

**APLIKASI PENJADWALAN KULIAH DENGAN  
MENERAPKAN METODE ALGORITMA STEEPEST-ASCENT  
*HILL CLIMBING* DI FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS BALE BANDUNG**

**SKRIPSI**

Karya Tulis sebagai syarat memperoleh  
Gelar Sarjana Komputer dari Fakultas Teknologi Informasi  
Universitas Bale Bandung

Disusun oleh:

**RIDWAN BUDIMAN**  
**NIM. C1A150004**



**PROGRAM STRATA 1  
PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS BALE BANDUNG  
BANDUNG  
2019**

**LEMBAR PERSETUJUAN PEMBIMBING**

**APLIKASI PENJADWALAN KULIAH DENGAN  
MENERAPKAN METODE ALGORITMA *STEEPEST-ASCENT*  
*HILL CLIMBING* DI FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS BALE BANDUNG**

Disusun oleh:

RIDWAN BUDIMAN  
NIM. C1A150004

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar  
**SARJANA KOMPUTER**

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2019

Disetujui oleh:

Pembimbing 1

Pembimbing 2

Yudi Herdiana, S.T., M.T.  
NIK. 04104808008

Nurul Imamah, S.T., M.T.  
NIK. 04104808121

**LEMBAR PENGESAHAN PENGUJI**

**APLIKASI PENJADWALAN KULIAH DENGAN  
MENERAPKAN METODE ALGORITMA *STEEPEST-ASCENT*  
*HILL CLIMBING* DI FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS BALE BANDUNG**

Disusun oleh:

Ridwan Budiman  
NIM. C1A150004

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar  
**SARJANA KOMPUTER**

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2019

Disetujui oleh:

Pengaji 1

Pengaji 2

Yaya Suharya, S.Kom., M.T.  
NIK. 01043170007

Denny Rusdianto, S.T., M.Kom.  
NIK. 04104808094

**LEMBAR PENGESAHAN PROGRAM STUDI**

**APLIKASI PENJADWALAN KULIAH DENGAN  
MENERAPKAN METODE ALGORITMA *STEEPEST-ASCENT*  
*HILL CLIMBING* DI FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS BALE BANDUNG**

Disusun oleh:

Ridwan Budiman  
NIM. C1A150004

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar  
**SARJANA KOMPUTER**

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2019

Mengetahui,  
Dekan,

Mengesahkan,  
Ketua Program Studi

Yudi Herdiana, S.T., M.T.  
NIK. 04104808008

Yaya Suharya, S.Kom., M.T.  
NIK. 01043170007

## **LEMBAR PERNYATAAN KEASLIAN**

Saya yang bertanda tangan dibawah ini:

Nama : RIDWAN BUDIMAN

NIM : C1A150004

Judul Skripsi : **APLIKASI PENJADWALAN KULIAH DENGAN MENERAPKAN ALGORITMA STEEPEST-ASCENT HILL CLIMBING DI FAKULTAS TEKNOLOGI INFORMASI UNIVERSITAS BALE BANDUNG.**

Menyatakan dengan sebenarnya bahwa penulisan skripsi ini berdasarkan hasil penelitian, pemikiran dan pemaparan asli dari saya sendiri, baik untuk naskah laporan maupun kegiatan *programming* yang tercantum sebagai bagian dari skripsi ini. Jika terdapat karya orang lain, saya mencantumkan sumber yang jelas.

Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidak benaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini dan sanksi lain sesuai dengan peraturan yang berlaku di FAKULTAS TEKNOLOGI INFORMASI UNIVERSITAS BALE BANDUNG.

Demikian surat pernyataan ini saya buat dalam keadaan sadar tanpa paksaan dari pihak manapun.

Baleendah, Agustus 2019

Yang membuat pernyataan,

Ridwan Budiman

NIM. C1A150004

## **ABSTRACT**

*Lecture scheduling is one part of the academic field which is very important to be able to carry out an activity of learning and teaching processes that are good for lectures at the University or College. Good lecture scheduling is what can be done by all students and lecturers. In the process of arranging the lecture schedule at the Faculty of Information Technology, University of Bale Bandung (UNIBBA), it is still manual so it requires a long time and the resulting schedule is often not as expected because it still has problems clashing both lecturers and teaching time. To help solve the problem, a lecture scheduling application was built by applying a steepest-ascent hill climbing algorithm. Steepest-Ascent Hill climbing algorithm as a method in the search process to help provide solutions to scheduling problems such as lecture scheduling. The application of the steepest-ascent hill climbing algorithm can obtain optimal course scheduling solutions and facilitate staff at the Information Technology Faculty so that in making class schedules in each semester.*

**Keywords:** Application, scheduling lectures, Guardianship, Steepest-Ascent Hill Climbing Algorithm, Study Plan Card.

## ABSTRAK

Penjadwalan kuliah merupakan salah satu bagian di bidang akademik yang sangat penting untuk dapat terlaksananya sebuah kegiatan proses belajar dan mengajar yang baik bagi perkuliahan di Universitas atau Perguruan Tinggi. Penjadwalan kuliah yang baik adalah yang dapat dilaksanakan oleh semua mahasiswa dan dosen pengajar. Pada proses penyusunan jadwal perkuliahan di Fakultas Teknologi Informasi Universitas Bale Bandung (UNIBBA) masih manual sehingga membutuhkan waktu yang lama dan jadwal yang dihasilkan sering tidak sesuai dengan yang diharapkan karena masih mengalami masalah bentrok baik pada dosen dan waktu mengajar. Untuk membantu menyelesaikan masalah maka dibangun aplikasi penjadwalan kuliah dengan menerapkan algoritma *steepest-ascent hill climbing*. Algoritma *steepest-ascent hill climbing* sebagai metode dalam proses pencarian untuk membantu memberikan solusi mengenai masalah-masalah penjadwalan seperti pada penjadwalan perkuliahan. Penerapan algoritma *steepest ascent hill climbing* ini dapat memperoleh solusi penjadwalan mata kuliah yang optimal serta memudahkan staff di Fakultas Teknologi Informasi dalam membuat jadwal kuliah di setiap semesternya.

**Kata Kunci:** Aplikasi, Penjadwalan Kuliah, Algoritma *Steepest-Ascent Hill Climbing*.

## KATA PENGANTAR

Segala puji bagi Allah SWT yang telah memberikan Rahmat, Karunia serta Hidayah\_Nya kepada kita semua. Shalawat serta salam semoga terlimpah curahkan kepada baginda kita Nabi Muhammad SAW yang kita nantikan syafa'atnya di akhirat nanti. Penulis mengucapkan syukur kepada Allah SWT atas nikmat sehat\_Nya, baik itu berupa sehat fisik maupun akal pikiran, sehingga penulis mampu menyelesaikan penulisan skripsi ini yang berjudul “APLIKASI PENJADWALAN KULIAH DENGAN MENERAPKAN ALGORITMA STEEPEST-ASCENT HILL CLIMBING DI FAKULTAS TEKNOLOGI INFORMASI UNIVERSITAS BALE BANDUNG ”.

Tujuan penulisan skripsi ini guna untuk memenuhi salah satu syarat dalam menyelesaikan jenjang Strata Satu (S1) di Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Bale Bandung.

Terselesaikannya penulisan skripsi ini tidak terlepas dari bantuan banyak pihak, sehingga pada kesempatan ni dengan segala kerendahan hati dan penuh rasa hormat menghaturkan terima kasih yang sebesar-besarnya kepada kedua orang tua yaitu Bapak Entoy Suhendar dan Ibu Pupu Yuhanah yang telah mencerahkan segenap kasih sayang dan perhatian moril maupun materil kepada penulis. Serta ucapan terima kasih penulis kepada:

1. Bapak Yudi Herdiana, S.T., M.T., selaku Dekan di Fakultas Teknologi Informasi dan juga selaku dosen pembimbing 1 yang telah membimbing dalam penulisan skripsi ini.
2. Bapak Yaya Suharya, S.Kom., M.T., selaku ketua Program Studi Teknik Informatika.
3. Ibu Nurul Imamah, S.T., M.T., selaku pembimbing 2 yang telah banyak meluangkan waktu untuk membantu dan membimbing penulis.
4. Segenap Bapak dan Ibu Dosen di Fakultas Teknologi Informasi yang sudah memberi masukan serta arahan kepada penulis.
5. Rekan-rekan mahasiswa angkatan 2015 terkhusus rekan-rekan Teknik Informatika.

