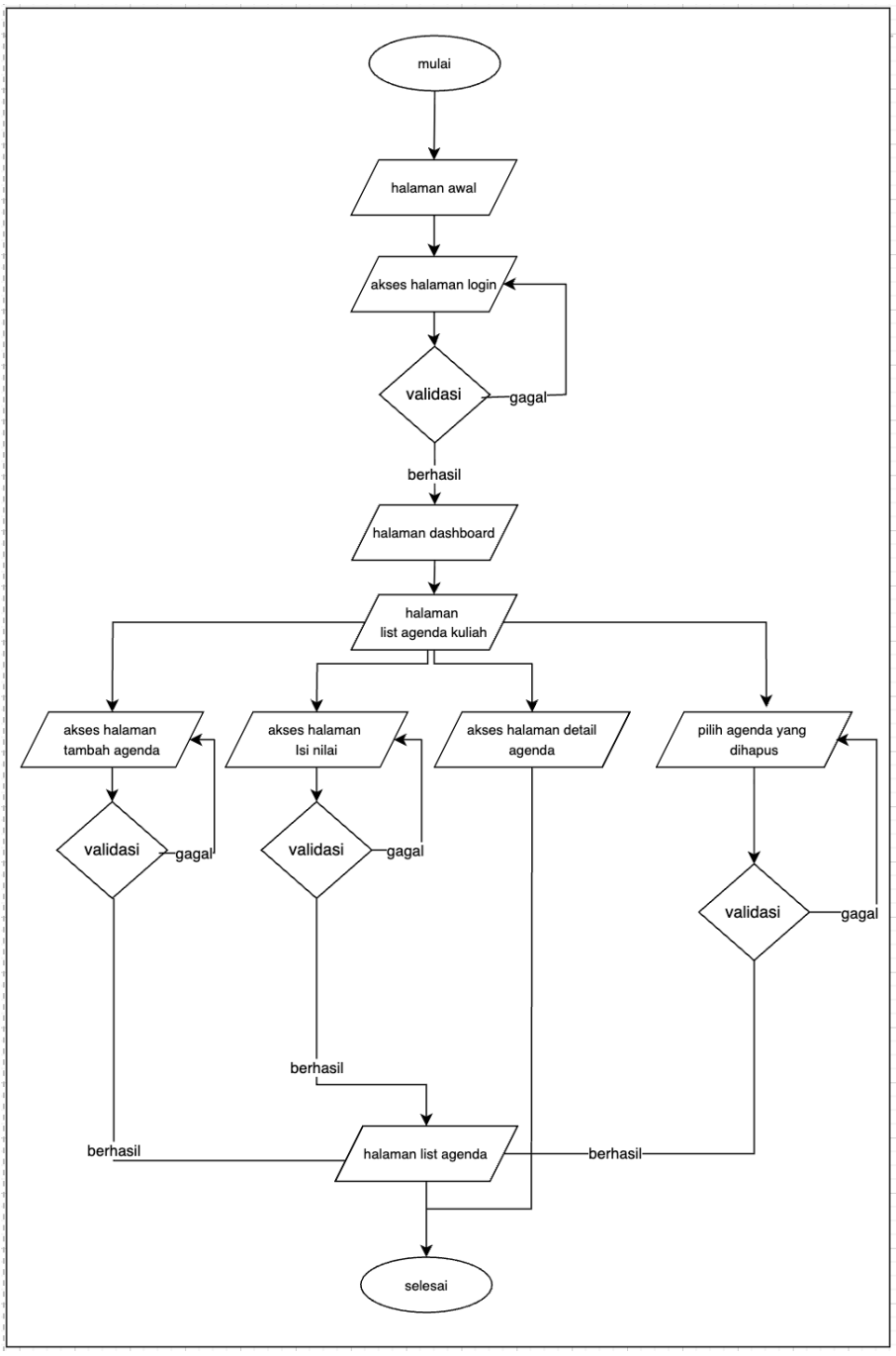
Tabel 4. 11 *Matriks* atur *user*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	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		1		1				4 – 1 = 3
7								1									1 – 1 = 0
8															1		1 – 1 = 0
9										1							1 – 1 = 0
10									1						1		2 – 1 = 1
11															1		1 – 1 = 0
12																	0

13													1			1 – 1 = 0
14												1		1		2 – 1 = 1
15															1	1 – 1 = 0
16																0

