

BAB I

PENDAHULUAN

A. Latar Belakang

Universitas Muhammadiyah Parepare merupakan universitas yang berdiri tahun 1999 di Provinsi Sulawesi Selatan Kota Parepare. sebagai sebuah lembaga pendidikan tinggi memiliki tujuan untuk memuliakan dan mencerdaskan kehidupan bangsa dan berkomitmen menyelenggarakan pendidikan tinggi nasional yang melaksanakan proses pembelajaran, penelitian, pengabdian kepada masyarakat, dan pembudayaan yang luhur. Dalam mencapai tujuan dan menjaga komitmen tersebut, berbagai cara ditempuh agar proses pendidikan berjalan dengan baik dan lancar. Mulai dari peningkatan kualitas tenaga pendidik dan kependidikan sampai dengan kelengkapan fasilitas penunjang proses pembelajaran. Melalui usaha peningkatan kualitas SDM dan fasilitas penunjang tersebut diharapkan kualitas pelayanan terhadap pengguna, dalam hal ini adalah mahasiswa, dapat tercapai dengan baik.

Salah satu bentuk peningkatan kualitas SDM adalah peningkatan pelayanan terhadap mahasiswa. Sedangkan dari sisi fasilitas salah satunya adalah dukungan sistem yang dapat mendukung proses pelayanan tersebut dapat berjalan optimal. Namun dalam penerapannya belum didukung oleh sistem yang dapat mengakomodir pelayanan tersebut, sehingga masih muncul permasalahan. Beberapa permasalahan yang sering muncul antara lain antrian konfirmasi ataupun komplain yang panjang, berkas komplain kadang terselip, dan mahasiswa harus

datang berulang kali untuk menanyakan status permasalahan yang dihadapinya. Hal tersebut tentunya membuat tingkat kepuasan mahasiswa menjadi menurun.

Berdasarkan uraian di atas, untuk menjaga kepuasan mahasiswa diperlukan sebuah solusi dari sisi sistem informasi. Solusi tersebut berupa pengembangan aplikasi *mobile* berbasis android untuk mengelola bantuan secara elektronik dengan fitur yang memungkinkan pengguna mengakses aplikasi secara *offline*. Solusi dengan fitur ini sangat diperlukan melihat kondisi dilapangan banyak penggunaan *smartphone* berbasis android, namun kualitas layanan internet dari penyedia layanan seluler yang terkadang masih sering naik turun.

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka penelitian ini akan mengambil sebuah topik berjudul “**Aplikasi Pengelola Bantuan Mahasiswa Berbasis Android Di Universitas Muhammadiyah Parepare**”. Dari sistem yang akan dibuat diharapkan kinerja lembaga menjadi lebih baik serta memberikan pelayanan yang optimal.

B. Rumusan Masalah

Dari latar belakang masalah yang ditemukan diatas maka pokok permasalahan yang terdapat dalam penelitian ini adalah Sebagai Berikut:

Bagaimana membuat Aplikasi Pengelola Bantuan Mahasiswa Berbasis Android Di Universitas Muhammadiyah Parepare ?

C. Batasan Masalah

Batasan masalah dalam penelitian tugas akhir ini adalah sebagai berikut :

1. penelitian ini meliputi pengelola bantuan beasiswa mahasiswa.
2. Aplikasi ini hanya menggunakan Bahasa pemrograman PHP dan android
3. Pengambilan data pada Universitas Muhammadiyah Parepare.

D. Tujuan Penelitian

Tujuan penelitian ini adalah Pengelola Bantuan Mahasiswa Berbasis Android Di Universitas Muhammadiyah Parepare dengan baik.

E. Manfaat Penelitian

Adapun manfaat penelitian yaitu sebagai berikut :

1. Manfaat bagi mahasiswa
Sebagai penerapan teori yang telah di dapatkan dari bangku kuliah dan diharapkan dapat menambah pengetahuan dan wawasan tentang bagaimana cara mengetahui informasi terkait beasiswa pada Universitas Muhammadiyah Parepare.
2. Manfaat bagi penulis
Menambah pengetahuan, pengalaman, serta wawasan tentang metode perkembangan dan melatih diri dalam menyelesaikan masalah untuk maju dan berkembang.

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 penelitiannya 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 penelitiannya 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. Beasiswa

Beasiswa adalah bentuk bantuan keuangan yang diberikan untuk membantu penerimanya dalam mengejar pendidikan sesuai bidang yang mereka kuasai. Bantuan keuangan ini dapat diberikan oleh lembaga pendidikan (sekolah maupun perguruan tinggi), yayasan, perusahaan, atau organisasi lain yang memiliki minat dalam mendukung perkembangan pendidikan dan mendorong pencapaian akademik yang tinggi. Untuk mendapatkan beasiswa, seorang pemberi beasiswa biasanya akan menetapkan sejumlah kriteria yang harus dipenuhi. Kriteria ini bisa mencakup prestasi akademik dengan standar nilai tinggi, bakat khusus di bidang tertentu seperti olahraga, seni, atau musik, kepemimpinan, keterlibatan dalam kegiatan sosial, hingga faktor keuangan seperti keadaan ekonomi. Jika dinyatakan berhasil, penerima beasiswa akan mendapatkan sejumlah benefit sesuai yang dijanjikan oleh pemberi beasiswa, seperti uang kuliah, buku, alat tulis, hingga biaya hidup sehari-hari. Tidak hanya itu, program beasiswa juga dapat memberikan kesempatan tambahan bagi para penerimanya, seperti kesempatan magang, mentorship, atau akses ke jaringan profesional yang dapat meningkatkan peluang karir mereka di masa depan. Penting untuk dicatat bahwa beasiswa bukanlah pinjaman, yang berarti penerima beasiswa tidak diharuskan untuk mengembalikan dana yang diberikan. Kendati begitu, pemberi beasiswa biasanya mengajukan sejumlah persyaratan yang harus dipenuhi oleh penerima beasiswa, seperti mempertahankan pencapaian akademik yang baik atau berpartisipasi dalam kegiatan yang ditetapkan oleh pemberi beasiswa. Secara umum, beasiswa adalah sebuah program yang dapat membantu para individu dengan kemampuan finansial

terbatas untuk mengakses pendidikan. Beasiswa memiliki peran penting dalam mendorong dan mendukung pengembangan sumber daya manusia yang berkualitas, memperluas peluang pendidikan, dan menciptakan dampak positif dalam masyarakat.

C. Jenis Beasiswa

1. Beasiswa KIP Kuliah

Macam-macam beasiswa yang bisa kamu coba daftarkan dan tersedia dalam perguruan tinggi adalah KIP Kuliah. Beasiswa tersebut merupakan program baru yang dibuat oleh pemerintah. Tujuannya adalah untuk menggantikan beasiswa bidikmisi dengan anggaran yang lebih besar. Tidak heran jika syarat dan targetnya sama dengan beasiswa untuk bidikmisi.

Biasanya, pendaftaran KIP Kuliah bersamaan dengan pendaftaran dalam SNMPTN. Jika kamu mendapatkan beasiswa tersebut, maka kamu bisa mendapat biaya hidup sebesar Rp 700.000 setiap semester. Sementara biaya pendidikan mencapai Rp 2.400.000 setiap semester yang dilakukan selama 4 tahun.

2. Beasiswa Berupa Bantuan Biaya Mahasiswa

Macam-macam beasiswa di perguruan tinggi yang bisa kamu coba adalah Bantuan Biaya Mahasiswa. Jenis beasiswa ini berasal dari Dirjen Dikti Kemdiknas dengan tujuan membantu ekonomi para mahasiswanya. Syarat yang dibutuhkan dalam beasiswa ini hampir sama dengan beasiswa PPA. Namun, ada beberapa syarat khusus lainnya, antara lain:

- Mempunyai nilai IPK minimal 2,5
- Melampirkan bukti surat keterangan tanda tidak mampu dari kepala desa atau lurah

D. Teknologi yg digunakan

1. 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 *elektronik (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*.

2. 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 berorientasi 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 pemrograman 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 *Smalltalk*, 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 berorientasi objek

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

3. Android (SDK)

Android SDK (*Software Development Kit*) adalah tools API (*Application 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 *platform* aplikasi netral, Android memberi anda kesempatan untuk membuat aplikasi yang kita butuhkan yang bukan aplikasi bawaan *Handphone/smartphone*. Beberapa fitur Android adalah :

- a. *Framework* 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 spesifikasi *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*.

4. 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.

5. 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 berorientasi 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) *Behaviorial 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*)

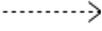
Ada empat *relationship* (hubungan) dalam *Unified Modelling Language* (UML):

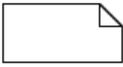
- 1) Ketergantungan (*dependency*) adalah hubungan dimana perubahan yang terjadi pada suatu elemen independent akan mempengaruhi elemen yang bergantung padanya.
- 2) Asosiasi adalah apa dan bagaimana yang menghubungkan antara objek satu dengan yang lainnya. Suatu bentuk asosiasi adalah agregasi yang menampilkan hubungan suatu objek dengan bagian-bagiannya.
- 3) Generalisasi adalah hubungan dimana objek anak berbagai perilaku dan struktur data dari objek yang ada di atasnya (objek induk). Arah dari objek induk ke objek anak dinamakan spesialisasi 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 :

Tabel 2. 1 Simbol *Use Case* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan

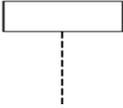
NO	GAMBAR	NAMA	KETERANGAN
			struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>Use Case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>Use Case</i> target memperluas perilaku dari <i>Use Case</i>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

Tabel 2. 2 Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).

2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Tabel 2. 3 Simbol *Sequence Diagram*

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

Tabel 2. 4 Simbol *StateChart Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>State</i>	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		<i>Initial Pseudo State</i>	Bagaimana objek dibentuk atau diawali
3		<i>Final State</i>	Bagaimana objek dibentuk dan dihancurkan
4		<i>Transition</i>	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		<i>Node</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Tabel 2. 5 Simbol *Actifity* Diagram

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

BAB III

METODOLOGI PENELITIAN

A. Waktu Dan Lokasi Penelitian

Pelaksanaan penelitian dilakukan pada bulan Oktober 2023 dan lokasi Universitas Muhammadiyah Parepare.

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 Perancangan Aplikasi Pengelola Bantuan Mahasiswa Berbasis Android Di Universitas Muhammadiyah Parepare adalah sebuah laptop dengan spesifikasi minimal yang dapat dilihat pada tabel 4 berikut :

Tabel 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

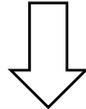
Perangkat lunak yang digunakan untuk membangun aplikasi ini dapat dilihat pada Tabel 5 berikut :

Tabel 3. 2 Spesifikasi Minimal Perangkat Lunak

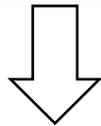
Jenis	Spesifikasi
Sistem Operasi	Windows 11
Bahasa Pemrograman	PHP dan Android Studio
Database	MySql
Tools	Notepad++, Netbeans, XAMPP

E. Kerangka Pikir

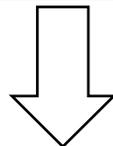
Salah satu bentuk peningkatan kualitas SDM adalah peningkatan pelayanan terhadap mahasiswa. Sedangkan dari sisi fasilitas salah satunya adalah dukungan sistem yang dapat mendukung proses pelayanan tersebut dapat berjalan optimal.



Beberapa permasalahan yang sering muncul antara lain antrian konfirmasi ataupun komplain yang panjang, berkas komplain kadang terselip, dan mahasiswa harus datang berulang kali untuk menanyakan status permasalahan yang dihadapinya. Hal tersebut tentunya membuat tingkat kepuasan mahasiswa menjadi menurun.

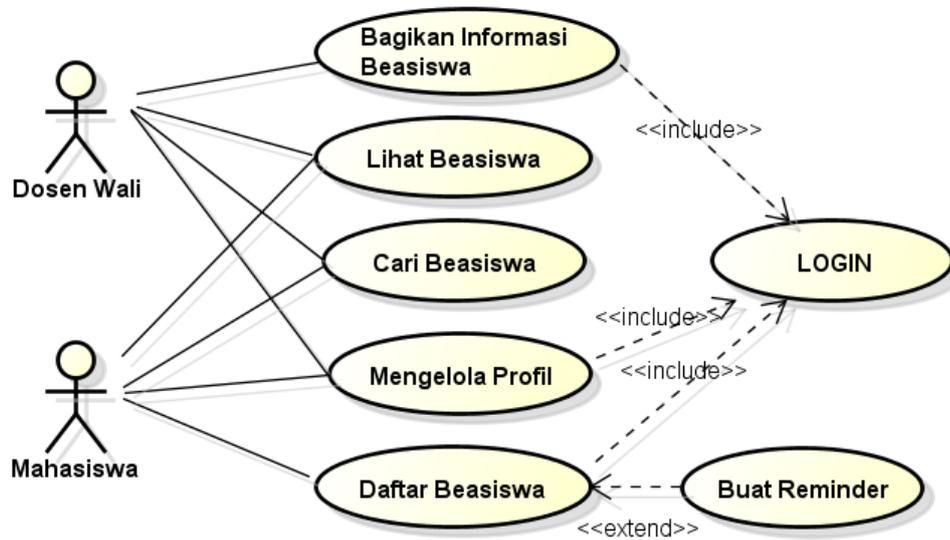


Solusi tersebut berupa pengembangan aplikasi *mobile* berbasis android untuk mengelola bantuan secara elektronik dengan fitur yang memungkinkan pengguna mengakses aplikasi secara *offline*. Solusi dengan fitur ini sangat diperlukan melihat kondisi dilapangan banyak penggunaan *smartphone* berbasis android



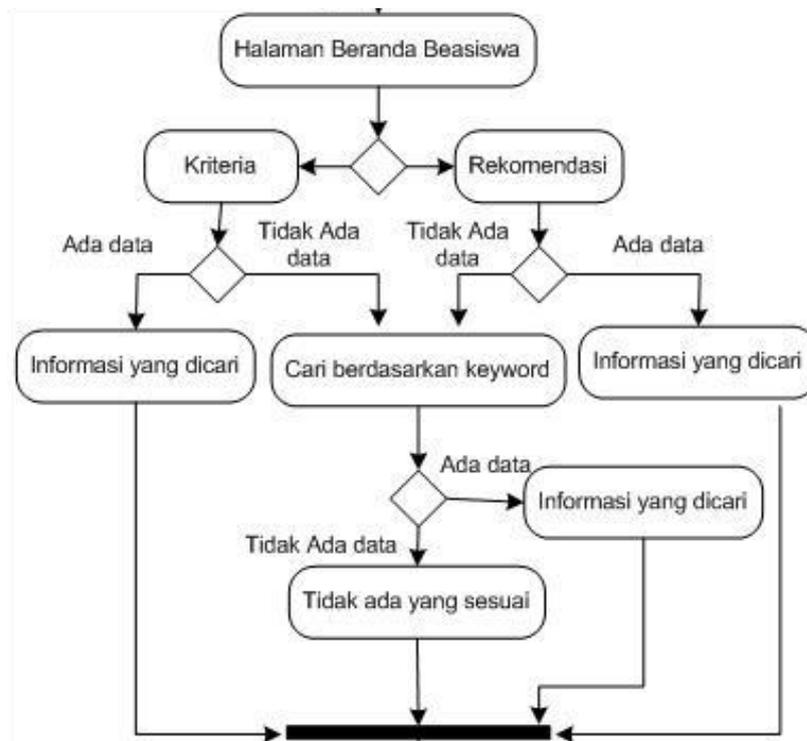
maka penelitian ini akan mengambil sebuah topik berjudul “Aplikasi Pengelola Bantuan Mahasiswa Berbasis Android Di Universitas Muhammadiyah Parepare

F. Desain Sistem



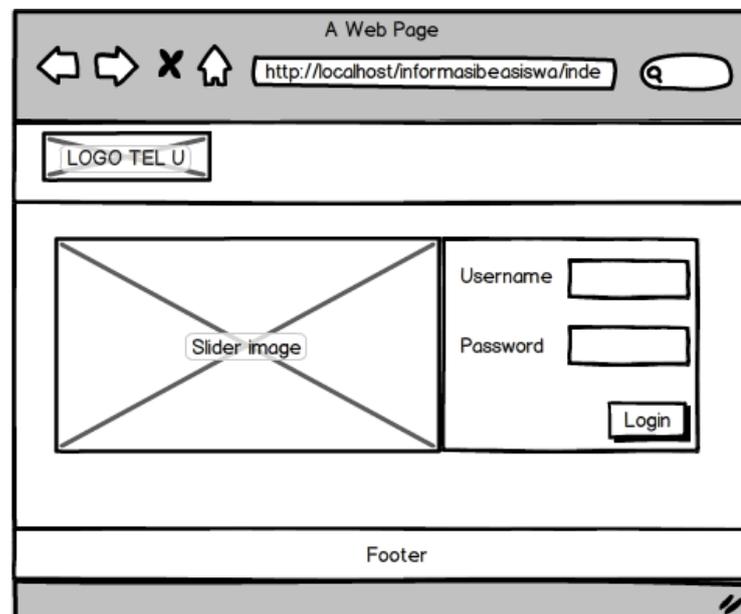
Gambar 3. 1 Use case

Pada gambar diatas digambarkan bahwa use case memiliki 2 aktor yaitu Dosen Wali dan Mahasiswa. Aktivitas yang dilakukan dosen wali meliputi : Lihat Beasiswa, Bagikan Informasi Beasiswa, Cari Beasiswa dan Mengelola Profil. Aktivitas yang dilakukan mahasiswa hampir sama dengan dosen wali kecuali Daftar Beasiswa. Aktivitas Bagikan Informasi Beasiswa, mengelola profil dan daftar beasiswa harus include ke login. Buat reminder extend ke Daftar beasiswa.

