

BAB I

PENDAHULUAN

A. Latar Belakang

Kegiatan perkuliahan umumnya dilakukan dengan tatap muka antara dosen dengan mahasiswa dalam waktu yang terbatas. Lamanya waktu perkuliahan tergantung dari berapa SKS mata kuliah tersebut. SKS (satuan kredit semester), adalah satuan yang digunakan dalam sistem kredit semester yaitu sistem penyelenggaraan perkuliahan di perguruan tinggi yang menyatakan beban studi mahasiswa, beban kerja dosen dan beban penyelenggaraan perkuliahan dengan menggunakan Satuan Kredit Semester (SKS). Besarnya 1 (satu) satuan kredit semester dalam kegiatan perkuliahan pada berbagai perguruan tinggi terkadang berbeda namun ada juga yang sama. Kebanyakan suatu Universitas menerapkan 1 (satu) SKS dalam kegiatan perkuliahan adalah setara dengan alokasi waktu 50 menit kegiatan tatap muka terjadwal antara dosen dengan mahasiswa. Terbatasnya waktu tatap muka antara dosen dengan mahasiswa sering disertai dengan keterbatasan ruang juga kesediaan waktu dari dosen-dosen kemudian ada banyak mata kuliah yang diadakan dalam 1 (satu) semester, akan sulit bagi staf di suatu jurusan dan membutuhkan waktu yang lama untuk menjadwalkan kegiatan perkuliahan tersebut.

Dalam proses penentuan jadwal kuliah ini terdapat banyak aspek yang harus dipertimbangkan yaitu waktu per SKS, waktu mahasiswa, ketersediaan dosen, mata kuliah di masing-masing semester dan ruangan kelas, tidak jarang terdapat tumpang

tindih jadwal satu sama lain baik dari sisi mahasiswanya, dosen ataupun ruangnya. Oleh karena itu dalam menentukan jadwal kuliah di setiap semesternya dibutuhkan pembangunan suatu perangkat lunak yang menerapkan suatu metode atau algoritma yang akan menghasilkan penjadwalan kuliah yang optimal. Algoritma yang diterapkan pada pembangunan perangkat lunak penjadwalan kuliah ini adalah algoritma dengan menggunakan algoritma *Ant Colony Optimization* (ACO). Dengan menggunakan algoritma ini maka sistem akan mencari cara tercepat untuk mencapai suatu penempatan jadwal optimal dengan kata lain, jadwal mahasiswa, dosen dan ruangan tidak akan mengalami bentrok untuk 1 (satu) mata kuliah dengan mata kuliah yang lain.

Berdasarkan uraian diatas, penulis ingin merancang sebuah aplikasi sebagai tugas akhir untuk memudahkan akademisi, dosen, dan mahasiswa dalam pembuatan jadwal kuliah dengan judul penelitian adalah **“implementasi aplikasi penjadwalan kuliah pada fakultas teknik universitas muhammadiyah parepare”**.

B. Rumusan Masalah

Berdasarkan latar belakang diatas, maka rumusan masalah yang akan dibahas yaitu Bagaimana mengimplementasi aplikasi penjadwalan mata kuliah pada fakultas teknik universitas muhammadiyah parepare ?

C. Batasan Masalah

Penelitian ini memiliki beberapa batasan masalah dengan harapan penelitian terfokus dengan batasan-batasan yang dibuat. Adapun batasan dalam penelitian ini sebagai berikut :

1. Aplikasi ini mengatur jadwal perkuliahan pada mata kuliah teori agar tidak terjadi tumpang tindih antara mahasiswa, dosen, ruangan dan waktu
2. Pengguna aplikasi ini mencakup level admin yang dimana dapat mengatur, menyusun jadwal dan pelaporan jadwal setiap dosen dan mahasiswa.

D. Tujuan Penelitian

Adapun tujuan dari penelitian tugas akhir ini adalah mengimplementasikan aplikasi penjadwalan mata kuliah pada fakultas teknik di universitas muhammadiyah parepare.

E. Manfaat Penelitian

Diharapkan dengan penelitian ini dapat diambil beberapa manfaat yang mencakup tiga hal pokok berikut:

1. Manfaat bagi Universitas

Dapat di optimalkan dan dimanfaatkan dengan baik aplikasi pembuatan jadwal mata kuliah dan dikembangkan lebih lanjut untuk menyelesaikan masalah jadwal kuliah.

2. Manfaat bagi Peneliti

Dapat mengembangkan wawasan keilmuan dan merealisasikan apa yang telah dipelajari di kampus dengan penelitian yang dilakukan dan diharapkan hasil penelitian ini yang berupa aplikasi dapat digunakan sebagaimana mestinya.

F. Sistematika Penulisan

Adapun sistematika penulisan pada tugas akhir ini yaitu :

BAB I PENDAHULUAN

Pada bab ini berisikan tentang uraian latar belakang masalah judul ini diangkat, rumusan masalah, Batasan masalah, tujuan penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Dalam bab ini berisi mengenai beberapa teori-teori yang digunakan sebagai referensi dalam penyusunan tugas akhir ini serta pembahasan mengenai algoritma yang dipakai.

BAB III METODE PENELITIAN

Bab ini membahas tentang jenis penelitian, lokasi dan waktu penelitian, teknik pengumpulan data, alat dan bahan penelitian, tahapan penelitian, jenis data, desain sistem yang digunakan.

BAB IV ANALISI DAN PERANCANGAN SISTEM

Bab ini penulis menjelaskan tentang rancangan analisis sistem yang diusulkan dengan menggunakan *use case*, perancangan *database*, dan rancangan *input/output*.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan dan saran.

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Penjadwalan (*Scheduling*)

Penjadwalan adalah aktivitas perencanaan untuk menentukan kapan dan dimana setiap operasi sebagai bagian dari pekerjaan secara keseluruhan harus dilakukan pada sumber daya yang terbatas.

Menurut Baker (1974) penjadwalan adalah proses untuk melakukan tugas dengan menggunakan sumber-sumber yang tersedia pada waktu yang telah ditetapkan.

Menurut Stevenson (1999) penjadwalan adalah membangun penentuan waktu penggunaan dari peralatan, fasilitas dan aktivitas manusia dalam suatu organisasi

Menurut Pinedo (2002) penjadwalan adalah proses pengambilan keputusan yang memegang peranan yang penting dalam manufaktur dan sistem produksi.

2. Universitas Muhammadiyah Parepare

Universitas Muhammadiyah Parepare (UMPAR) secara resmi berdiri pada tanggal 10 Mei 1999 bertepatan dengan tanggal 24 Muharram 1420 H berdasarkan Surat Keputusan Mendikbud RI Nomor 86/D/O/1999 tanggal 10 Mei 1999, sebagai perubahan bentuk dari Sekolah Tinggi Keguruan dan Ilmu Pendidikan (STKIP) Muhammadiyah Parepare.

UMPAR dalam usia 22 tahun telah menapaki dua fase yaitu; fase perintisan, dan fase pengembangan. Fase perintisan ditandai dengan usaha peralihan dari Sekolah Tinggi menjadi Universitas. Sekolah Tinggi yakni STKIP Muhammadiyah Parepare pada saat itu membina tiga program studi, yakni program studi Pendidikan Matematika, Pendidikan Bahasa Inggris dan Pendidikan Luar Sekolah (PLS). Usaha perubahan bentuk menjadi Universitas yang diprakarsai oleh Drs. Said Amir Anjala, MM sekaligus sebagai Rektor Pertama.

Sampaisaat ini UMPAR telah membina 7 Fakultas diantaranya:

a. Fakultas Keguruan dan Ilmu Pendidikan (FKIP)

Fakultas Keguruan dan Ilmu Pendidikan Universitas Muhammadiyah Parepare (FKIP UMPAR) merupakan salah satu fakultas yang keberadaannya menyertai sejarah pendirian Universitas Muhammadiyah Parepare (UMPAR). Diawali sebagai Filial Fakultas Ilmu Pendidikan (FIP) Universitas Muhammadiyah Makassar secara resmi beroperasi pada tanggal 9 September 1963 dengan membuka Jurusan Pendidikan Umum (PU). Pada Tahun 1970, Jurusan Pendidikan Umum kemudian diganti menjadi Jurusan Pendidikan Sosial (Pensos), hingga pada Tahun 1974 mengalami perubahan dari Fakultas Filial menjadi FIP UNISMUH Cabang Parepare dengan status terdaftar SK. Kopertis Wilayah VII No. III Tahun 1974. Selanjutnya pada tahun 1978 berubah bentuk menjadi Sekolah Tinggi Keguruan dan Ilmu Pendidikan (STKIP) Muhammadiyah Parepare dengan Status terdaftar No. 0353/O/1978, tanggal 30 November 1978 dan diperbaharui kembali dengan ketetapan Menteri Pendidikan dan Kebudayaan Nomor: 48/O/1981 tanggal 22 Januari 1981.

Berbagai langkah strategis terus dilakukan disepanjang kurun waktu tersebut, khususnya pada pemenuhan kebutuhan jurusan yang diminati, salah satunya ditunjukkan dengan penyesuaian penamaan Jurusan Pendidikan Sosial menjadi Jurusan Pendidikan Luar Sekolah (PLS) dengan SK Mendikbud Nomor: 0392/O/1980. Selanjutnya pada tahun 1986, beberapa jurusan ditetapkan beroperasi sesuai SK Mendikbud No. 041/O/1990, yaitu: Jurusan Pendidikan Bahasa dan Seni (Prodi Pendidikan Bahasa Inggris), dan Jurusan Pendidikan Matematika dan Ilmu Pengetahuan Alam (Prodi Pendidikan Matematika).

Proses pembenahan terus dilakukan sebagai upaya peningkatan kualitas perguruan tinggi, hingga tanggal 10 Mei 1999 berdasarkan SK Mendikbud Nomor: 86/D/0/1999 STKIP Muhammadiyah Parepare mengalami perubahan bentuk menjadi Universitas Muhammadiyah Parepare (UMPAR). Memperhatikan ketetapan SK tersebut, maka beberapa program studi yang merupakan ruang lingkup Fakultas Keguruan dan Ilmu Pendidikan resmi dinyatakan terdaftar.

Fakultas Keguruan dan Ilmu Pendidikan Universitas Muhammadiyah Parepare (FKIP UMPAR) dalam perjalanannya terus berbenah diri dan melakukan berbagai upaya pengembangan dan peningkatan mutu. Salah satu aspek pencapaian hingga ditahun 2012, FKIP UMPAR membuka satu program studi, yaitu program studi pendidikan biologi sesuai Keputusan Menteri Pendidikan Kebudayaan Republik Indonesia, nomor: 337/E/O/2012 tentang penyelenggaraan program studi pendidikan biologi pada Universitas Muhammadiyah Parepare.

b. Fakultas Ilmu Kesehatan (FIKES)

Fakultas Ilmu Kesehatan Universitas Muhammadiyah Parepare merupakan salah satu fakultas dalam lingkup Universitas Muhammadiyah Parepare yang berdiri pada tanggal 09 Agustus 2004. Dengan demikian kegiatan operasional kegiatannya dimulai pada Tahun Akademik 2004/2005 dengan status terdaftar berdasarkan Surat Keputusan DIKTI No. 3062/D/T/2004 tentang Pendirian Program Studi Kesehatan Masyarakat. Dalam upaya mempertahankan kesinambungan dan kualitas penyelenggaraan pendidikan di perguruan tinggi, maka Fakultas Ilmu Kesehatan yang membawahi satu Program Studi yakni Program Studi Ilmu Kesehatan Masyarakat yang membina beberapa konsentrasi terus berbenah diri sesuai dengan perkembangan dengan tujuan agar kepercayaan masyarakat dapat diwujudkan melalui kerja nyata.

Legalitas izin operasional Program Studi Ilmu Kesehatan Masyarakat tertuang dalam Surat Keputusan DIKTI Nomor No.2360/D/T/200 Tanggal 09 Agustus 2004 tentang pendirian Program Studi Kesehatan Masyarakat Universitas Muhammadiyah Parepare dan mendapatkan Perpanjangan Izin Penyelenggaraan Program Studi berdasarkan Surat Keputusan Perpanjangan Izin Penyelenggaraan Program Studi No.274/D/T/2004, Tanggal 12 Pebruari 2007. Dan Program Studi Kesehatan Masyarakat terakreditasi oleh Badan Akreditasi Nasional (BAN) dengan ditandai dengan keluarnya SK BAN Nomor. 027/BAN-PT/Ak-XII/SI/IX/2009 Tertanggal 11 September 2009.

Fakultas Kesehatan Masyarakat (FKM) berganti Nama menjadi Fakultas Ilmu Kesehatan pada Tahun 2009 berdasarkan SK. Rektor UMPAR. Fakultas Ilmu Kesehatan yang membina Program Studi Kesehatan Masyarakat dengan 4 (empat)

Konsentrasi yaitu, Administrasi dan Kebijakan Kesehatan, Epidemiologi, Gizi serta Kesehatan Lingkungan. Program Studi Kesehatan Masyarakat Fakultas Ilmu Kesehatan Universitas Muhammadiyah Parepare saat ini telah Terakreditasi B Berdasarkan Surat Keputusan (SK) Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT) Nomor. 451/BAN-PT/Akred/S/XI/2014 tertanggal 15 November 2014.

c. Fakultas Hukum (FH)

Berdasarkan hasil survey dari beberapa instansi Pendidikan pemerintah di kota Parepare dan kabupaten/kota sekitarnya (Dinas Pendidikan Kota Parepare, Kab. Sidrap, Kab. Pinrang, Kab. Barru dan Kab. Enrekang), serta wawancara dengan beberapa *stakeholder* menyatakan Kota Parepare dan sekitarnya membutuhkan wadah untuk menciptakan akademisi dan praktisi dibidang hukum, olehnya itu diusulkan pembentukan Program studi Ilmu Hukum dengan surat resmi 27 mei 2012 dan pada tahun 2015 disahkan dengan Keputusan Kementrian Riset Teknologi dan Pendidikan dengan Berdasarkan surat keputusan Kementerian Riset Tekhnologi dan Pendidikan Tinggi Nomor :126/M/Kp/III/2015 tertanggal 31 Maret 2015 tentang Ijin penyelenggaraan program studi Ilmu Hukum Sarjana Strata Satu (S1). Program studi ini menaungi beberapa konsentrasi sebagai kekhususan seperti Hukum Lingkungan, Hukum Ekonomi Bisnis, Hukum Pidana dan Hukum Perdata,dll.

Pada awalnya program studi Ilmu Hukum berada dibawah naungan Fakultas Ekonomi dengan alasan untuk peningkatan profesionalisme sumberdaya di bidang hukum yang outputnya diharapkan bisa mendukung program pemerintah khususnya dalam menghasilkan generasi muda yang menguasai ilmu pengetahuan

dan teknologi terkhusus bidang ilmu-ilmu hukum maka dianggap perlu untuk mendirikan Fakultas Hukum, selain itu Keinginan *stakeholder* beserta masyarakat Parepare dan sekitarnya untuk pendirian/pembentukan Fakultas Hukum (FH) yang menaungi program studi Ilmu Hukum pada Universitas Muhammadiyah Parepare (UMPAR) telah dibuktikan dengan surat dukungan yang diberikan langsung oleh masyarakat umum dan masing-masing kepala Instansi ditambah dengan dukungan dari beberapa sekolah, aparat penegak hukum dan para siswa-siswa SMA atau sederajatnya.

Berdasarkan hasil rapat kerja Universitas Muhammadiyah Parepare di Taman Wisata Bantimurung, Maros Tanggal 28-29 Agustus 2016 disepakati pembentukan Fakultas Hukum (FH) dan ditindaklanjuti dengan surat pengusulan pendirian Fakultas Hukum dengan nomor surat 835/III.3.AU/C/2016, Kepada Rektor Universitas Muhammadiyah Parepare dan pada 3 Januari 2017 terbentuklah Fakultas Hukum (FH) dengan Keputusan Rektor Universitas Muhammadiyah Parepare Nomor: 023/KEP/III.3.AU/D/2017 tentang Pendirian Fakultas Hukum Universitas Muhammadiyah Parepare.