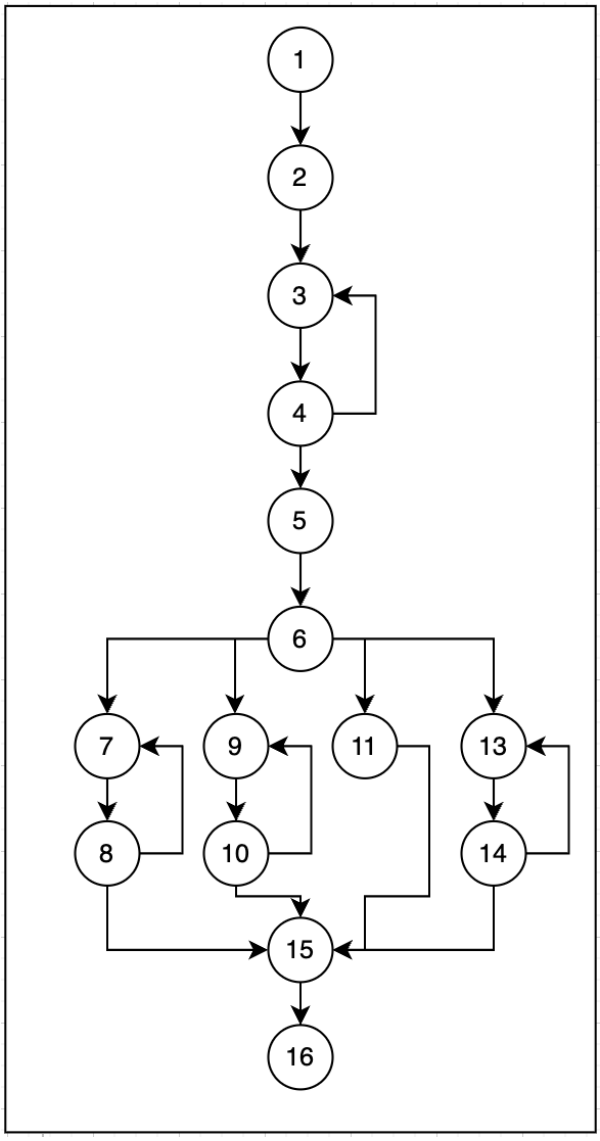
d. *Whitebox* pengujian atur agenda kuliah

1) *Flowchart*



Gambar 4. 41 *Flowchart* atur agenda kuliah

2) *Flowgraph*



Gambar 4. 42 *Flowgraph* atur agenda kuliah

Berdasarkan gambar 4. 42 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 16

N (*node*) = 20

P (predikat *node*) = 4

Penyelesaian :

V(G) = E – N + 2

= 20 – 16 + 2

= 6

Predikat (P) = P + 1

= 4 + 1

= 5

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 6*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1–2–3–4–5–6–7–8–15–16