6. Teman-teman seperjuangan Deni Hidayat, Jeri Rismawan, Irfan Wahyu Permana, Gillar Fathonan.
7. Dan semua teman-teman atau pihak-pihak yang tidak bisa penulis sebutkan satu persatu. Terima kasih atas semuanya.

Penulis menyadari bahwa masih banyak kekurangan dari skripsi ini, baik dari materi maupun teknik penyajiannya, mengingat kurangnya pengetahuan dan pengalaman penulis. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan.

Bandung, Agustus 2019

Penulis

## DAFTAR ISI

<i>ABSTRACT</i> .....	vi
ABSTRAK .....	vii
KATA PENGANTAR .....	viii
DAFTAR ISI.....	x
DAFTAL TABEL .....	xiii
DAFTAR GAMBAR .....	xiv
DAFTAR LAMPIRAN .....	xvii
BAB I PENDAHULUAN .....	1
1.1    Latar Berlakang .....	1
1.2    Rumusan Masalah .....	2
1.3    Batasan Masalah.....	2
1.4    Tujuan.....	3
1.5    Metodologi Penelitian .....	3
1.6    Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA.....	6
2.1    Landasan Teori.....	6
2.2    Dasar Teori .....	7
2.2.1    Pengertian Penjadwalan .....	7
2.2.2    Teknik Heuristik.....	8
2.2.3    Pengertian Aplikasi .....	14
2.2.4 <i>Database</i> .....	14
2.2.5 <i>Database Management System (DBMS)</i> .....	17
2.2.6    MySQL.....	19
2.2.7    XAMPP .....	20
2.2.8    Bahasa Pemograman .....	20
2.2.9 <i>Object-Oriented Programming (OOP)</i> .....	21
2.2.10    Java.....	25
2.2.11 <i>Java Development Kit (JDK)</i> .....	26
2.2.12    NetBeans .....	26

2.2.13	<i>Unified Modeling Language (UML)</i> .....	27
2.2.14	JavaFx Scene Builder.....	34
2.2.15	<i>Model-Driven Development (MDD)</i> .....	34
	BAB III METODOLOGI PENELITIAN.....	38
3.1	Kerangka Berfikir.....	38
3.2	Penjelasan Skema Penelitian .....	39
3.2.1	Investigasi Pendahuluan.....	39
3.2.2	Analisis Masalah .....	41
3.2.3	Analisa Kebutuhan .....	41
3.2.4	Desain.....	43
3.2.5	Kontruksi.....	45
3.2.6	Implementasi .....	46
3.2.7	Pembuatan Laporan.....	46
	BAB IV ANALISIS DAN PERANCANGAN .....	49
4.1	Analisis .....	49
4.1.1	Analisis Masalah .....	49
4.1.2	Instrumen penelitian.....	54
4.1.3	Analisis Pengguna .....	56
4.1.4	Analisis <i>User Interface</i> .....	56
4.1.5	Fitur-Fitur .....	57
4.1.6	Analisa Data.....	57
4.2	Perancangan.....	66
4.2.1	Struktur Menu .....	66
4.2.2	<i>Use Case Diagram</i> .....	67
4.2.3	<i>Activity Diagram</i> .....	69
4.2.4	<i>Class Diagram</i> .....	79
4.2.5	<i>Sequence Diagram</i> .....	80
4.2.6	Struktur Tabel.....	84
4.2.7	Perancangan <i>User Interface</i> .....	86
	BAB V IMPLEMENTASI DAN PENGUJIAN .....	104
5.1	Implementasi .....	104
5.1.1	Spesifikasi Sistem .....	104

5.1.2	Implementasi Antar Muka.....	106
5.2	Pengujian .....	115
5.2.1	Rencana Pengujian .....	115
5.2.2	Kasus dan Hasil Pengujian.....	119
BAB VI KESIMPULAN DAN SARAN .....		127
6.1	Kesimpulan.....	127
6.2	Saran .....	127
DAFTAR PUSTAKA .....		128
LAMPIRAN .....		130

## **DAFTAL TABEL**

Tabel 2.1 Simbol-Simbol <i>Class Diagram</i> .....	28
Tabel 2.2 Simbol-Simbol <i>Use Case Diagram</i> .....	30
Tabel 2.3 Simbol-Simbol <i>Activity Diagram</i> .....	31
Tabel 2.4 Simbol-Simbol <i>Sequenci Diagram</i> .....	32
Tabel 4.1 Tabel Jadwal Semester Ganjil 2015.....	49
Tabel 4.2 Data Jadwal Semester Ganjil 2017 .....	51
Tabel 4.3 Data Jadwal Semester Ganjil 2018 .....	53
Tabel 4.4 Perangkat Keras yang digunakan .....	55
Tabel 4.5 Perangkat Lunak yang digunakan. ....	55
Tabel 4.6 Jenis Bentrok.....	57
Tabel 4.7 Data Kelas .....	58
Tabel 4.8 Jadwal.....	63
Tabel 4.9 Variabel Penentu Nilai Bobot Dosen .....	64
Tabel 4.10 Jadwal Random.....	64
Tabel 4.11 Aksi Dosen.....	68
Tabel 4.12 Aksi Staff TU .....	69
Tabel 4.13 Struktur Tabel User .....	84
Tabel 4.14 Struktur Tabel Dosen .....	84
Tabel 4.15 Struktur Tabel Matakuliah .....	85
Tabel 4.16 Struktur Tabel Mahasiswa (MPS).....	85
Tabel 4.17 Struktur Tabel Kesediaan Mengajar .....	85
Tabel 4.18 Struktur Tabel Ruangan .....	85
Tabel 4.19 Struktur Tabel Hari .....	86
Tabel 4.20 Struktur Tabel Kelas .....	86
Tabel 4.21 Struktur Tabel Jadwal .....	86
Tabel 5.1 Perangkat Keras Yang Digunakan .....	104
Tabel 5.2 Minimum Perangkat Keras .....	105
Tabel 5.3 Rencana Pengujian .....	115
Tabel 5.4 Pengujian Aplikasi .....	119

## DAFTAR GAMBAR

Gambar 2.1 Alur Algoritma <i>Hill Climbing</i> .....	9
Gambar 2.2 <i>Tree</i> Untuk <i>Depth First Search</i> .....	13
Gambar 2.3 Skema DBMS.....	18
Gambar 2.4 Konsep Dasar Pemrograman Berorientasi Objek.....	23
Gambar 2.5 Simbol-Simbol <i>Componen Diagram</i> .....	29
Gambar 2.6 Simbol-Simbol <i>Object Diagram</i> .....	30
Gambar 2.7 Simbol-Simbol <i>State Machine Diagram</i> .....	32
Gambar 2.8 <i>Model-Driven Development</i> (MDD). .....	36
Gambar 3.1 Kerangka Berfikir.....	38
Gambar 4.1 Pseudo Code Algoritma <i>Steepest-Ascent Hill Climbing</i> .....	60
Gambar 4.2 Pencarian Solusi dengan Algoritma SAHC. ....	61
Gambar 4.3 Pseudocode Fungsi Rata-Rata Mengajar Dosen. ....	66
Gambar 4.4 Struktur Menu .....	67
Gambar 4.5 Use Case Diagram Aplikasi Penjadwalan.....	68
Gambar 4.6 Activity Diagram Proses Login.....	70
Gambar 4.7 Activity Diagram Proses Input dan Hapus Data Dosen .....	70
Gambar 4.8 Activity Diagram Pencarian dan Hapus Data Dosen .....	71
Gambar 4.9 Activity Diagram Tambah Dan Edit Data Matakuliah.....	71
Gambar 4.10 Activity Diagram Pencarian dan Hapus Data Matakuliah .....	72
Gambar 4.11 Activity Diagram Input dan Edit Kesediaan Mengajar .....	72
Gambar 4.12 Activity Diagram Pencarian dan Hapus Data Mengajar .....	73
Gambar 4.13 Activity Diagram Input dan Edit Mahasiswa.....	73
Gambar 4.14 Activity Diagram Pencarian dan Hapus Data Mahasiswa .....	74
Gambar 4.15 Activity Diagram Input dan Edit Ruangan.....	74
Gambar 4.16 Activity Diagram Pencarian dan Hapus Ruangan.....	75
Gambar 4.17 Activity Diagram Tambah dan Edit Data Waktu.....	75
Gambar 4.18 Activity Diagram Pencarian dan Hapus Data Waktu.....	76
Gambar 4.19 Activity Diagram Tambah dan Edit Data Kelas.....	76
Gambar 4.20 Activity Diagram Pencarian Dan Hapus Data Kelas .....	77
Gambar 4.21 Activity Diagram Tambah dan Ubah Password.....	77