- d. Fakultas Pertanian, Peternakan dan Perikanan (FAPETRIK)
- e. Fakultas Ekonomi dan Bisnis (FEB)
- f. Fakultas Agama Islam (FAI)

Fakultas Agama Islam Universitas Muhammadiyah Parepare (FAI UM Parepare) merupakan salah satu fakultas di Universitas Muhammadiyah Parepare (UM Parepare) yang secara resmi berdiri pada tanggal 21 Juni 2002. Berdasarkan Surat Keputusan Direktur Jenderal Pendidikan Tinggi, Departemen Pendidikan

Nasional Republik Indonesia Nomor: Dj.II/148/2002, Fakultas Agama Islam telah diizinkan menyelenggarakan pendidikan dengan jenjang Program Sarjana.

Pada fase ini Fakultas Agama Islam yang semula memiliki 2 (dua) Program Studi yaitu Program Studi Pendidikan Agama Islam dan Program Studi Bimbingan dan Penyuluhan Islam. Disusul Program Studi Perbankan Syariah dengan keputusan Direktur Jenderal Pendidikan Islam melalui SK Nomor: 1891 Tahun 2012. Program Studi ini menjadi terfavorit dengan adanya terobosan yang dilakukan yaitu peningkatan kualitas dosen, menyediakan laboratorium bank mini dan komputer, dan menjalin kerjasama dengan sejumlah bank yang ada di Sulawesi Selatan khususnya di daerah Kota Parepare, setelah Program Studi Perbankan Syariah, kemudian disusul pembukaan Program Studi baru yaitu Program Studi Pendidikan Guru Raudhatul Athfal (PGRA) berdasarkan keputusan Direktur Jenderal Pendidikan Islam SK Nomor: 466 Tahun 2013.

g. Fakultas Teknik (FT)

Masing-masing Fakultas membina Program Studi, meliputi: Pendidikan Matematika, Pendidikan Bahasa Ingeris, Pendidikan Non Formal, Pendidikan Biologi dan Program Pendidikan Guru (PPG) untuk FKIP, Studi Pembangunan, Akuntansi, prodi Manajemen untuk FEB, Teknik Sipil, Teknik Elektro, Teknik Informatika dan Perencanaan Wilayah Kota (PWK) untuk FATEK, Agribisnis, Agroteknologi, Budidaya Perairan dan Nutrisi Makanan Ternak untuk FAPETRIK, AKK dan Epidemiologi dan Gizi untuk FIKES, Pendidikan Agama Islam, Bimbingan penyuluhan Islam, dan Perbankan Syariah untuk FAI. Ilmu Hukum untuk FAKULTAS HUKUM. Untuk Program Pascasarjana (2007) membina

Program Studi, diantaranya Program Studi Agribisnis, Pendidikan Bahasa Inggris dan Pendidikan Agama Islam, sementara untuk Program Doktor yakni Program Studi Pendidikan Agama Islam.

Penyelenggara pendidikan di UMPAR tidak lepas daripada visi, misi, dan tujuan rencana kebijakan pengembangan yang tertuang dalam RENSTRA UMPAR, terutama pada bidang akademik, sumberdaya dan mutu. Hingga saat ini UMPAR telah mengalami 5 kali pergantian Rektor dan Saat ini (2020-2024) UMPAR dipimpin oleh Dr. H.M. Nasir S, M.Pd, (Rektor), Asram AT. Jadda, S.H.I., M.Hum (Wakil Rektor I), Arham, S.E., M.Ak (Wakil Rektor II). Hamsyah, S.T., M.T (Wakil Rektor III). Muhammad Nur Maallah, S.Ag., M.A. (Wakil Rektor IV).

4. Aplikasi Desktop

Aplikasi desktop adalah suatu aplikasi yang dapat berjalan sendiri atau independen tanpa menggunakan browser atau koneksi internet di suatu computer otonom dengan sistem operasi atau platform tertentu (Konixbam, 2009).

Menurut Prasetyo (2008) aplikasi berbasis desktop dikembangkan untuk dijalankan di masing-masing klien (komputer pengakses aplikasi pengolahan database). Database diletakkan di server sedangkan aplikasinya diinstal di masing-masing klien. Bahasa pemrograman yang digunakan untuk aplikasi tipe ini biasanya adalah Borland Delphi, Visual Basic, Java, Netbeans dan sebagainya. Pada aplikasi berbasis desktop, aplikasi dibangun dengan menggunakan tools tertentu, kemudian dikompilasi. Hasilnyadapatlangsungdigunakandalamkomputer.

5. Dart

Dart adalah sebuah bahasa pemrograman yang dikembangkan oleh Google dan merupakan bahasa pemrograman resmi untuk *Flutter*, sebuah *UI toolkit* dan aplikasi multiplatform dari Google. *Flutter* sendiri telah digunakan oleh berbagai perusahaan besar seperti Google, Alibaba.com, dan Tencent karena dapat menghemat waktu dan tenaga dengan cara cukup membutuhkan satu codebase untuk mengembangkan aplikasi di berbagai platform, daripada harus menghabiskan waktu untuk membuat codebase terpisah untuk masing-masing platform.

- a. Dart merupakan bahasa yang wajib Anda kuasai untuk mengembangkan aplikasi *Flutter*.
- b. Dengan Dart, Anda hanya perlu satu codebase untuk dapat mengembangkan aplikasi pada berbagai platform seperti web, Android, dan iOS.
- c. Bahasa Dart dirancang supaya familier dengan bahasa pemrograman lain sehingga mudah bagi yang sudah mengerti bahasa pemrograman lain maupun bagi yang baru memulai perjalanannya sebagai developer.
- d. Penggunaan bahasa Dart itu gratis (open source) dan dikembangkan oleh komunitas developer ahli yang aktif dan terbuka.
- e. Dart adalah bahasa yang dioptimalkan untuk pengembangan UI secara cepat dan produktif pada banyak platform.

6. Git For Windows

GIT adalah singkatan dari *Group Inclusive Tour*, merupakan version control yang digunakan programmer untuk berkolaborasi dalam membuat atau mengembangkan aplikasi. Tujuan penggunaan *GIT* adalah untuk memudahkan

pengelolaan source code program, seperti adanya perubahan pada baris kode yang diganti atau ditambahkan.

Version Control adalah aplikasi yang mengelola dan mencatat setiap perubahan pada file. Setiap perubahan yang terjadi akan dicatat lengkap dengan informasi siapa yang merubah, kapan perubahan dilakukan, dan isi perubahannya.

GIT adalah perangkat lunak yang digunakan oleh Linus Torvald untuk membuat kernel Linux pada tahun 2005. Selanjutnya, GIT dikelola oleh Junio Hamano, seorang programmer asal Jepang. Git merupakan tool gratis dan bisa diinstal di perangkat dengan system operasi Mac, Windows, Linux, hingga Solaris.

GIT adalah aplikasi dengan banyak fungsi yang akan Anda rasakan manfaatnya ketika Anda mulai melakukan coding. Beberapa fungsi GIT sebagai berikut:

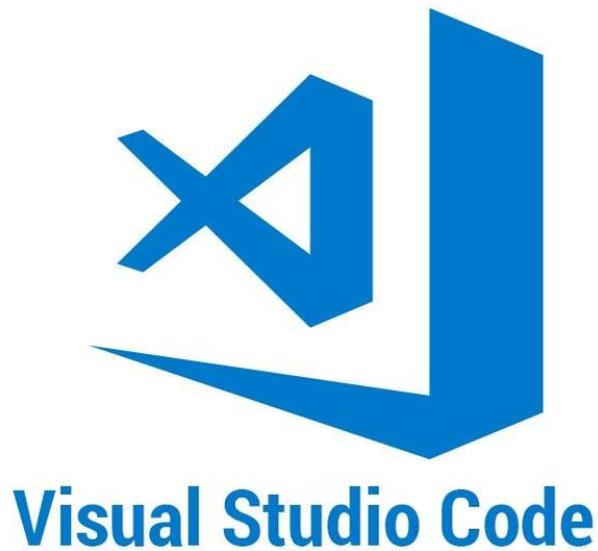
- a. Kolaboratif Dengan menggunakan GIT, Anda dapat bekerja bersama programmer lain dalam membuat atau mengembangkan sistem. Anda tidak perlu khawatir source code Anda akan tercampur maupun rusak, karena setiap programmer akan bekerja di dalam branch-nya masing-masing. Perubahan yang terjadi di dalam code program juga akan lebih mudah dilacak. Menariknya, jika code program selesai dibuat, maka setiap code program dari masing-masing programmer dapat disatukan dengan mudah.
- b. *Open Source* GIT adalah software yang bersifat open source. Hal ini berarti Anda dapat menggunakan aplikasi GIT secara gratis, tanpa biaya tambahan apapun.

- c. Membantu Organisir Dengan menggunakan GIT Anda dapat membuat folder project berdasarkan versinya, misal project v1, v2 dan seterusnya. Kemudian, Anda akan memiliki 1 project dengan menggunakan database khusus yang berisi semua file versi projek yang Anda miliki.
- d. Sebagai Platform yang Memiliki Fleksibilitas Ada banyak aplikasi GIT yang bisa Anda gunakan. Seperti Gitlab, GitHub, Bitbucked, SourceForge. Git menjadi platform yang memiliki fleksibilitas, dapat digunakan untuk mengerjakan berbagai projek Bersama tim Anda.
- e. Backup GIT memiliki fungsi untuk backup. Dengan fungsi ini, Anda dapat dengan mudah melakukan restore script program Kembali ke keadaan sebelumnya.

7. Visual Studio Code

Visual Studio Code (VS Code) ini adalah sebuah editor ringan dan handal yang dibuat oleh *Microsoft* untuk sistem operasi multiplatform, artinya tersedia juga untuk versi *Linux, Mac*, dan *Windows*. Teks editor ini secara langsung mendukung Bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang via *Marketplace Visual studio code* (seperti C++, C#, Python, Go, Java, dst).

Banyak sekali fitur-fitur yang disediakan oleh *Visual Studio Code*, diantaranya *Intellisense, Git Integration, Debugging*, dan fitur ekstensi yang menambah kemampuan teks editor. Fitur-fitur tersebut akan terus bertambah seiring dengan bertambahnya versi *Visual Studio Code*.



Gambar 2.1 Logo *Visual Studio Code*

8. Flutter

Flutter merupakan sebuah *SDK* untuk pengembangan aplikasi mobile yang dikembangkan oleh *Google* untuk membangun aplikasi yang memiliki kinerja tinggi serta dapat dipublikasi ke *platform Android* dan *IOS* dari *codebase* tunggal. *Flutter* dapat dengan mudah dipelajari karena menggunakan Bahasa pemrograman *Dart* yang pastinya terasa familier jika sudah terbiasa menggunakan bahasa pemrograman *Java* atau *Javascript*. Selain itu *Flutter* juga menyertakan kerangka *reactive-functional*, mesin *render2D*, *widget* siap pakai, dan tools untuk pengembangan (Muslim, Sari, R dan Rahmayuda, S, 2022)

Ada begitubanyakkerangka yang bisa digunakan untuk mengembangkan aplikasi lintas *platform*, seperti *React Native*, *Nativescript*, dan *Fuse*. Namun yang membedakannya adalah, *Flutter* tidak menggunakan *Webview* maupun *widget* bawaan, *Flutter* punya mesin render sendiri untuk menampilkan *widget*-nya, hal ini

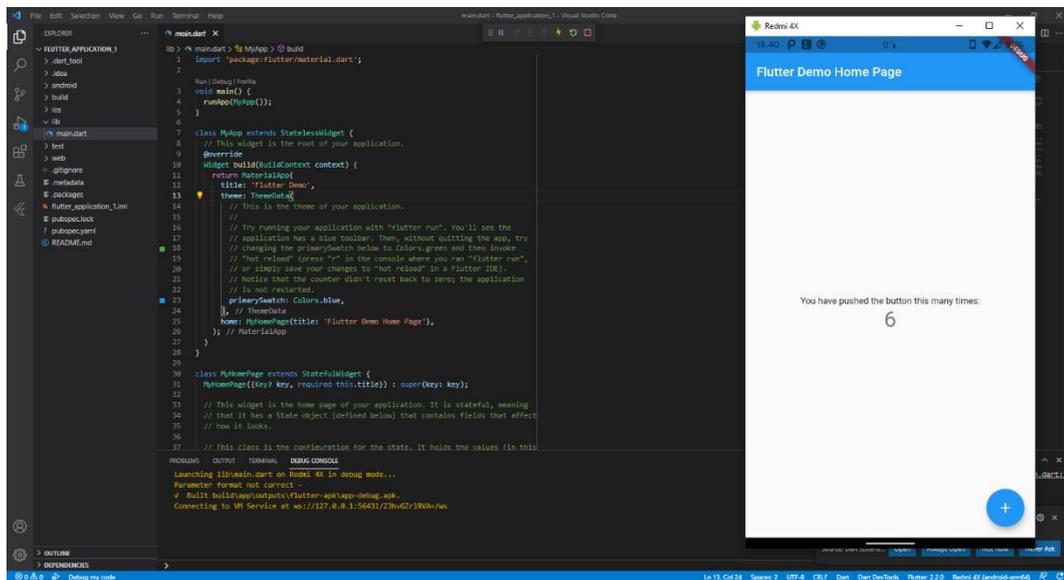
menguntungkan *developer* yang ingin memiliki tampilan UI unik yang konsisten pada semua perangkat karena tidak bergantung pada *widget* bawaan OEM. Kinerja *Flutter* yang tinggi ini tentunya didukung oleh berbagai teknologi terbaik. *Flutter* dibuat dengan *C*, *C++*, *Dart*, *Skia* untuk mesin render 2D, *Mojo IPC*, dan *Blink* untuk sistem render.

Cara kerja *Flutter* pada platform *Android* yaitu, kode *C/C++* dikompilasi menggunakan *AndroidSDK*, sebagian besar kerangka dan kode aplikasi dijalankan dalam bentuk kode *native* yang dikompilasi oleh *Dart compiler*. Sedangkan pada platform *iOS*, kode dikompilasi dengan *LLVM* dan aplikasi dijalankan dengan kumpulan instruksi *native* tanpa *interpreter*.

Flutter dapat berjalan pada sistem operasi *Android* 4.1 atau lebih tinggi dan *iOS* 8 atau lebih tinggi; dan dapat dijalankan pada perangkat asli maupun *Androidemulator*, serta *iOSSimulator*. Semua *developer* pastinya menginginkan kinerja aplikasi yang paling optimal. *Flutter* menjanjikan kinerja yang sempurna, aplikasi yang dibangun dengan *Flutter* dapat berjalan secara konstan pada tingkat *60 frame per detik* hingga *120fps*.

Alasan utama mengapa *Flutter* tidak menggunakan *widget* bawaan yaitu, *Flutter* tidak ingin kinerjanya dibatasi dan bergantung pada kinerja *widget* bawaan sehingga aplikasi kualitas tinggi dapat dihasilkan. *Flutter* sudah dipaketkan dengan kode pengaksesan *platform service* dan *API*, seperti sensor, penyimpanan lokal dan lain sebagainya. Namun, untuk menghindari kendala, pengaksesan *API* untuk platform yang spesifik, dapat digunakan dengan plugin untuk integrasi. (Sari, 2022).

Berikut merupakan beberapa dokumentasi mengenai *Flutter*:



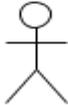
Gambar 2.2 Tampilan *flutter*

9. Use Case Diagram

Diagram *use case* menyajikan interaksi antara *use case* dan aktor, Dimana aktor dapat berupa orang, peralatan, atau sistem lain yang berinteraksi dengan sistem yang sedang di bangun. *Use case* menggambarkan fungsionalitas system atau persyaratan-persyaratan yang harus dipenuhi system dari pandangan pemakai. (Sholih, 2006).