Path 2 = 1–2–3–4–5–6–9–10–15–16

Path 3 = 1–2–3–4–5–6–11–15–16

Path 4 = 1–2–3–4–5–6–13–14–15–16

Path 5 = 1–2–3–4–3–4–5–6–11–15–16

Path 6 = 1–2–3–4–5–6–9–10–9–10–15–16

(4) Grafik *matriks edukasi* berhasil

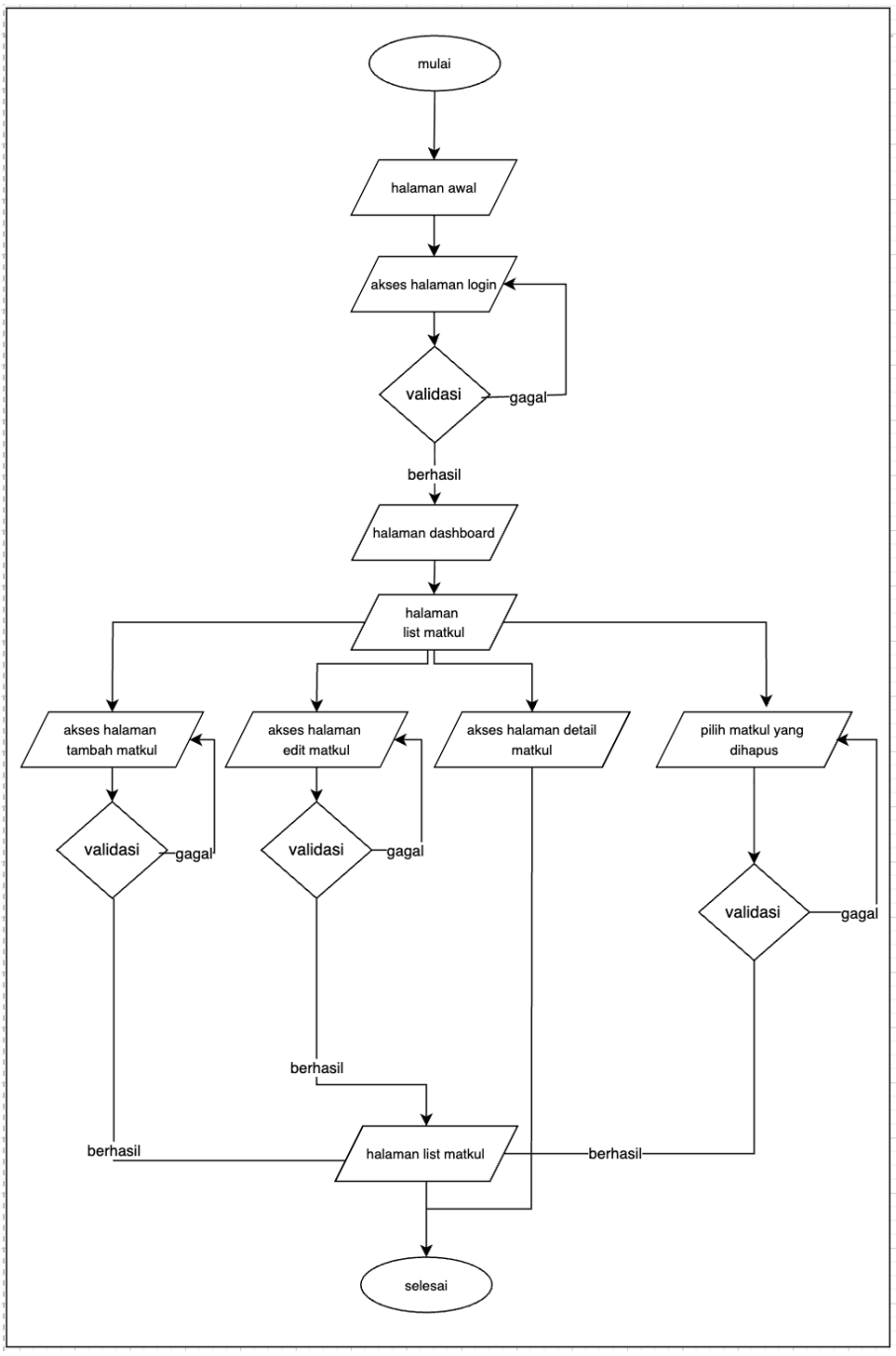
Tabel 4. 12 *Matriks* atur agenda kuliah

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	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		1		1				4 – 1 = 3
7								1									1 – 1 = 0
8															1		1 – 1 = 0
9										1							1 – 1 = 0
10									1						1		2 – 1 = 1
11															1		1 – 1 = 0
12																	0

13													1			1 – 1 = 0
14												1		1		2 – 1 = 1
15															1	1 – 1 = 0
16																0