Gambar 4.22 Activity Diagram Input dan Update Jadwal Kuliah .....	78
Gambar 4.23 Activity Diagram Pencarian Dan Hapus Jadwal .....	78
Gambar 4.24 Activity Diagram Proses Optimasi Jadwal Kuliah.....	79
Gambar 4.25 Class Diagram Penjadwalan Yang Diusulkan.....	79
Gambar 4.26 Sequence Diagram Login .....	80
Gambar 4.27 Sequence Diagram Input Dosen .....	80
Gambar 4.28 Sequence Diagram Pengelolaan Matakuliah.....	81
Gambar 4.29 Sequence Diagram Pengelolaan Mahasiswa .....	81
Gambar 4.30 Sequence Diagram Pengelolaan Ruangan.....	82
Gambar 4.31 Sequence Diagram Pengelolaan Waktu .....	82
Gambar 4.32 Sequence Diagram Data Kesediaan mengajar .....	83
Gambar 4.33 Sequence Diagram Pengelolaan Kelas.....	83
Gambar 4.34 Sequence Diagram Penjadwalan SAHC. ....	84
Gambar 4.35 Desain Form Login .....	87
Gambar 4.36 Desain Form Menu Utama .....	88
Gambar 4.37 Desain Form Pengelolaan Dosen .....	89
Gambar 4.38 Desain Form Input Dosen .....	90
Gambar 4.39 Desain Form Matakuliah.....	91
Gambar 4.40 Desain Form Input Matakuliah .....	92
Gambar 4.41 Desain Form Mahasiswa .....	93
Gambar 4.42 Desain Form Kesediaan Mengajar Dosen .....	94
Gambar 4.43 Desain Form Input Mengajar .....	95
Gambar 4.44 Desain Form Pengelolaan Data Ruangan .....	96
Gambar 4.45 Desain Form Waktu .....	97
Gambar 4.46 Desain Form Pengelolaan Data Kelas .....	98
Gambar 4.47 Desain Form Input Kelas .....	99
Gambar 4.48 Desain Form Pengelolaan Data User .....	100
Gambar 4.49 Desain Form Input Data User .....	101
Gambar 4.50 Desain Form Input Jadwal Kuliah.....	101
Gambar 4.51 Desain Form Pengelolaan Jadwal Kuliah.....	102
Gambar 5.1 Tampilan Struktur Menu Login.....	106
Gambar 5.2 Tampilan Halaman Menu Utama .....	107

Gambar 5.3 Tampilan Pengelolaan Dosen.....	107
Gambar 5.4 Tampilan Input Dosen.....	108
Gambar 5.5 Tampilan Pengelolaan Matakuliah.....	108
Gambar 5.6 Tampilan Input Matakuliah.....	109
Gambar 5.7 Tampilan Pengelolaan Mahasiswa .....	109
Gambar 5.8 Tampilan Pengelolaan Ruangan.....	110
Gambar 5.9 Tampilan Pengelolaan Waktu .....	110
Gambar 5.10 Tampilan Kesediaan Mengajar.....	111
Gambar 5.11 Input Kesediaan Mengajar .....	111
Gambar 5.12 Halaman Pengelolaan Kelas.....	112
Gambar 5.13 Form Input Kelas.....	112
Gambar 5.14 Halaman Pengelolaan Jadwal.....	113
Gambar 5.15 Form Input Jadwal.....	113
Gambar 5.16 Tampilan Pengelolaan user .....	114
Gambar 5.17 Form Input User .....	114

## **DAFTAR LAMPIRAN**

Lampiran 1 Listing Program .....	130
Lampiran 2 Biodata Penulis .....	147

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Berlakang**

Penjadwalan mata kuliah merupakan salah satu bagian di bidang akademik yang sangat penting untuk dapat terlaksananya sebuah kegiatan belajar mengajar yang baik bagi perkuliahan di Universitas atau Perguruan Tinggi. Pada penjadwalan perkuliahan meliputi dosen, mahasiswa, mata kuliah, ruangan/kelas dan waktu. Penjadwalan kuliah yang baik adalah penjadwalan yang bisa dilaksanakan oleh semua pihak yang terkait pada proses belajar dan mengajar di perkuliahan, baik itu dosen sebagai pengajar dan mahasiswa yang belajar.

Fakultas Teknologi Informasi adalah salah satu fakultas yang terdapat di Universitas Bale Bandung (UNIBBA) yang berlokasi di Jl.R.A.A Wiranatakusuma No. 7 Baleendah. Jumlah mahasiswa yang ada di jurusan Teknik Informatika dan Sistem Informasi akan semakin bertambah dari tahun ke tahun dan ini akan berdampak pada ruang lingkup jurusan antara mahasiswa dosen dan ruangan.

Pada pengelolaan dan penyusunan jadwal perkuliahan di Fakultas Teknologi Informasi sering terdapat berbagai masalah yang timbul ketika proses pembuatan jadwal yaitu sering terjadinya bentrok baik pada waktu maupun dosen mengajar, dan ruang. Sebagai contoh dosen dijadwalkan pada dua kelas yang berbeda dalam waktu yang sama maupun dua dosen dijadwalkan pada kelas yang sama dalam waktu yang sama. Hal itu membuat proses pelaksanaan kegiatan matakuliah menjadi sedikit terhambat.

Masalah-masalah penjadwalan yang sering terjadi tersebut dapat dihindari dengan perhitungan penjawalan yang tepat dan teliti dan mencakup semua kemungkinan yang dapat dibuat, oleh karena itu membutuhkan waktu yang lama dan ketelitian yang tinggi. Pembuatan jadwal dapat dilakukan secara komputerisasi dengan menggunakan algoritma/metode tertentu. Metode yang dapat digunakan dalam pembuatan jadwal perkuliahan adalah salah satunya

dengan metode pencarian heuristik. Pencarian heuristik terdiri dari *Generate and Test.* (Pembangkitan dan Pengujian), *Algoritma Hill Climbing*, *Steepest-Ascent Hill Climbing*, *Best First Search*, dan *Simulated Annealing*.

Dengan adanya masalah pada penjadwalan mata kuliah maka akan dibangun sistem penjadwalan yang dapat menghasilkan jadwal yang efisien. Algoritma *Steepest-Ascent Hill Climbing* sebagai metode dalam proses penjadwalan yang optimal dan cepat sehingga meminimalisasi bentrok pada jadwal perkuliahan. Maka penelitian ini akan membuat aplikasi penjadwalan dengan menerapkan metode algoritma *Steepest-Ascent Hill Climbing*.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut permasalahan yang di angkat yaitu:

1. Bagaimana mengimplementasikan algoritma *steepest-ascent hill climbing*.
2. Bagaimana membuat aplikasi penjadwalan yang efektif dengan menerapkan algoritma *steepest-ascent hill climbing* sehingga diperoleh jadwal mata kuliah dengan kombinasi terbaik untuk pasangan mata kuliah dan dosen pengajar secara keseluruhan.

## 1.3 Batasan Masalah

Batasan-batasan dari aplikasi yang akan dibangun adalah:

1. Aplikasi hanya sebatas penyusunan jadwal perkuliahan.
2. Aplikasi berbasis dekstop.
3. Aplikasi menggunakan bahasa pemograman Java.
4. Aplikasi menggunakan MySQL sebagai database
5. Aplikasi sebatas penginputan data, edit data, hapus data dan proses optimasi data jadwal bentrok.
6. Jadwal yang di optimasi berdasarkan data kesediaan mengajar masing-masing dosen pengajar.

## 1.4 Tujuan

Adapun tujuan dari penelitian adalah:

1. Mempermudah dalam penyusunan jadwal, efesensi waktu serta meringankan beban tenaga kerja..
2. Menghasilkan aplikasi penjadwalan perkuliahan yang efektif dan terstruktur.