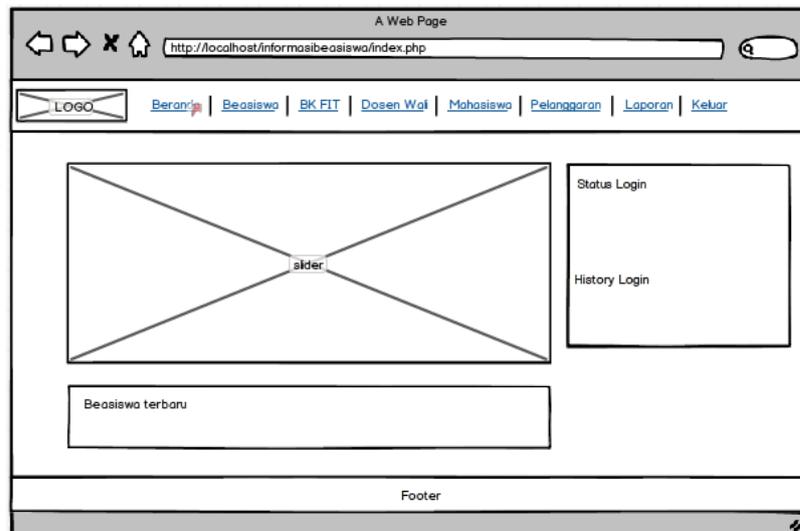


Gambar 3. 2 Diagram

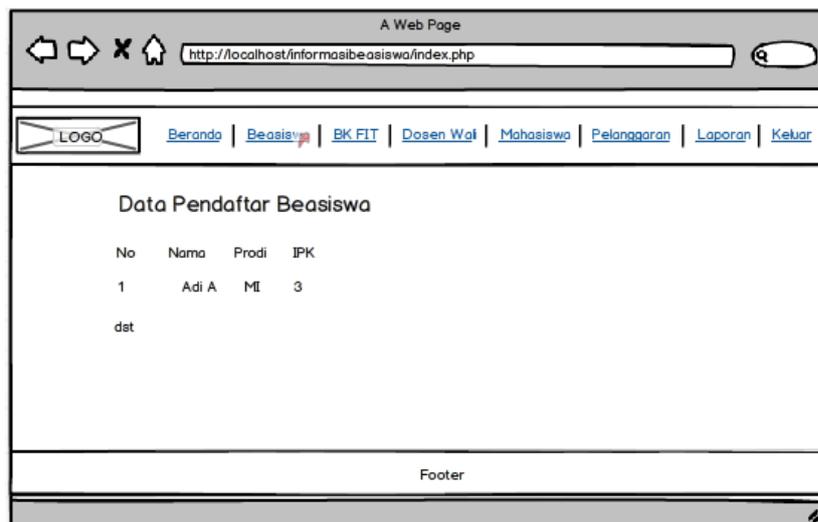
G. Rancangan Interface



Gambar 3. 3 rancangan *login*



Gambar 3. 4 rancangan *dashboard*



Gambar 3. 5 rancangan halaman

BAB IV

PEMBAHASAN DAN HASIL

A. Analisis Aliran Data UML

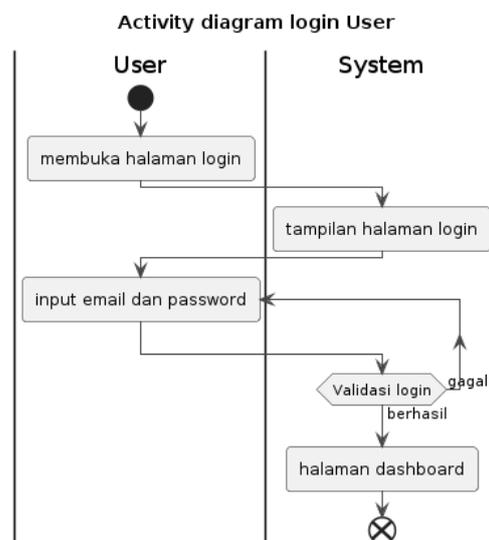
Dalam bagian ini, akan dibahas dua diagram UML yang menggambarkan aliran data dan interaksi dalam aplikasi pengelola bantuan mahasiswa, yaitu *Activity diagram* dan *Sequence diagram*.

1. Activity diagram

Activity diagram digunakan untuk memodelkan aliran kerja atau aliran kontrol dari sebuah proses bisnis atau *use case*. Diagram ini menunjukkan aktivitas-aktivitas yang terjadi dalam sistem dan urutan eksekusinya.

a. Activity diagram login

1) Activity diagram login

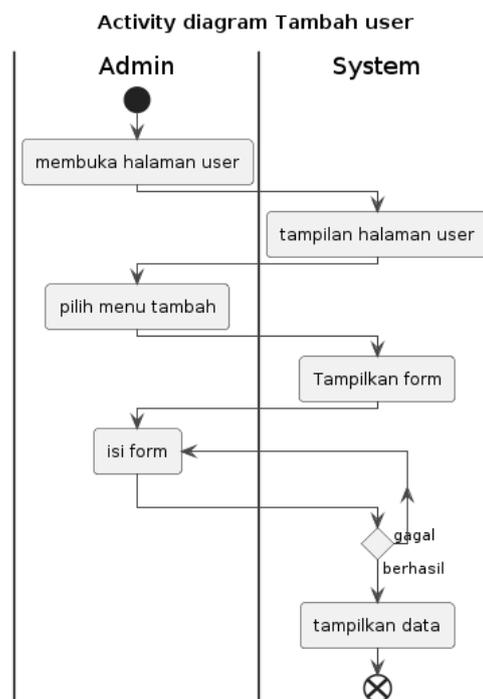


Gambar 4. 1 Activity diagram login

Pada gambar 4. 1 menjelaskan cara masuk login. *pengguna* harus mengunjungi website terlebih dahulu. Setelah itu, formulir login akan muncul di layar, mengharuskan *pengguna* memasukkan alamat email dan kata sandi pengguna. Setelah itu, sistem akan memvalidasi data dan masuk ke halaman *dashboard* jika semuanya sudah benar, kembali ke halaman login jika tidak.

b. Activity diagram admin

1) Activity diagram tambah pengguna

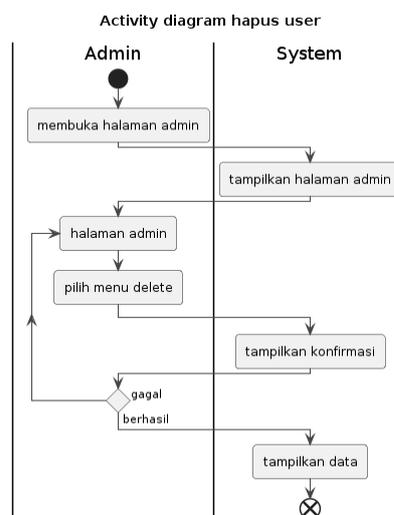


Gambar 4. 2 Activity diagram tambah pengguna

Pada gambar 4. 2 menjelaskan langkah-langkah yang digunakan oleh *admin* untuk menambahkan user. Setelah *admin* membukanya, sistem menampilkan halaman *admin*. Sistem kemudian menampilkan halaman menu yang dipilih admin, dan admin mengklik tombol

tambah pengguna setelah memilih item menu tambah pengguna. Sistem kemudian akan menampilkan form tambah pengguna. Setelah itu admin diminta untuk mengisi form tambah pengguna. Setelah selesai, sistem akan memverifikasi info pengguna dan, jika semuanya sudah diperiksa, menampilkan halaman menu untuk pengguna yang baru ditambahkan. *Admin* perlu mengisi kembali formulir penambahan pengguna dengan benar jika tidak berfungsi.

2) *Activity diagram hapus pengguna*

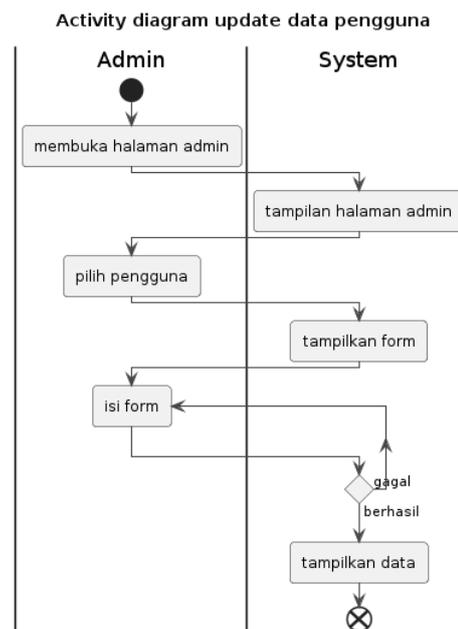


Gambar 4. 3 *Activity diagram hapus pengguna*

Pada gambar 4. 3 menjelaskan prosedur yang digunakan *admin* untuk menghapus user. Setelah *admin* membukanya, sistem menampilkan halaman *admin*. Selanjutnya *admin* memilih pengguna mana yang akan dihapus, sistem akan menampilkan pengguna yang dipilih *admin*, dan *admin* mengklik opsi hapus pengguna. Setelah itu, sistem akan menampilkan konfirmasi penghapusan. Admin kemudian

memverifikasi. Pengguna akan dihapus jika admin menyetujui; jika tidak, pengguna akan dibatalkan.

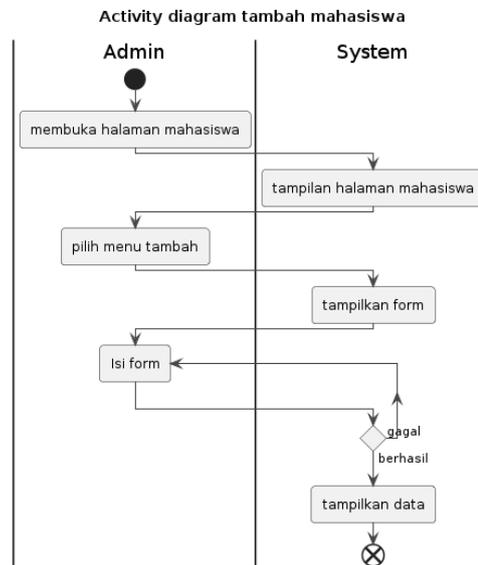
3) Activity diagram edit pengguna



Gambar 4. 4 Activity diagram edit pengguna

Pada gambar 4.4 menjelaskan untuk mengubah data pengguna, *admin* harus menuju ke halaman admin dan *admin* harus menekan tombol edit untuk menuju ke form pengisian data pengguna. Tindakan ini akan meminta sistem untuk menampilkan form pengisian data pengguna. Kemudian, setelah pengguna mengisi form, sistem akan memvalidasi apakah form telah terisi atau belum. Jika form telah terisi maka sistem akan menuju kembali ke halaman admin dan jika belum terisi maka sistem akan kembali menampilkan form pengisiannya serta menampilkan error.

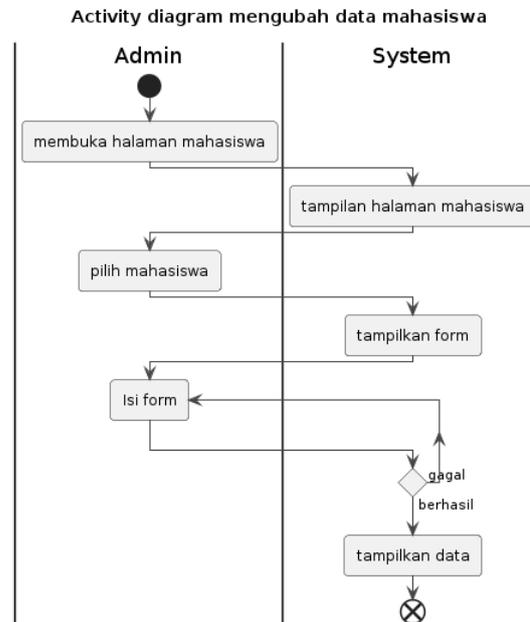
4) Activity diagram tambah mahasiswa



Gambar 4. 5 Activity diagram tambah mahasiswa

Pada gambar 4. 5 menjelaskan langkah-langkah yang digunakan oleh *admin* untuk menambahkan mahasiswa. Setelah *admin* membukanya, sistem menampilkan halaman mahasiswa. Sistem kemudian menampilkan halaman menu yang dipilih admin, dan admin mengklik tombol tambah setelah memilih item menu tambah. Sistem kemudian akan menampilkan form tambah mahasiswa. Setelah itu admin diminta untuk mengisi form tambah mahasiswa. Setelah selesai, sistem akan memverifikasi info mahasiswa dan, jika semuanya sudah diperiksa, menampilkan halaman menu untuk mahasiswa yang baru ditambahkan. *Admin* perlu mengisi kembali formulir penambahan mahasiswa dengan benar jika tidak berfungsi.

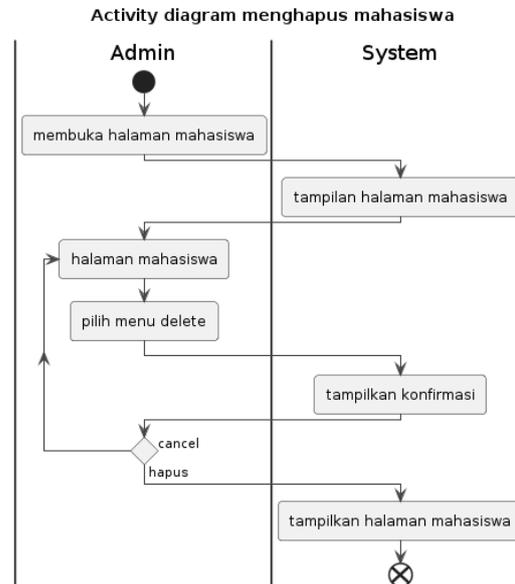
5) Activity diagram edit data mahasiswa



Gambar 4. 6 Activity diagram edit data mahasiswa

Pada gambar 4.6 menjelaskan untuk mengubah data mahasiswa, *admin* harus menuju ke halaman admin dan *admin* harus menekan tombol edit untuk menuju ke form pengisian data mahasiswa. Tindakan ini akan meminta sistem untuk menampilkan form pengisian data mahasiswa. Kemudian, setelah mahasiswa mengisi form, sistem akan memvalidasi apakah form telah terisi atau belum. Jika form telah terisi maka sistem akan menuju kembali ke halaman admin dan jika belum terisi maka sistem akan kembali menampilkan form pengisiannya serta menampilkan error.

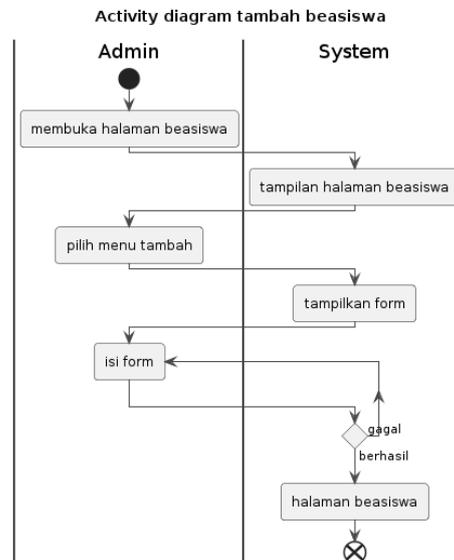
6) Activity diagram hapus mahasiswa



Gambar 4. 7 Activity diagram hapus mahasiswa

Pada gambar 4. 7 menjelaskan prosedur yang digunakan *admin* untuk menghapus mahasiswa. Setelah *admin* membukanya, sistem menampilkan halaman *admin*. Selanjutnya *admin* memilih mahasiswa mana yang akan dihapus, sistem akan menampilkan mahasiswa yang dipilih *admin*, dan *admin* mengklik opsi hapus mahasiswa. Setelah itu, sistem akan menampilkan konfirmasi penghapusan. Admin kemudian memverifikasi. Mahasiswa akan dihapus jika admin menyetujui; jika tidak, mahasiswa akan dibatalkan.

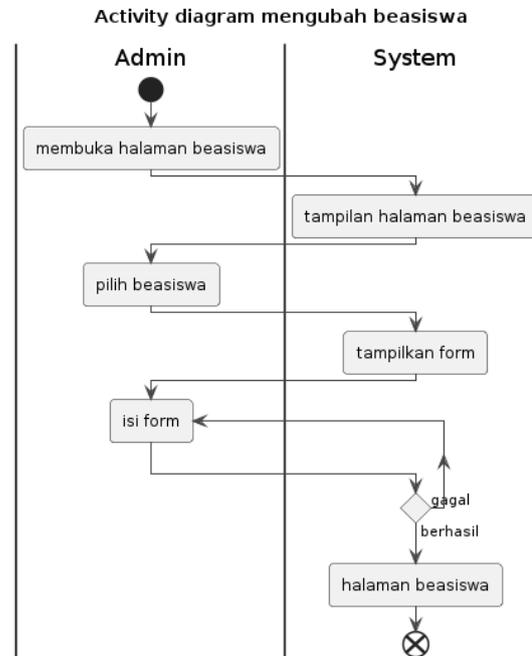
7) Activity diagram tambah beasiswa



Gambar 4. 8 Activity diagram tambah beasiswa

Pada gambar 4. 8 menjelaskan langkah-langkah yang digunakan oleh *admin* untuk menambahkan beasiswa. Setelah *admin* membukanya, sistem menampilkan halaman beasiswa. Sistem kemudian menampilkan halaman menu yang dipilih admin, dan admin mengklik tombol tambah setelah memilih item menu tambah. Sistem kemudian akan menampilkan form tambah beasiswa. Setelah itu admin diminta untuk mengisi form tambah beasiswa. Setelah selesai, sistem akan memverifikasi info beasiswa dan, jika semuanya sudah diperiksa, menampilkan halaman menu untuk beasiswa yang baru ditambahkan. *Admin* perlu mengisi kembali formulir penambahan beasiswa dengan benar jika tidak berfungsi.