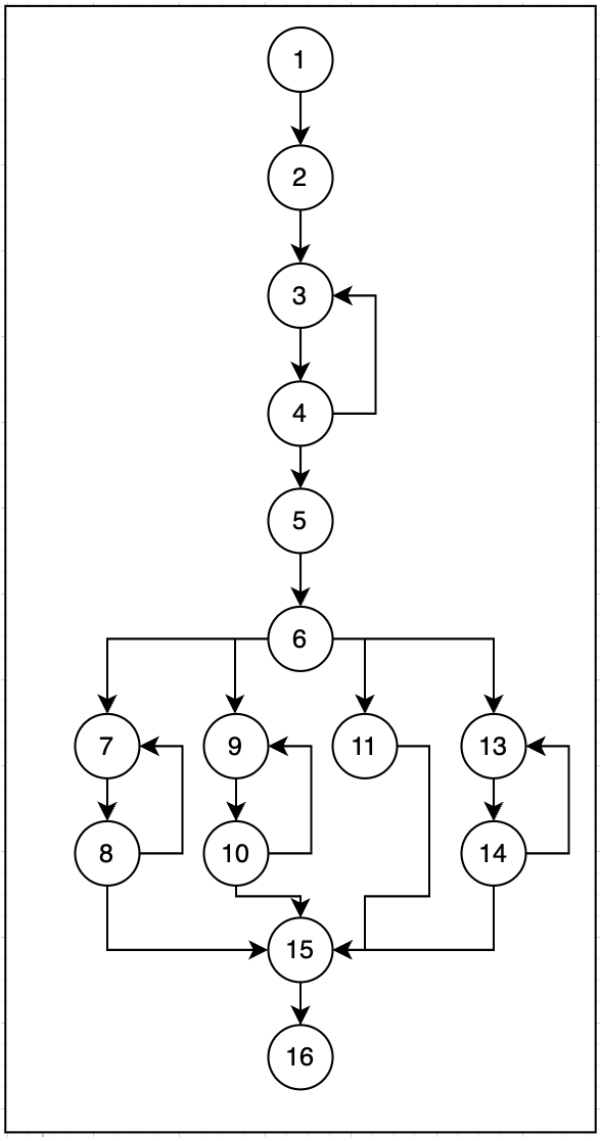
e. *Whitebox* pengujian atur matkul

1) *Flowchart*



Gambar 4. 43 *Flowchart* atur matkul

2) *Flowgraph*



Gambar 4. 44 *Flowgraph* atur matkul

Berdasarkan gambar 4. 44 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 16

N (*node*) = 20

P (predikat *node*) = 4

Penyelesaian :

V(G) = E – N + 2

= 20 – 16 + 2

= 6

Predikat (P) = P + 1

= 4 + 1

= 5

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 6*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1–2–3–4–5–6–7–8–15–16