## 1.5 Metodologi Penelitian

Dalam menyusun laporan tugas akhir ini, penyusun menggunakan beberapa teknik pengumpulan data, antara lain :

1. Metode Pengumpulan data

Yaitu teknik pengumpulan data melalui tinjauan langsung pada objek pembahasan guna mendapatkan data yang konkret dengan cara :

- a. Studi Lapangan

- 1). Wawancara

Penulis mengadakan tanya jawab dan dialog langsung dengan staff TU di Fakultas Teknologi Informasi yang bertujuan untuk mencari dan menemukan informasi terkait proses masalah yang akan diteliti.

- 2). Observasi

Dalam metode ini penulis melakukan pengamatan langsung terhadap objek yang akan diteliti yang bertujuan untuk mengetahui dan mendapatkan informasi secara langsung bagaimana sistem yang sedang berjalan di Fakultas Teknologi Informasi

- 3). Studi Pustaka

Yaitu teknik pengumpulan data yang didapat dari buku-buku panduan atau referensi yang sesuai dengan masalah yang dibahas, dengan cara mempelajari dan menelaah isi literatur melalui pendekatan pada masalah yang sebenarnya atau yang berhubungan dengan tugas yang diberikan.

## 2. Analisis Masalah

Analisis dapat diartikan sebagai upaya mengolah data menjadi informasi, sehingga data tersebut dapat dengan mudah dipahami untuk menjawab masalah-masalah yang berkaitan dengan kegiatan penelitian.

## 3. Perancangan

Proses perancangan aplikasi membagi persyaratan dalam sistem perangkat keras atau perangkat lunak.

## 4. Implementasi

Perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program

## **1.6 Sistematika Penulisan**

Untuk memberikan gambaran secara jelas, maka dirancang sebuah sistematika penulisan pada laporan tugas akhir, adapun sistematika penulisan yang akan dibahas dalam penulisan laporan tugas akhir ini sebagai berikut:

## **BAB I PENDAHULUAN**

Pada bab ini dijelaskan beberapa hal yang mendasari atau yang melatar belakangi penulisan tugas akhir ini. Di dalam bab ini dideskripsikan tentang latar belakang, rumusan masalah, batasan, tujuan, metodologi penelitian, dan sistematika penulisan.

## **BAB II TINJAUAN PUSTAKA**

Pada bab ini berisi tentang landasan teori yang mendasari pembahasan secara detail dan digunakan sebagai dasar untuk menganalisis, turunan hasil-hasil penelitian terdahulu, dan dasar teori yang menjelaskan teori-teori yang relevan dengan kegiatan obsevasi/ dilakukan atau masalah yang diteliti.

### **BAB III METODOLOGI**

Bab ini menjelaskan secara singkat metode yang digunakan untuk memperoleh data yang diperlukan atas permasalahan yang ada terdiri dari kerangka pemikiran, sumber data dan jenis data serta metode analisis data.

### **BAB IV ANALISIS DAN PERANCANGAN**

Pada bab ini akan dibahas mengenai analisis dan perancangan sistem aplikasi yang akan dibuat. Pokok bahasan yang terdapat dalam bab ini adalah deskripsi produk, kebutuhan khusus, kebutuhan fungsionalitas, dan perancangan arsitektur dari aplikasi penjadwalan perkuliahan.

### **BAB V IMPLEMENTASI DAN PENGUJIAN**

Pada bab ini akan dilakukan implementasi dan pengujian terhadap sistem. Tahapan ini dilakukan setelah perancangan selesai dilakukan dan selanjutnya akan diimplementasikan pada bahasa pemrograman. Setelah diimplementasikan maka dilakukan pengujian terhadap sistem dan dilihat kekurangan-kekurangan pada aplikasi untuk pengembangan sistem selanjutnya.

### **BAB VI KESIMPULAN DAN SARAN**

Bab ini merupakan bagian akhir dari laporan penelitian yang berisi mengenai kesimpulan dan saran yang dibutuhkan terhadap hasil akhir dari penelitian yang telah dibuat.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Landasan Teori.

##### 1. “Penerapan Metode Algoritma Genetika Untuk Penjadwalan Mengajar”

Penjadwalan adalah bentuk kegiatan dalam proses belajar dan mengajar di semua sekolah untuk para pengajar khususnya di SMK Giri Pandwa. Dalam aspek kegiatan mengejar sangat penting untuk kelancaran proses belajar disekolah dan terkadang waktu sering bertumpukan untuk para pengajar. Oleh karena itu perlu adanya penjadwalan khusus para pengajar ataupun guru untuk menghindari terjadinya tumpukan. Penelitian ini menggunakan algoritma genetika untuk menyelesaikan masalah optimasi kompleks dan cukup efektif dan efisien digunakan untuk pembuatan jadwal mengajar dibandingkan cara manual. (I Gusti Ayu Desi Saryanti, I Kadek wijanegara 2017)

##### 2. “Penjadwalan Perkuliahan Menggunakan Algoritma *Hill Climbing*”

Program Studi Teknik Informatika di Universitas Pamulang memiliki jumlah kelas, dosen, mata kuliah, dan waktu perkuliahan yang cukup banyak, sehingga sering terjadinya bentrok antar jadwal. Dalam penyelesaiannya digunakan metode algoritma *hill climbing* untuk melakukan proses penjadwalan. Pada penelitian ini dilakukan secara eksperimen, yaitu dengan melakukan percobaan dan pembuatan aplikasi di laboratorium. Dari penelitian yang dilakukan, dapat disimpulkan bahwa algoritma *hill climbing* dapat memperoleh solusi penjadwalan mata kuliah yang optimal. Dan proses pengembangan yang mungkin untuk dilakukan, pihak peneliti memberikan masukan pada proses pengecekan solusi jadwal ditambahkan fungsi agar tidak dilakukan proses pengecekan kembali pada solusi jadwal yang sama dan dapat ditambahkan fungsi untuk laporan beban akademik dosen disetiap semesternya. (Aida Putri Deswanti, Aries Saifudin 2016).

### 3. “Pengembangan Sistem Penjadwalan Kuliah Menggunakan Algoritma *Steepest Ascent Hill Climbing*”

Salah satu faktor untuk meningkatkan pelayanan akademik adalah penjadwalan kuliah. Proses penjadwalan kuliah bukan merupakan suatu kegiatan yang mudah untuk dilakukan, karena dalam menyususun jadwal antara matakuliah, waktu, dosen dan ruangan. Akan tetapi, banyak kemungkinan kombinasi yang perlu diperhatikan. Oleh karena itu perlu dibangun sistem penjadwalan kuliah dengan menggunakan algoritma *steepest ascent hill climbing* yang lebih cepat, efektif dan optimal sehingga jadwal yang telah diproses tidak akan mengalami bentrok. Dan pada hasilnya dapat disimpulkan bahwa algoritma *steepest ascent hill climbing* mampu membantu mempercepat dalam proses penjadwalan Fakultas Sains dan Teknologi Universitas Teknologi Yogyakarta jika dibandingkan dengan proses yang dilakukan saat ini secara manual satu persatu dalam menjadwalkan kuliah. (Shoffan Saifullah, Arief Hermawan 2017)

## 2.2 Dasar Teori

### 2.2.1 Pengertian Penjadwalan

Penjadwalan merupakan aktivitas perencanaan untuk menentukan kapan dan dimana setiap operasi sebagai bagian dari pekerjaan harus dilakukan pada sumber daya dan waktu yang terbatas mencakup kepada pengurutan aktivitas, pengalokasian aktivitas pada fasilitas dan pemetaan aktivitas.

Menurut Kamus Besar Bahasa Indonesia (KBBI), jadwal adalah pembagian waktu berdasarkan rencana urutan kerja; daftar atau tabel kegiatan atau rencana kegiatan dengan pembagian pelaksanaan yang terperinci. Sedangkan pengetian penjadwalan itu sendiri adalah proses/cara/perbuatan menjadwalkan atau memasukan kedalam jadwal.

Menurut (Pinedo, 2012), penjadwalan dapat didefinisikan sebagai proses pengalokasian sumber daya untuk mengerjakan sekumpulan tugas dalam jangka waktu tertentu dengan 2 arti penting sebagai berikut.

- a. Penjadwalan merupakan fungsi pengambilan keputusan untuk membuat atau menentukan jadwal.
- b. Penjadwalan merupakan suatu teori yang berisi sekumpulan prinsip dasar, model, teknik, dan kesimpulan logis dalam proses pengambilan keputusan yang memberikan pengertian dalam fungsi penjadwalan.