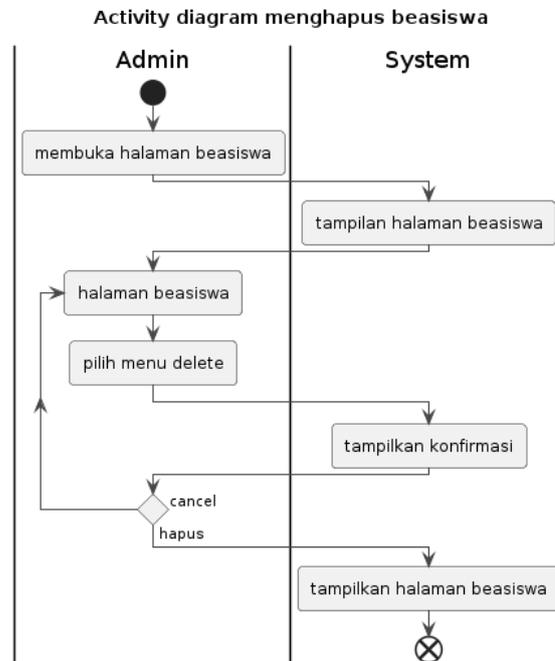
8) Activity diagram edit beasiswa



Gambar 4.9 Activity diagram edit beasiswa

Pada gambar 4.9 menjelaskan untuk mengubah data beasiswa, *admin* harus menuju ke halaman admin dan *admin* harus menekan tombol edit untuk menuju ke form pengisian data beasiswa. Tindakan ini akan meminta sistem untuk menampilkan form pengisian data beasiswa. Kemudian, setelah beasiswa mengisi form, sistem akan memvalidasi apakah form telah terisi atau belum. Jika form telah terisi maka sistem akan menuju kembali ke halaman admin dan jika belum terisi maka sistem akan kembali menampilkan form pengisiannya serta menampilkan error.

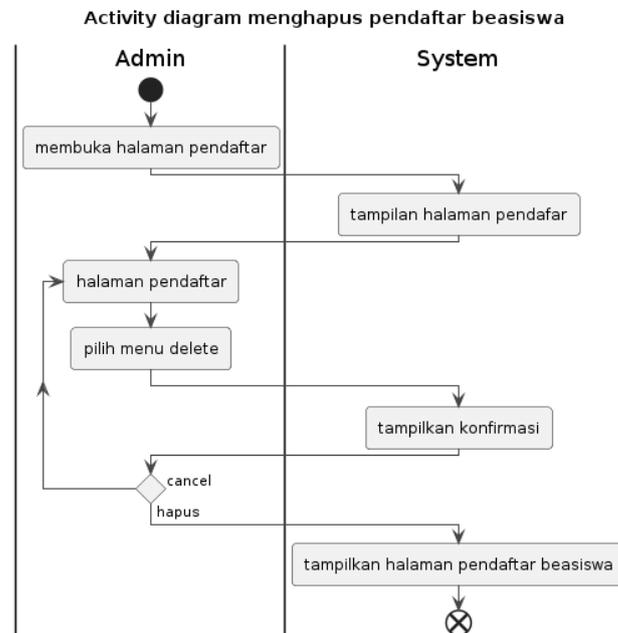
9) Activity diagram hapus beasiswa



Gambar 4. 10 Activity diagram hapus beasiswa

Pada gambar 4. 10 menjelaskan prosedur yang digunakan *admin* untuk menghapus beasiswa. Setelah *admin* membukanya, sistem menampilkan halaman *admin*. Selanjutnya *admin* memilih beasiswa mana yang akan dihapus, sistem akan menampilkan beasiswa yang dipilih *admin*, dan *admin* mengklik opsi hapus beasiswa. Setelah itu, sistem akan menampilkan konfirmasi penghapusan. Admin kemudian memverifikasi. Beasiswa akan dihapus jika admin menyetujui; jika tidak, beasiswa akan dibatalkan.

10) Activity diagram hapus pendaftar beasiswa

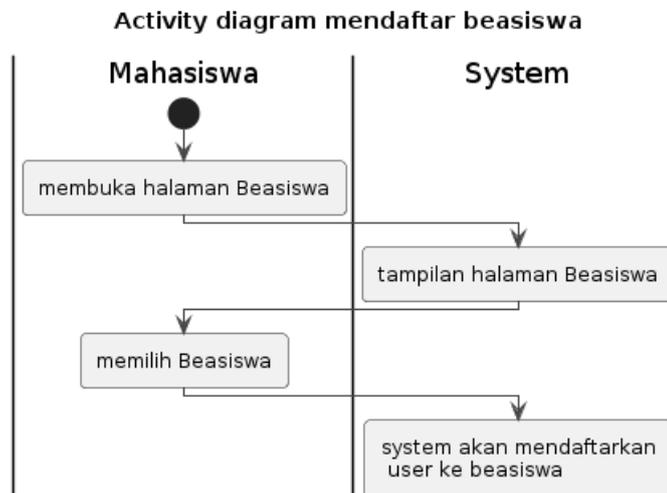


Gambar 4. 11 Activity diagram hapus pendaftar beasiswa

Pada gambar 4. 11 menjelaskan prosedur yang digunakan *admin* untuk menghapus pendaftar beasiswa. Setelah *admin* membukanya, sistem menampilkan halaman *admin*. Selanjutnya *admin* memilih pendaftar beasiswa mana yang akan dihapus, sistem akan menampilkan pendaftar beasiswa yang dipilih *admin*, dan *admin* mengklik opsi hapus pendaftar beasiswa. Setelah itu, sistem akan menampilkan konfirmasi penghapusan. Admin kemudian memverifikasi. Pendaftar beasiswa akan dihapus jika admin menyetujui; jika tidak, pendaftar beasiswa akan dibatalkan.

c. *Activity diagram* beasiswa

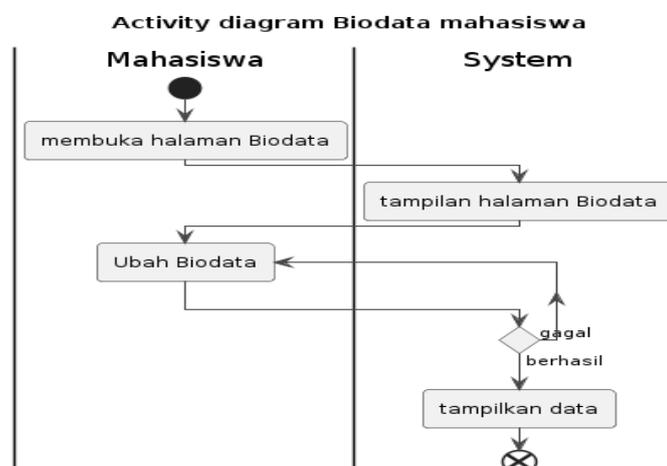
1) *Activity diagram* mendaftar beasiswa



Gambar 4. 12 *Activity diagram* mendaftar beasiswa

Pada gambar 4. 12 untuk mendaftar beasiswa, mahasiswa harus memasuki halaman beasiswa. setelah itu mahasiswa di haruskan memilih beasiswa yg mau di daftar dan system akan mendaftarkan beasiswa yg dipilih.

2) *Activity diagram* mengubah data mahasiswa



Gambar 4. 13 *Activity diagram* mengubah data mahasiswa

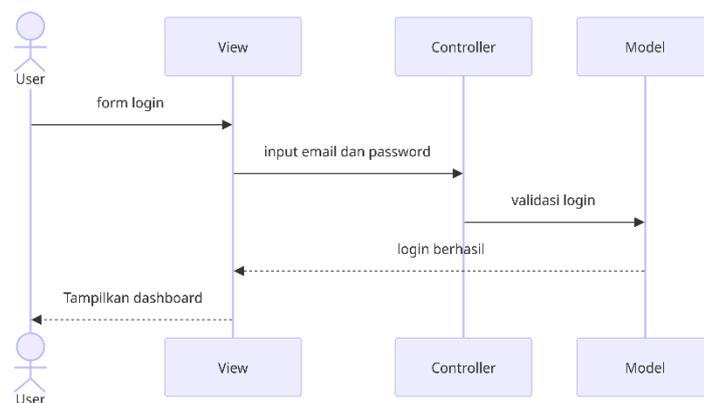
Pada gambar 4. 13 untuk mengubah data mahasiswa, mahasiswa harus menuju ke halaman profil. Tindakan ini akan meminta sistem untuk menampilkan form pengisian biodata. Kemudian, setelah mahasiswa mengisi form , sistem akan memvalidasi apakah form telah terisi atau belum. Jika form telah terisi maka system akan menuju kembali ke halaman home dan jika belum terisi maka system akan kembali menampilkan form pengisiannya serta menampilkan error .

2. *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*

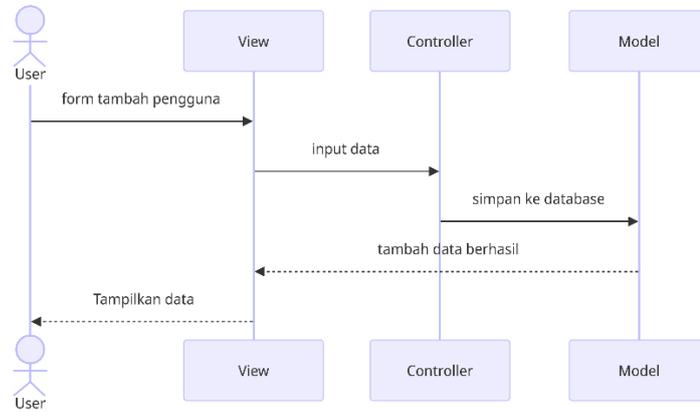
1) *Sequence diagram login*



Gambar 4. 14 *Sequence diagram login*

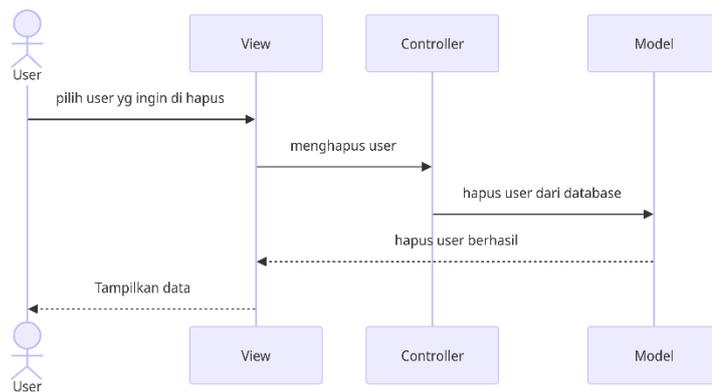
b. Sequence diagram admin

2) Sequence diagram tambah user



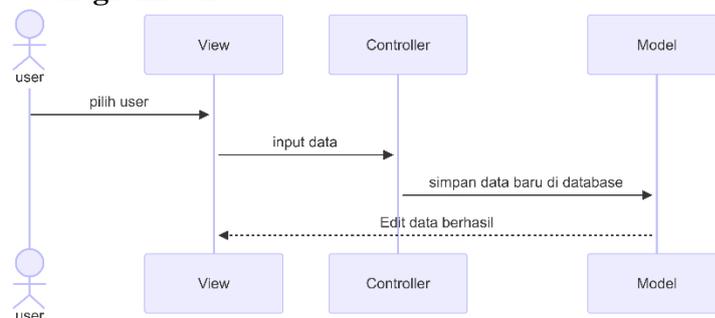
Gambar 4. 15 Sequence diagram tambah user

3) Sequence diagram hapus user



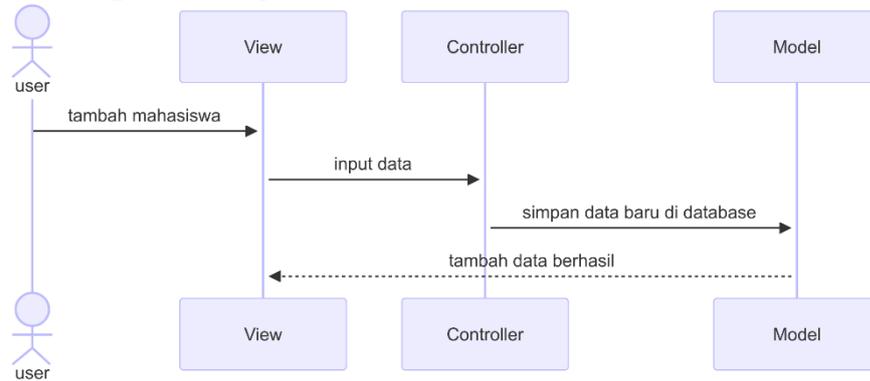
Gambar 4. 16 Sequence diagram hapus user

4) Sequence diagram edit user



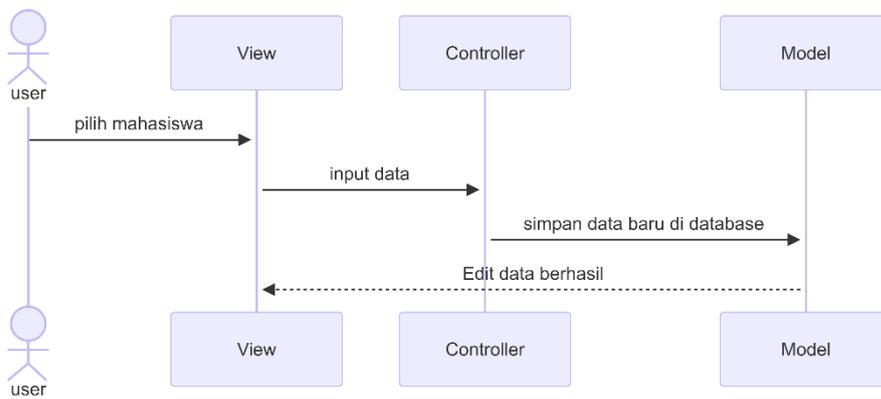
Gambar 4. 17 Sequence diagram edit user

5) Sequence diagram tambah mahasiswa



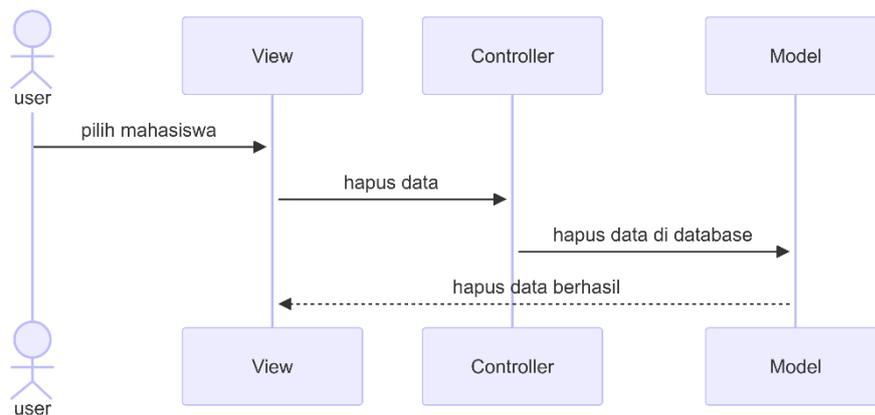
Gambar 4. 18 Sequence diagram tambah mahasiswa

6) Sequence diagram edit mahasiswa



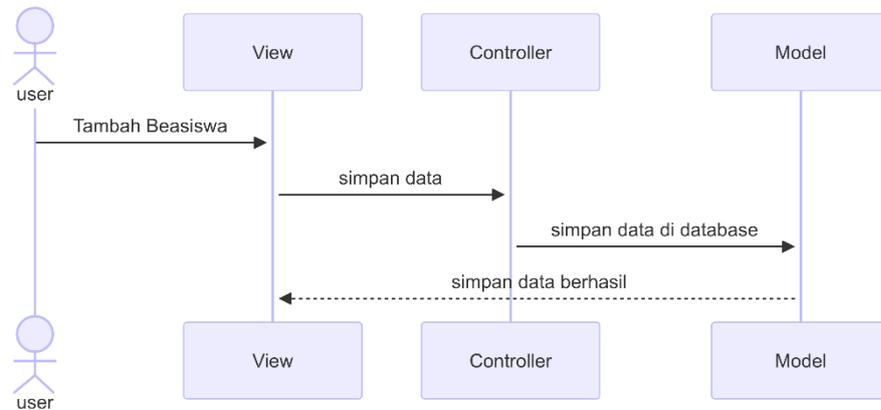
Gambar 4. 19 Sequence diagram edit mahasiswa

7) Sequence diagram hapus mahasiswa



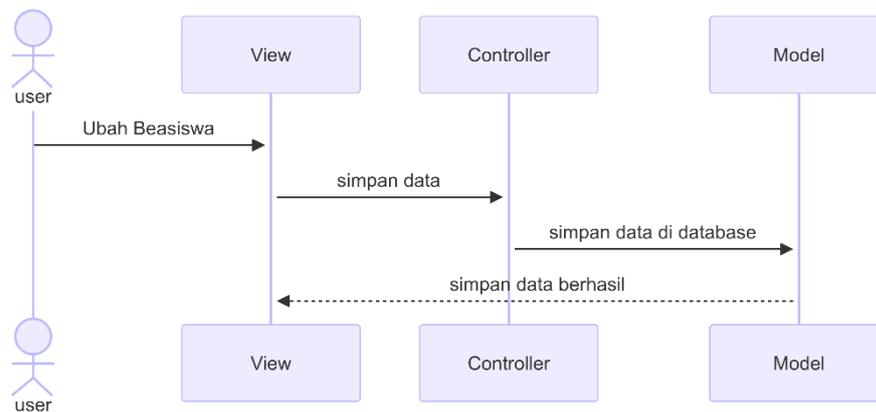
Gambar 4. 20 Sequence diagram hapus mahasiswa

8) *Sequence diagram* Tambah Beasiswa



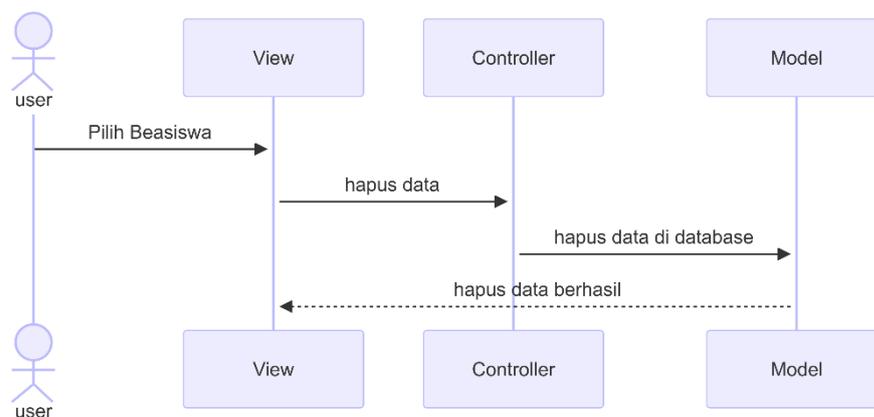
Gambar 4. 21 *Sequence diagram* tambah beasiswa