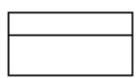
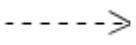
Adapun simbol-simbol Use Case Diagram antara lain :

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 (<i>independent</i>).
3		<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>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
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 aktor

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

Tabel 2.2 Simbol *Class Diagram*

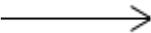
NO.	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan 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 aktor
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*

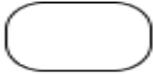
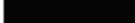
NO.	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi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

Tabel 2.4 Simbol *State Chart Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		State	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 Activity Diagram*

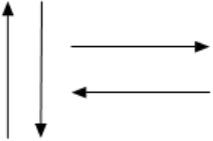
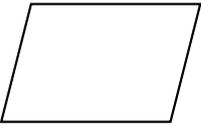
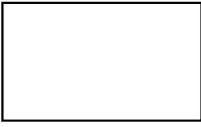
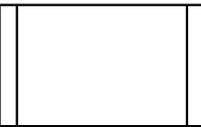
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node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

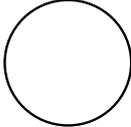
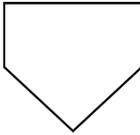
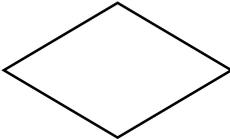
10. Flowchart

“Flowchart atau bagan alir adalah representasi grafik dari sistem yang mendeskripsikan relasi fisik diantara entitas – entitas intinya. Bagan alir dapat digunakan untuk menyajikan aktivitas manual, aktivitas pemrosesan komputer, atau keduanya. Bagan alir dokumen (document flowchart) digunakan untuk menggambarkan elemen–elemen dari sistem manual, termasuk catatan akuntansi

(dokumen, jurnal, buku besar, dan file), departemen organisasi yang terlibat dalam proses dan aktivitas (baik yang bersifat administratif maupun fisik) yang dilakukan Simbol – simbol yang digunakan dalam flowchart (Hasyim, 2021). ” adalah sebagai berikut:

Tabel 2. 6 Simbol-simbol flowchart

No.	Simbol	Keterangan
1.		Simbol ini digunakan untuk menghubungkan langkah-langkah dalam <i>flowchart</i> dan menunjukkan urutan operasi.
2.		Simbol ini digunakan ketika data dimasukkan (<i>input</i>) atau dikeluarkan (<i>output</i>) dari proses.
3.		Simbol ini menggambarkan aktivitas atau tugas yang harus diselesaikan.
4.		Simbol proses terdefinisi digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan di tempat lain (<i>procedure</i>).

No.	Simbol	Keterangan
5.		Simbol ini digunakan untuk memulai dan mengakhiri <i>flowchart</i> .
6.		Simbol ini digunakan untuk mengindikasikan bahwa aliran berlanjut ke titik lain dalam diagram.
7.		Simbol penghubung / <i>connector symbol</i> adalah simbol yang digunakan untuk menunjukkan sambungan dari alir atau proses yang terputus dan masih dalam halaman yang sama.
8.		Simbol ini menggambarkan percabangan dalam alur proses berdasarkan kondisi atau pernyataan logis.
9.		Simbol ini menggambarkan dokumen yang dibuat, digunakan, atau dihasilkan dalam langkah-langkah tertentu.
10.		Simbol <i>database</i> digunakan untuk mewakili <i>database</i>

B. Kajian Penelitian Terdahulu

1. Rudy Hartadi, Arief Hidayat, Victor G Utomo (2016), "Perancangan Aplikasi Penjadwalan Mata Kuliah". Di penelitian ini peneliti menggunakan Microsoft SQL server 2008 R2 sebagai pengembang aplikasinya.

2. Salim, Akbar (2016), “Aplikasi Jadwal Mata Kuliah Teknik Informatika Dan Sistem Informasi dan Penjadwalan Ruang Kuliah Berbasis Desktop”. Di penelitian ini peneliti menggunakan pemrograman *Delphi* sebagai pengembang aplikasi dan *Microsoft access* sebagai tempat pengolahan *database*.
3. Fauzi (2015) dengan judul penelitian “Optimasi Penjadwalan Mata kuliah Dengan Menggunakan Algoritma Genetika”. Pada Penelitian ini, akan diuji coba metode optimasi dalam pembuatan jadwal perkuliahan yaitu algoritma Genetika. Algoritma Genetika merupakan pendekatan komputasional untuk menyelesaikan masalah yang dimodelkan dengan proses biologi dari evolusi. Parameter-parameter Algoritma genetika yang memengaruhi jadwal perkuliahan yang dihasilkan adalah jumlah individu, probabilitas *crossover*, probabilitas mutase serta metode seleksi, *crossover*, *mutation* yang digunakan. Pengujian dilakukan dengan cara mencari nilai parameter-parameter algoritma genetika yang paling optimal dalam jadwal perkuliahan. Hasil penelitian menunjukkan bahwa dengan 100 jumlah generasi, jumlah individu, 30% probabilitas *crossover* dan 50 probabilitas *mutation* dapat menghasilkan jadwal yang paling optimal.

Berdasarkan dari kedua penelitian di atas, penelitian ini memiliki perbedaan yakni: objek penelitian, fitur, algoritma, dan Bahasa pemrograman yang digunakan.

BAB III

METODOLOGI PENELITIAN

A. Jenis Penelitian

Jenis penelitian yang digunakan yaitu:

- a. Penelitian lapangan, yang mana penelitian ini dilakukan secara langsung terhadap objek yang akan diteliti. Dalam penelitian lapangan, yang dilakukan penulis yakni melakukan pengumpulan data dengan Teknik wawancara dan observasi.
- b. Penelitian Pustaka yang mana penelitian ini dilakukan dengan menggunakan beberapa buku sebagai referensi untuk menulis

B. Lokasi dan Waktu Penelitian

Lokasi penelitian ini di Universitas Muhammadiyah Parepare beralamat di Jln. Jend. Ahmad Yani KM. 6, Bukit Harapan, Soreang, Kota Parepare, Sulawesi Selatan 91112, dan waktu penelitian ini akan dilaksanakan selama kurang lebih ± 2 Bulan pada tahun 2022.

C. Alat dan Bahan Penelitian

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

1. Alat penelitian

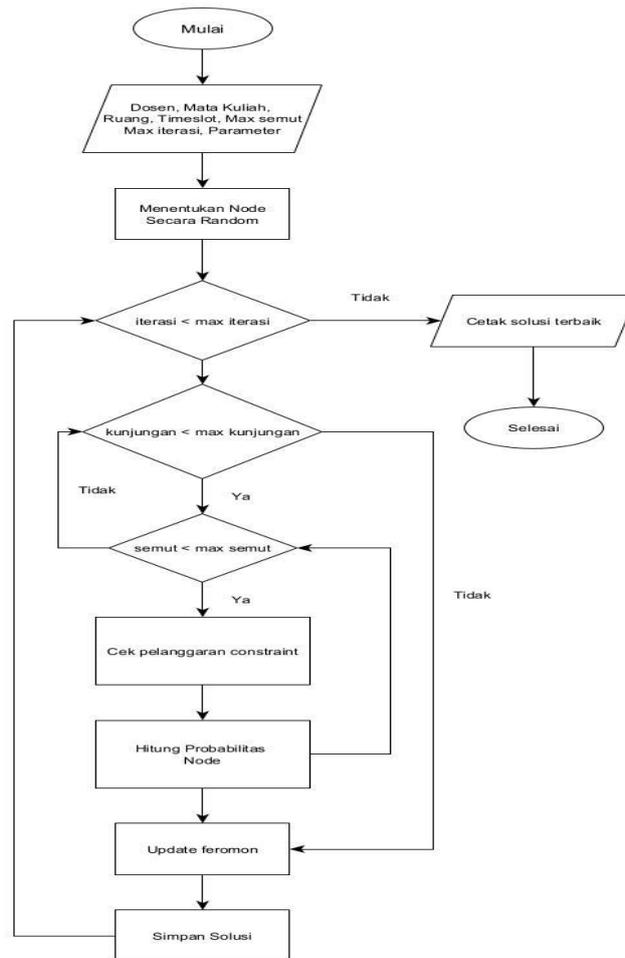
a. Perangkat keras yang digunakan selama proses penelitian yaitu:

- 1) Laptop Asus
- 2) Printer Epson L1110

b. Alat penelitian yang digunakan berupa *software* yaitu :

- 1) Sistem Operasi : *Windows 11 Home Single Language*
- 2) Aplikasi program : *Visual Studio Code*

D. Desain Sistem



Gambar 3.1 *flowchart* desain sistem

Pada gambar 3.1 menjelaskan *flowchart* desain sistem yang akan dibuat bekerja. Proses dimulai dengan proses input data kegiatan matakuliah (dosen, matakuliah, ruang, dan parameter untuk proses algoritma). Proses dilanjutkan dengan proses pengecekan apakah iterasi, kunjungan, dan semut yang sedang berjalan melebihi batas maksimal yang sudah di input sebelumnya. Setelah pengecekan, akan ada Batasan atau *constraint* untuk penjadwalan. Batasan atau *constraint* adalah peraturan yang diperoleh dari penjadwalan secara manual yang telah dilakukan sebelumnya oleh bagian akademik fakultas.

Langkah selanjutnya adalah penghitungan probabilitas dari setiap node yang dilewati dan pembaharuan feromon di jalur yang telah dilewati. Selanjutnya, Solusi perjalanan akan disimpan dan pengecekan apabila max iterasi sudah tidak melebihi iterasi maka akan dilanjutkan dengan ekstraksi Solusi terbaik. Pada bagian ini algoritma *Ant Colony Optimization* akan bekerja untuk membantu optimasi saat menghasilkan jadwal pengajaran dari data yang telah di input oleh pengguna sehingga membantu mengurangi adanya kesalahan dalam penjadwalan secara otomatis.

E. Metode Pengumpulan Data

1. Observasi

Merupakan metode pengumpulan data dengan melakukan pengamatan langsung mengenai sistem yang telah ada, yang kemudian mencatat dengan sebaik mungkin.

2. Wawancara

Wawancara adalah bentuk komunikasi antara penulis dengan *responden*. Komunikasi ini berlangsung dalam tanya jawab kepada pihak yang bersangkutan.

3. Studi Pustaka

Mengumpulkan data dengan mempelajari masalah yang berhubungan dengan objek yang diteliti bersumber dari buku-buku pedoman, literatur yang disusun oleh para ahli untuk melengkapi data yang diperlukan dalam penelitian baik secara *offline* maupun *online*.

F. Teknik Pengumpulan Data

Metode pengumpulan data dilakukan dengan melalui studi Pustaka, terutama yang berhubungan dengan data-data sekunder. Sementara data primer dilakukan melalui studi lapangan yaitu berupa :

1. Observasi, yaitu mengumpulkan data dengan cara mengetahui informasi dan data awal tentang keadaan objek penelitian.
2. Dokumentasi, yaitu mengumpulkan data dalam bentuk dokumen atau catatan tertulis.

G. Teknik Analisis Data

Jenis data yang dikumpulkan dalam penelitian ini adalah data primer dan data sekunder yang relevan dengan masalah yang akan dibahas.

1. Data Primer

Data Primer adalah data yang berasal atau data yang diperoleh langsung dari sumber-sumber yang berada dilapangan yang berasal dari wawancara dan observasi.

2. Data Sekunder

Data sekunder adalah data yang diperoleh tidak secara langsung dari objek penelitian. Penelitian mendapatkan data yang sudah jadi yang dikumpulkan oleh pihak lain dengan berbagai cara atau metode baik secara komersial maupun non komersial.

H. Metode Pengujian

Pengujian unit digunakan untuk menguji setiap modul untuk menjamin setiap modul menjalankan fungsinya dengan baik. Ada 2 metode untuk melakukan unit testing, yaitu:

1. *White Box Testing*

Coba *white box testing* merupakan metode perancangan *testcase* yang menggunakan structural untuk mendapatkan *testcase*, test ini digunakan untuk meramal cara kerja perangkat lunak secara rinci kepada *logic path* (jalur logika), perangkat lunak di tes dengan kondisi dan perulangan secara fisik.

Contoh pengujian white box testing ini merupakan peringatan ketika user menginputkan password user yang salah, untuk kesalahan semacam ini akan memberikan suatu informasi kepada user mengenai kesalahan yang di lakukan.

2. *Black Box*

Berdasarkan hasil uji coba yang dilakukan, seluruh navigasi dan tombol fasilitas program lainnya serta proses yang di jalankan tidak terjadi kesalahan, tetapi aplikasi mempunyai aturan-aturan yang sudah di tetapkan dan harus di ikuti karena apabila di hiraukan maka Sistem akan menolak perintah yang tidak sesuai seperti kesalahan ketika user belum menginput data yang harusnya di input sesuai

ketentuan sistem yang di jalankan dan system memberikan informasi kepada user karena data yang ingin diproses belum lengkap atau tidak memenuhi ketentuan untuk proses selanjutnya.

BAB IV

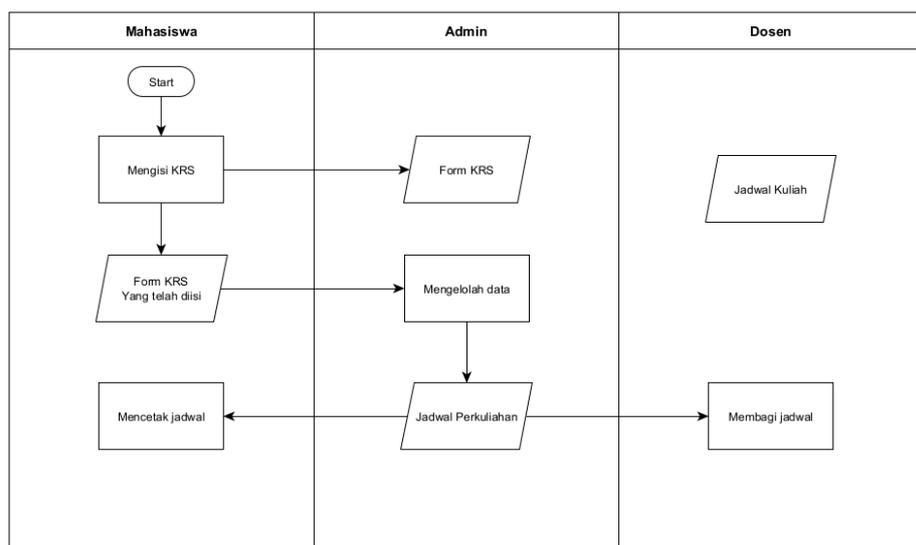
HASIL DAN PEMBAHASAN

A. Hasil

Pada bagian ini akan dipaparkan Analisis Sistem yang berjalan saat ini pada pembuatan jadwal kuliah Universitas Muhammadiyah Parepare serta sistem yang diusulkan.

1. Analisa Sistem yang berjalan.

Menganalisis sistem yang sedang berjalan bertujuan untuk mengetahui lebih jelas bagaimana carakerja sistem dan masalah yang akan dihadapi system untuk dijalankan landasan usulan perencanaan sistem. Berdasarkan pengamatan dan wawancara yang penulislakukan, kegiatan jadwal matakuliah Universitas Muhammadiyah Parepare dilakukan secara manual karena belum adanya sistem yang mengatur kegiatan ini.



Gambar 4.1 Analisa sistem yang berjalan

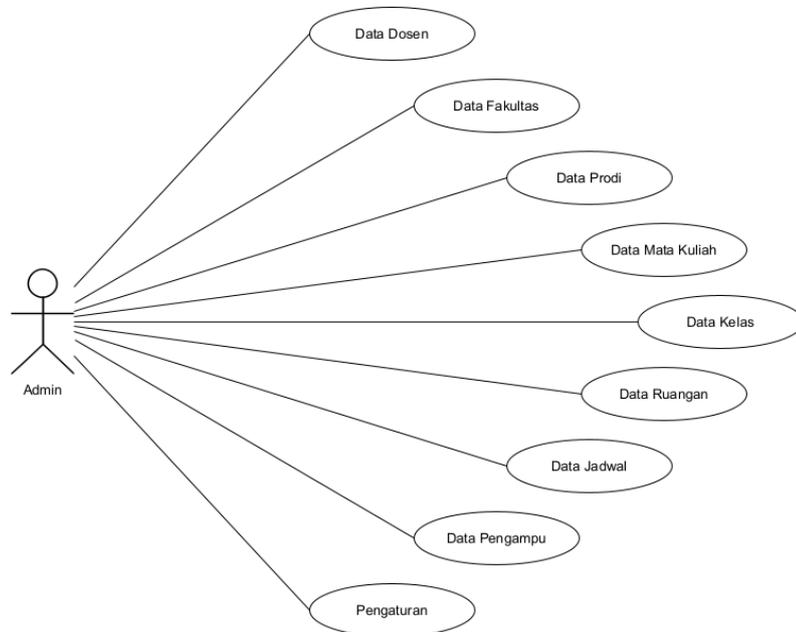
Penjadwalan matakuliah pada Universitas Muhammadiyah Parepare terbagi menjadi beberapa proses yang dilakukan berdasarkan urutan. Adapun analisis prosedur proses jadwal mata kuliah yang berjalan.