### **2.2.2 Teknik Heuristik**

Heuristik adalah seni dan ilmu pengetahuan yang berhubungan dengan suatu penemuan. Heuristik yang berkaitan dengan pemecahan masalah adalah cara menunjukkan pemikiran seseorang dalam melakukan proses pemecahan sampai masalah tersebut berhasil dipecahkan. Ini berbeda dari algoritma dimana hanya digunakan sebagai pedoman, bertentangan dengan prosedur invarian.

Heuristik adalah suatu proses yang mungkin dapat menyelesaikan masalah tetapi tidak ada jaminan bahwa solusi yang dicari selalu dapat ditemukan. Dengan kata lain, heuristik adalah teknik yang mengembangkan efisiensi dalam proses pencarian, namun dengan kemungkinan mengorbankan kelengkapan (*completeness*).

Fungsi heuristik digunakan untuk mengevaluasi keadaan-keadaan problema inividial dan menentukan seberapa jauh hal tersebut dapat digunakan untuk mendapatkan solusi yang diinginkan.

Ada 4 metode pencarian heuristik :

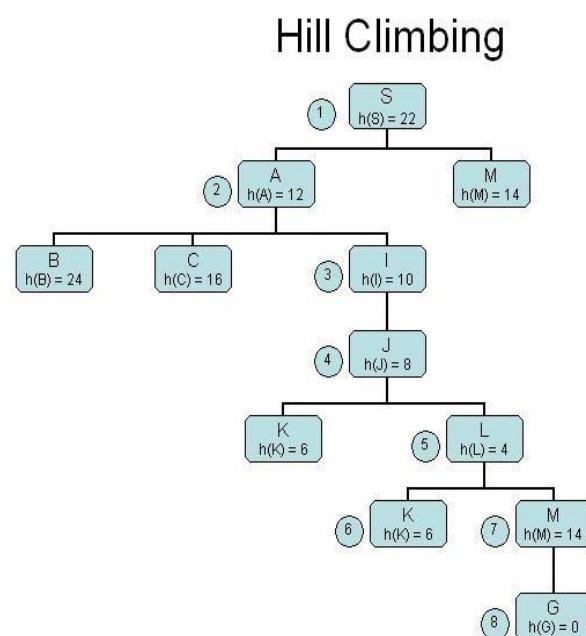
- a. Pembangkitan & Pengujian (*Generate and Test*)
- b. Pendakian Bukit (*Hill Climbing*)
- c. Pencarian Terbaik Pertama (*Best First Search*)
- d. *Simulated Annealing*

#### **1. Algoritma *Hill Climbing***

Algoritma adalah suatu prosedur yang jelas untuk menyelesaikan suatu persoalan dengan menggunakan langkah-langkah tertentu dan terbatas jumlahnya. Instruksi yang jelas untuk memperoleh keluaran yang diinginkan dari suatu masukan dalam jumlah waktu yang terbatas (Suarga 2012:1).

Algoritma *hill climbing* adalah teknik optimasi matematis yang termasuk dalam kategori teknik pencarian lokal. Metode ini hampir sama dengan pembangkitan dan pengujian, hanya saja proses pengujian dilakukan dengan menggunakan fungsi *heuristic*. Pembangkitan keadaan berikutnya tergantung pada *feedback* dari prosedur pengetesan. Tes yang berupa fungsi heuristik ini akan menunjukkan seberapa baiknya nilai terkaan yang diambil terhadap keadaan-keadaan lainnya yang mungkin.

Metode *hill climbing* merupakan variasi dari *depth-first search*. Dengan metode ini, eksplorasi terhadap keputusan dilakukan dengan cara *depth-first search* dengan mencari *path* yang bertujuan untuk menurunkan *cost* untuk menuju kepada *goal*/keputusan, yaitu dengan selalu memilih nilai heuristik terkecil. Dalam metode heuristik *hill climbing*, terdapat dua jenis *Hill Climbing* yang sedikit berbeda, yakni *Simple Hill Climbing* (Hill climbing Sederhana) dan *Steepest-Ascent Hill Climbing* (Hill Climbing dengan memilih kemiringan yang tajam/ curam).



Gambar 2.1 Alur Algoritma *Hill Climbing*

Dasar penyelesaian permasalahan dengan mengikuti langkah berikut ini :

1. Mulai dari keadaan awal, lakukan pengujian: jika merupakan tujuan, maka berhenti; dan jika tidak, lanjutkan dengan keadaan sekarang sebagai keadaan awal.
2. Kerjakan langkah-langkah berikut sampai solusinya ditemukan atau sampai tidak ada operator baru yang akan di aplikasikan pada keadaan sekarang
  - a. Cari operator yang belum pernah digunakan; gunakan operator ini untuk mendapatkan keadaan yang baru.
  - b. Evaluasi keadaan baru tersebut:
    - 1). Jika keadaan baru merupakan tujuan, keluar.
    - 2). Jika bukan tujuan, namun nilainya lebih baik daripada keadaan sekarang, maka jadikan keadaan baru tersebut menjadi keadaan sekarang.
    - 3). Jika keadaan baru tidak lebih baik daripada keadaan sekarang, maka lanjutkan iterasi.

## **2. Algoritma *Steepest-Ascent Hill Climbing***

Algoritma *steepest-ascent hill climbing* adalah salah satu algoritma yang sering digunakan untuk memecahkan masalah optimasi. Secara harafiah *steepest* berarti tertinggi, sedangkan *ascent* adalah kenaikan, sehingga *steepest ascent hill climbing* dapat diartikan sebagai kenaikan tertinggi, dimana prinsip dasar dari algoritma ini adalah mencari kenaikan tertinggi dari solusi-solusi yang mungkin terjadi untuk mencapai nilai yang paling optimal untuk solusi terbaik (Thiang, dkk, 2009).

Algoritma *steepest-ascent hill climbing* merupakan pengembangan dari algoritma *simple hill climbing*. Perbedaannya terletak pada *simple hill climbing*, *next state* akan ditentukan dengan membandingkan *current state* dengan satu *successor* saja, proses perbandingan dini dimulai dari sebelah kiri. Apabila ditemukan *successor* baru yang lebih baik dari *current state* maka *successor* tersebut akan menjadi *next state*. Sedangkan pada *steepest-ascent hill climbing* dalam menentukan *next state*, *current state* akan dibandingkan dengan semua *successor* yang paling

baik serta mendekati hasil optimasi yang diharapkan. Selain itu *steepest-ascent hill climbing* urutan penggunaan operator tidak berpengaruh terhadap solusi sedangkan pada *simple hill climbing* urutan penggunaan operator sangat berpengaruh.

Algoritma *steepest ascent hill climbing*:

1. Evaluasi keadaan awal (Initial State). Jika keadaan awal sama dengan tujuan (Goal state) maka kembali pada *initial state* dan berhenti berproses. Jika tidak maka *initial state* tersebut jadikan sebagai *current state*.
2. Mulai dengan *current state* = *initial state*.
3. Dapatkan semua pewaris (*successor*) yang dapat dijadikan *next state* pada *current statenya* dan evaluasi *successor* tersebut dengan fungsi evaluasi dan beri nilai pada setiap *successor* tersebut. Jika salah satu dari *successor* tersebut mempunyai nilai yang lebih baik dari *current state* maka jadikan *successor* dengan nilai yang paling baik tersebut sebagai *new current state*. Lakukan operasi ini terus menerus hingga tercapai *current state* = *goal state* atau tidak ada perubahan pada *current statenya*.

Struktur umum dari suatu algoritma genetika dapat didefinisikan dengan langkah-langkah sebagai berikut (Suhartono, 2015):

- a. Menghasilkan atau membangkitkan populasi awal (*generate*)

Populasi awal di-*generate* secara random, dimana populasi tersebut terdiri dari beberapa kromosom yang telah didefinisikan sehingga dapat dijadikan solusi awal. Populasi tersebut terdiri dari beberapa kromosom yang mewakili solusi yang diinginkan.