9) *Sequence diagram* edit beasiswa



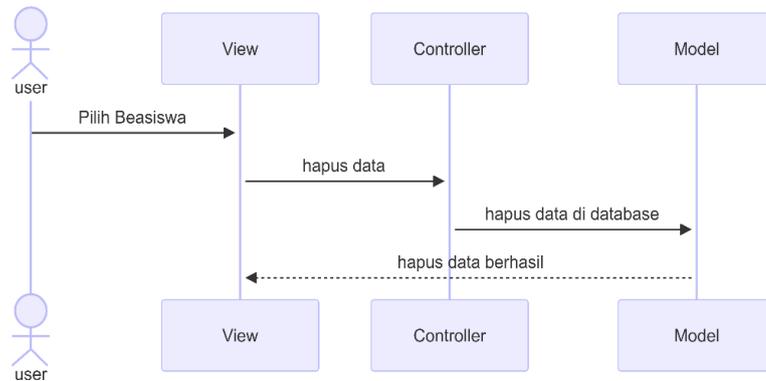
Gambar 4. 22 *Sequence diagram* edit beasiswa

10) *Sequence diagram* hapus beasiswa



Gambar 4. 23 *Sequence diagram* hapus beasiswa

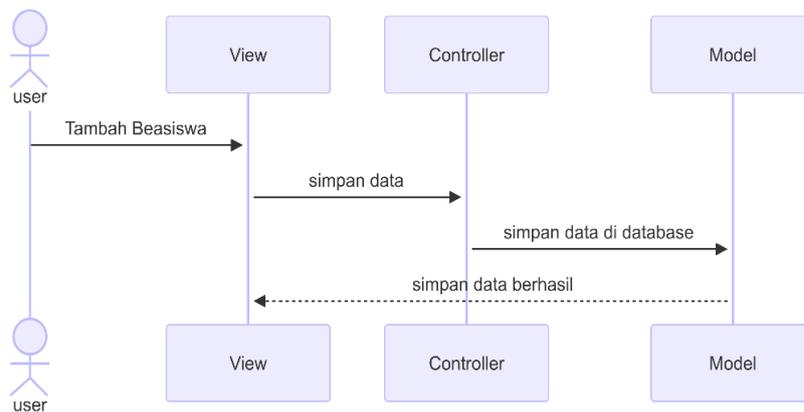
11) Sequence diagram hapus pendaftar beasiswa



c. Sequence diagram mahasiswa

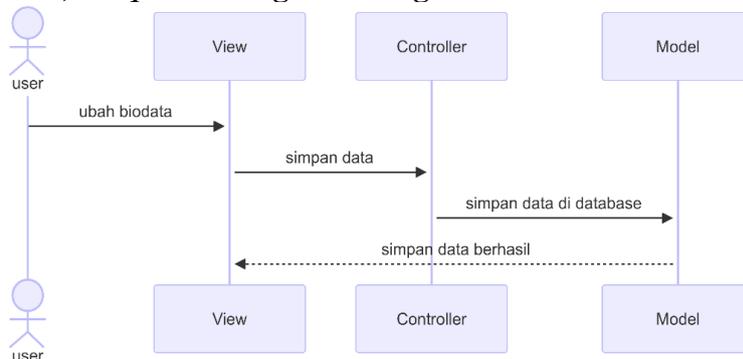
Gambar 4. 24 Sequence diagram hapus pendaftar beasiswa

1) Sequence diagram mendaftar beasiswa



Gambar 4. 25 Sequence diagram mendaftar beasiswa

2) Sequence diagram mengubah data beasiswa

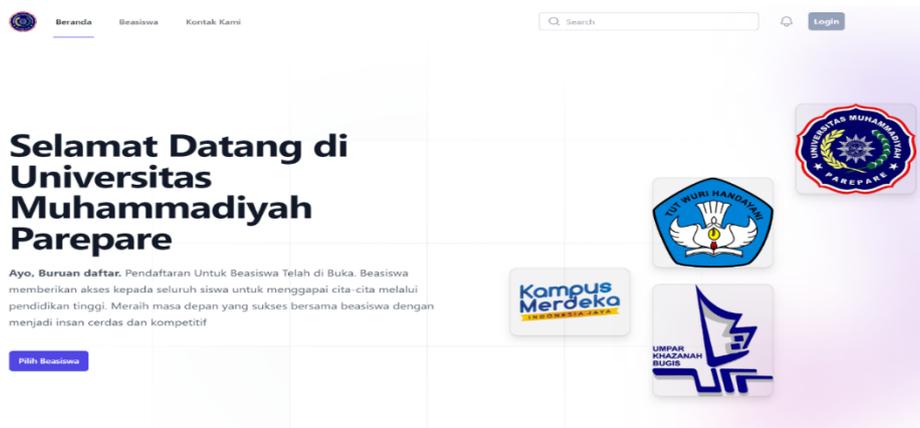


Gambar 4. 26 Sequence diagram ubah biodata

B. Detail Sistem

a. Halaman *welcome*

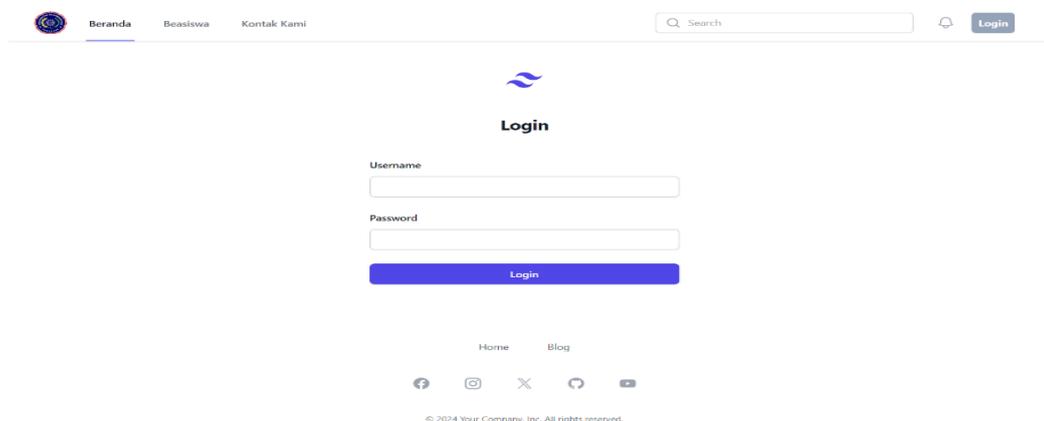
Halaman *welcome* merupakan halaman awal yg muncul Ketika pertama kali mengakses aplikasi.



Gambar 4. 27 Halaman *welcome*

b. Halaman *login*

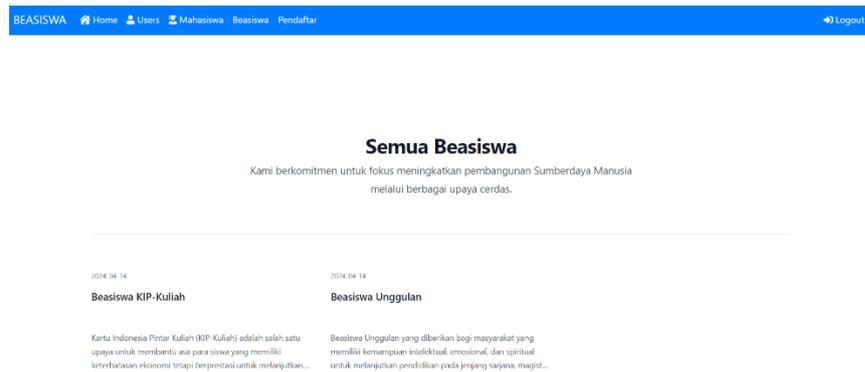
Halaman *login* merupakan halaman utama yang digunakan oleh *user* untuk mendapatkan akses ke halaman *dashboard*.



Gambar 4. 28 Halaman *login*

c. Halaman *dashboard* utama

Halaman *dashboard* utama merupakan halaman yg bisa diakses oleh semua *user*.



Gambar 4. 29 Halaman *dashboard* utama

d. Halaman tambah *User*

Halaman tambah *user* merupakan halaman untuk menambah *user*. berbeda dengan register, halaman ini hanya khusus untuk admin untuk menambahkan *user*.

Gambar 4. 30 Halaman tambah *User*

e. Halaman *Users*

Halaman *user* merupakan halaman untuk melihat semua *user*.

No.	Username	Password	Level	
1	101	25f9e794323b453885f5181f1b624d0b	Mahasiswa	
2	221280118	25f9e794323b453885f5181f1b624d0b	Mahasiswa	
3	admin	21232f297a57a5a743894a0e4a801fc3	Admin	

Gambar 4. 31 Halaman *Users*

f. Halaman edit *User*

Halaman edit user merupakan halaman untuk mengubah data user. halaman ini berguna jika dalam penambahan User terdapat kesalahan ketik.

Gambar 4. 32 Halaman edit *User*

g. Halaman tambah mahasiswa

Halaman tambah mahasiswa merupakan halaman untuk menambah mahasiswa. berbeda dengan register, halaman ini hanya khusus untuk admin untuk menambahkan mahasiswa.

Data Mahasiswa							
<div style="display: flex; justify-content: space-between; align-items: center;"> Tambah Show <input type="text" value="10"/> entries Search: <input type="text"/> </div>							
NIM	Nama Mahasiswa	Alamat	No.Telepon	Fakultas	Prodi	IPK	
101	Arjuna	SIDRAP	085xxx	TEKNIK	INFORMATIKA-A	3.55	 
221280118	Fadlullah	Parepare	085256953376	Teknik	Informatika	3.0	 
Showing 1 to 2 of 2 entries						Previous <input type="text" value="1"/> Next	

Gambar 4. 33 Halaman tambah mahasiswa

h. Halaman edit mahasiswa

Halaman edit mahasiswa merupakan halaman untuk mengubah data mahasiswa.halaman ini berguna jika dalam penambahan mahasiswa terdapat kesalahan ketik.

Update Data Mahasiswa	
NIM	<input type="text" value="101"/>
Nama	<input type="text" value="Arjuna"/>
Alamat	<input type="text" value="SIDRAP"/>
No. Telepon	<input type="text" value="085xxx"/>
Fakultas	<input type="text" value="TEKNIK"/>
Prodi	<input type="text" value="INFORMATIKA-A"/>
IPK	<input type="text" value="3.55"/>
<input type="button" value="Update"/> <input type="button" value="Batal"/>	

Gambar 4. 34 Halaman edit mahasiswa

i. Halaman mahasiswa

Halaman mahasiswa merupakan halaman untuk melihat semua mahasiswa.

Data Mahasiswa							
<div style="display: flex; justify-content: space-between; align-items: center;"> Tambah Show 10 entries Search: <input type="text"/> </div>							
NIM	Nama Mahasiswa	Alamat	No.Telepon	Fakultas	Prodi	IPK	
101	Arjuna	SIDRAP	085xxx	TEKNIK	INFORMATIKA-A	3.55	 
221280118	Fadlullah	Parepare	085256953376	Teknik	Informatika	3.0	 
Showing 1 to 2 of 2 entries						Previous 1 Next	

Gambar 4. 35 Halaman mahasiswa

j. Halaman Tambah beasiswa

Halaman tambah mahasiswa merupakan halaman untuk menambah mahasiswa.berbeda dengan register, halaman ini hanya khusus untuk admin untuk menambahkan mahasiswa.

Tambah Artikel Beasiswa	
Judul	<input type="text"/>
Isi	<input type="text"/>
Tanggal	<input type="text" value="dd/mm/yyyy"/> 
<div style="display: flex; justify-content: space-between;"> Simpan Batal </div>	

Gambar 4. 36 halaman tambah beasiswa

k. Halaman Beasiswa

Halaman Beasiswa merupakan halaman untuk melihat semua beasiswa.

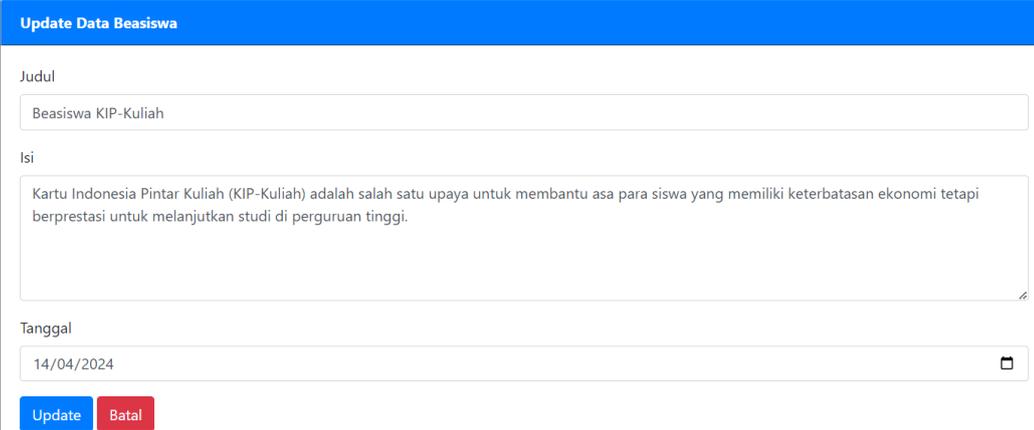


No.	Judul	Isi	Tanggal	
1	Beasiswa KIP-Kuliah	Kartu Indonesia Pintar Kuliah (KIP-Kuliah) adalah salah satu upaya untuk membantu asa para siswa yan	2024-04-14	  
2	Beasiswa Unggulan	Beasiswa Unggulan yang diberikan bagi masyarakat yang memiliki kemampuan intelektual, emosional, dan	2024-04-14	  

Gambar 4. 37 Halaman Beasiswa

l. Halaman edit beasiswa

Halaman edit beasiswa merupakan halaman untuk mengubah data beasiswa. halaman ini berguna jika dalam penambahan beasiswa terdapat kesalahan ketik.



Gambar 4. 38 halaman edit beasiswa

m. Halaman pendaftar beasiswa

Halaman ini merupakan halaman untuk melihat semua pendaftar beasiswa.

Data Pendaftar Beasiswa									
No.	NIM	Nama	Alamat	Fakultas	Telepon	Prodi	IPK	Beasiswa	
1	101	Arjuna	SIDRAP	TEKNIK	085xxx	INFORMATIKA-A	3.55	Beasiswa KIP-Kuliah	
2	101	Arjuna	SIDRAP	TEKNIK	085xxx	INFORMATIKA-A	3.55	Beasiswa Unggulan	
3	101	Arjuna	SIDRAP	TEKNIK	085xxx	INFORMATIKA-A	3.55	Beasiswa Unggulan	
4	101	Arjuna	SIDRAP	TEKNIK	085xxx	INFORMATIKA-A	3.55	Beasiswa KIP-Kuliah	
5	221280118	Fadlullah	Parepare	Teknik	085256953376	Informatika	3.0	Beasiswa KIP-Kuliah	
6	221280118	Fadlullah	Parepare	Teknik	085256953376	Informatika	3.0	Beasiswa Unggulan	
7	101	Arjuna	SIDRAP	TEKNIK	085xxx	INFORMATIKA-A	3.55	Beasiswa Unggulan	
8	101	Arjuna	SIDRAP	TEKNIK	085xxx	INFORMATIKA-A	3.55	Beasiswa KIP-Kuliah	
9	101	Arjuna	SIDRAP	TEKNIK	085xxx	INFORMATIKA-	3.55	Beasiswa	

Gambar 4. 39 halaman pendaftar beasiswa

n. Halaman daftar beasiswa

Halaman ini adalah halaman untuk mendaftar beasiswa.

BEASISWA [Home](#) [Profile](#) Logout

Semua Beasiswa

Kami berkomitmen untuk fokus meningkatkan pembangunan Sumberdaya Manusia melalui berbagai upaya cerdas.

2024-04-14

Beasiswa KIP-Kuliah

Kartu Indonesia Pintar Kuliah (KIP-Kuliah) adalah salah satu upaya untuk membantu asa para siswa yang memiliki keterbatasan ekonomi tetapi berprestasi untuk melanjutkan...

2024-04-14

Beasiswa Unggulan

Beasiswa Unggulan yang diberikan bagi masyarakat yang memiliki kemampuan intelektual, emosional, dan spiritual untuk melanjutkan pendidikan pada jenjang sarjana, magist...