- a. Mahasiswa melakukan pengisian kartu rencana studi (KRS) pada form KRS, matakuliah apa yang diambil maupun yang mau diulang Kembali untuk semester selanjutnya. Kemudian KRS dikumpul di masing-masing fakultas.
- b. Pada form KRS yang telah diisi mahasiswa selanjutnya dari pihak fakultas menyeter atau mengumpulkan data KRS ke akademik masing-masing kemudian ketahap admin untuk mengolah data KRS mahasiswa menjadi sebuah jadwal perkuliahan mahasiswa, baik untuk semester selanjutnya maupun yang mengulang.
- c. Pada tahap selanjutnya data KRS (kartu rencana studi) yang telah terolah menjadi sebuah jadwal perkuliahan selanjutnya admin dari pihak fakultas membagikan jadwal perkuliahan kepada mahasiswa dan dosenpengajar.

Setelah dilakukan analisis sistem yang berlaku, secara garis besar ditarik Kesimpulan bahwa secara fungsional sistem yang berjalan dapat memenuhi tujuan meskipun masih terdapat beberapa kekurangan yang harus diperbaiki dan ditingkatkan lagi. Dari deskripsi sistem yang berjalaniatas, terlihat proses pengolahan data yang dilakukan lamanya penyusunan jadwal, sehingga lambatnya proses jadwal matakuliah.

2. Analisis sistem yang diusulkan

a. Use case Diagram aplikasi



Gambar 4.2 Use case diagram aplikasi

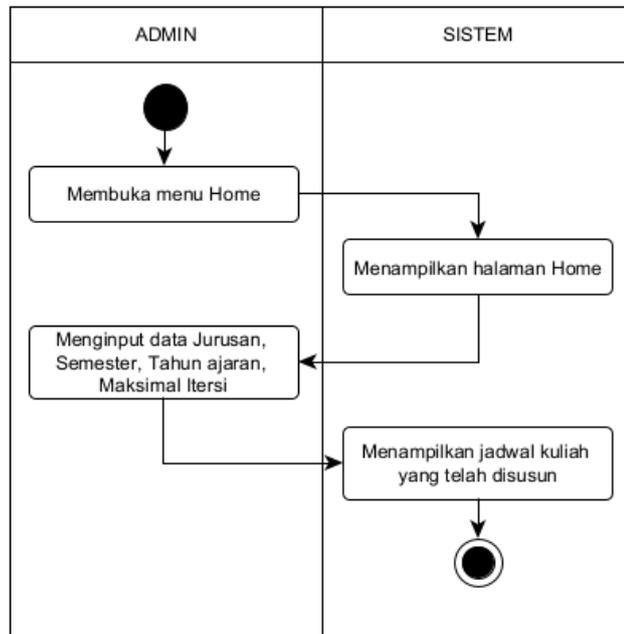
Penjelasan *Use case Diagram* Aplikasi:

Tabel 4.1 *Use Case Diagram*

Nama Use case	Deskripsi Use case
Data Dosen	<i>Use case</i> ini menggambarkan proses untuk menambahkan data dosen
Data Fakultas	<i>Use case</i> ini menggambarkan proses untuk menambahkan data fakultas
Data Prodi	<i>Use case</i> ini menggambarkan proses untuk menambahkan data prodi
Data Mata Kuliah	<i>Use case</i> ini menggambarkan proses untuk menambahkan data matakuliah
Data Kelas	<i>Use case</i> ini menggambarkan proses untuk menambahkan data kelas
Data Ruangan	<i>Use case</i> ini menggambarkan proses untuk menambahkan data Ruangan
Data Jadwal	<i>Use case</i> ini menggambarkan proses untuk menambahkan data Jadwal
Data Pengampu	<i>Use case</i> ini menggambarkan proses untuk menambahkan data Pengampu
Pengaturan	<i>Use case</i> ini digunakan untuk mengubah pengaturan

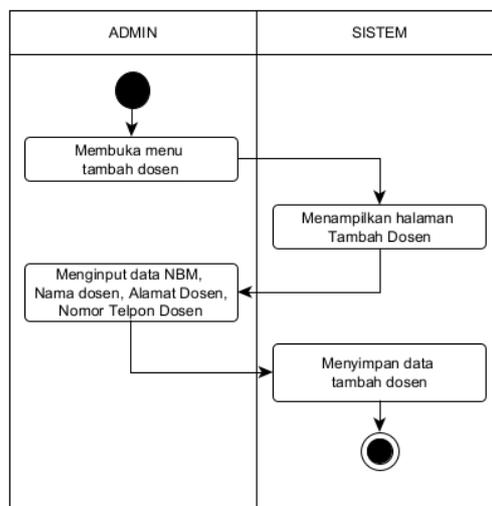
3. Activity Diagram

b. Activity Diagram Halaman Home



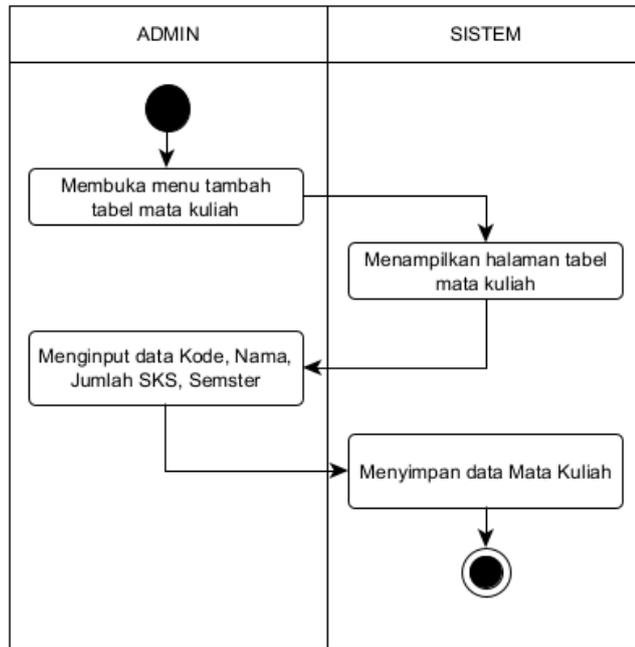
Gambar 4.3Activity Diagram Home

c. Activity Diagram Halaman Dosen



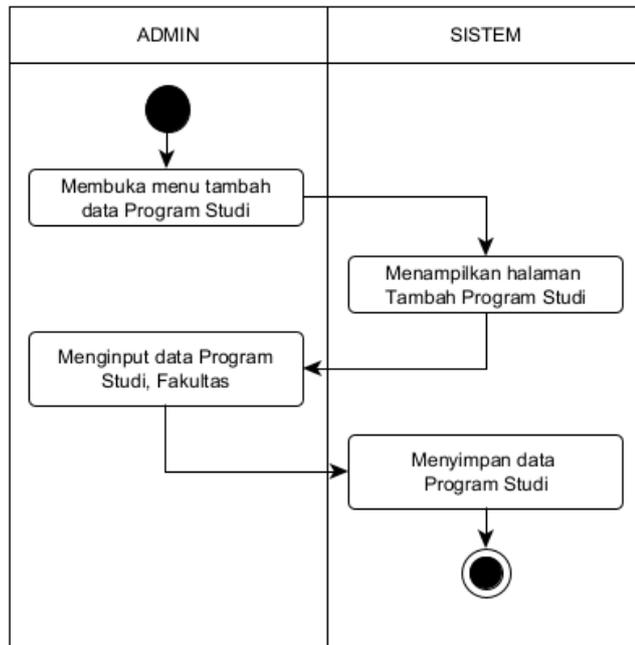
Gambar 4.4Activity DiagramDosen

d. Activity Diagram HalamanMatakuliah



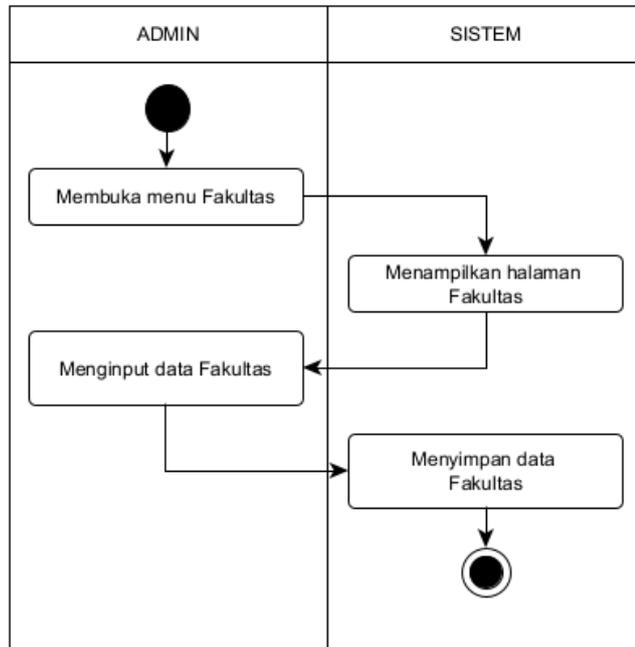
Gambar 4.5 Activity Diagram Matakuliah

e. Activity Diagram Halaman Program Studi



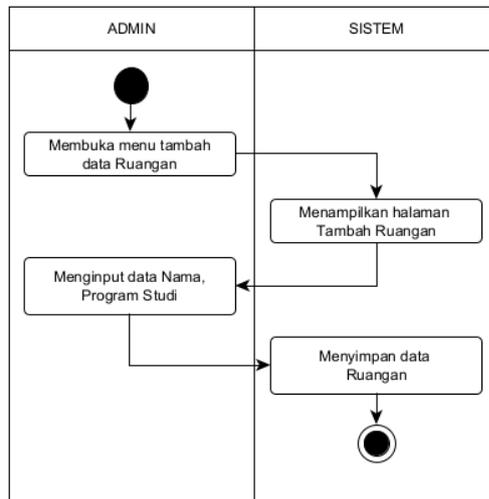
Gambar 4.6 Activity Diagram Program Studi