Path 2 = 1–2–3–4–5–6–9–10–15–16

Path 3 = 1–2–3–4–5–6–11–15–16

Path 4 = 1–2–3–4–5–6–13–14–15–16

Path 5 = 1–2–3–4–3–4–5–6–11–15–16

Path 6 = 1–2–3–4–5–6–9–10–9–10–15–16

(4) Grafik *matriks edukasi* berhasil

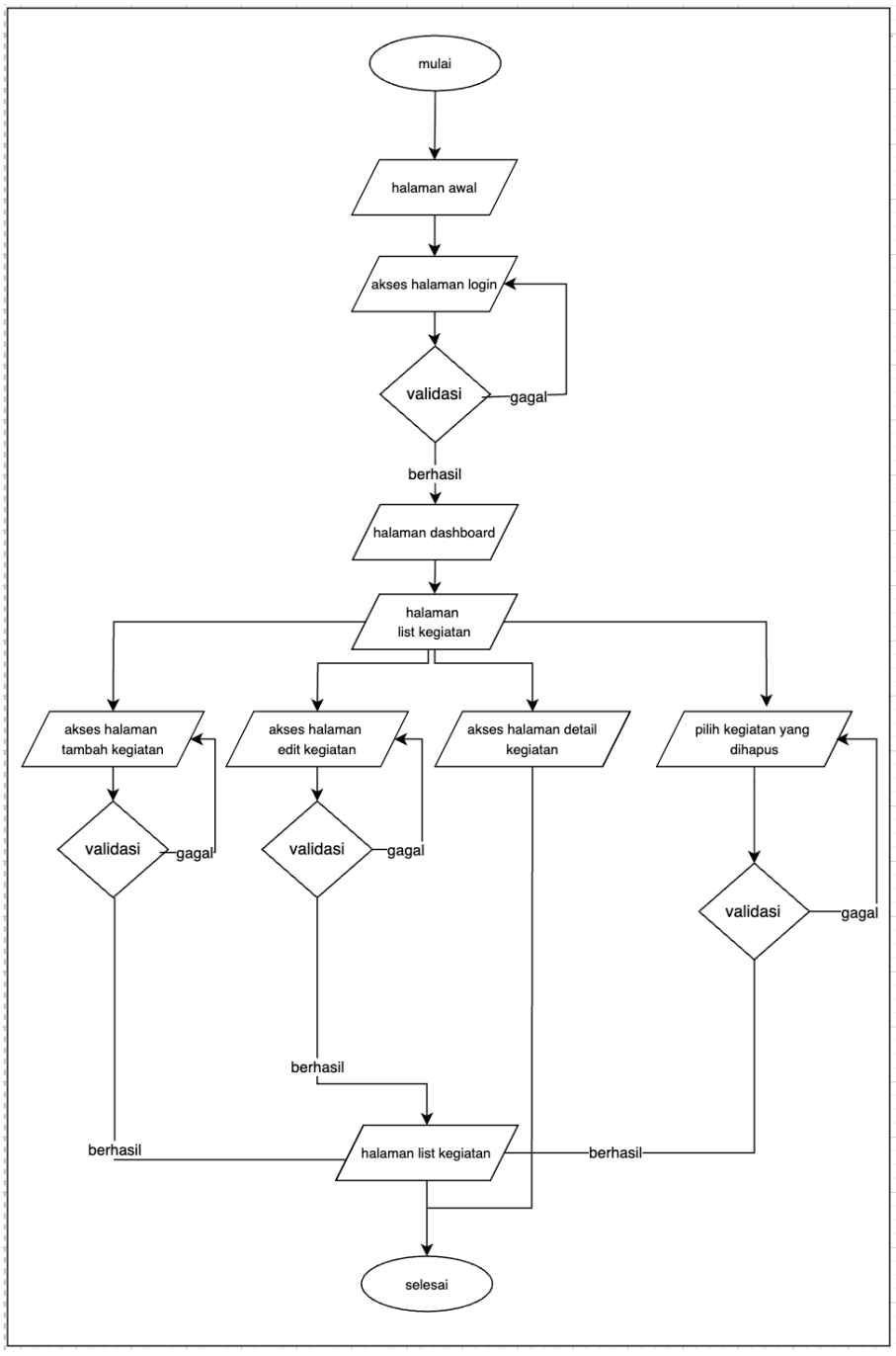
Tabel 4. 13 *Matriks* atur matkul

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	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		1		1				4 – 1 = 3
7								1									1 – 1 = 0
8															1		1 – 1 = 0
9										1							1 – 1 = 0
10									1						1		2 – 1 = 1
11															1		1 – 1 = 0
12																	0

13													1			1 – 1 = 0
14												1		1		2 – 1 = 1
15															1	1 – 1 = 0
16																0

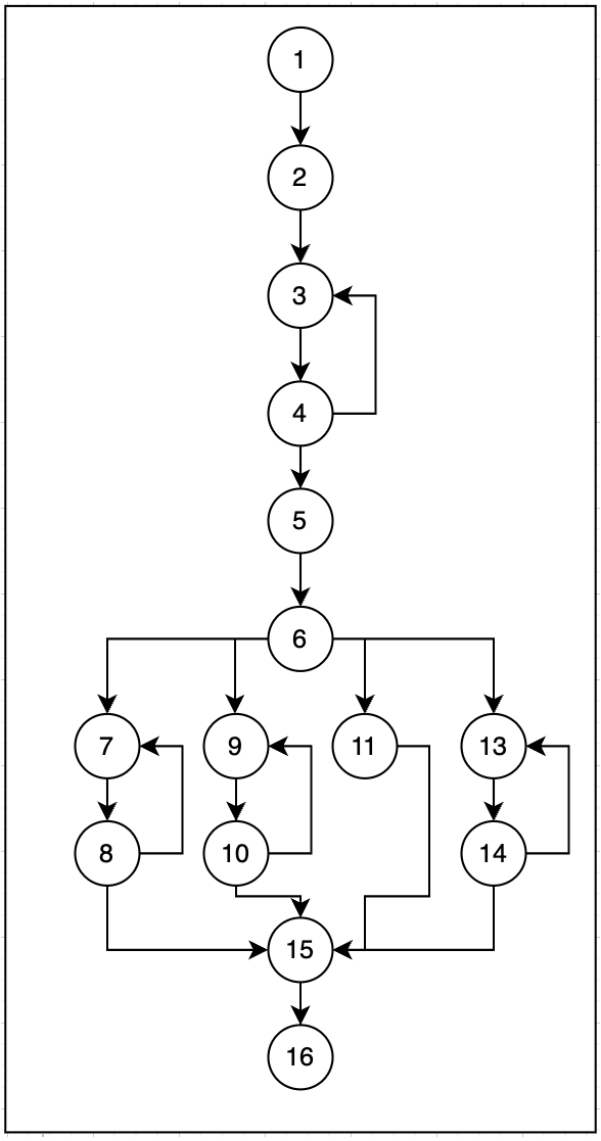
f. *Whitebox* pengujian atur kegiatan

1) *Flowchart*



Gambar 4. 45 *Flowchart* atur kegiatan

2) *Flowgraph*



Gambar 4. 46 *Flowgraph* atur kegiatan

Berdasarkan gambar 4. 46 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 16

N (*node*) = 20

P (predikat *node*) = 4

Penyelesaian :

V(G) = E – N + 2

= 20 – 16 + 2

= 6

Predikat (P) = P + 1

= 4 + 1

= 5

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 6*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1–2–3–4–5–6–7–8–15–16