Gambar 4. 40 halaman daftar beasiswa

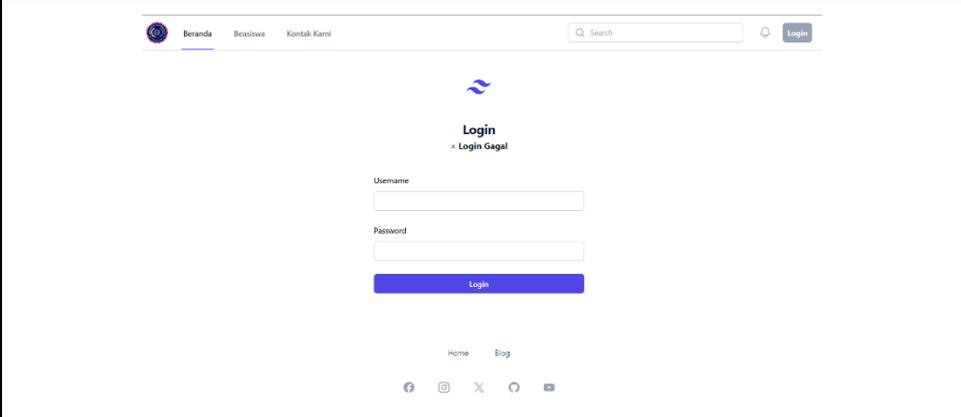
C. Pengujian Sistem

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

1. *Black box*

a. *Black box testing kesalahan username dan password*

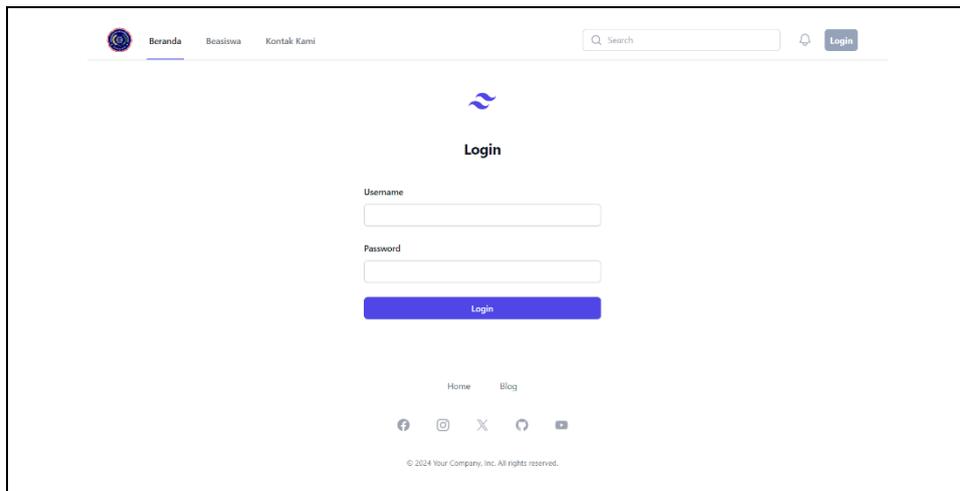
Tabel 4. 1 *Black box testing kesalahan username dan password*

Tes faktor	Hasil	Keterangan
Memasukkan <i>username</i> atau <i>password</i> yang tidak sesuai	✓	Berhasil, ketika <i>username</i> atau <i>password</i> tidak sesuai tampil <i>login gagal</i>
Screenshot		
		

b. *Black box testing login berhasil*

Tabel 4. 2 *Black box testing login berhasil*

Tes faktor	Hasil	Keterangan
Memasukkan <i>username</i> atau <i>password</i> yang benar	✓	Sistem berhasil menampilkan halaman dashboard.
Screenshot		



c. Black box testing dashboard utama

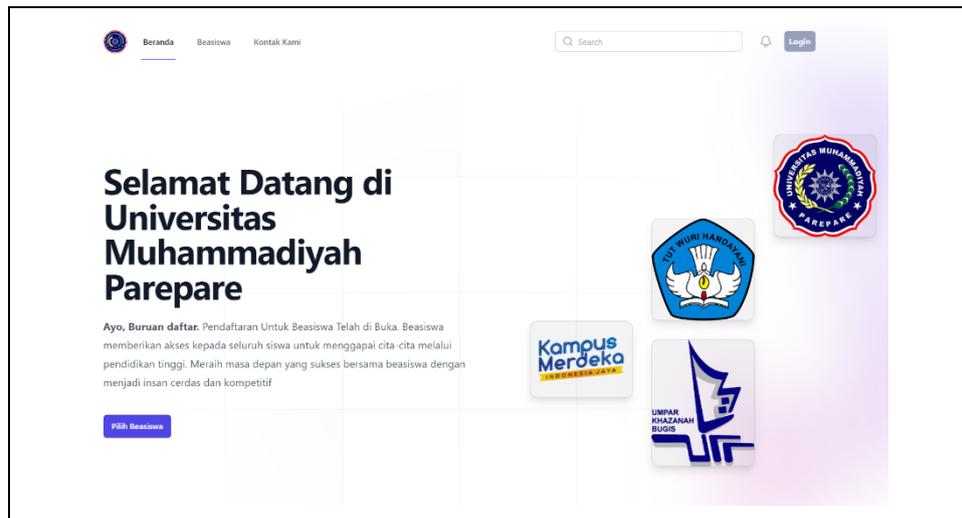
Tabel 4. 3 Black box testing dashboard utama

Tes faktor	Hasil	Keterangan
User melakukan login mengakses halaman dashboard	✓	Halaman dashboard akan tampil
<i>Screenshot</i>		

d. Black box testing halaman awal

Tabel 4. 4 blackbox testing halaman awal

Tes faktor	Hasil	Keterangan
User mengakses halaman awal	✓	Halaman awal akan tampil
<i>Screenshot</i>		



e. *Black box testing tambah user*

Tabel 4.5 Black box testing tambah

Tes faktor	Hasil	Keterangan
Admin mengisi form tambah dan menekan tombol simpan.	✓	User akan bertambah .
<i>Screenshot</i>		

f. *Black box testing Users*

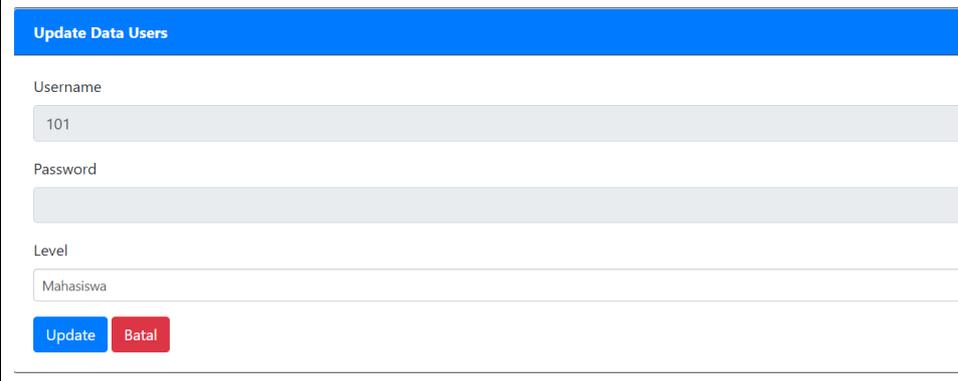
Tabel 4.6 Black box testing Users

Tes faktor	Hasil	Keterangan
Admin mengakses halaman Users dashboard.	✓	Halaman Users dashboard akan tampil
<i>Screenshot</i>		

Data Users				
<div style="display: flex; justify-content: space-between; align-items: center;"> Tambah Show 10 entries Search: <input type="text"/> </div>				
No.	Username	Password	Level	
1	101	25f9e794323b453885f5181f1b624d0b	Mahasiswa	 
2	221280118	25f9e794323b453885f5181f1b624d0b	Mahasiswa	 
3	admin	21232f297a57a5a743894a0e4a801fc3	Admin	 
Showing 1 to 3 of 3 entries				<div style="display: flex; justify-content: flex-end; align-items: center;"> Previous 1 Next </div>

g. Black box testing Edit user

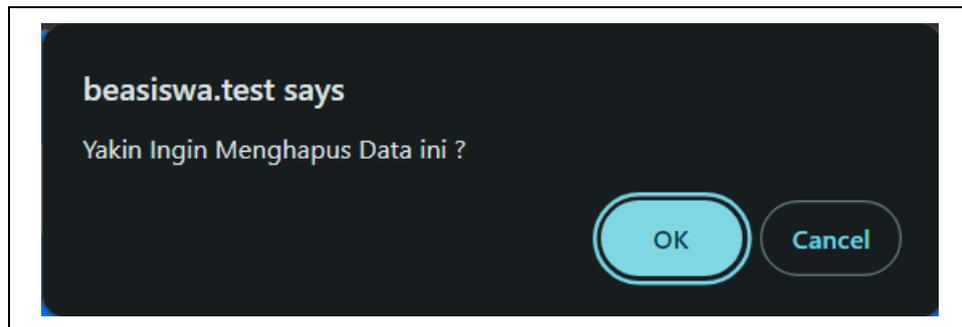
Tabel 4. 7 Black box testing edit user

Tes faktor	Hasil	Keterangan
Admin mengedit data User	✓	Data user yg di edit akan berubah.
<i>Screenshot</i>		
		

h. Black box testing hapus User

Tabel 4. 8 Black box testing halaman laporan

Tes faktor	Hasil	Keterangan
Admin menghapus User yg dipilih.	✓	User yg dipilih admin akan dihapus.
<i>Screenshot</i>		



i. *Black box testing* tambah mahasiswa

Tabel 4. 9 *Black box testing* tambah mahasiswa

Tes faktor	Hasil	Keterangan
<i>Admin</i> mengisi <i>form</i> tambah dan menekan tombol simpan.	✓	Mahasiswa akan bertambah.
<i>Screenshot</i>		

j. *Black box testing* edit mahasiswa

Tabel 4. 10 *Black box testing* edit mahasiswa

Tes faktor	Hasil	Keterangan
<i>Admin</i> mengisi <i>form</i> edit dan menekan tombol simpan.	✓	Mahasiswa akan berubah.
<i>Screenshot</i>		

Update Data Mahasiswa

NIM

Nama

Alamat

No. Telepon

Fakultas

Prodi

IPK

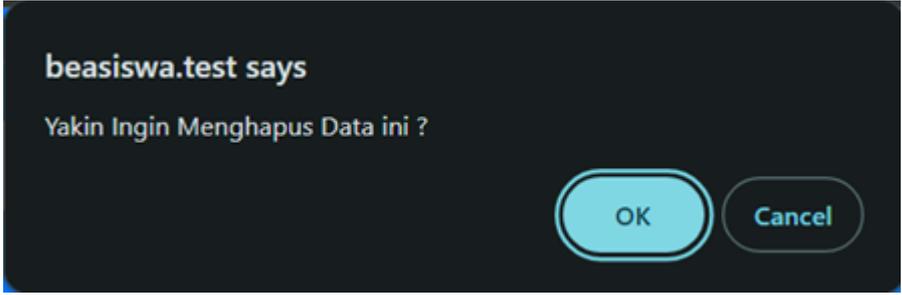
k. Black box testing mahasiswa

Tabel 4. 11 Black box testing mahasiswa

Tes faktor	Hasil	Keterangan																								
<i>Admin</i> mengakses halaman dashboard mahasiswa.	✓	Halaman dashboard mahasiswa akan tampil.																								
Screenshot																										
<div style="border: 1px solid black; padding: 10px; width: fit-content; margin: auto;"> <div style="background-color: #007bff; color: white; padding: 5px; text-align: center;">Data Mahasiswa</div> <div style="display: flex; justify-content: space-between; align-items: center; margin-top: 10px;"> <div style="background-color: #007bff; color: white; padding: 5px 10px; border-radius: 5px;">Tambah</div> <div style="text-align: right;"> Show 10 entries Search: <input style="width: 80px;" type="text"/> </div> </div> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th style="width: 10%;">NIM</th> <th style="width: 15%;">Nama Mahasiswa</th> <th style="width: 15%;">Alamat</th> <th style="width: 15%;">No.Telepon</th> <th style="width: 10%;">Fakultas</th> <th style="width: 10%;">Prodi</th> <th style="width: 10%;">IPK</th> <th style="width: 15%;"></th> </tr> </thead> <tbody> <tr> <td>101</td> <td>Arjuna</td> <td>SIDRAP</td> <td>085xxx</td> <td>TEKNIK</td> <td>INFORMATIKA-A</td> <td>3.55</td> <td style="text-align: center; vertical-align: middle;"> <div style="display: flex; flex-direction: column; gap: 5px;"> <div style="background-color: #ffc107; padding: 2px 5px; border-radius: 3px;">✎</div> <div style="background-color: #dc3545; padding: 2px 5px; border-radius: 3px;">✖</div> </div> </td> </tr> <tr> <td>221280118</td> <td>Fadlullah</td> <td>Parepare</td> <td>085256953376</td> <td>Teknik</td> <td>Informatika</td> <td>3.0</td> <td style="text-align: center; vertical-align: middle;"> <div style="display: flex; flex-direction: column; gap: 5px;"> <div style="background-color: #ffc107; padding: 2px 5px; border-radius: 3px;">✎</div> <div style="background-color: #dc3545; padding: 2px 5px; border-radius: 3px;">✖</div> </div> </td> </tr> </tbody> </table> <div style="display: flex; justify-content: space-between; align-items: center; margin-top: 10px;"> Showing 1 to 2 of 2 entries Previous 1 Next </div> </div>			NIM	Nama Mahasiswa	Alamat	No.Telepon	Fakultas	Prodi	IPK		101	Arjuna	SIDRAP	085xxx	TEKNIK	INFORMATIKA-A	3.55	<div style="display: flex; flex-direction: column; gap: 5px;"> <div style="background-color: #ffc107; padding: 2px 5px; border-radius: 3px;">✎</div> <div style="background-color: #dc3545; padding: 2px 5px; border-radius: 3px;">✖</div> </div>	221280118	Fadlullah	Parepare	085256953376	Teknik	Informatika	3.0	<div style="display: flex; flex-direction: column; gap: 5px;"> <div style="background-color: #ffc107; padding: 2px 5px; border-radius: 3px;">✎</div> <div style="background-color: #dc3545; padding: 2px 5px; border-radius: 3px;">✖</div> </div>
NIM	Nama Mahasiswa	Alamat	No.Telepon	Fakultas	Prodi	IPK																				
101	Arjuna	SIDRAP	085xxx	TEKNIK	INFORMATIKA-A	3.55	<div style="display: flex; flex-direction: column; gap: 5px;"> <div style="background-color: #ffc107; padding: 2px 5px; border-radius: 3px;">✎</div> <div style="background-color: #dc3545; padding: 2px 5px; border-radius: 3px;">✖</div> </div>																			
221280118	Fadlullah	Parepare	085256953376	Teknik	Informatika	3.0	<div style="display: flex; flex-direction: column; gap: 5px;"> <div style="background-color: #ffc107; padding: 2px 5px; border-radius: 3px;">✎</div> <div style="background-color: #dc3545; padding: 2px 5px; border-radius: 3px;">✖</div> </div>																			

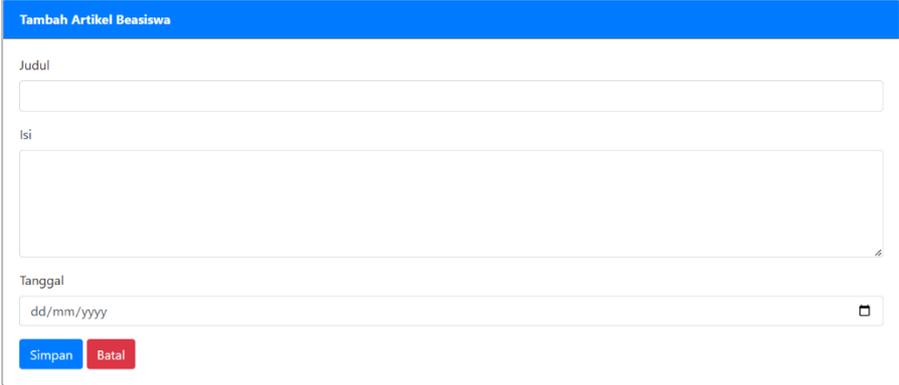
l. *Black box testing* hapus mahasiswa

Tabel 4. 12 *Black box testing* hapus mahasiswa.

Tes faktor	Hasil	Keterangan
<i>Admin</i> menghapus mahasiswa yg dipilih.	✓	mahasiswa yg dipilih admin akan dihapus.
<i>Screenshot</i>		
		

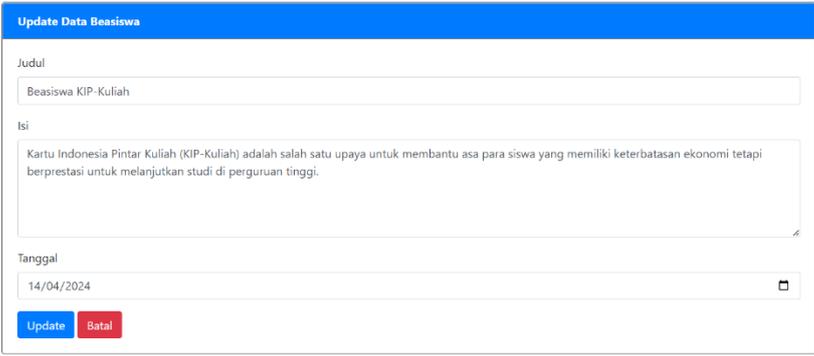
m. *Black box testing* tambah beasiswa

Tabel 4. 13 *Black box testing* tambah laporan.