f. Activity Diagram Halaman Fakultas



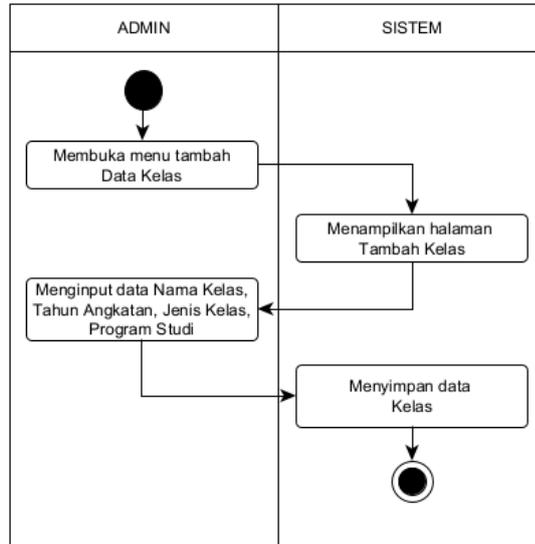
Gambar 4.7 *Activity Diagram* Fakultas

g. *Activity Diagram* Halaman Ruangan



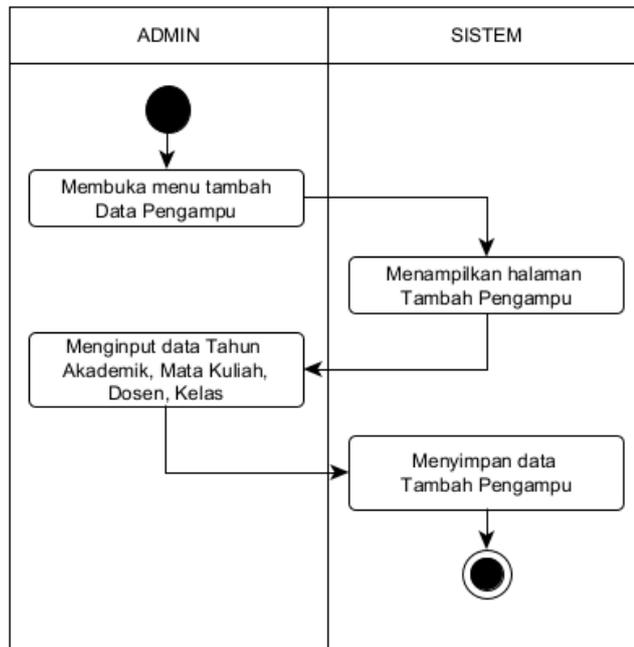
Gambar 4.8 *Activity Diagram* Ruangan

h. *Activity Diagram* Halaman Kelas



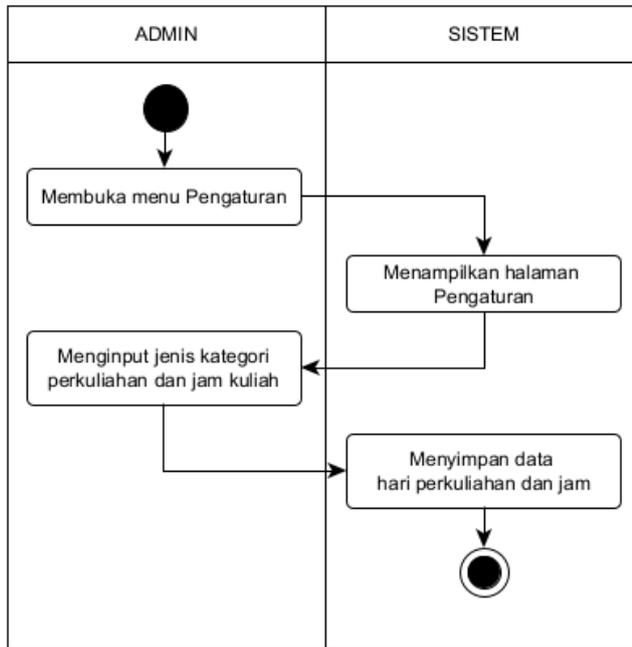
Gambar 4.9 *Activity Diagram* Kelas

i. *Activity Diagram* Halaman Pengampu



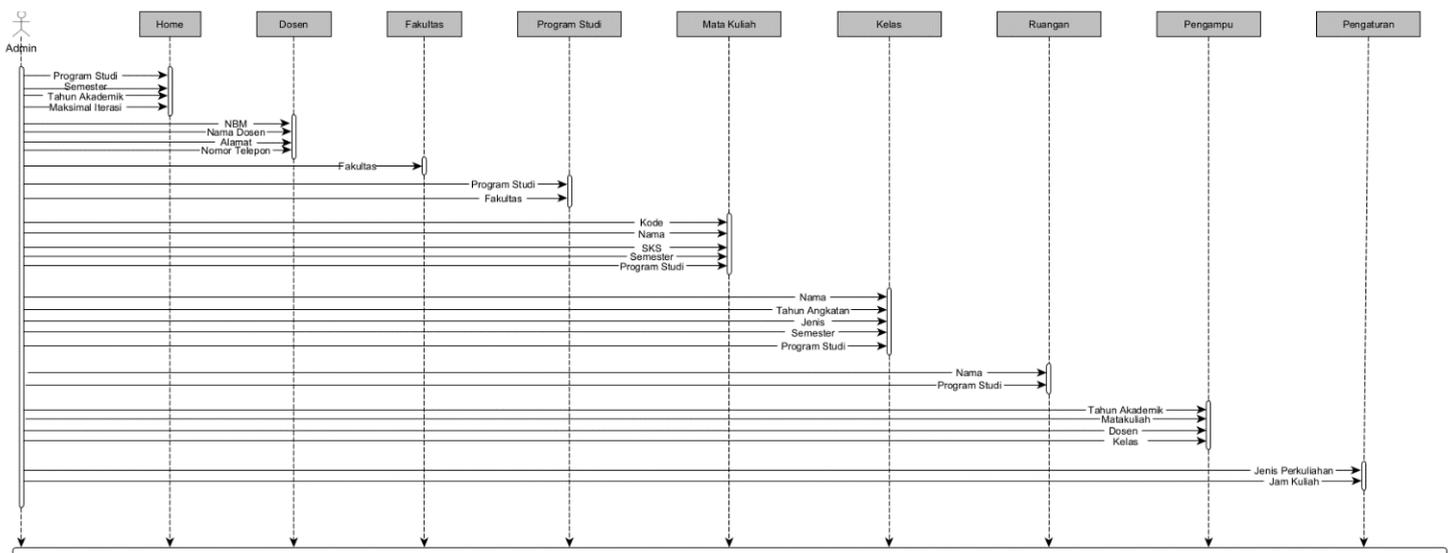
Gambar 4.10 *Activity Diagram* Pengampu

j. *Activity Diagram* Halaman Pengaturan



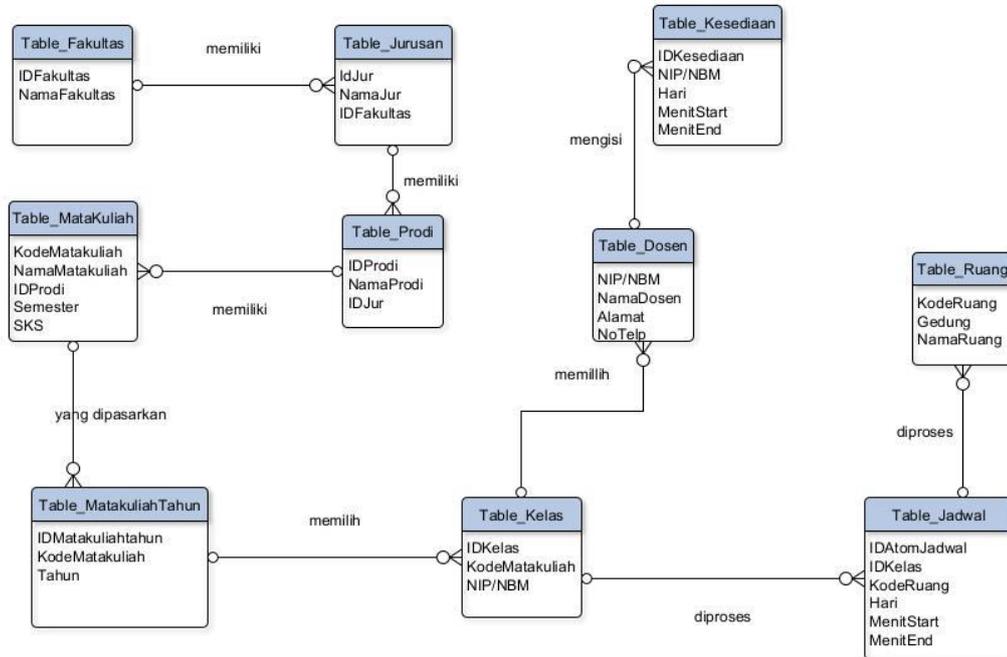
Gambar 4.11 Activity Diagram Pengaturan

4. Sequence diagram



Gambar 4.12 Sequence Diagram

5. Class diagram



Gambar 4.13 Class Diagram

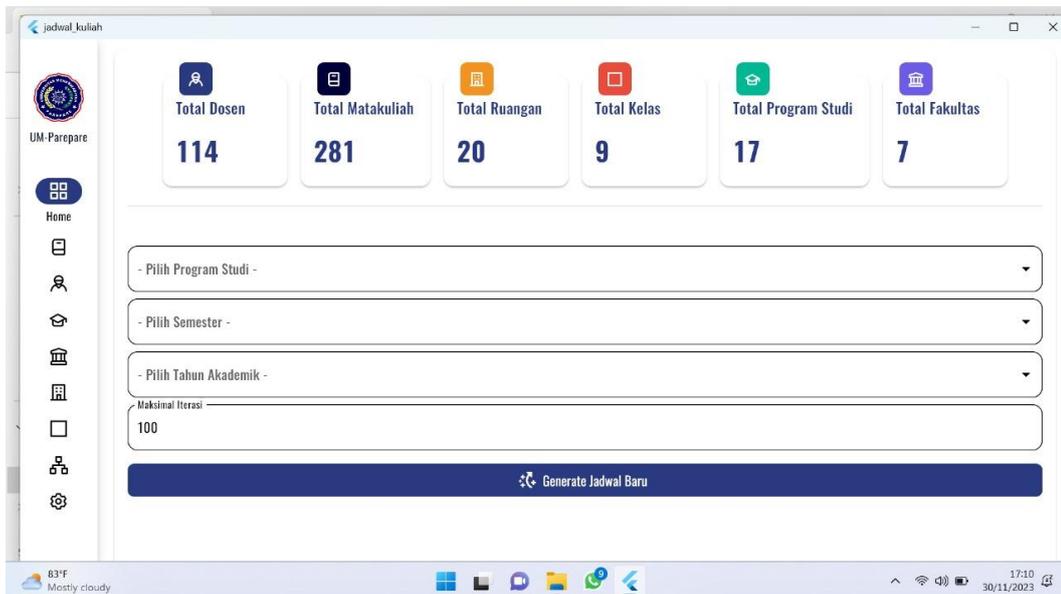
B. Pembahasan

Pada bagian ini akan dibahas mengenai hasil rancangan perangkat dan proses yang dilakukan untuk membuat aplikasi penjadwalan kuliah.

1. Detail Aplikasi

a. Halaman Home Aplikasi

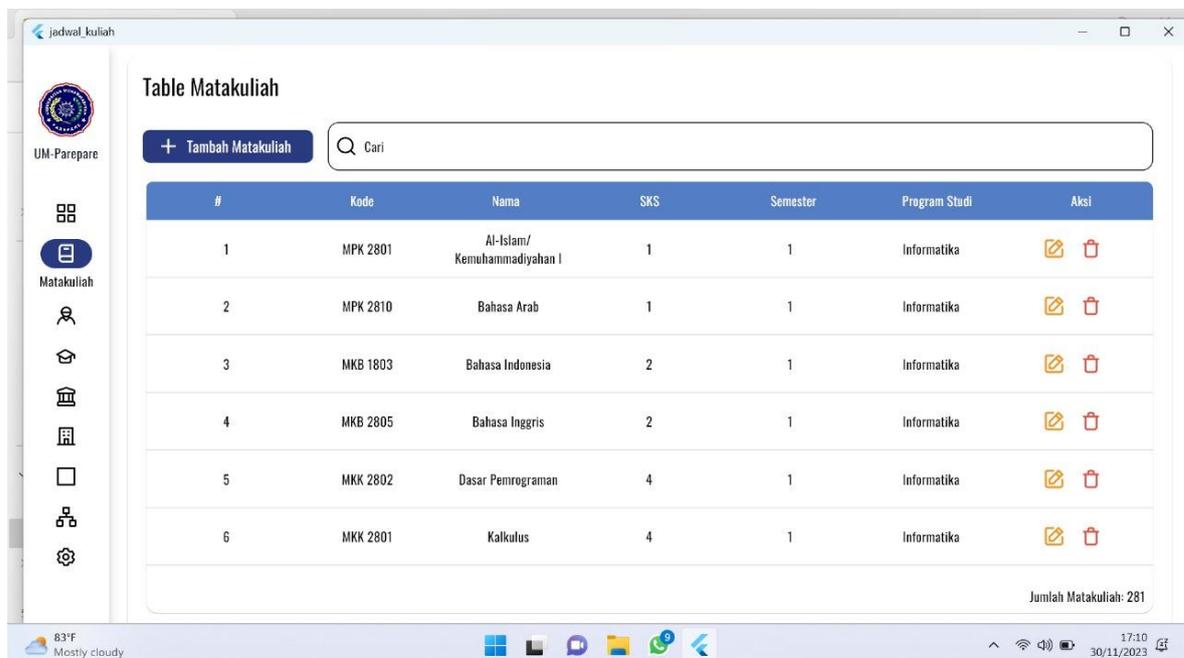
Pada halaman *Home* aplikasi merupakan halaman untuk menambahkan data program studi, semester, tahun akademik dan jumlah iterasi untuk menghasilkan jadwalkuliah.



Gambar 4.14 Halaman Homeaplikasi

b. Halaman Matakuliah

Pada halaman Matakuliah merupakan halaman yang menampilkan data matakuliah dan dapat menambahkan data baru terdiri dari kode, Nama, SKS, semester dan program studi.



Gambar 4.15 Halaman Matakuliah

c. Halaman Dosen

Pada halaman Dosen merupakan halaman yang menampilkan data dosen dan halaman untuk menambahkan dosen dengan memasukkan data NBM, Nama, alamat, dan nomor telepon.

The screenshot shows a web application interface for managing lecturers. The main content area is titled "Table Dosen" and features a search bar and a "+ Tambah Dosen" button. Below these is a table with the following data:

#	NBM	Nama	Alamat	Nomor Telepon	Aksi
1	1	Dosen 1	-	-	[Edit] [Delete]
2	2	Dosen 2	-	-	[Edit] [Delete]
3	3	Dosen 3	-	-	[Edit] [Delete]
4	4	Dosen 4	-	-	[Edit] [Delete]
5	5	Dosen 5	-	-	[Edit] [Delete]
6	6	Dosen 6	-	-	[Edit] [Delete]

At the bottom right of the table, it says "Jumlah Dosen: 114". The interface also includes a sidebar with navigation icons and a Windows taskbar at the bottom.

Gambar 4.16 Halaman Dosen

d. Halaman Program Studi

Pada halaman Program Studi merupakan halaman yang menampilkan data Program Studi dan halaman untuk menambahkan Program Studi dengan memasukkan data program studi dan fakultas.

#	Program Studi	Fakultas	Aksi
1	Informatika	Teknik	
2	Sipil	Teknik	
3	Elektro	Teknik	
4	Perencanaan Wilayah dan Kota	Teknik	
5	Peternakan	Peternakan dan Perikanan	
6	Budaya Perairan	Peternakan dan Perikanan	

Jumlah Program Studi: 17

Gambar 4.17 Program Studi

e. Halaman Fakultas

Pada halaman Fakultas merupakan halaman yang menampilkan data fakultas dan halaman untuk menambahkan fakultas dengan memasukkan data fakultas.

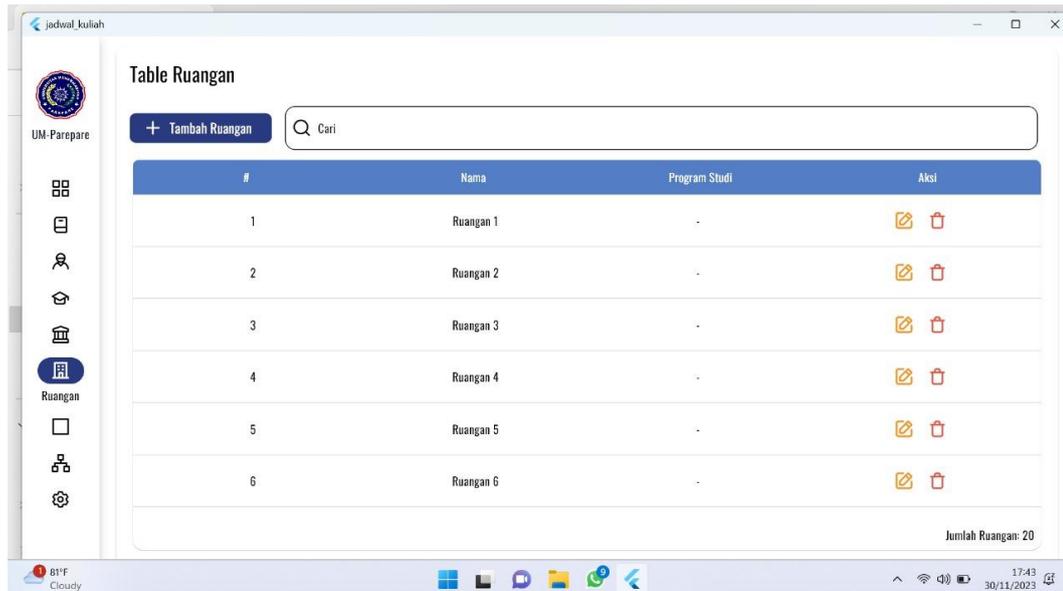
#	Fakultas	Aksi
1	Teknik	
2	Ekonomi	
3	Keguruan dan Ilmu Pendidikan	
4	Peternakan dan Perikanan	
5	Hukum	
6	Ilmu Kesehatan	

Jumlah Fakultas: 7

Gambar 4.18 Halaman Fakultas

f. Halaman Ruangan

Pada halaman Ruangan merupakan halaman yang menampilkan data Ruangan dan halaman untuk menambahkan data ruangan dengan memasukkan data nama ruangan, program studi.



Gambar 4.19 Halaman Ruangan

g. Halaman Kelas

Pada halaman Kelas merupakan halaman yang menampilkan data kelas dan halaman untuk menambahkan data kelas dengan memasukkan data Nama kelas, tahun angkatan, semester, jenis kuliah, program studi.

UM-Parepare

+ Tambah Kelas

Q Cari

#	Nama	Tahun Angkatan	Semester	Jenis	Program Studi	Aksi
1	A, B, C, D, E, F	2023	1	Reguler	Pendidikan Agama Islam	
2	A, B, C, D, E	2023	1	Reguler	Informatika	
3	F	2023	1	Non Reguler	Informatika	
4	A, B, C, D, E	2022	3	Reguler	Informatika	
5	F	2022	3	Non Reguler	Informatika	
6	A, B, C, D, E	2021	5	Reguler	Informatika	

Jumlah Kelas: 9

81°F Cloudy 17:43 30/11/2023

Gambar 4.20 Halaman Kelas

h. Halaman Pengampu

Pada halaman Pengampu merupakan halaman yang menampilkan data pengampu dan halaman untuk menambahkan data pengampu dengan memasukkan data tahun akademik, matakuliah, dosen, kelas.

UM-Parepare

+ Tambah Pengampu

Q Cari

#	Tahun Akademik	Matakuliah	Dosen	Kelas	Aksi
1	2023 - 2024	Cloud Computing	Dosen 99	A, B, C, D, E, F	
2	2023 - 2024	Pengkodean dan Kompresi Data	Dosen 28	A, B, C, D, E, F	
3	2023 - 2024	AI-Islam/Kemuhammadiyah III	Dosen 37	A, B, C, D, E, F	
4	2023 - 2024	Bahasa Inggris	Dosen 18	A, B, C, D, E, F	
5	2023 - 2024	Bahasa Indonesia	Dosen 97	A, B, C, D, E, F	
6	2023 - 2024	Teknik Pengembangan Game	Dosen 81	A, B, C, D, E, F	

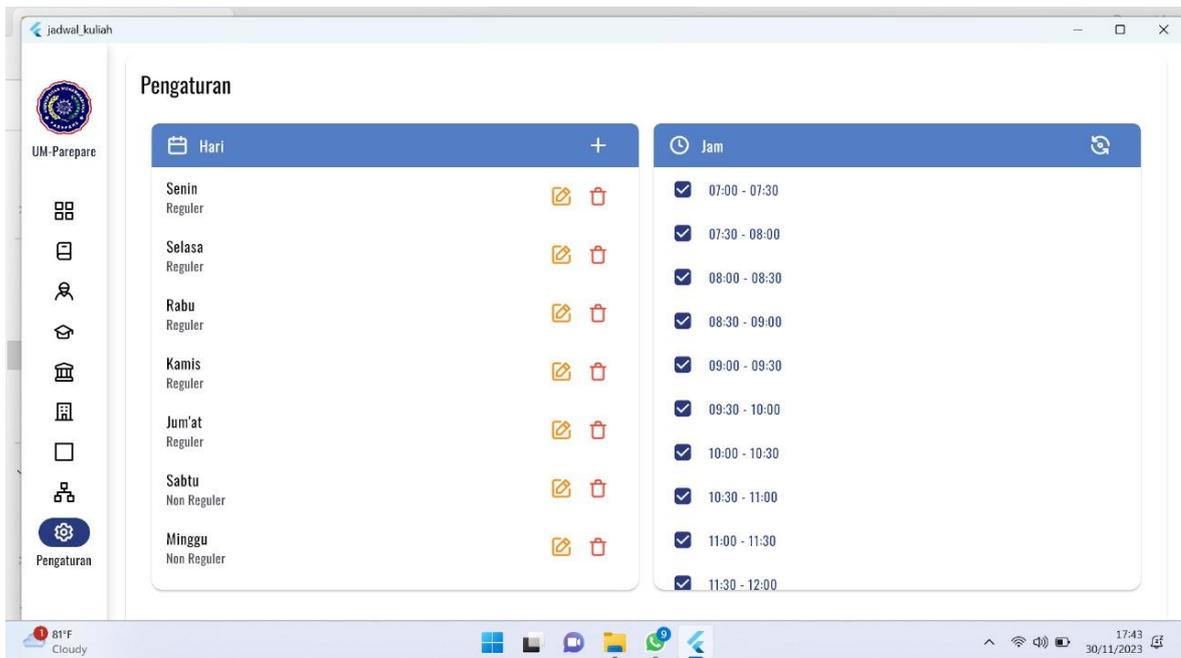
Jumlah Pengampu: 32

81°F Cloudy 17:43 30/11/2023

Gambar 4.21 Halaman Pengampu

i. Halaman Pengaturan

Pada halaman Pengaturan merupakan halaman untuk menampilkan data pengaturan dan halaman untuk mengubah pengaturan.