Path 2 = 1–2–3–4–5–6–9–10–15–16

Path 3 = 1–2–3–4–5–6–11–15–16

Path 4 = 1–2–3–4–5–6–13–14–15–16

Path 5 = 1–2–3–4–3–4–5–6–11–15–16

Path 6 = 1–2–3–4–5–6–9–10–9–10–15–16

(4) Grafik *matriks edukasi* berhasil

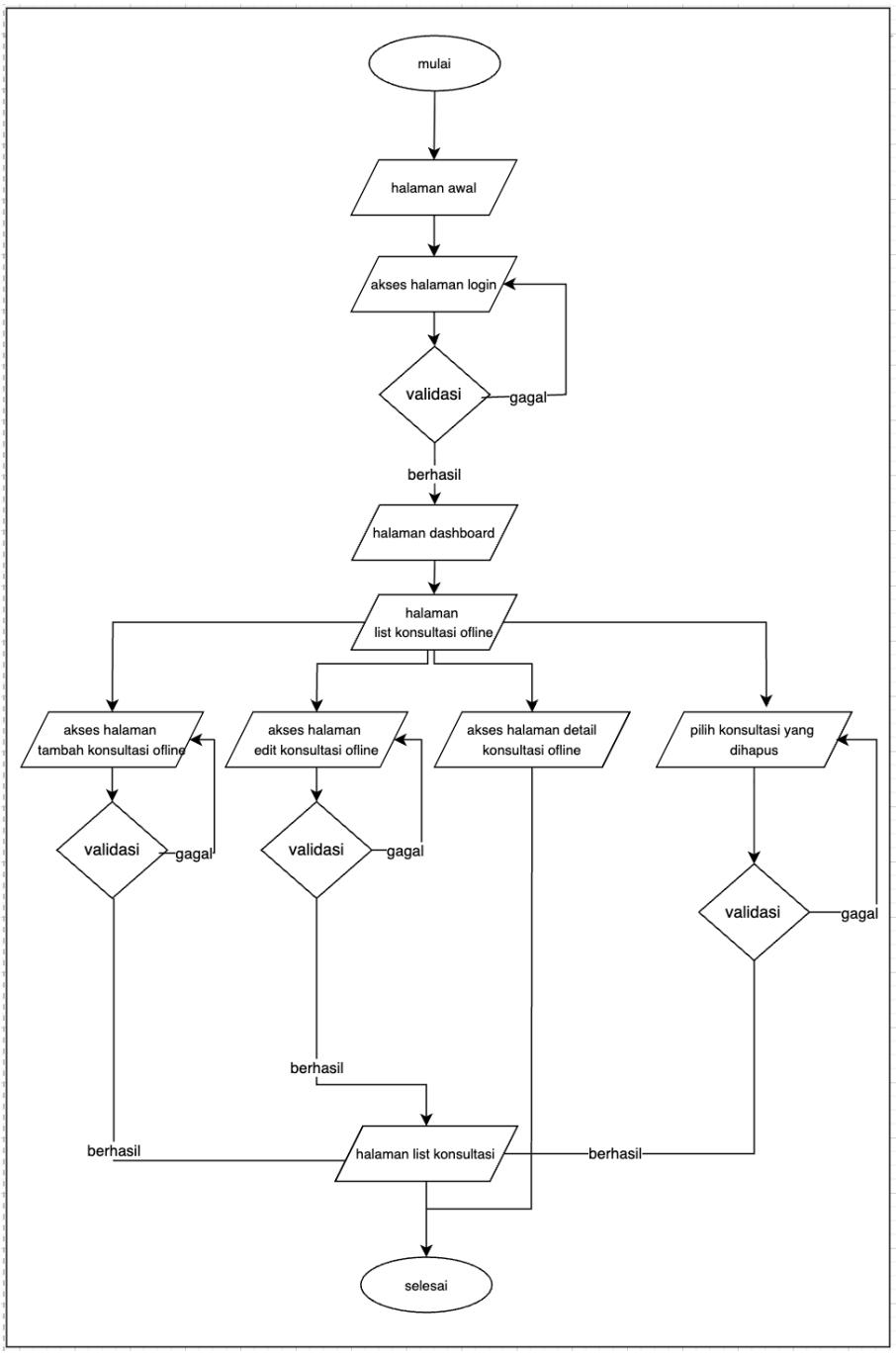
Tabel 4. 14 *Matriks* atur kegiatan

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	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		1		1				4 – 1 = 3
7								1									1 – 1 = 0
8															1		1 – 1 = 0
9										1							1 – 1 = 0
10									1						1		2 – 1 = 1
11															1		1 – 1 = 0
12																	0

13													1			1 – 1 = 0
14												1		1		2 – 1 = 1
15															1	1 – 1 = 0
16																0