Tes faktor	Hasil	Keterangan
<i>Admin</i> mengisi form tambah dan menekan tombol simpan.	✓	Beasiswa akan bertambah.
<i>Screenshot</i>		
		

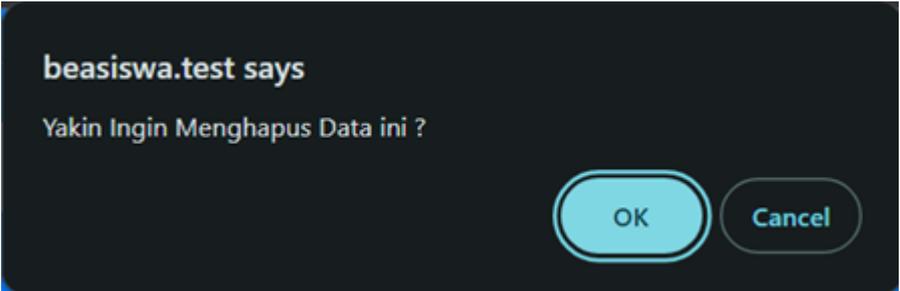
n. Black box edit Beasiswa

Tabel 4. 14 Black box testing edit beasiswa

Tes faktor	Hasil	Keterangan
<i>Admin</i> mengisi form edit dan menekan tombol simpan.	✓	Beasiswa akan berubah.
Screenshot		
		

o. Black box hapus Beasiswa

Tabel 4. 15 Black box Testing hapus Beasiswa

Tes faktor	Hasil	Keterangan
<i>Admin</i> menghapus beasiswa yg dipilih.	✓	beasiswa yg dipilih admin akan dihapus.
Screenshot		
		

p. Black box Beasiswa

Tabel 4. 16 Blackbox testing beasiswa

Tes faktor	Hasil	Keterangan
<i>Admin</i> mengakses halaman beasiswa dashboard.	✓	Halaman beasiswa dashboard akan tampil.
Screenshot		

Data Beasiswa																			
<p>Tambah Beasiswa</p> <p>Show 10 entries Search: <input type="text"/></p> <table border="1"> <thead> <tr> <th>No.</th> <th>Judul</th> <th>Isi</th> <th>Tanggal</th> <th></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Beasiswa KIP-Kuliah</td> <td>Kartu Indonesia Pintar Kuliah (KIP-Kuliah) adalah salah satu upaya untuk membantu asa para siswa yan</td> <td>2024-04-14</td> <td> </td> </tr> <tr> <td>2</td> <td>Beasiswa Unggulan</td> <td>Beasiswa Unggulan yang diberikan bagi masyarakat yang memiliki kemampuan intelektual, emosional, dan</td> <td>2024-04-14</td> <td> </td> </tr> </tbody> </table> <p>Showing 1 to 2 of 2 entries Previous <input type="button" value="1"/> Next</p>					No.	Judul	Isi	Tanggal		1	Beasiswa KIP-Kuliah	Kartu Indonesia Pintar Kuliah (KIP-Kuliah) adalah salah satu upaya untuk membantu asa para siswa yan	2024-04-14	 	2	Beasiswa Unggulan	Beasiswa Unggulan yang diberikan bagi masyarakat yang memiliki kemampuan intelektual, emosional, dan	2024-04-14	 
No.	Judul	Isi	Tanggal																
1	Beasiswa KIP-Kuliah	Kartu Indonesia Pintar Kuliah (KIP-Kuliah) adalah salah satu upaya untuk membantu asa para siswa yan	2024-04-14	 															
2	Beasiswa Unggulan	Beasiswa Unggulan yang diberikan bagi masyarakat yang memiliki kemampuan intelektual, emosional, dan	2024-04-14	 															

q. **Black box** pendaftar beasiswa

Tabel 4. 17 Blackbox testing pendaftar beasiswa

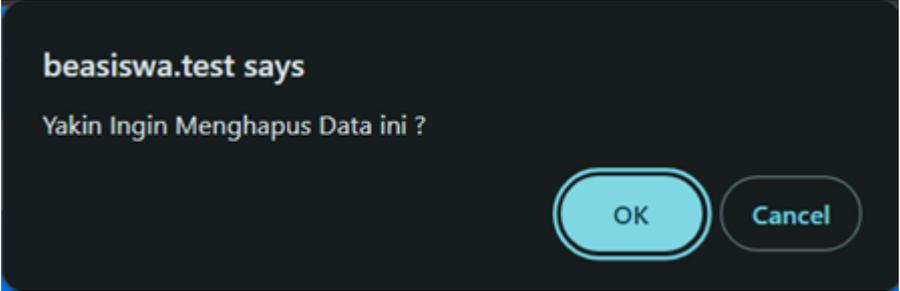
Tes faktor	Hasil	Keterangan
Admin halaman beasiswa . mengakses pendaftar .	✓	Halaman pendaftar beasiswa akan tampil.

Screenshot

Data Pendaftar Beasiswa																																																																																																													
<p>Show 10 entries Search: <input type="text"/></p> <table border="1"> <thead> <tr> <th>No.</th> <th>NIM</th> <th>Nama</th> <th>Alamat</th> <th>Fakultas</th> <th>Telepon</th> <th>Prodi</th> <th>IPK</th> <th>Beasiswa</th> <th></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>101</td> <td>Arjuna</td> <td>SIDRAP</td> <td>TEKNIK</td> <td>085xxx</td> <td>INFORMATIKA-A</td> <td>3.55</td> <td>Beasiswa KIP-Kuliah</td> <td></td> </tr> <tr> <td>2</td> <td>101</td> <td>Arjuna</td> <td>SIDRAP</td> <td>TEKNIK</td> <td>085xxx</td> <td>INFORMATIKA-A</td> <td>3.55</td> <td>Beasiswa Unggulan</td> <td></td> </tr> <tr> <td>3</td> <td>101</td> <td>Arjuna</td> <td>SIDRAP</td> <td>TEKNIK</td> <td>085xxx</td> <td>INFORMATIKA-A</td> <td>3.55</td> <td>Beasiswa Unggulan</td> <td></td> </tr> <tr> <td>4</td> <td>101</td> <td>Arjuna</td> <td>SIDRAP</td> <td>TEKNIK</td> <td>085xxx</td> <td>INFORMATIKA-A</td> <td>3.55</td> <td>Beasiswa KIP-Kuliah</td> <td></td> </tr> <tr> <td>5</td> <td>221280118</td> <td>Fadlullah</td> <td>Parepare</td> <td>Teknik</td> <td>085256953376</td> <td>Informatika</td> <td>3.0</td> <td>Beasiswa KIP-Kuliah</td> <td></td> </tr> <tr> <td>6</td> <td>221280118</td> <td>Fadlullah</td> <td>Parepare</td> <td>Teknik</td> <td>085256953376</td> <td>Informatika</td> <td>3.0</td> <td>Beasiswa Unggulan</td> <td></td> </tr> <tr> <td>7</td> <td>101</td> <td>Arjuna</td> <td>SIDRAP</td> <td>TEKNIK</td> <td>085xxx</td> <td>INFORMATIKA-A</td> <td>3.55</td> <td>Beasiswa Unggulan</td> <td></td> </tr> <tr> <td>8</td> <td>101</td> <td>Arjuna</td> <td>SIDRAP</td> <td>TEKNIK</td> <td>085xxx</td> <td>INFORMATIKA-A</td> <td>3.55</td> <td>Beasiswa KIP-Kuliah</td> <td></td> </tr> <tr> <td>9</td> <td>101</td> <td>Arjuna</td> <td>SIDRAP</td> <td>TEKNIK</td> <td>085xxx</td> <td>INFORMATIKA-</td> <td>3.55</td> <td>Beasiswa</td> <td></td> </tr> </tbody> </table>										No.	NIM	Nama	Alamat	Fakultas	Telepon	Prodi	IPK	Beasiswa		1	101	Arjuna	SIDRAP	TEKNIK	085xxx	INFORMATIKA-A	3.55	Beasiswa KIP-Kuliah		2	101	Arjuna	SIDRAP	TEKNIK	085xxx	INFORMATIKA-A	3.55	Beasiswa Unggulan		3	101	Arjuna	SIDRAP	TEKNIK	085xxx	INFORMATIKA-A	3.55	Beasiswa Unggulan		4	101	Arjuna	SIDRAP	TEKNIK	085xxx	INFORMATIKA-A	3.55	Beasiswa KIP-Kuliah		5	221280118	Fadlullah	Parepare	Teknik	085256953376	Informatika	3.0	Beasiswa KIP-Kuliah		6	221280118	Fadlullah	Parepare	Teknik	085256953376	Informatika	3.0	Beasiswa Unggulan		7	101	Arjuna	SIDRAP	TEKNIK	085xxx	INFORMATIKA-A	3.55	Beasiswa Unggulan		8	101	Arjuna	SIDRAP	TEKNIK	085xxx	INFORMATIKA-A	3.55	Beasiswa KIP-Kuliah		9	101	Arjuna	SIDRAP	TEKNIK	085xxx	INFORMATIKA-	3.55	Beasiswa	
No.	NIM	Nama	Alamat	Fakultas	Telepon	Prodi	IPK	Beasiswa																																																																																																					
1	101	Arjuna	SIDRAP	TEKNIK	085xxx	INFORMATIKA-A	3.55	Beasiswa KIP-Kuliah																																																																																																					
2	101	Arjuna	SIDRAP	TEKNIK	085xxx	INFORMATIKA-A	3.55	Beasiswa Unggulan																																																																																																					
3	101	Arjuna	SIDRAP	TEKNIK	085xxx	INFORMATIKA-A	3.55	Beasiswa Unggulan																																																																																																					
4	101	Arjuna	SIDRAP	TEKNIK	085xxx	INFORMATIKA-A	3.55	Beasiswa KIP-Kuliah																																																																																																					
5	221280118	Fadlullah	Parepare	Teknik	085256953376	Informatika	3.0	Beasiswa KIP-Kuliah																																																																																																					
6	221280118	Fadlullah	Parepare	Teknik	085256953376	Informatika	3.0	Beasiswa Unggulan																																																																																																					
7	101	Arjuna	SIDRAP	TEKNIK	085xxx	INFORMATIKA-A	3.55	Beasiswa Unggulan																																																																																																					
8	101	Arjuna	SIDRAP	TEKNIK	085xxx	INFORMATIKA-A	3.55	Beasiswa KIP-Kuliah																																																																																																					
9	101	Arjuna	SIDRAP	TEKNIK	085xxx	INFORMATIKA-	3.55	Beasiswa																																																																																																					

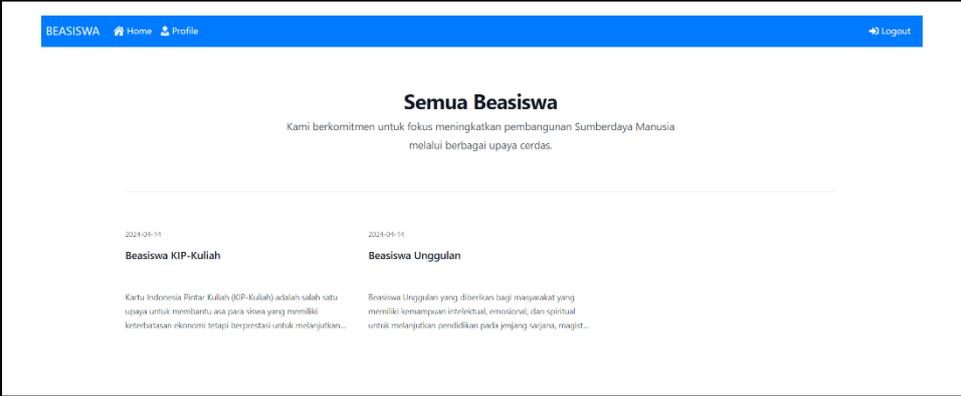
r. *Blackbox* hapus pendaftar beasiswa

Tabel 4. 18 *Blackbox testing* hapus pendaftar beasiswa

Tes faktor	Hasil	Keterangan
<i>Admin</i> menghapus pendaftar beasiswa yg dipilih.	✓	Pendaftar beasiswa yg dipilih admin akan dihapus.
<i>Screenshot</i>		
		

s. *Blackbox* mendaftar beasiswa

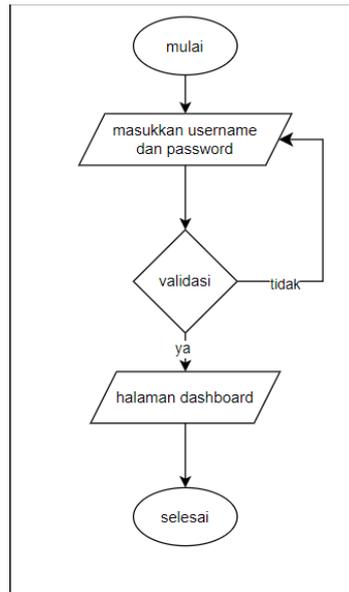
Tabel 4. 19 *Blackbox testing* mendaftar beasiswa

Tes faktor	Hasil	Keterangan
<i>Mahasiswa</i> mengklik beasiswa yg ingin di daftar.	✓	Sistem akan mendaftarkan beasiswa yg dipilih.
<i>Screenshot</i>		
		

2. White box

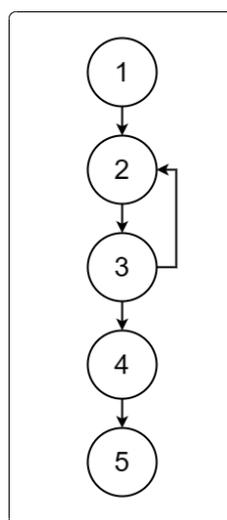
a. White box testing kesalahan Username dan password

1) Flowchart



Gambar 4. 41 Flowchart kesalahan Username dan password

2) Flowgraph



Gambar 4. 42 Flowgraph kesalahan Username dan password

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

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

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E (\text{edge}) = 5$$

$$N (\text{node}) = 5$$

$$P (\text{Predikat } \textit{node}) = 1$$

Penyelesaian :

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

$$= 5 - 5 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

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

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

$$\text{Path 1} = 1 - 2 - 3 - 2$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5$$

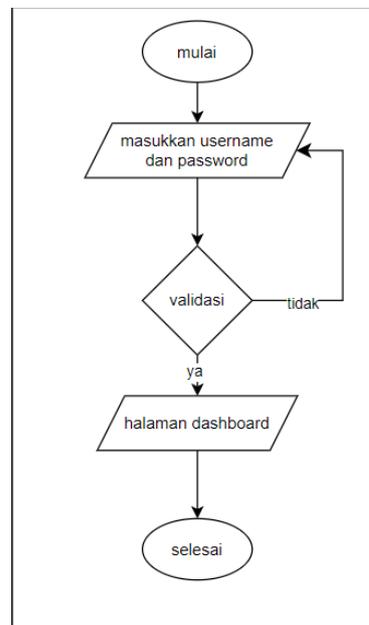
(4) Grafik matriks kesalahan *username* dan *password*

Tabel 4. 20 Grafik matriks kesalahan *username* dan *password*

	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)					$1 + 1 = 2$

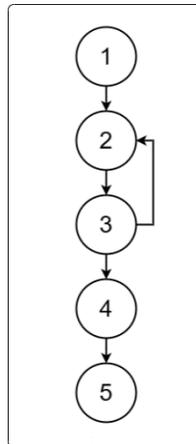
b. *White box testing login* berhasil

1) *Flowchart*



Gambar 4. 43 *Flowchart login* berhasil

2) Flowgraph



Gambar 4. 44 Flowgraph login berhasil

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

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

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 5$$

$$N \text{ (node)} = 5$$

$$P \text{ (predikat node)} = 1$$

Penyelesaian :

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

$$= 5 - 5 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

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

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

$$\text{Path 1} = 1 - 2 - 3 - 2$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5$$

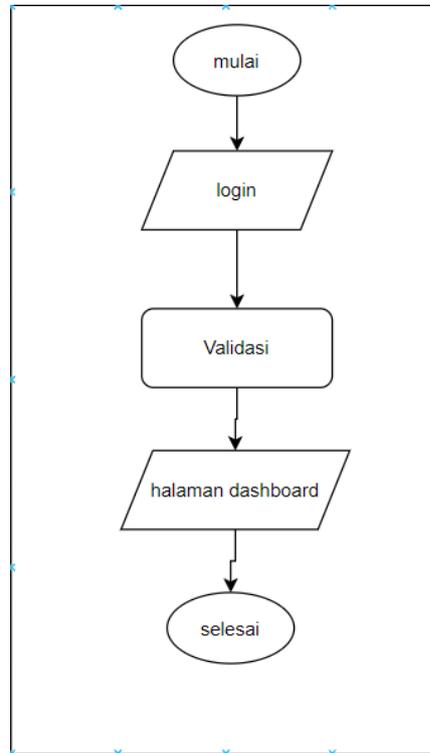
(4) Grafik matriks *login* berhasil

Tabel 4. 21 Grafik matriks *login* berhasil

	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)					$1 + 1 = 2$

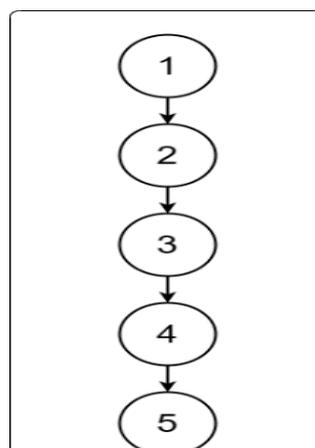
c. *White box testing halaman dashboard utama*

1) *Flowchart*



Gambar 4. 45 *Flowchart halaman dashboard utama*

2) *Flowgraph*



Gambar 4. 46 *Flowgraph dashboard utama*

Berdasarkan gambar 4. 47 diatas, dilakukan perhitungan sebagai berikut:

(5) Menghitung *cyclomatic complexity* $V(G)$ pada *egde* dan *node*

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E (\text{edge}) = 5$$

$$N (\text{node}) = 5$$

$$P (\text{predikat node}) = 1$$

Penyelesaian :

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

$$= 5 - 5 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

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

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

$$\text{Path 1} = 1 - 2 - 3 - 4 - 5$$

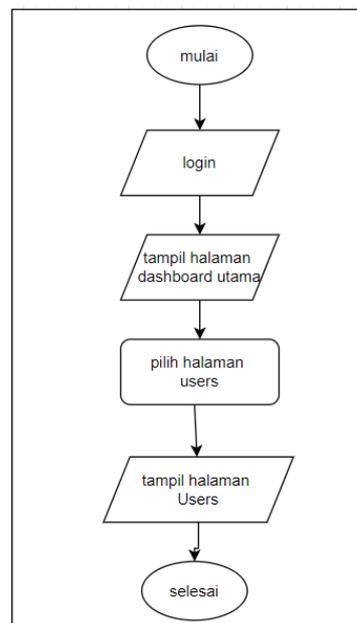
(8) Grafik matriks *halaman dashboard utama*

Tabel 4. 22 Grafik matriks halaman *dashboard* utama

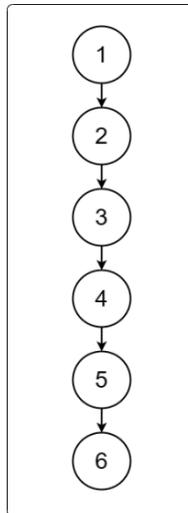
	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3				1		$1 - 1 = 0$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$1 + 1 = 2$