- b. Membentuk generasi baru

Membuat generasi baru menggunakan tiga operator yaitu operator reproduksi/seleksi, *crossover* dan mutasi. Tahapan tersebut dilakukan secara berulang-ulang sehingga diperoleh sejumlah kromosom yang cukup untuk menghasilkan generasi baru dimana generasi baru tersebut merupakan presentasi dari solusi baru. Generasi baru ini dikenal dengan istilah anak (*offspring*).

c. Evaluasi solusi

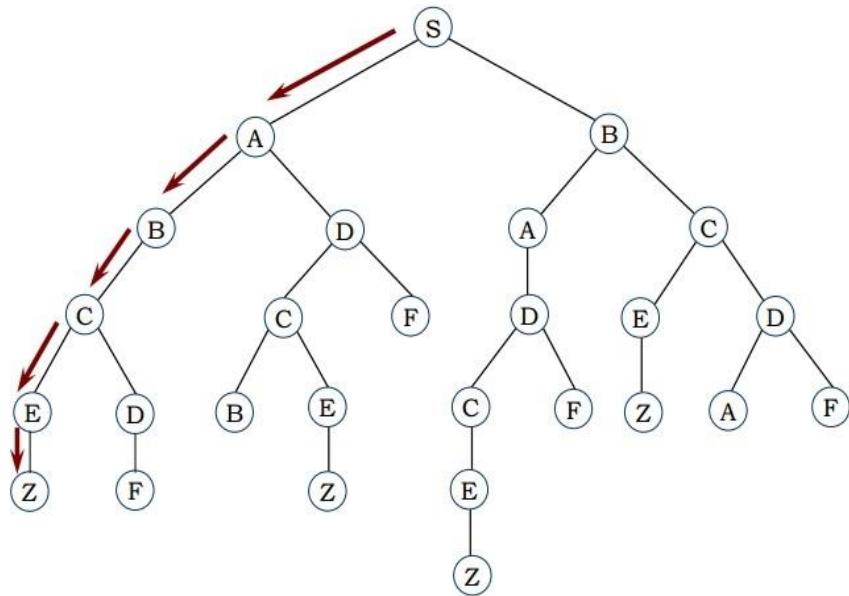
Pada tiap generasi, kromosom akan diukur dengan fungsi *fitness*. Nilai *fitness* suatu kromosom menggambarkan kualitas kromosom dalam populasi tersebut. Proses ini akan mengevaluasi setiap populasi dengan menghitung nilai *fitness* setiap kromosom dan mengevaluasinya sampai terpenuhi kriteria berhenti. Apabila kriteria berhenti belum terpenuhi maka akan dibentuk lagi generasi baru dengan mengulangi langkah b. beberapa kriteria berhenti yang sering digunakan antara lain:

1. Berhenti pada generasi tertentu.
2. Berhenti setelah beberapa generasi berturut-turut didapatkan nilai *fitness* tertinggi.
3. Berhenti pada n generasi dimana nilai *fitness* dari populasi tidak mengalami perubahan. Berhenti pada generasi tertentu.

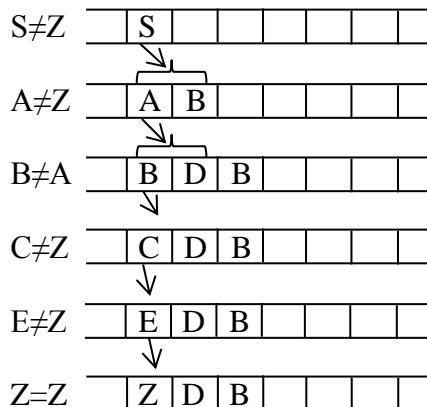
**3. Depth-First Search (DFS)**

DFS (*Depth-First Search*) adalah salah satu algoritma penelusuran struktur *graf/pohon* berdasarkan kedalaman. Pencarian dilakukan pada satu node dalam setiap level dari yang paling kiri. Jika pada level yang paling dalam, solusi belum ditemukan, maka pencarian dilanjutkan pada node sebelah kanan. Node yang kiri dapat dihapus dari memori. Jika pada level yang paling dalam tidak ditemukan solusi, maka pencarian dilanjutkan pada level sebelumnya. Demikian seterusnya sampai ditemukan solusi. Jika solusi ditemukan maka tidak diperlukan *backtracking* (penelusuran balik untuk mendapatkan solusi yang diinginkan).

- Algoritma:
  1. Buat sebuah antrian, inisialisasi node pertama dengan *Root* dari *tree*.
  2. Bila node pertama, jika  $\neq$  GOAL, node dihapus dihanti dengan anak-anaknya dengan urutan *Lchild*.
  3. Bila node pertama = GOAL, selesai.



Gambar 2.2 Tree Untuk Depth First Search.



Lintasan yang di dapat : S – A – B – C – E – Z

- Keuntungan
  1. Membutuhkan memori yang relative kecil, karena hanya node-node pada lintasan yang aktif saja yang akan disimpan.
  2. Menemukan solusi tanpa harus menguji lebih banyak lagi dalam ruang keadaan.
- Kelemahan
  1. Kemungkinan terjebak pada optimal lokal.
  2. Hanya akan mendapatkan satu solusi pada setiap pencarian.

### 2.2.3 Pengertian Aplikasi

Menurut (Anisyah, 2000:30), aplikasi adalah penerapan, penggunaan atau penambahan. Dari pengertian aplikasi dapat disimpulkan bahwa aplikasi merupakan software yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data.

Aplikasi dalam suatu paket biasanya memiliki antar muka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi. Seringkali mereka memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna.

### 2.2.4 *Database*

#### A Pengertian *Database*

*Database* adalah kumpulan berbagai data dan informasi yang tersimpan dan tersusun di dalam komputer secara sistematik yang dapat diperiksa, diolah atau dimanipulasi dengan menggunakan program komputer untuk mendapatkan informasi dari basis data tersebut.

Secara formal, “*database*” mengacu pada sekumpulan data terkait dan cara mengelolanya. Akses ke data ini biasanya disediakan oleh “sistem manajemen basis data” (DBMS) yang terdiri dari satu set perangkat lunak komputer terintegrasi yang memungkinkan pengguna untuk berinteraksi dengan satu atau lebih basis data dan menyediakan akses ke semua data yang terkandung dalam basis data (meskipun pembatasan mungkin ada yang membatasi akses ke data tertentu). DBMS menyediakan berbagai fungsi yang memungkinkan entri, penyimpanan, dan pengambilan informasi dalam jumlah besar dan menyediakan cara untuk mengelola bagaimana informasi tersebut diorganisasikan. Karena hubungan erat diantara mereka, istilah “*database*” sering digunakan dengan santai untuk merujuk ke *database* dan DBMS yang digunakan untuk memanipulasinya. (Sumber: <https://en.m.wikipedia.org/wiki/Database>).

## B Fungsi *Database*

1. Suatu data dapat dikelompokan dengan tujuan mempermudah proses identifikasi data, pengelompokan dapat dilakukan dengan berbagai macam cara seperti membuat beberapa tabel atau dengan *field* yang berbeda-beda.
2. Menghindari tanda ganda yang tersimpan. Suatu *software DBMS* dapat disetting agar mampu mengenali duplikasi data yang terjadi saat di *input*. Hal ini dikarenakan sifat *database* yang dapat diakses oleh lebih dari satu pengguna.
3. Mempermudah penggunaan hampir disemua sisi seperti memasukan data baru, mengupdate atau bahkan menghapus data yang sudah tidak diperlukan lagi. Didukung dengan tampilan atau tatap muka yang sudah disediakan menggunakan aplikasi tertentu.
4. Menjaga kualitas data dan informasi yang diakses sesuai dengan yang di *input*.
5. Membantu proses penyimpanan data besar.

## C Manfaat *Database*

1. Kecepatan dan kemudahan

*Database* memiliki kemampuan dalam menyeleksi data sehingga menjadi suatu kelompok yang terurut dengan cepat, sehingga dapat, menghasilkan informasi yang dibutuhkan secara cepat pula.

2. Tidak Terjadi Redudansi Basis Data

*Database* bisa membantu meminimalkan redundansi data. Redundansi sendiri merupakan terjadinya data-data ganda dalam berkas-berkas yang berbeda.

3. Integritas Terjaga

*Database* memastikan integritas data yang tinggi dimana *database* akan memastikan keakuratan, aksesibilitas, konsistensi, dan juga kualitas tinggi pada suatu data.

4. Independensi Data Terjaga

Menjaga independensi data dimana orang lain dapat merubah data meskipun data bisa diakses.

5. Kemudahan Berbagi Data

Menggunakan perangkat lunak *database* bisa digunakan untuk berbagi data atau informasi dengan sesama pengguna lainnya.