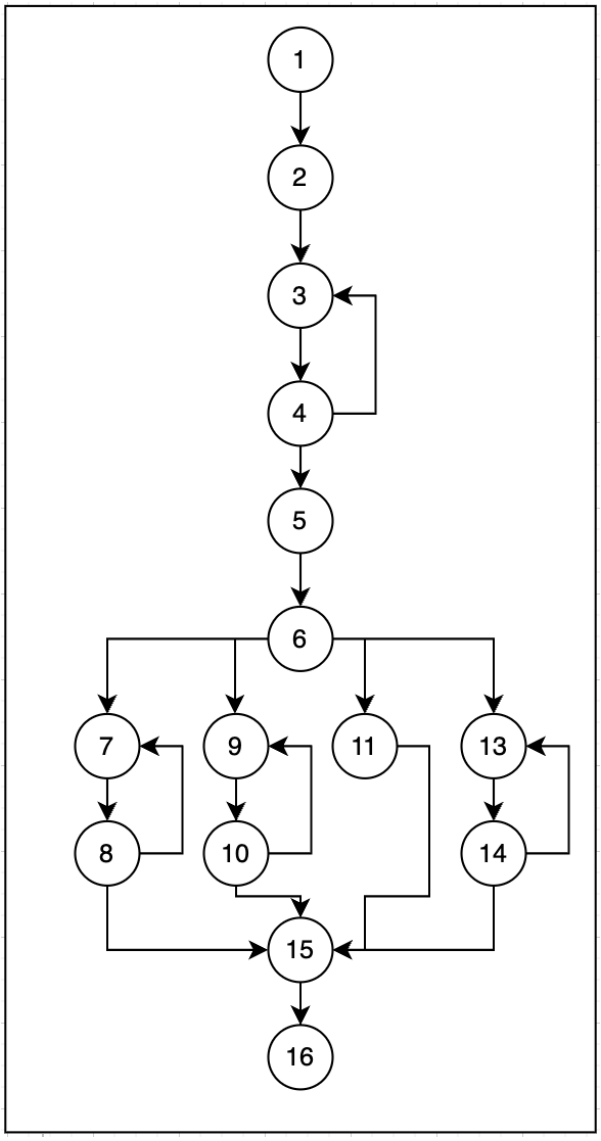
g. *Whitebox* pengujian konsultasi *offline*

1) *Flowchart*



Gambar 4. 47 *Flowchart* konsultasi *offline*

2) *Flowgraph*



Gambar 4. 48 *Flowgraph* konsultasi *offline*

Berdasarkan gambar 4. 48 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 16

N (*node*) = 20

P (predikat *node*) = 4

Penyelesaian :

V(G) = E – N + 2

= 20 – 16 + 2

= 6

Predikat (P) = P + 1

= 4 + 1

= 5

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 6*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1–2–3–4–5–6–7–8–15–16

Path 2 = 1–2–3–4–5–6–9–10–15–16

Path 3 = 1–2–3–4–5–6–11–15–16

Path 4 = 1–2–3–4–5–6–13–14–15–16

Path 5 = 1–2–3–4–3–4–5–6–11–15–16

Path 6 = 1–2–3–4–5–6–9–10–9–10–15–16

(4) Grafik *matriks edukasi* berhasil

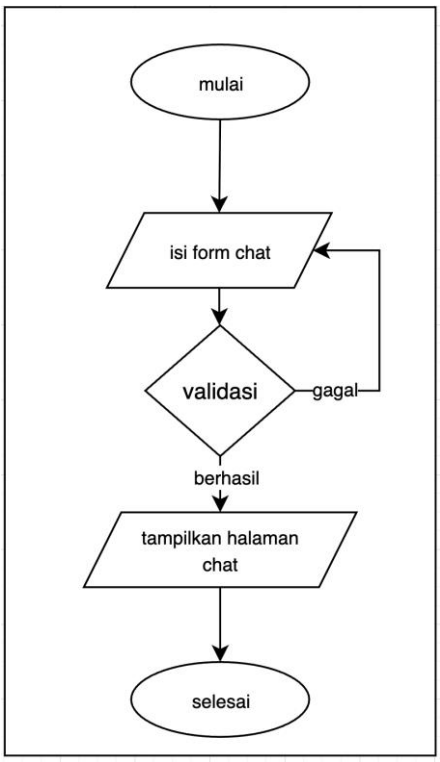
Tabel 4. 15 *Matriks* konsultasi *offline*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	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		1		1				4 – 1 = 3
7								1									1 – 1 = 0
8															1		1 – 1 = 0
9										1							1 – 1 = 0
10									1						1		2 – 1 = 1
11															1		1 – 1 = 0
12																	0