- *White box testing halaman Users*

1) *Flowchart*

**Gambar 4. 47 *Flowchart* halaman user**

2) *Flowgraph*



Gambar 4. 48 *Flowgraph* halaman users

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

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

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 4$$

$$N \text{ (node)} = 5$$

$$P \text{ (Predikat node)} = 0$$

Penyelesaian :

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

$$= 4 - 5 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas memiliki *region* = 1

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

$$\text{Path 1} = 1 - 2 - 3 - 4 - 5$$

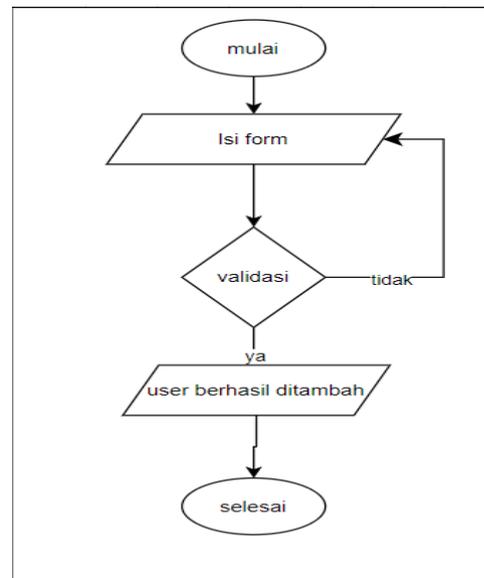
(4) Grafik matriks halaman Users

Tabel 4. 23 Grafik matriks halaman Users

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

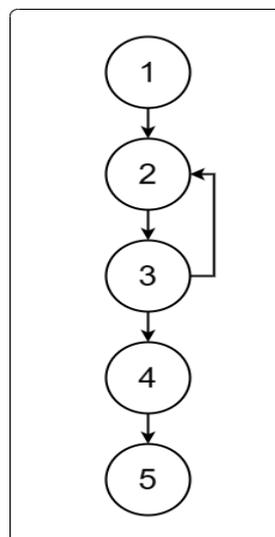
- **White box testing tambah User**

1) *Flowchart*



Gambar 4. 49 *Flowchart* tambah user

2) *Flowgraph*



Gambar 4. 50 *Flowgraph* tambah user

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

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

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 5$$

$$N \text{ (node)} = 5$$

$$P \text{ (predikat node)} = 1$$

Penyelesaian :

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

$$= 5 - 5 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

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

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

$$\text{Path 1} = 1 - 2 - 3 - 2$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5$$

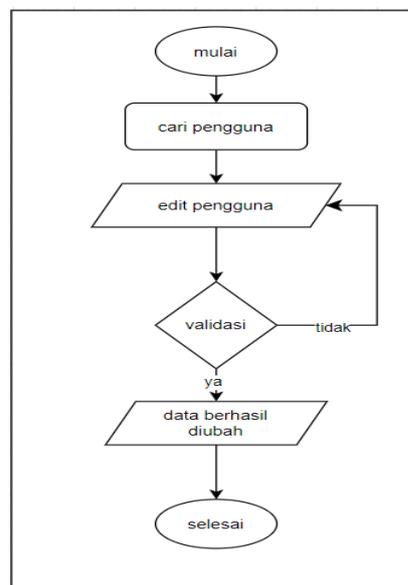
(1) Grafik matriks ubah data]

Tabel 4. 24 Grafik matriks tambah user

	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)					$1 + 1 = 2$

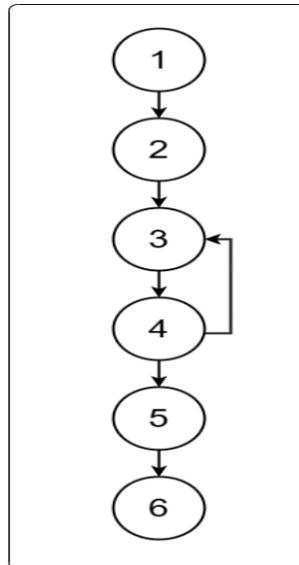
- *White box testing edit user*

3) Flowchart



Gambar 4. 51 Flowchart edit user

4) *Flowgraph*



Gambar 4. 52 *Flowgraph* edit user

Berdasarkan gambar 4. 52 diatas, dilakukan perhitungan sebagai berikut:

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

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 6$$

$$N \text{ (node)} = 6$$

$$P \text{ (Predikat node)} = 1$$

Penyelesaian :

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

$$= 6 - 6 + 2$$

$$= 2$$

$$\begin{aligned}
 \text{Predikat (P)} &= P + 1 \\
 &= 1 + 1 \\
 &= 2
 \end{aligned}$$

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

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

$$\text{Path 1} = 1 - 2 - 3 - 4 - 3$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5 - 6$$

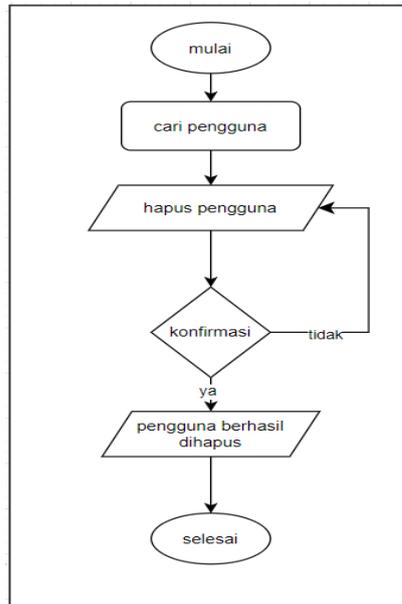
(5) Grafik matriks ubah *user*

Tabel 4. 25 Grafik matriks ubah *user*

	1	2	3	4	5	6	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							0	
	SUM (E + 1)							$1 + 1 = 2$

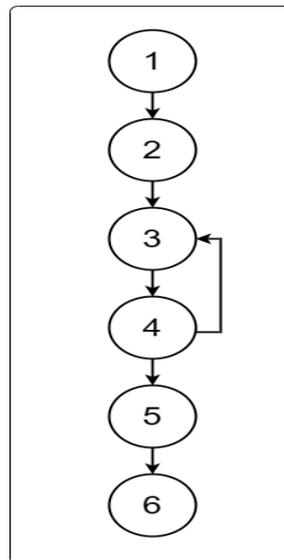
- **White box testing hapus user**

1) **Flowchart**



Gambar 4. 53 Flowchart hapus user

2) **Flowgraph**



Gambar 4. 54 Flowgraph hapus user

Berdasarkan gambar 4. 54 diatas, dilakukan perhitungan sebagai berikut:

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

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 6$$

$$N \text{ (node)} = 6$$

$$P \text{ (Predikat node)} = 1$$

Penyelesaian :

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

$$= 6 - 6 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

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

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

$$\text{Path 1} = 1 - 2 - 3 - 4 - 3$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5 - 6$$

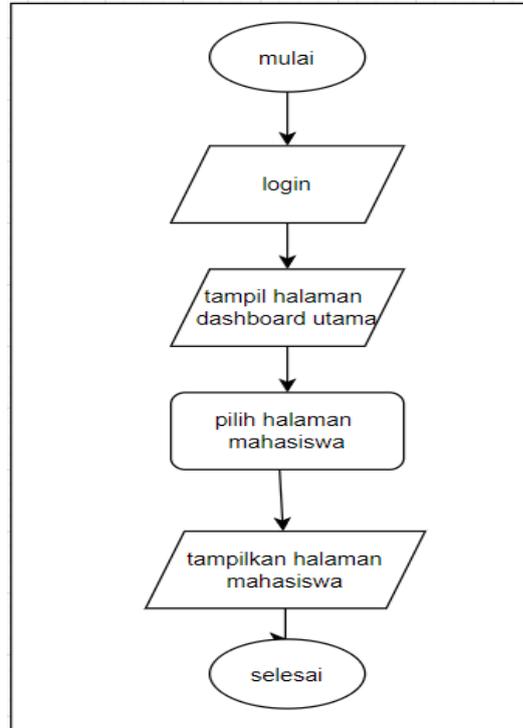
(4) Grafik matriks hapus *user*

Tabel 4. 26 Grafik matriks hapus *user*

	1	2	3	4	5	6	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							0	
	SUM (E + 1)							$1 + 1 = 2$

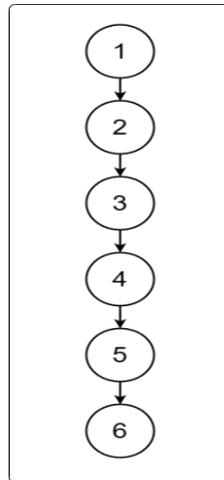
- **White box testing halaman mahasiswa**

1) Flowchart



Gambar 4. 55 Flowchart halaman mahasiswa

2) *Flowgraph*



Gambar 4. 56 *Flowgraph* halaman mahasiswa

Berdasarkan gambar 4. 56 diatas, dilakukan perhitungan sebagai berikut:

(5) Menghitung *cyclomatic complexity* $V(G)$ pada *egde* dan *node*.

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 4$$

$$N \text{ (node)} = 5$$

$$P \text{ (Predikat node)} = 0$$

Penyelesaian :

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

$$= 4 - 5 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

(6) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas memiliki *region* = 1

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

$$\text{Path 1} = 1 - 2 - 3 - 4 - 5$$

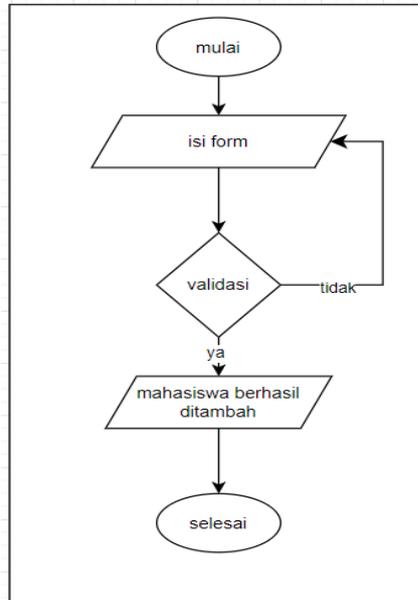
(8) Grafik matriks halaman mahasiswa

Tabel 4. 27 Grafik matriks halaman mahasiswa

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

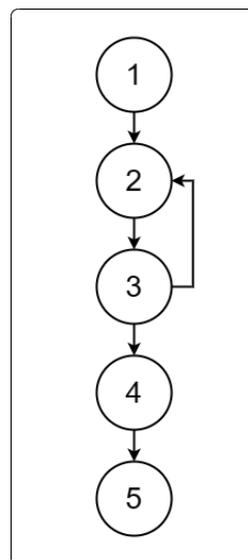
- *White box testing* tambah mahasiswa

1) *Flowchart*



Gambar 4. 57 *Flowchart* tambah mahasiswa

2) *Flowgraph*



Gambar 4. 58 *Flowgraph* tambah mahasiswa

Berdasarkan gambar 4. 58 diatas, dilakukan perhitungan sebagai berikut:

(4) Menghitung *cyclomatic complexity* $V(G)$ pada *egde* dan *node*

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 5$$

$$N \text{ (node)} = 5$$

$$P \text{ (predikat node)} = 1$$

Penyelesaian :

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

$$= 5 - 5 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

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

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

$$\text{Path 1} = 1 - 2 - 3 - 2$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5$$

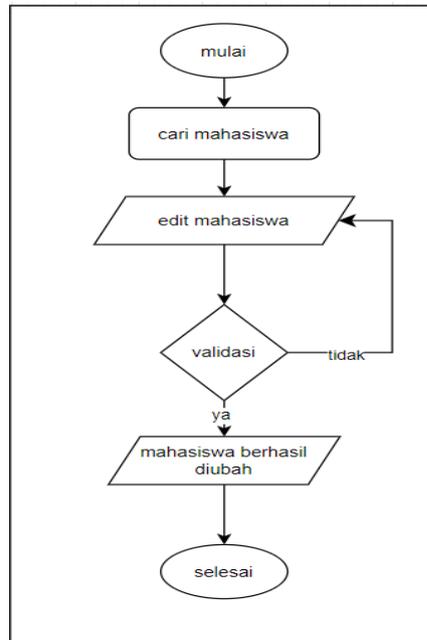
(6) Grafik matriks tambah mahasiswa

Tabel 4. 28 Grafik matriks tambah mahasiswa

	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)					$1 + 1 = 2$

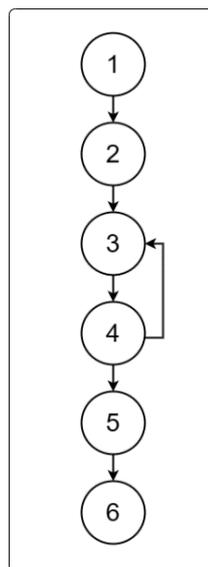
- **White box testing edit mahasiswa**

1) **Flowchart**



Gambar 4. 59 Flowchart edit mahasiswa

2) **Flowgraph**



Gambar 4. 60 Flowgraph edit mahasiswa

Berdasarkan gambar 4. 60 diatas, dilakukan perhitungan sebagai berikut:

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

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E (\textit{edge}) = 6$$

$$N (\textit{node}) = 6$$

$$P (\textit{Predikat node}) = 1$$

Penyelesaian :

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

$$= 6 - 6 + 2$$

$$= 2$$

$$\textit{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

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

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

$$\textit{Path 1} = 1 - 2 - 3 - 4 - 3$$

$$\textit{Path 2} = 1 - 2 - 3 - 4 - 5 - 6$$

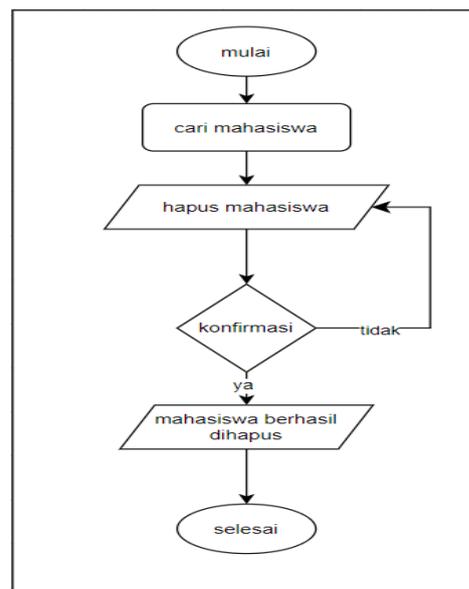
(10) Grafik matriks ubah mahasiswa

Tabel 4. 29 Grafik matriks edit mahasiswa

	1	2	3	4	5	6	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							0	
	SUM (E + 1)							$1 + 1 = 2$

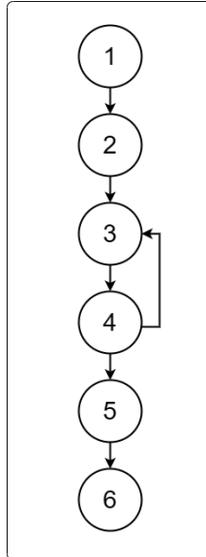
- *White box testing* hapus mahasiswa

1) *Flowchart*



Gambar 4. 61 Flowchart hapus mahasiswa

2) Flowgraph



Gambar 4. 62 Flowgraph hapus mahasiswa

Berdasarkan gambar 4. 62 diatas, dilakukan perhitungan sebagai berikut:

(5) Menghitung *cyclomatic complexity* $V(G)$ pada *egde* dan *node*

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 6$$

$$N \text{ (node)} = 6$$

$$P \text{ (Predikat node)} = 1$$

Penyelesaian :

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

$$= 6 - 6 + 2$$

$$= 2$$

$$\begin{aligned}
 \text{Predikat (P)} &= P + 1 \\
 &= 1 + 1 \\
 &= 2
 \end{aligned}$$

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

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

$$\text{Path 1} = 1 - 2 - 3 - 4 - 3$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5 - 6$$

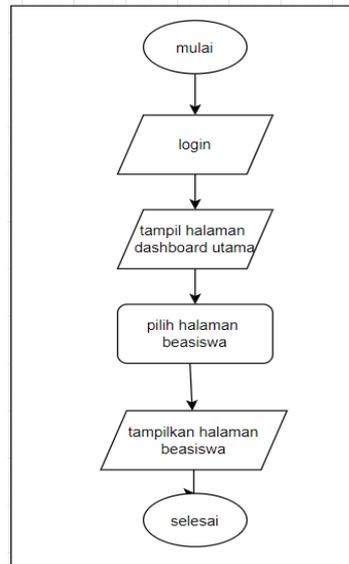
(8) Grafik matriks hapus mahasiswa

Tabel 4. 30 Grafik matriks hapus mahasiswa

	1	2	3	4	5	6	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							0	
	SUM (E + 1)							$1 + 1 = 2$

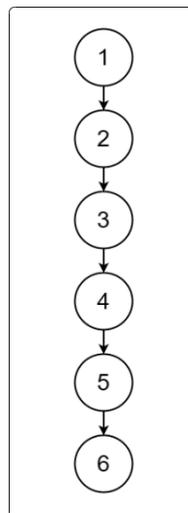
- **White box testing halaman beasiswa**

1) **Flowchart**



Gambar 4. 63 Flowchart halaman beasiswa

2) **Flowgraph**



Gambar 4. 64 Flowgraph halaman beasiswa

Berdasarkan gambar 4. 64 diatas, dilakukan perhitungan sebagai berikut:

(9) Menghitung *cyclomatic complexity* $V(G)$ pada *egde* dan *node*.