6. Menjaga Keamanan Data

*Database* menjamin keamanan suatu informasi dan data, dimana kita bisa menyisipkan kode akses untuk data-data tertentu yang tidak bisa diakses bersama.

## D *Type Database*

1. *Analytical Database*

*Database* untuk menyimpan informasi dan data yang diambil dari operasional dan eksternal *database*.

2. *Operatuation Database*

*Database* yang menyimpan data mendetail yang dibutuhkan untuk mendukung operasi satu organisasi secara keseluruhan.

3. *Distributed Database*

Kelompok kerja lokal *database* dan departemen di berbagai kantor dan lokasi kerja yang lainnya. *Database* ini terdapat dua segmen yaitu user *database* dan operasional yang datanya digunakan dan dihasilkan pada situs itu sendiri.

4. *Data Warehouse*

Sebuah gudang data yang menyimpan berbagai data dari tahun-tahun sebelumnya hingga saat ini.

5. *End-user Database*

Basis data pengguna akhir yang terdiri dari berbagai file data yang dikembangkan dari *end-user* dalam *workstation* mereka.

6. *Real Time Database*

Sistem pengolahan yang dirancang dalam menangani beban kerja suatu negara yang bisa berubah-ubah, mengandung data terus menerus dan sebagian tidak berpengaruh terhadap waktu.

#### 7. *Document Oriented Database*

Salah satu perangkat lunak komputer yang dibuat untuk sebuah aplikasi dan berorientasi pada dokumen.

#### 8. *In Memory Database*

Merupakan salah satu program komputer yang dirangkai untuk sebuah aplikasi yang berorientasi pada dokumen. Sistem ini dapat diterapkan sebagai lapisan diatas *database* relasional maupun objek data.

#### 9. *Hypermedia Database*

Merupakan sekumpulan halaman multimedia yang saling berkaitan satu sama lain, yang terdiri dari *homepage*, dan *hyperlink* multimedia seperti gambar, text.

#### 10. *Navigational Database*

Pada navigasi *database*, *queries* menemukan benda bagi yang mengikuti referensi dari objek tertentu.

#### 11. *External Database*

*Database* yang menyediakan akses ke luar, data pribadi online.

#### 12. *Relational Database*

Merupakan standar komputasi bisnis, dan basis data yang paling umum dipakai saat ini.

### **2.2.5 *Database Management System (DBMS)***

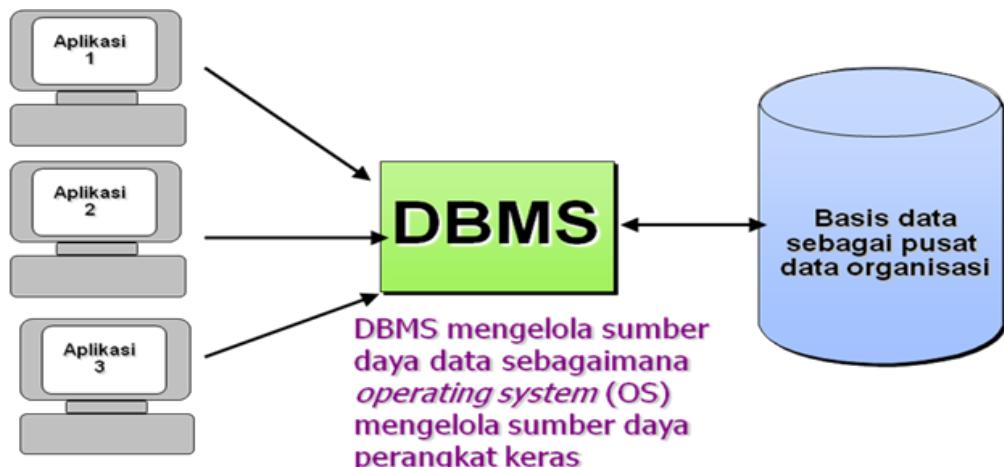
DBMS adalah singkatan dari “*Database Management System*” yaitu sistem pengorganisasian dan sistem pengolahan *database* pada komputer. DBMS merupakan perangkat lunak (*software*) yang dipakai untuk membangun basis data yang berbasis komputerisasi.

DBMS (*Database Management System*) ini juga dapat membantu dalam memelihara serta mengolah data dalam jumlah yang besar, dengan menggunakan DBMS bertujuan agar tidak dapat menimbulkan kekacauan dan dapat dipakai oleh user sesuai dengan kebutuhan.

DBMS adalah perantara *user* dengan basis data, untuk dapat berinteraksi dengan DBMS dapat menggunakan bahasa basis data yang sudah ditentukan oleh perusahaan DBMS. Bahasa Basis data umumnya terdiri dari berbagai macam intruksi yang diformulasikan sehingga intruksi tersebut dapat diproses oleh DBMS.

Tujuan DBMS:

- Dapat digunakan bersama-sama
- Kecepatan serta kemudahan dalam mengakses data
- Efisiensi ruang penyimpanan data
- Untuk menangani data dalam jumlah yang besar atau banyak
- Untuk menghilangkan duplikasi data dan juga inkonsistensi data
- Untuk keamanan Data.



Gambar 2.3 Skema DBMS.

Perintah tersebut umumnya digunakan oleh user, adapun bahasa yang digunakan dibagi kedalam 2 (dua) macam diantaranya:

1) DDL (*Data Definition Language*)

*Data Definition Language* yaitu dipakai untuk menggambarkan desain dari

basis data secara menyeluruh. DDL dapat dipakai untuk membuat tabel baru, memuat indeks maupun mengubah tabel. Hasil dari kompilasi DDL akan disimpan di kamus data.

## 2) DML (*Data Manipulation Language*)

*Data Manipulation Language* yaitu dipakai untuk memanipulasi dan mengambil data pada suatu basis data, misalnya seperti menambah data yang baru kedalam suatu basis data, menghapus data pada suatu basis data dan mengubah data pada suatu basis data.

### 2.2.6 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang multialur, multi pengguna. MySQL sebenarnya merupakan turunan dari salah satu konsep utama dalam database untuk pemilihan atau seleksi dan pemasukan data yang memungkinkan pengoprasiannya data dikerjakan secara mudah dan otomatis.

Berikut kelebihan dan kekurangan MySQL :

Kelebihan:

- Free atau gratis sehingga dapat dengan mudah untuk mendapatkannya
- MySQL stabil dan tangguh dalam pengoprasiannya
- MySQL mempunyai sistem keamanan yang cukup baik
- Sangat mendukung transaksi dan mempunyai banyak dukungan dari komunitas
- Sangat fleksibel dengan berbagai macam program
- Perkembangan dari MySQL sangat cepat.

Kekurangan:

- Kurang mendukung koneksi bahasa pemrograman seperti Visual Basic (VB), Foxpro, Delphi sebab koneksi ini menyebabkan field yang dibaca harus sesuai dengan koneksi dari bahasa pemrograman visual tersebut.

- Data yang dapat ditangani belum besar dan belum mendukung windowing function.

### 2.2.7 XAMPP

XAMPP adalah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan control panel dari beberapa program. Yang mempunyai fungsi sebagai server yang berdiri sendiri (localhost), yang terdiri dari program MySQL database, Apache HTTP Server, dan penerjemah ditulis dalam bahasa pemrograman PHP dan Perl.

Nama XAMPP merupakan singkatan dari X (empat sistem operasi), Apache, MySQL, PHP dan Perl. Program ini tersedia di bawah GNU General Public License dan bebas, adalah mudah untuk menggunakan web server yang dapat melayani tampilan halaman web yang dinamis. Bagian dari XAMPP yang penting untuk dipahami:

- **htdoc** adalah folder di mana Anda meletakkan file yang akan dijalankan, seperti file PHP, HTML dan script lainnya.
- **phpMyAdmin** adalah bagian untuk mengelola *database* MySQL yang dikomputer.
- Untuk membukanya, membuka browser dan ketik alamat `http://localhost / phpMyAdmin`, halaman phpMyAdmin akan muncul.
- **Control Panel** yang berfungsi untuk mengelola layanan (service) XAMPP. Seperti stop service (berhenti), atau mulai (mulai).

### 2.2.8 Bahasa Pemograman

Bahasa pemograman (*programming language*) adalah teknik komunikasi standar untuk mengekspresikan intruksi kepada komputer. Layaknya bahasa manusia, setiap bahasa memiliki tata tulis dan aturan tertentu. Bahasa pemograman memfasilitasi seorang programer secara tepat menetapkan data apa yang sedang dilakukan oleh komputer selanjutnya, bagaimana data tersebut

disimpan dan dikirim, dan apa yang akan dilakukan apabila terjadi kondisi yang variatif.