13													1			1 – 1 = 0
14												1		1		2 – 1 = 1
15															1	1 – 1 = 0
16																0

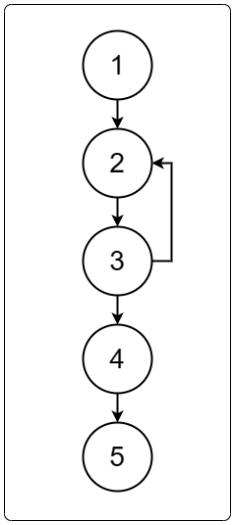
h. *Whitebox* pengujian konsultasi *online*

1) *Flowchart*



Gambar 4. 49 *Flowchart* konsultasi *online*

2) *Flowgraph*



Gambar 4. 50 *Flowgraph* konsultasi *online*

Berdasarkan gambar 4. 50 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexcity* V(G) pada *egde* dan *node*

Pada rumus : V(G) = E – N + 2

E (*edge*) = 5

N (*node*) = 5

P (predikat *node*) = 1

Penyelesaian :

V(G) = E – N + 2

= 5 – 5 + 2

= 2

Predikat (P) = P + 1

= 1 + 1

= 2

(2) Berdasarkan perhitungan *cyclomatic complexcity* dari *flowgraph* diatas memiliki *region = 2*

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 2

Path 2 = 1 – 2 – 3 – 4 – 5

(4) Grafik *matriks login* berhasil

Tabel 4. 16 *Matriks* konsultasi *online*

	1	2	3	4	5	E-1
1		1				1 - 1 = 0
2			1			1 – 1 = 0
3		1		1		2 – 1 = 1
4					1	1 – 1 = 0

5						0
	SUM (E + 1)					1 + 1 = 2

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan penelitian yang telah dilakukan, kesimpulan yang dapat ditarik adalah sebagai berikut:

1. Dalam penelitian yang telah dilakukan, penulis berhasil membuat sebuah sistem untuk membantu penasehat akademik (pa) fakultas teknik dalam mengelola konsutasi mahasiswanya.
2. Sistem ini dibangun dengan menggunakan Bahasa pemograman *PHP* untuk *backend* dan *React js* untuk *frontend* yang diintegrasikan melalui api. Visual *Studio code* digunakan sebagai *text Editor* dan *Mysql* sebagai sistem manajemen *database*.

B. Saran

Pada penelitian ini penulis menyadari bahwa masih ada beberapa kekurangan yang sangat di perlu perbaikan dan pengembangan di penelitian selanjutnya. Oleh karena itu, penulis memiliki beberapa saran untuk pengembangan selanjutnya, sebagai berikut:

1. Integrasi Kecerdasan buatan(AI) untuk mempermudah jalannya aplikasi.

DAFTAR PUSTAKA

Adi nugroho. (2011). *Perancangan dan implementasi sistem basis data*. yokyakarta: Andi.

Andi widi (2021) Ponsel cerdas (bahasa inggris: *Smartphone*) adalah telepon genggam yang mempunyai kemampuan dengan penggunaan dan fungsi yang menyerupai komputer

Choirul Anna Nur Afifah dan Salsabila Romadona (2022) dalam penelitianya yang berjudul "Pengembangan Aplikasi Sehatgram Berbasis *Smartphone* Dalam Mengatasi Dan Mencegah Obesitas Pada Remaja"

Rita Astari (2018) dalam penelitianya yang berjudul "Rancang Bangun Aplikasi Pengelolaan Hidup Sehat Berbasis *Android*"

Bidanku. Proses Terjadinya Kehamilan. http://bidanku.com/terjadinya-kehamilan (30 April 2024).

Chapman, Vicky. Asuhan Kebidanan Persalinan dan Kelahiran. Jakarta : Penerbit Buku Kedokteran EGC. 2006.

Departemen Agama. Al Qur'an Dan Terjemahannya. Semarang: CV. Toha Putra.1989.

Hasnawati. Aplikasi Penghitung Kalender Menstruasi Otomatis Berdasarkan Hari Pertama Haid Terakhir dengan Platform *Android*. Makassar : UIN Alauddin Makassar. 2013.

Herry.ActivityDiagram.http://her0satr.staff.ub.ac.id/category/uncategorized/page/3/ (30 April 2024)

Hidayat, Alimul. Asuhan Neonatus, Bayi, & Balita. Jakarta : Penerbit BukuKedokteran EGC. 2009.