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 4$$

$$N \text{ (node)} = 5$$

$$P \text{ (Predikat node)} = 0$$

Penyelesaian :

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

$$= 4 - 5 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

(10) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas memiliki *region* = 1

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

$$\text{Path 1} = 1 - 2 - 3 - 4 - 5$$

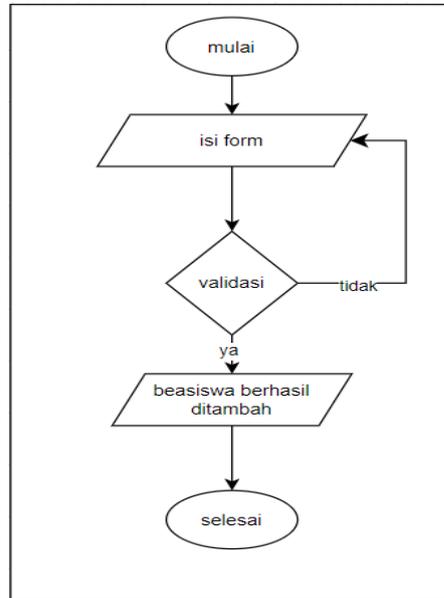
(12) Grafik matriks halaman beasiswa

Tabel 4. 31 Grafik matriks halaman beasiswa

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

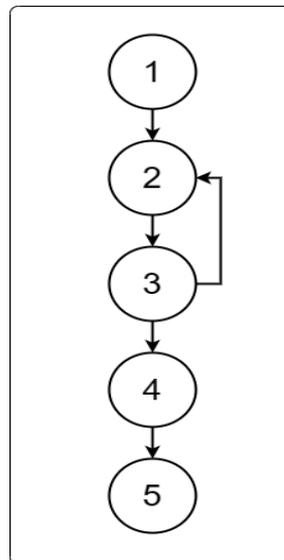
- *White box testing* tambah beasiswa

1) *Flowchart*



Gambar 4. 65 *Flowchart* tambah beasiswa

2) *Flowgraph*



Gambar 4. 66 *Flowgraph* tambah beasiswa

Berdasarkan gambar 4. 66 diatas, dilakukan perhitungan sebagai berikut:

(7) Menghitung *cyclomatic complexity* $V(G)$ pada *egde* dan *node*

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 5$$

$$N \text{ (node)} = 5$$

$$P \text{ (predikat node)} = 1$$

Penyelesaian :

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

$$= 5 - 5 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

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

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

$$\text{Path 1} = 1 - 2 - 3 - 2$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5$$

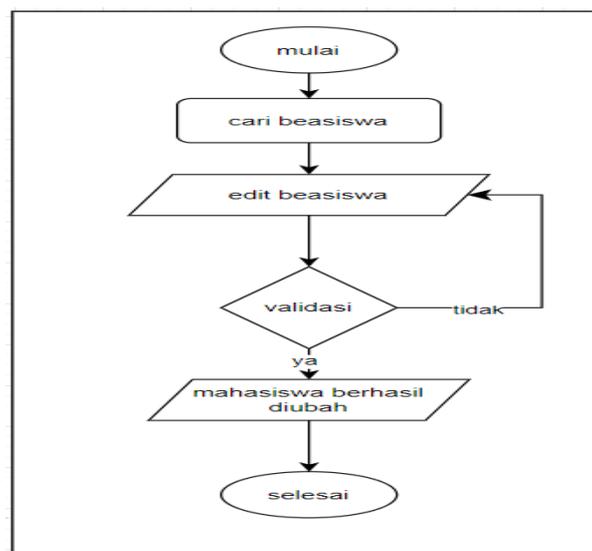
(11) Grafik matriks tambah beasiswa

Tabel 4. 32 Grafik matriks tambah beasiswa

	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)					$1 + 1 = 2$

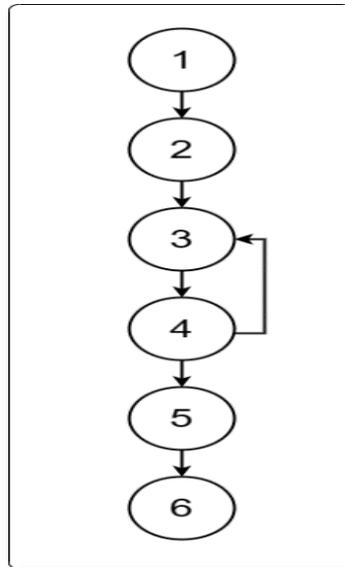
- *White box testing edit beasiswa*

1) *Flowchart*



Gambar 4. 67 Flowchart edit beasiswa

2) *Flowgraph*



Gambar 4. 68 *Flowgraph* edit beasiswa

Berdasarkan gambar 4. 68 diatas, dilakukan perhitungan sebagai berikut:

(12) Menghitung *cyclomatic complexity* $V(G)$ pada *egde* dan *node*

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 6$$

$$N \text{ (node)} = 6$$

$$P \text{ (Predikat node)} = 1$$

Penyelesaian :

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

$$= 6 - 6 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

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

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

$$\text{Path 1} = 1 - 2 - 3 - 4 - 3$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5 - 6$$

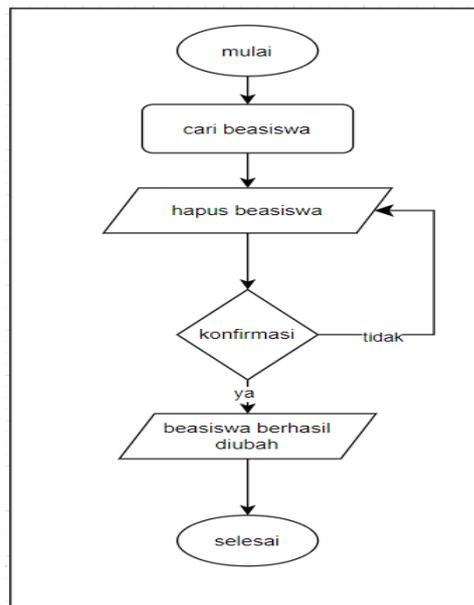
(15) Grafik matriks ubah beasiswa

Tabel 4. 33 Grafik matriks edit beasiswa

	1	2	3	4	5	6	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							0	
	SUM (E + 1)							$1 + 1 = 2$

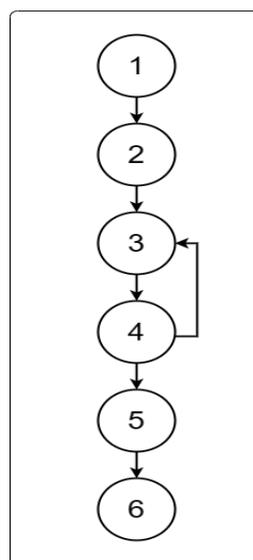
- **White box testing hapus beasiswa**

1) *Flowchart*



Gambar 4. 69 *Flowchart* hapus beasiswa

2) *Flowgraph*



Gambar 4. 70 *Flowgraph* hapus beasiswa

Berdasarkan gambar 4. 70 diatas, dilakukan perhitungan sebagai berikut:

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

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E (\textit{edge}) = 6$$

$$N (\textit{node}) = 6$$

$$P (\textit{Predikat node}) = 1$$

Penyelesaian :

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

$$= 6 - 6 + 2$$

$$= 2$$

$$\textit{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

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

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

$$\textit{Path 1} = 1 - 2 - 3 - 4 - 3$$

$$\textit{Path 2} = 1 - 2 - 3 - 4 - 5 - 6$$

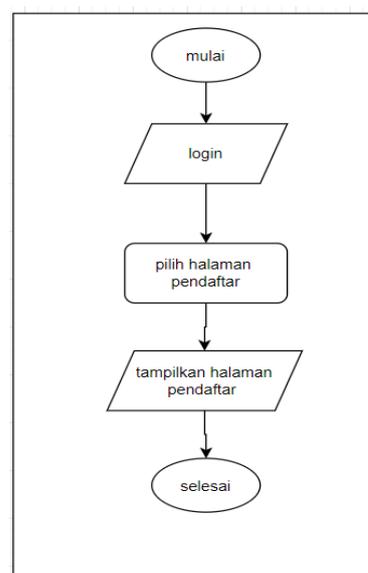
(12) Grafik matriks hapus beasiswa

Tabel 4. 34 Grafik matriks hapus beasiswa

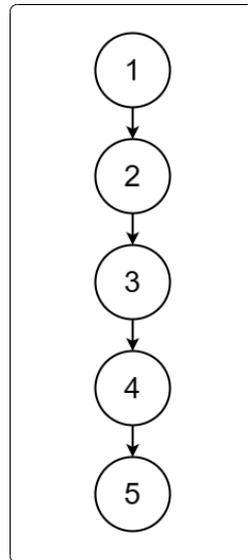
	1	2	3	4	5	6	E-1	
1		1					$1 - 1 = 0$	
2			1				$1 - 1 = 0$	
3				1			$1 - 1 = 0$	
4			1		1		$2 - 1 = 1$	
5						1	$1 - 1 = 0$	
6							0	
	SUM (E + 1)							$1 + 1 = 2$

- **White box testing halaman pendaftar beasiswa**

1) *Flowchart*

**Gambar 4. 71 Flowchart halaman pendaftar beasiswa**

2) *Flowgraph*



Gambar 4. 72 *Flowgraph* halaman pendaftar beasiswa

Berdasarkan gambar 4. 72 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *egde* dan *node*.

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 4$$

$$N \text{ (node)} = 5$$

$$P \text{ (Predikat node)} = 0$$

Penyelesaian :

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

$$= 4 - 5 + 2$$

$$= 1$$

$$\begin{aligned}
 \text{Predikat (P)} &= P + 1 \\
 &= 0 + 1 \\
 &= 1
 \end{aligned}$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas memiliki *region = 1*

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

$$\text{Path 1} = 1 - 2 - 3 - 4 - 5$$

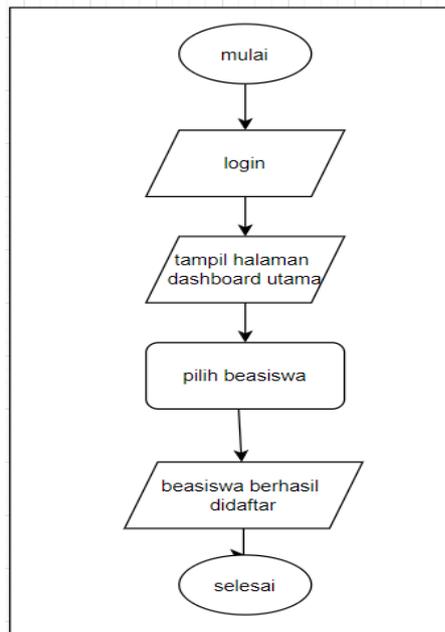
(4) Grafik matriks halaman pendaftar beasiswa

Tabel 4. 35 Grafik matriks halaman pendaftar beasiswa

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

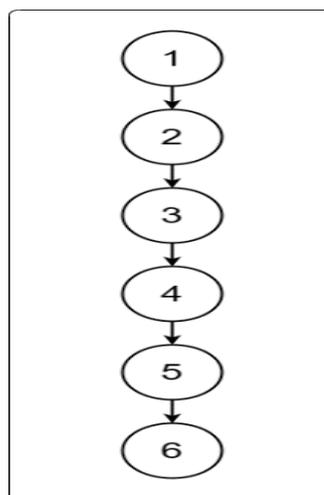
- **White box testing halaman mendaftar beasiswa**

1) *Flowchart*



Gambar 4. 73 *Flowchart* halaman mendaftar beasiswa

2) *Flowgraph*



Gambar 4. 74 *Flowgraph* halaman mendaftar beasiswa

Berdasarkan gambar 4. 74 diatas, dilakukan perhitungan sebagai berikut:

- (13) Menghitung *cyclomatic complexity* $V(G)$ pada *egde* dan *node*.

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E (\text{edge}) = 4$$

$$N (\text{node}) = 5$$

$$P (\text{Predikat node}) = 0$$

Penyelesaian :

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

$$= 4 - 5 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

- (14) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas memiliki *region = 1*

- (15) *Independent path* pada *flowgraph* yaitu:

$$\text{Path 1} = 1 - 2 - 3 - 4 - 5$$

- (16) Grafik matriks halaman mendaftar beasiswa

Tabel 4. 36 Grafik matriks halaman mendaftar beasiswa

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

BAB V

PENUTUP

A. Kesimpulan

Penelitian yang berjudul "Aplikasi Pengelolaan Bantuan Mahasiswa Berbasis Android di Universitas Muhammadiyah Parepare" ini bertujuan untuk mengembangkan sebuah aplikasi yang dapat membantu mahasiswa dalam mengelola bantuan yang diterima. Berikut kesimpulan yang dapat diambil dari penelitian ini:

1. Pengembangan Aplikasi:

- Penelitian ini menghasilkan aplikasi berbasis Android untuk pengelolaan bantuan mahasiswa.
- Aplikasi dilengkapi fitur manajemen data bantuan, pengguna, dan pendaftaran bantuan.
- Metode pengembangan menggunakan Rapid Application Development (RAD) dengan tahapan: perencanaan kebutuhan, lokakarya desain RAD, dan implementasi.
- Penilaian kelayakan aplikasi oleh ahli materi mendapat nilai rata-rata 4,33 (sangat layak) dan ahli media 4,08 (layak).

2. Implementasi:

- Penelitian dilaksanakan di Universitas Muhammadiyah Parepare.

- Perangkat keras yang digunakan adalah laptop Asus dengan spesifikasi minimal.
 - Metode pengumpulan data meliputi observasi, wawancara, dan studi pustaka.
 - Aplikasi dirancang untuk meningkatkan efisiensi pengelolaan dan pemanfaatan bantuan mahasiswa.
3. Keterbatasan dan Saran Pengembangan:
- Beberapa fitur aplikasi masih memerlukan penyempurnaan lebih lanjut.
 - Diperlukan implementasi dan pengujian aplikasi dalam lingkungan yang lebih luas.

B. Saran

Pada penelitian ini, penulis mengakui adanya beberapa kekurangan yang memerlukan perbaikan dan pengembangan lebih lanjut. Untuk itu, penulis menawarkan beberapa saran untuk penelitian yang akan datang, yaitu:

1. **Pengembangan Aplikasi:** Penelitian ini mengembangkan aplikasi berbasis Android yang dapat membantu mahasiswa dalam mengelola bantuan yang diterima. Aplikasi ini dilengkapi dengan fitur-fitur seperti pengelolaan data bantuan, pengelolaan pengguna, dan pengelolaan pendaftaran bantuan.
2. **Metode Pengembangan:** Penelitian ini menggunakan metode Rapid Application Development (RAD) yang terdiri dari fase requirements

planning, RAD design workshop, dan implementation. Penilaian kelayakan aplikasi dilakukan oleh ahli materi dan ahli media.

3. **Hasil Penelitian:** Hasil penelitian menunjukkan bahwa aplikasi ini dapat membantu mahasiswa dalam mengelola bantuan yang diterima dengan lebih efektif dan efisien. Penilaian kelayakan aplikasi mendapat nilai rata-rata sebesar 4,33 tergolong “sangat layak” untuk materi dan 4,08 tergolong “layak” untuk media.
4. **Implementasi:** Penelitian ini dilakukan di Universitas Muhammadiyah Parepare dengan menggunakan perangkat keras laptop Asus yang memiliki spesifikasi minimal. Penelitian ini juga menggunakan metode pengumpulan data yang meliputi observasi, wawancara, dan studi pustaka.
5. **Manfaat:** Penelitian ini dapat memberikan manfaat bagi mahasiswa dalam mengelola bantuan yang diterima. Aplikasi ini dapat membantu mahasiswa dalam mengatur dan mengelola data bantuan dengan lebih baik, serta meningkatkan efisiensi dalam penggunaan bantuan.
6. **Keterbatasan:** Penelitian ini masih memiliki beberapa keterbatasan, seperti perlu adanya penyempurnaan lebih lanjut terhadap fitur pada aplikasi yang belum aktif serta implementasi aplikasi pada remaja.

DAFTAR PUSTAKA

- BADARUDDIN NASUTION (2021), dalam penelitiannya yang berjudul “Perancangan Aplikasi Web Layanan Bimbingan Konseling Beasiswa Di Universitas Pembangunan Panca Budi
- Direktorat Pendidikan Kesetaraan, “Pendidikan Kesetaraan Mencerahkan Anak Bangsa” Dirjen PLS DEPDIKNAS. 2006.
- Hikmawati, Fenti, 2010. *Bimbingan Konseling* (Edisi Revisi), Jakarta: Raja Grafindo Persada,
- Kadir Abdul. 2006. *Dasar Aplikasi Database MySQL Delphi*. Yogyakarta: C.V ANDI OFFSET (Penerbit Andi).
- Masyhuri, M. Z. 2008. *Metodologi penelitian pendekatan praktis dan aplikatif*. Bandung: PT Refika Aditama.
- Nugroho, Adi. 2005. *Analisis dan Perancangan Sistem Informasi dengan Metodologi Berorientasi Objek*. Informatika. Bandung
- Nur Fitriyah Rahmawati, 2009 “*Implementasi Model Homeschooling Dalam Mengatasi Keterbatasan Pendidikan Formal*” Skripsi Universitas Negeri Yogyakarta:.
- PURNOMO JATI NUGROHO (2021), dalam penelitiannya yang berjudul “APLIKASI BIMBINGAN KONSELING SISWA DI SMK SUDIRMAN 1 WONOGIRI BERBASIS WEB
- Rahman Hakim, Arief, 2007, *Homeschooling; Rumah Kelasku, Dunia Sekolahku*, Jakarta: PT Kompas Media Nusantara,.
- Ramadhan Arif, S.Kom. 2006. *Pemrograman Web Database dengan PHP dan MySQL*. Jakarta: PT Elex Media Komputindo.
- Santoso, Harip. 2010. *Aplikasi Web/asp.net + cd*. Jakarta : Elex Media Komputindo.
- Sugiyono. 2001. *Metode Penelitian kuantitatif kualitatif dan R&D*. Bandung: Alfabeta.
- Sukanto, Rosa Ariani. 2009. *Analisa dan Desain Sistem Informasi*. Bandung: Informatika.
- Sutarman, 2007. *Membangun Aplikasi Web dengan PHP & MySQL*. Graha Ilmu. Yogyakarta

YUDI PERMANDES (2020) melakukan penelitian yang berjudul “RANCANG BANGUN SISTEM INFORMASI BIMBINGAN DAN KONSELING PADA SMA NEGERI 1 PAMPANGAN BERBASIS WEB