Gambar 4.22 Halaman Pengaturan

2. Pengujian Sistem

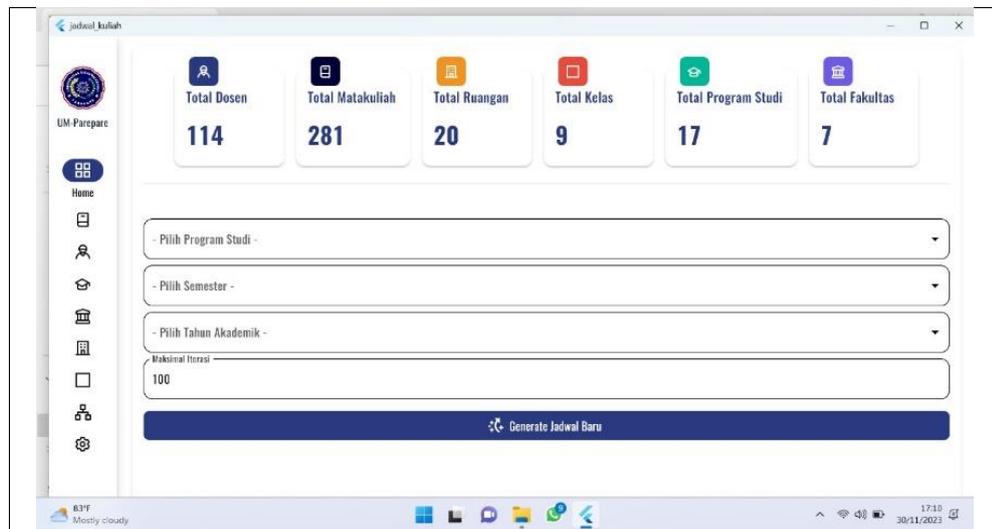
Pengujian aplikasi dilakukan dengan menggunakan dua metode pengujian yaitu pengujian *Black Box* dan pengujian *White Box*.

a) *Black Box*

1) *Black Box Home*

Tabel 4.2 Pengujian *Black Box Home*

<i>Test Faktor</i>	<i>Hasil</i>	<i>Kesimpulan</i>
Tampilan halaman <i>Home</i> saat proses pembuatan jadwal kuliah	✓	Berhasil membuat jadwal kuliah
<i>Screen Shot</i>		

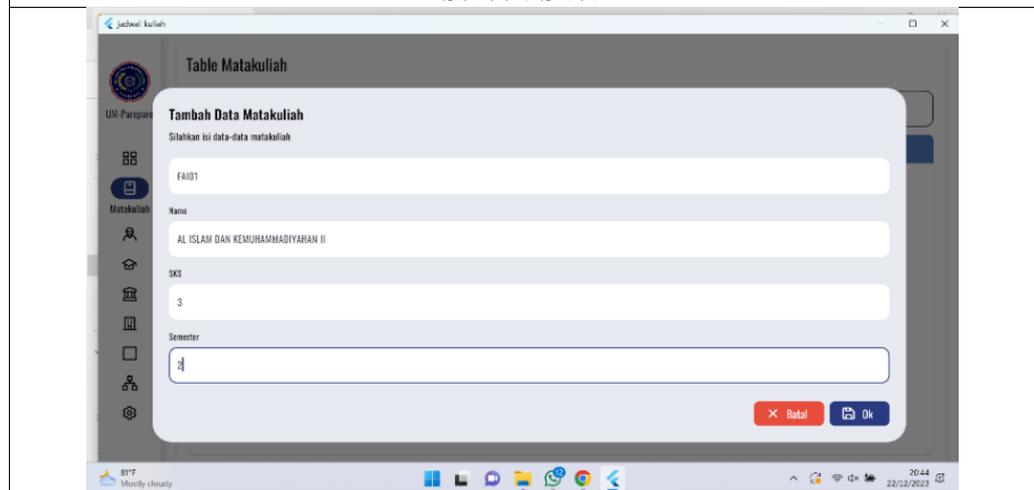


2) *Black Box* Matakuliah

Tabel 4.3 Pengujian *Black Box* Matakuliah

Test Faktor	Hasil	Kesimpulan
Tampilan halaman Matakuliah saat proses <i>Admin</i> menambahkan Matakuliah	✓	Berhasil menambahkan Matakuliah

Screen Shot



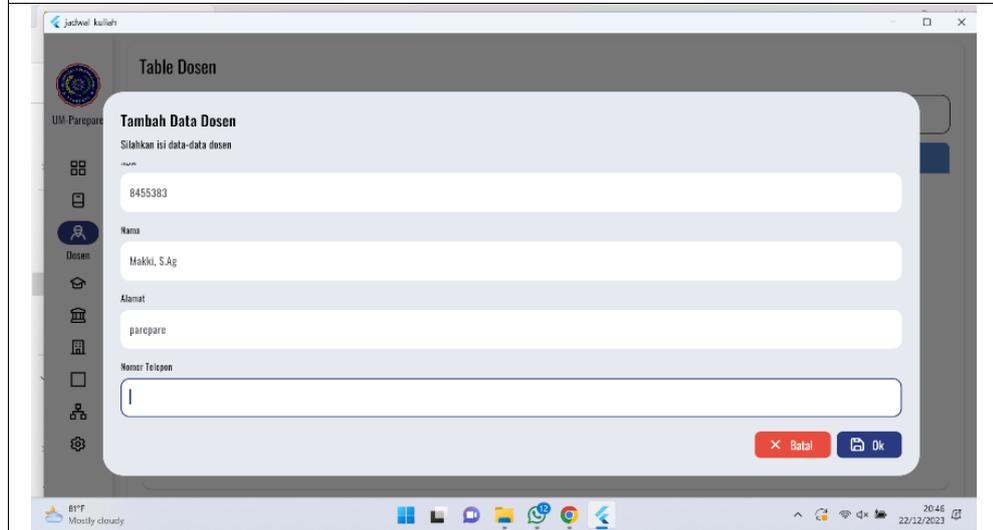
3) *Black Box* Dosen

Tabel 4.4 Pengujian *Black Box* Dosen

Test Faktor	Hasil	Kesimpulan
--------------------	--------------	-------------------

Tampilan halaman Dosen saat proses Admin menambahkan data Dosen	✓	Berhasil menambahkan data Dosen
---	---	---------------------------------

Screen Shot

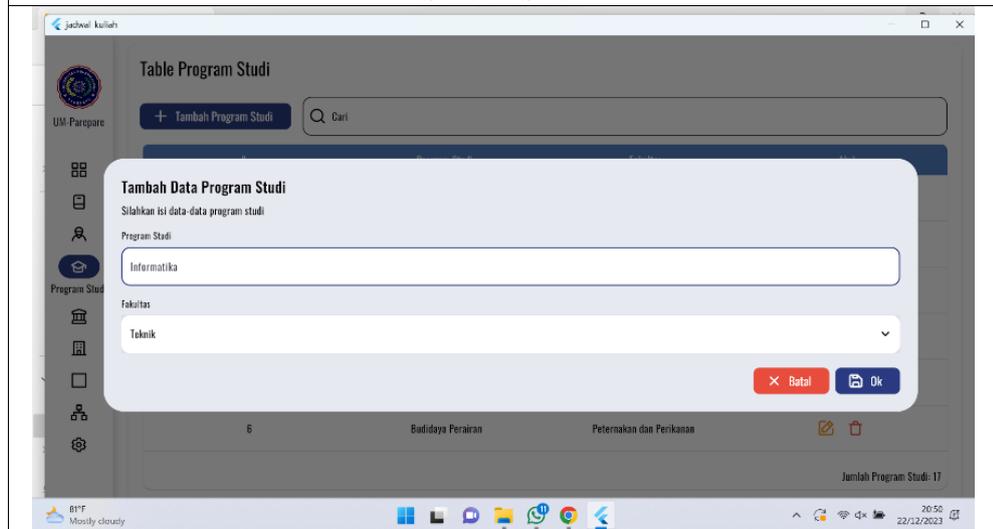


4) *Black Box* Program Studi

Tabel 4.5 Pengujian *Black Box* program studi

Test Faktor	Hasil	Kesimpulan
Tampilanhalaman Program Studi saat proses Admin menambahkan data Program Studi	✓	Berhasilmenambahkan data Program Studi

Screen Shot

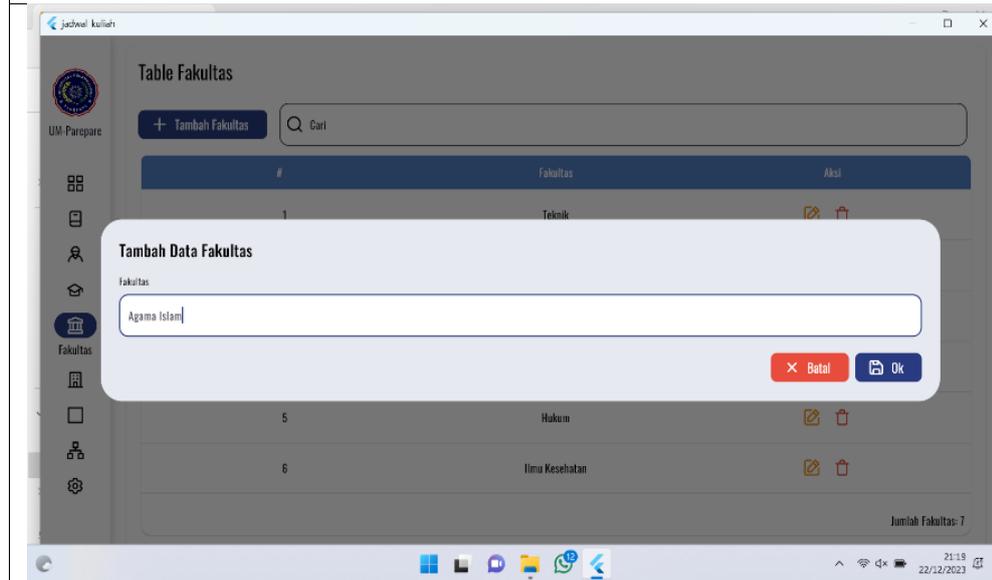


5) *Black Box* Fakultas

Tabel 4.6 Pengujian *Black Box* fakultas

Test Faktor	Hasil	Kesimpulan
Tampilan halaman Fakultas saat proses <i>Admin</i> menambahkan data Fakultas	✓	Berhasil menambahkan Fakultas

Screen Shot

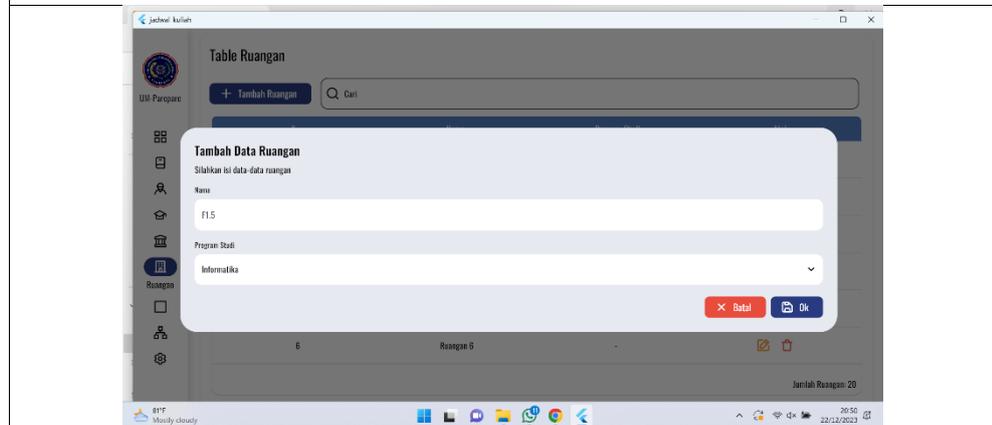


6) *Black Box* Ruangan

Tabel 4.7 Pengujian *Black Box* Ruangan

Test Faktor	Hasil	Kesimpulan
Tampilan halaman Ruangan saat proses <i>Admin</i> menambahkan data Ruangan	✓	Berhasil menambahkan Data ruangan

Screen Shot



7) *Black Box* Kelas

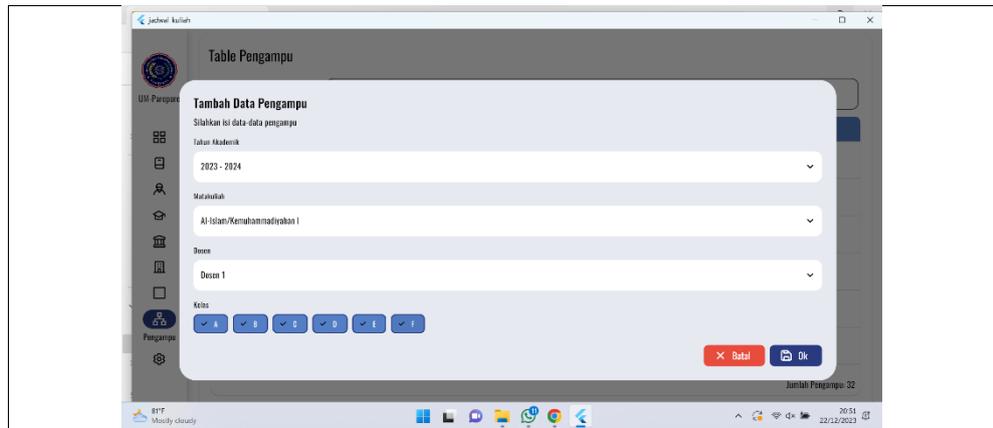
Tabel 4.8 Pengujian *Black Box* Kelas

Test Faktor	Hasil	Kesimpulan
Tampilan halaman Kelas saat proses Admin menambahkan data Kelas	✓	Berhasil menambahkan data Kelas
Screen Shot		
		

8) *Black Box* Pengampu

Tabel 4.9 Pengujian *Black Box* pengampu

Test Faktor	Hasil	Kesimpulan
Tampilan halaman Pengampu saat proses Admin menambahkan data Pengampu	✓	Berhasil menambahkan data Pengampu
Screen Shot		

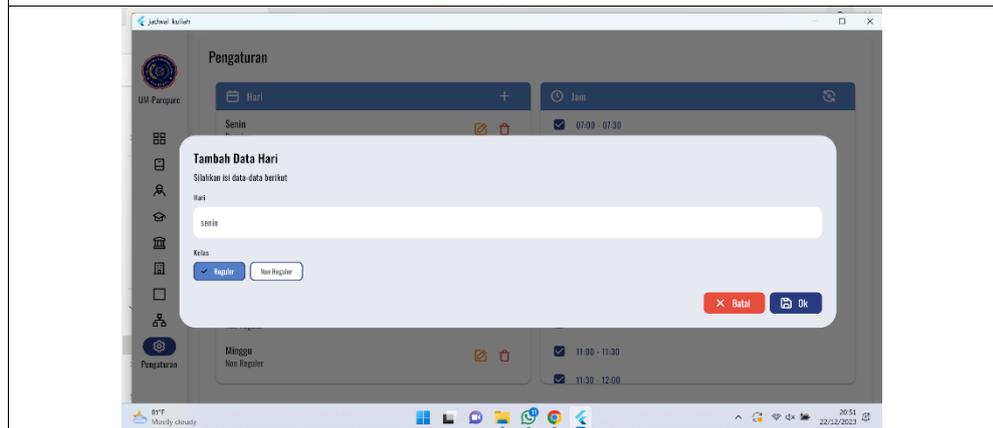


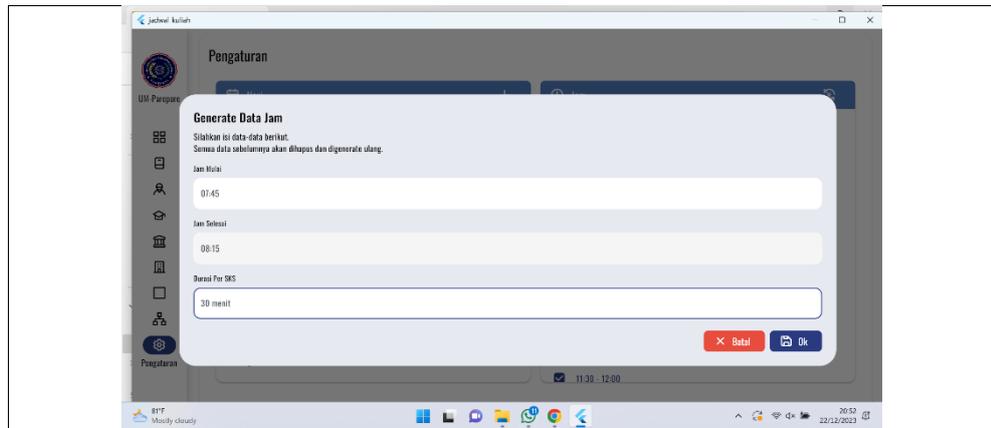
9) *Black Box* Pengaturan

Tabel 4.10 Pengujian *Black Box* Pengaturan

Test Faktor	Hasil	Kesimpulan
Tampilan halaman Pengaturan saat proses <i>Admin</i> menambahkan dan mengubah pengaturan hari dan pengaturan jam	✓	Berhasil menambahkan dan mengubah data Pengaturan

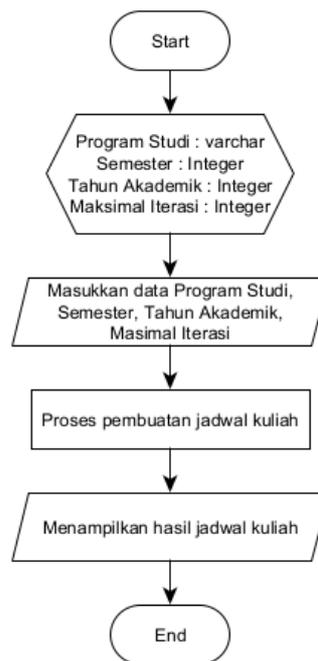
Screen Shot





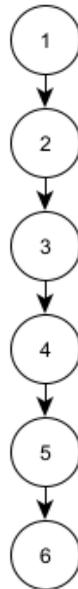
a. *White Box*

2) *Flowchart dan Flowgraph Home Aplikasi*



Gambar 4.23 *Flowchart Home Aplikasi*

Dari *flowchart* menu *Home* Aplikasi yang dilakukan untuk pengujian *software*, maka ditentukan *flowgraph* sebagai berikut:



Gambar 4.24 *Flowgraph HomeAplikasi*

Dari *flowgraph* menu *Home Aplikasi* yang diatas dapat dilakukan proses perhitungan sebagaiberikut:

- a) Menghitung *cyclomatic complexity* $V(G)$ asal *Edge* dan *Node*:

Menggunakan rumus: $V(G) = E - N + 2$

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

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

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

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

$= 5 - 6 + 2$

$= 1$

Predikat (P) $= P + 1$

$= 1 + 1$

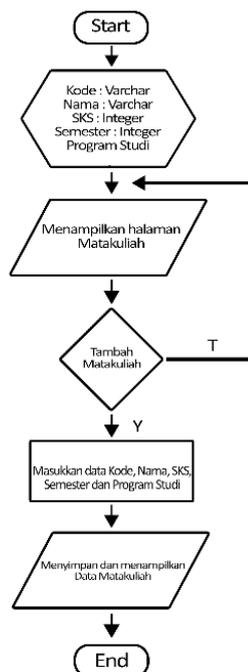
$= 2$

- b) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas mempunyai *Region*= 2
- c) *Independent Path* pada *flowgraph* diatasialah:
Path 1 = 1 - 2 - 3 - 4 - 5 - 6
- d) *GrafikMatriksHomeAplikasi*

Tabel 4.11 *Matriks HomeAplikasi*

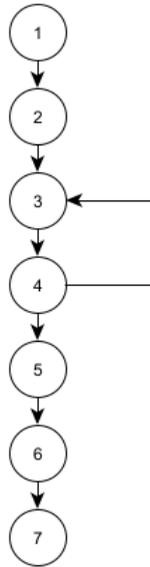
	1	2	3	4	5	6	E - 1
1		1					1 - 1 = 0
2			1				1 - 1 = 0
3				1			1 - 1 = 0
4					1		1 - 1 = 0
5						1	1 - 1 = 0
6							0
Sum = (E + 1)							0 + 1 = 1

3) *Flowchart* dan *flowgraph* menu Matakuliah



Gambar 4.25 *Flowchart* Matakuliah

Dari *flowchart* Matakuliah Aplikasi yang dilakukan untuk pengujian *software*, maka ditentukan *flowgraph* sebagai berikut:



Gambar 4.26 *Flowgraph* Matakuliah

Dari *flowgraph* menu Matakuliah Aplikasi yang diatas dapat dilakukan proses perhitungan sebagai berikut:

- a) Menghitung *cyclomatic complexity* $V(G)$ asal *Edge* dan *Node*:

Menggunakan rumus: $V(G) = E - N + 2$

E (*edge*) = 7

N (*Node*) = 7

P (*Predikat Node*) = 1

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

$= 7 - 7 + 2$

$= 2$

Predikat (P) = $P + 1$

$= 1 + 1$

$$= 2$$

b) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas mempunyai *Region*= 2

c) *Independent Path* pada *flograph* diatasialah:

$$Path 1 = 1 - 2 - 3 - 4 - 5 - 6 - 7$$

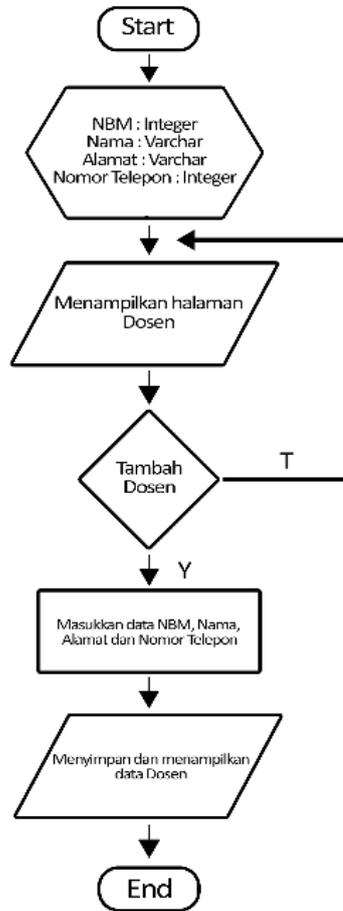
$$Path 2 = 1 - 2 - 3 - 4 - 3 - 4 - 5 - 6 - 7$$

d) Grafik Matriks Matakuliah Aplikasi

Tabel 4.12 Matriks Matakuliah

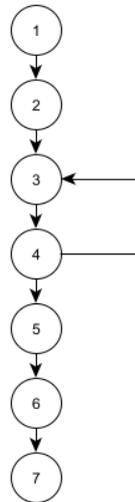
	1	2	3	4	5	6	7	E - 1
1		1						1 - 1 = 0
2			1					1 - 1 = 0
3				1				1 - 1 = 0
4			1		1			2 - 1 = 0
5						1		1 - 1 = 0
6							1	1 - 1 = 0
7								
Sum = (E + 1)								0 + 1 = 1

4) *Flowchart* dan *flowgraph* menu Dosen



Gambar 4.27 Flowchart Dosen

Dari *flowgraph* Dosen yang dilakukan untuk pengujian *software*, maka ditentukan *flowgraph* sebagai berikut:



Gambar 4.28 *Flowgraph* Dosen

Dari *flowgraph* menu Dosen yang di atas dapat dilakukan proses perhitungan sebagai berikut:

- a) Menghitung *cyclomatic complexity* $V(G)$ asal *Edge* dan *Node*:

Menggunakan rumus: $V(G) = E - N + 2$

$E(\text{edge}) = 7$

$N(\text{Node}) = 7$

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

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

$= 7 - 7 + 2$

$= 2$

Predikat (P) $= P + 1$

$= 1 + 1$

$= 2$

- b) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* di atas mempunyai *Region* = 2

c) *Independent Path* pada *flowgraph* diatasialah:

Path 1 = 1 - 2 - 3 - 4 - 5 - 6 - 7

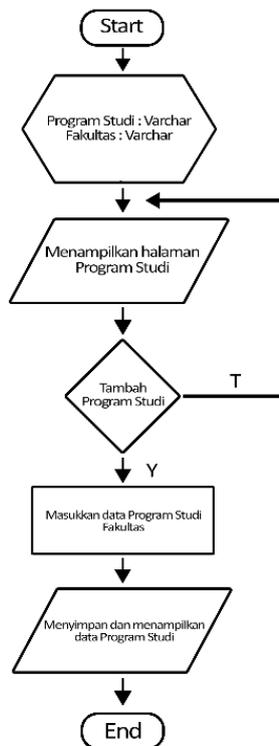
Path 2 = 1 - 2 - 3 - 4 - 3 - 4 - 5 - 6 - 7

d) Grafik Matriks Dosen Aplikasi

Tabel 4.13 Matriks Dosen

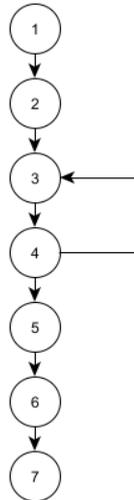
	1	2	3	4	5	6	7	E - 1
1		1						1 - 1 = 0
2			1					1 - 1 = 0
3				1				1 - 1 = 0
4			1		1			2 - 1 = 0
5						1		1 - 1 = 0
6							1	1 - 1 = 0
7								
Sum = (E + 1)								0 + 1 = 1

5) *Flowchart* dan *flowgraph* menu Program Studi



Gambar 4.29 *Flowchart* Program Studi

Dari *flowgraph* Program Studi yang dilakukan untuk pengujian *software*, maka ditentukan *flowgraph* sebagai berikut:



Gambar 4.30 *Flowgraph* Program Studi

Dari *flowgraph* menu Program Studi yang di atas dapat dilakukan proses perhitungan sebagai berikut:

- a) Menghitung *cyclomatic complexity* $V(G)$ asal *Edge* dan *Node*:

Menggunakan rumus: $V(G) = E - N + 2$

$$E(\text{edge}) = 7$$

$$N(\text{Node}) = 7$$

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

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

$$= 7 - 7 + 2$$

$$= 2$$

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

$$= 1 + 1$$

$$= 2$$

b) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas mempunyai *Region*= 2

c) *Independent Path* pada *flograph* diatasialah:

Path 1 = 1 - 2 - 3 - 4 - 5 - 6 - 7

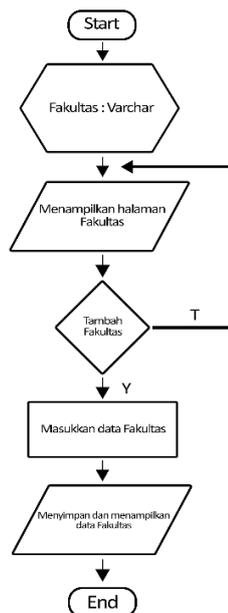
Path 2 = 1 - 2 - 3 - 4 - 3 - 4 - 5 - 6 - 7

d) Grafik Matriks Program Studi

Tabel 4.14 Matriks Program Studi

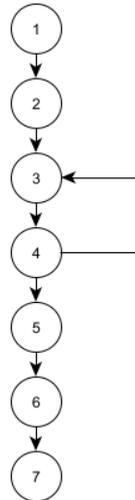
	1	2	3	4	5	6	7	E - 1
1		1						1 - 1 = 0
2			1					1 - 1 = 0
3				1				1 - 1 = 0
4			1		1			2 - 1 = 0
5						1		1 - 1 = 0
6							1	1 - 1 = 0
7								
Sum = (E + 1)								0 + 1 = 1

6) *Flowchart* dan *flowgraph* menu Fakultas



Gambar 4.31 *Flowchart* Fakultas

Dari *flowgraph* Fakultas yang dilakukan untuk pengujian *software*, maka ditentukan *flowgraph* sebagai berikut:



Gambar 4.32 *Flowgraph* Fakultas

Dari *flowgraph* menu Fakultas yang di atas dapat dilakukan proses perhitungan sebagai berikut:

- a) Menghitung *cyclomatic complexity* $V(G)$ asal *Edge* dan *Node*:

Menggunakan rumus: $V(G) = E - N + 2$

$$E(\text{edge}) = 7$$

$$N(\text{Node}) = 7$$

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

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

$$= 7 - 7 + 2$$

$$= 2$$

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

$$= 1 + 1$$

$$= 2$$

b) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas mempunyai *Region*= 2

c) *Independent Path* pada *flograph* diatasialah:

$$Path 1 = 1 - 2 - 3 - 4 - 5 - 6 - 7$$

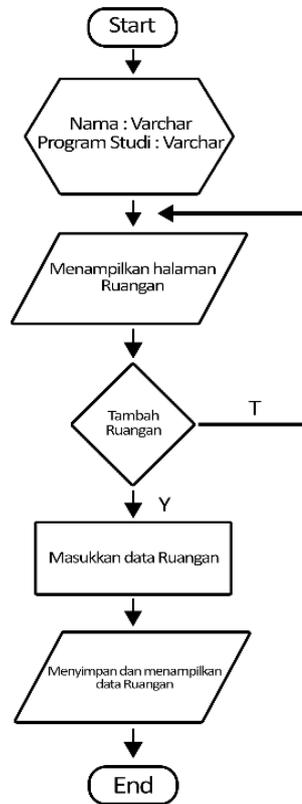
$$Path 2 = 1 - 2 - 3 - 4 - 3 - 4 - 5 - 6 - 7$$

d) Grafik Matriks Fakultas Aplikasi

Tabel 4.15 Matriks Fakultas

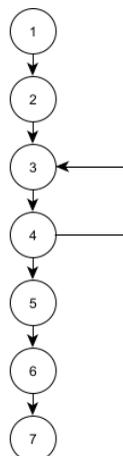
	1	2	3	4	5	6	7	E - 1
1		1						1 - 1 = 0
2			1					1 - 1 = 0
3				1				1 - 1 = 0
4			1		1			2 - 1 = 0
5						1		1 - 1 = 0
6							1	1 - 1 = 0
7								
Sum = (E + 1)								0 + 1 = 1

7) *Flowchart* dan *flowgraph* menu Ruangan



Gambar 4.33*Flowchart*Ruangan

Dari *flowgraph* Ruangan yang dilakukan untuk pengujian *software*, maka ditentukan *flowgraph* sebagai berikut:



Gambar 4.34*Flowgraph*Ruangan

Dari *flowgraph* menu Ruangan Aplikasi yang di atas dapat dilakukan proses perhitungan sebagai berikut:

- a) Menghitung *cyclomatic complexity* $V(G)$ asal *Edge* dan *Node*:

Menggunakan rumus: $V(G) = E - N + 2$

$E(\text{edge}) = 7$

$N(\text{Node}) = 7$

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

Penyelesaian: $V(G) = E - N + 2$
 $= 7 - 7 + 2$
 $= 2$

Predikat (P) $= P + 1$
 $= 1 + 1$
 $= 2$

- b) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* di atas mempunyai *Region* = 2

- c) *Independent Path* pada *flowgraph* di atas ialah:

Path 1 $= 1 - 2 - 3 - 4 - 5 - 6 - 7$

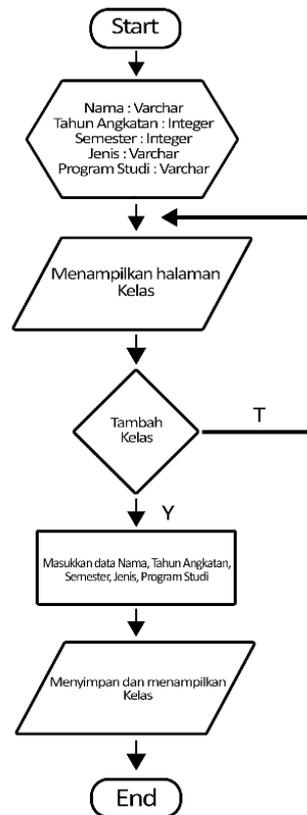
Path 2 $= 1 - 2 - 3 - 4 - 3 - 4 - 5 - 6 - 7$

- d) *Grafik Matriks* Ruangan Aplikasi

Tabel 4.16 Matriks Ruangan

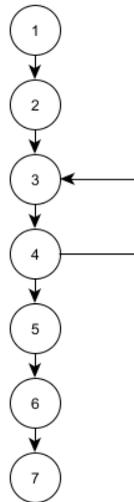
	1	2	3	4	5	6	7	E - 1
1		1						1 - 1 = 0
2			1					1 - 1 = 0
3				1				1 - 1 = 0
4			1		1			2 - 1 = 0
5						1		1 - 1 = 0
6							1	1 - 1 = 0
7								
Sum = (E + 1)								0 + 1 = 1

8) *Flowchart* dan *flowgraph* menu Kelas



Gambar 4.35 *Flowchart* Kelas

Dari *flowgraph* Kelas yang dilakukan untuk pengujian *software*, maka ditentukan *flowgraph* sebagai berikut:



Gambar 4.36 *Flowgraph* Kelas

Dari *flowgraph* menu Kelas yang di atas dapat dilakukan proses perhitungan sebagai berikut:

- a) Menghitung *cyclomatic complexity* $V(G)$ asal *Edge* dan *Node*:

Menggunakan rumus: $V(G) = E - N + 2$

$E(\text{edge}) = 7$

$N(\text{Node}) = 7$

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

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

$$= 7 - 7 + 2$$

$$= 2$$

Predikat (P) = $P + 1$

$$= 1 + 1$$

$$= 2$$

- b) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* di atas mempunyai *Region* = 2

c) *Independent Path* pada *flowgraph* diatasialah:

Path 1 = 1 - 2 - 3 - 4 - 5 - 6 - 7

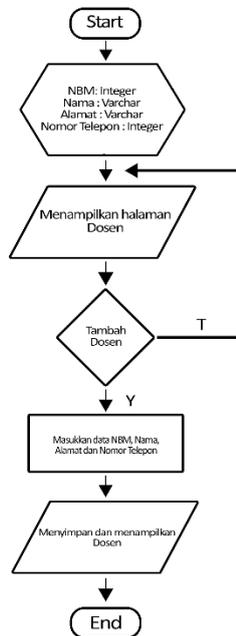
Path 2 = 1 - 2 - 3 - 4 - 3 - 4 - 5 - 6 - 7

d) *Grafik Matriks Kelas Aplikasi*

Tabel 4.17 *Matriks Kelas*

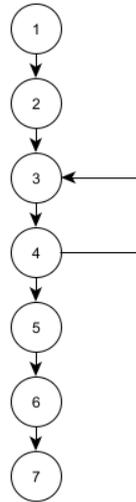
	1	2	3	4	5	6	7	E - 1
1		1						1 - 1 = 0
2			1					1 - 1 = 0
3				1				1 - 1 = 0
4			1		1			2 - 1 = 0
5						1		1 - 1 = 0
6							1	1 - 1 = 0
7								
Sum = (E + 1)								0 + 1 = 1

9) *Flowchart* dan *flowgraph* menu Pengampu



Gambar 4.37 *Flowchart* Pengampu

Dari *flowgraph* Pengampu yang dilakukan untuk pengujian *software*, maka ditentukan *flowgraph* sebagai berikut:



Gambar 4.38 *Flowgraph* Pengampu

Dari *flowgraph* menu Pengampuyang diatas dapat dilakukan proses perhitungan sebagai berikut:

- a) Menghitung *cyclomatic complexity* $V(G)$ asal *Edge* dan *Node*:

Menggunakan rumus: $V(G) = E - N + 2$

$$E(\text{edge}) = 7$$

$$N(\text{Node}) = 7$$

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

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

$$= 7 - 7 + 2$$

$$= 2$$

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

$$= 1 + 1$$

$$= 2$$

b) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas mempunyai *Region*= 2

c) *Independent Path* pada *flograph* diatasialah:

Path 1 = 1 - 2 - 3 - 4 - 5 - 6 - 7

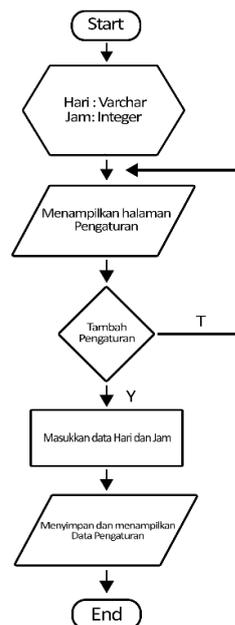
Path 2 = 1 - 2 - 3 - 4 - 3 - 4 - 5 - 6 - 7

d) *GrafikMatriks*PengampuAplikasi

Tabel 4.18*Matriks*Pengampu

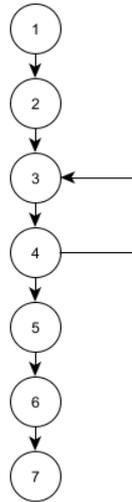
	1	2	3	4	5	6	7	E - 1
1		1						1 - 1 = 0
2			1					1 - 1 = 0
3				1				1 - 1 = 0
4			1		1			2 - 1 = 0
5						1		1 - 1 = 0
6							1	1 - 1 = 0
7								
Sum = (E + 1)								0 + 1 = 1

10) *Flowchart* dan *flowgraph* menu Pengaturan



Gambar 4.39*Flowchart*Pengaturan

Dari *flowgraph* Pengaturan yang dilakukan untuk pengujian *software*, maka ditentukan *flowgraph* sebagai berikut:



Gambar 4.40 *Flowgraph* Pengaturan

Dari *flowgraph* menu Pengaturan yang di atas dapat dilakukan proses perhitungan sebagai berikut:

- a) Menghitung *cyclomatic complexity* $V(G)$ asal *Edge* dan *Node*:

Menggunakan rumus: $V(G) = E - N + 2$

$$E(\text{edge}) = 7$$

$$N(\text{Node}) = 7$$

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

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

$$= 7 - 7 + 2$$

$$= 2$$

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

$$= 1 + 1$$

$$= 2$$

b) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph* diatas mempunyai *Region*= 2

c) *Independent Path* pada *flograph* diatasialah:

Path 1 = 1 - 2 - 3 - 4 - 5 - 6 - 7

Path 2 = 1 - 2 - 3 - 4 - 3 - 4 - 5 - 6 - 7

d) Grafik Matriks Pengaturan Aplikasi

Tabel 4.19 Matriks Pengaturan

	1	2	3	4	5	6	7	E - 1
1		1						1 - 1 = 0
2			1					1 - 1 = 0
3				1				1 - 1 = 0
4			1		1			2 - 1 = 0
5						1		1 - 1 = 0
6							1	1 - 1 = 0
7								
Sum = (E + 1)								0 + 1 = 1

PENJABARAN RUMUS

Dalam memilih rute perjalanan (titik yang akan dikunjungi selanjutnya), semut menggunakan persamaan peluang dimana :

P_{ij} = Peluang transisi dari titik i ke j pada saat t.

$\tau_{ij}(t)$ = intensitas jejak feromon pada garis (i,j) pada saat t.

$\eta_{ij}(t)$ = Visibilitas node i dan j pada saat t, $\eta_{ij} = \frac{1}{d(C_i, C_j)}$

α = parameter untuk mengendalikan tingkat kepentingan relatif dari jejak feromon

β = parameter untuk mengendalikan tingkat kepentingan relatif dari visibilitas.

Setelah didapatkan peluang perpindahan dari setiap titik, kemudian dihitung nilai kumulatif dari setiap titik yaitu dengan menjumlahkan peluang titik awal sampai titik yang dihitung.

Setelah semua semut melalui semua titik yang tersedia, proses berikutnya adalah melakukan perubahan nilai intensitas feromon antar titik. Hal ini dilakukan karena jumlah semut yang melewati setiap garis berbeda-beda, semakin banyak semut yang melewati suatu garis maka intensitas feromonnya akan semakin tinggi begitu juga sebaliknya. Berikut persamaan yang digunakan untuk menghitung nilai perubahan intensitas feromon antar titik :

$\mathcal{T}_{ij}(t)$ = Intensitas jejak feromon pada garis (i,j) pada saat t

$\mathcal{T}_{ij}(t+N)$ = Intensitas jejak feromon setelah semut menyelesaikan perjalanan dalam satu siklus

$$\Delta t_{ij}^k(t,t+N)$$

= intensitas jejak feromon yang ditinggalkan oleh semut k pada

garis (i,j) pada interval t dan t+1

L_k = panjang lintasan yang dilalui semut k

ρ = koefisien penguapan feromon

N = banyak titik

m = jumlah semut pada suatu titik

Q = konstanta jumlah feromon.

Untuk mendapatkan hasil penjadwalan kuliah yang optimal dengan menggunakan algoritma semut dibutuhkan parameterparameter antara lain:

- 2) Intensitas jejak semut antar titik (τ_{ij}) dan perubahannya ($\Delta\tau_{ij}$). τ_{ij} harus diinisialisai sebelum memulai siklus. τ_{ij} digunakan dalam persamaan probabilitas titik yang akan dikunjungi. $\Delta\tau_{ij}$ diinisialisasi setelah selesai satu siklus. $\Delta\tau_{ij}$ digunakan untuk menentukan τ_{ij} untuk siklus selanjutnya.
- 3) Tetapan siklus semut (Q). Q merupakan konstanta yang digunakan dalam persamaan untuk menentukan $\Delta\tau_{ij}$. Nilai Q ditentukan oleh pengguna.
- 4) Tetapan pengendali intensitas jejak semut (α). α digunakan dalam persamaan probabilitas titik yang akan dikunjungi yang berfungsi sebagai pengendali intensitas jejak semut. Nilai α ditentukan oleh pengguna.
- 5) Tetapan pengendali visibilitas (β). β digunakan dalam persamaan probabilitas titik yang akan dikunjungi dan berfungsi sebagai pengendali visibilitas. Nilai β ditentukan oleh pengguna.
- 6) Visibilitas antar titik (η_{ij}). η digunakan dalam persamaan probabilitas titik yang akan dikunjungi. Nilai η merupakan hasil dari $1/d_{ij}$ (jarak antar titik).
- 7) Banyak semut (m). m merupakan banyak semut yang akan melakukan siklus dalam algoritma semut. Nilai m ditentukan oleh pengguna.
- 8) Tetapan penguapan jejak semut (ρ). ρ digunakan untuk menentukan τ_{ij} untuk siklus selanjutnya. Nilai ρ ditentukan oleh pengguna.
- 9) Jumlah siklus maksimum (NC_{max}). NC_{max} adalah jumlah maksimum siklus yang akan berlangsung. Siklus akan berhenti sesuai dengan NC_{max} yang telah ditentukan atau telah konvergen. Nilai NC_{max} ditentukan oleh pengguna.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Aplikasi jadwal matakuliah Universitas Muhammadiyah Parepare dapat membantu proses jadwal kuliah lebih cepat. Hal ini dibuktikan dengan pengujian aplikasi, Dimana tidak terjadinya tumpang tindih jadwal matakuliah, cepatnya pengelolaan data dan cepatnya melakukan penyusunan jadwal.

B. Saran

Adapun beberapa saran yang di usulkannya yaitu:

1. Pengembangan aplikasi tersebut lebih di perluas, tidak hanya terbatas pada kegiatan pengolahan jadwal kuliah saja tetapi juga membahas pengolahan data yang lain.
2. Adanya pemeliharaan terhadap aplikasi yang telah dibuat agar tetap terjaga dengan dengan baik dengan cara melakukan perbaikan aplikasi program tersebut jika terdapat kesalahan
3. Aplikasi jadwal matakuliah dapat ditambahkan jadwal kuliah setiap mahasiswa

DAFTAR PUSTAKA

Aliklas, Q., Satyahadewi, N., & Perdana, H. (2019). Penerapan Algoritma Max-Min Ant System Dalam Penyusunan Jadwal Mata Kuliah Di Jurusan Matematika Fmipa Untan. *BIMASTER: Buletin Ilmiah Matematika, Statistika dan Terapannya*, 8(2).

- AnugrahA, D. (2011). OptimalisasiPenjadwalan Mata Kuliahmenggunakan Ant Colony Optimization (Doctoral dissertation, Universitas Brawijaya).
- Hartadi, R., & Hidayat, A. (2016). Perancangan Aplikasi Penjadwalan Mata Kuliah (Studi Kasus: STMIK Provisi Semarang). *Bianglala Informatika*, 4(1).
- Hasyim, A. H. (2021). *Dasar Pemrograman*. CV. Bangun Bumitama.
- Kho, Budi. 2018. *Pengertian Penjadwalan Menurut Para Ahli*. From <https://ilmumanajemenindustri.com/pengertian-penjadwalan-scheduling-dalam-proses-produksi.html> . (Diakses 21 Mei 2022).
- Noviasari, I., Rusli, A., & Hansun, S. (2018). Penerapan Algoritma ACO untuk Penjadwalan Kuliah Pengganti pada Perguruan Tinggi (Studi Kasus: Program Studi Informatika, Universitas Multimedia Nusantara). *Ultima InfoSys: Jurnal Ilmu Sistem Informasi*, 9(2), 79-85.
- Pamela Alfa Adelia Darmadji. “Pengembangan aplikasi sistem informasipenjadwalan perkuliahan elektronik berbasis web dengan sms gateway”.
- Sari, Y., Alkaff, M., Wijaya, E. S., Soraya, S., & Kartikasari, D. P. (2019). Optimasi Penjadwalan Mata Kuliah Menggunakan Metode Algoritma Genetika dengan Teknik Tournament Selection. *J. Teknol. Inf. dan Ilmu Komput*, 6(1), 85.
- Siddiq, A. (2022). Analisis Penjadwalan Menggunakan Aplikasi Microsoft Project 2010 (Studi Kasus: Ruang Terbuka Hijau (Rth) Kabupaten Wajo) (Vol. 2, Issue 2). [Http://Jurnal.Umpar.Ac.Id/Index.Php/Karajata](http://Jurnal.Umpar.Ac.Id/Index.Php/Karajata)•38
- Skripsi. Depok: Fakultas Teknik Universitas Indonesia, 2008.
- Pinedo, M. 2002. *Scheduling – Theory, Algorithms, and Systems 2nd Edition*. Prentice Hall. New Jersey.
- Salim, A. (2016). *Aplikasi Jadwal Mata Kuliah Teknik Informatika dan Sistem Informasi dan Penjadwalan Ruangan Kuliah Berbasis Dekstop* (Doctoral dissertation, Universitas Islam Negeri Alauddin Makassar).
- Sidik, R., Fitriawati, M., Mauluddin, S., & Nursikuwagus, A. (2018). Model Penerapan Algoritma Ant Colony Optimization (Aco) Untuk Optimasi Sistem Informasi Penjadwalan Kuliah. *Jurnal Teknologi dan Informasi*, 8(2), 120-132.
- Silalahi, A., Santoni, M. M., & Muliawati, A. (2020). Pembuatan Aplikasi Penjadwalan Mata Kuliah Menggunakan Algoritma Ant Colony Optimization (Studi Kasus: Program Studi Informatika Fakultas Ilmu

Komputer Universitas Pembangunan Nasional Veteran Jakarta). *Informatik: Jurnal Ilmu Komputer*, 16(3), 148-155.

Suhartono, E. (2015). Optimasi penjadwalan mata kuliah dengan algoritma genetika (studi kasus di amik jtc semarang). *Jurnal Ilmiah INFOKAM*, 11(5).