Bahasa pemrograman dapat diklasifikasikan menjadi 3 yaitu tingkat rendah, menengah, dan tingkat tinggi. Pergeseran tingkat dari rendah menjadi tinggi menunjukkan kedekatan terhadap “bahasa manusia”.

Kategori Bahasa Pemograman diantaranya:

1. Bahasa Pemograman Tingkat Rendah

Bahasa pemograman generasi pertama. Bahasa jenis ini sangat sulit dimengerti karena instruksiannya menggunakan bahasa mesin. Disebut juga bahasa assembly karena merupakan bahasa dengan pemetaan satu-persatu terhadap intruksi komputer. Setiap intruksi assembly diterjemahkan dengan menggunakan assembler.

2. Bahasa Pemograman Tingkat Menengah

Bahasa pemograman generasi kedua dimana penggunaan intruksi telah mendekati bahasa sehari-hari, walaupun masih cukup sulit dimengerti karena menggunakan singkatan-singkatan seperti STO yang berarti simpan (STORE) dan MOV yang berarti pindah (MOVE). Yang tergolong dalam bahasa ini adalah Fortran.

3. Bahasa Pemograman Tingkat Tinggi

Merupakan bahasa pemograman yang mempunyai ciri-ciri mudah dimengerti karena kedekatannya terhadap bahasa sehari-hari. Sebuah pernyataan program diterjemahkan kepada sebuah atau beberapa mesin dengan menggunakan compiler. Sebagai contoh adalah: JAVA, C++, .NET.

### **2.2.9 *Object-Oriented Programming (OOP)***

Pemograman berorientasi objek (*Object-Oriented Programming*) adalah paradigma pemograman yang berorientasikan kepada objek yang merupakan suatu metode dalam pembuatan program, dengan tujuan untuk menyelesaikan kompleksnya berbagai masalah program yang terus meningkat. Objek

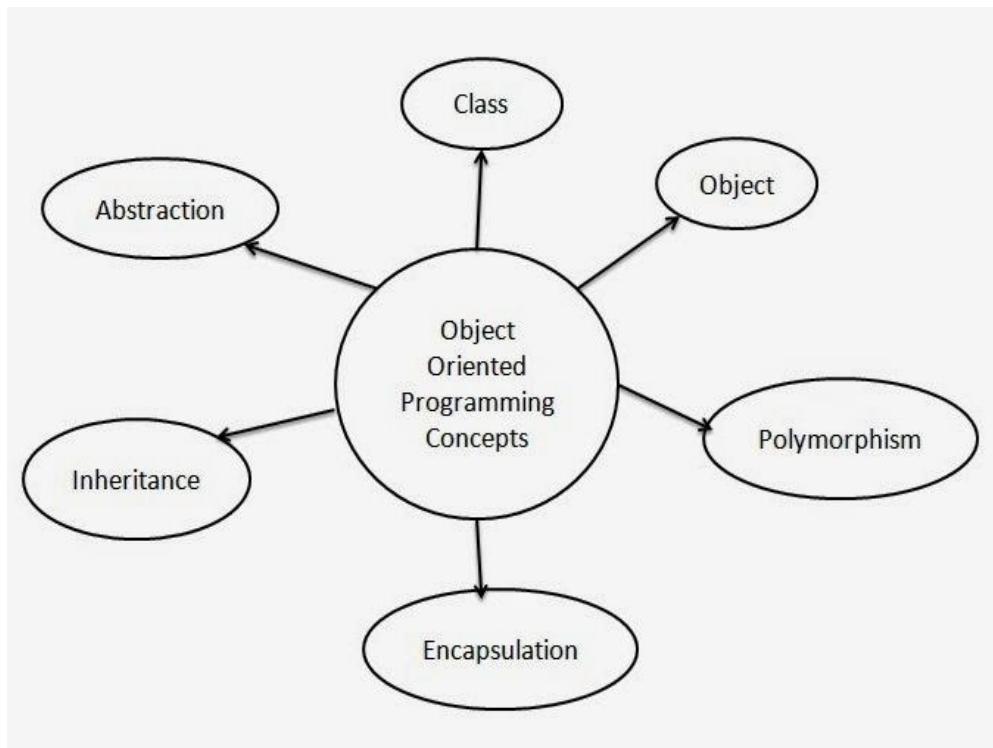
adalah *entitas* yang memiliki atribut, karakter (*bahavour*) dan kadang kala disertai kondisi (*state*) (Douglas, 1992).

Ide dasar pada OOP adalah mengkombinasikan data dan fungsi untuk mengakses data menjadi sebuah kesatuan unit yang dikenal dengan nama objek. Objek adalah struktur data yang terdiri dari bidang data dan metode bersama dengan interaksi mereka untuk merancang aplikasi dan program komputer. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya.

Pemograman berorientasi objek dalam melakukan suatu pemecahan masalah tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Sebagai contoh sebuah departemen yang memiliki seorang manager, sekertaris, petugas administrasi data dan lainnya. Jika manager ingin memperoleh data dari bagian administrasi maka manager tersebut tidak harus mengambilnya tetapi dapat meminta petugas bagian administrasi untuk mengambilnya. Pada kasus tersebut seorang manager tidak harus mengetahui bagaimana cara mengambil data tersebut tetapi manager bisa mendapatkan data tersebut melalui objek petugas administrasi. Jadi untuk menyelesaikan suatu masalah dengan kolaborasi antar objek-objek yang ada karena setiap objek memiliki deskripsi tugasnya sendiri.

Dengan menginstruksikan tugas khusus dan terbatas pada setiap objek, programmer dapat memecahkan masalah pemograman menjadi bagian-bagian kecil sehingga dapat mengatasi kompleksitas permasalahan dengan lebih mudah. Setiap objek umumnya memiliki tiga sifat, yaitu *method*, keadaan operasi, dan identitas. Sebuah objek memiliki perilaku yang disebut *method* dan dapat menyimpan suatu informasi sebagai hasil operasi sebelumnya. Informasi tersebut menentukan bagaimana objek melakukan operasi selanjutnya.

## A Konsep Dasar Pemrograman Berorientasi Objek



Gambar 2.4 Konsep Dasar Pemograman Berorientasi Objek.

### 1. Kelas (*Class*)

Kelas merupakan penggambaran satu set objek yang memiliki atribut yang sama. Kelas mirip dengan tipe data ada pemrograman non objek, akan tetapi lebih komprehensif karena terdapat struktur sekaligus karakteristiknya. Kelas baru dapat dibentuk lebih spesifik dari kelas pada umumnya. Kelas merupakan jantung dalam pemograman berorientasi objek.

### 2. Objek (*Object*)

Objek merupakan teknik dalam mentyelesaikan masalah yang kerap muncul dalam pengembangan perangkat lunak. Teknik ini merupakan teknik yang efektif dalam menemukan cara yang tepat dalam membangun sistem dan menjadi metode yang paling banyak dipakai oleh para pengembang perangkat lunak. Orientasi objek merupakan teknik

pemodelan sistem riil yang berbasis objek. Objek adalah entitas yang memiliki atribut, karakter dan kadang kala disertai kondisi. Objek mempresentasikan sesuai kenyataan seperti siswa, mempresentasikan dalam bentuk konsep seperti merek dagang, juga bisa menyatakan visualisasi seperti bentuk huruf (*font*).

### 3. Abstraksi (*Abstraction*)

Kemampuan sebuah program untuk melewati aspek informasi yang diolah adalah kemampuan untuk fokus pada inti permasalahan. Setiap objek dalam sistem melayani berbagai model dari pelaku abstrak yang dapat melakukan kerja, laporan dan perubahan serta berkomunikasi dengan objek lain dalam sistem, tanpa harus menampakan kelebihan diterapkan.

### 4. Enkapsulasi (Pembungkus)

Enkapsulasi adalah proses memastikan pengguna sebuah objek tidak dapat menggantikan keadaan dari sebuah objek dengan cara yang tidak sesuai prosedur. Artinya, hanya metode yang terdapat dalam objek tersebut yang diberi izin untuk mengakses keadaan yang diinginkan. Setiap objek mengakses *interface* yang menyebutkan bagaimana objek lainnya dapat berintergrasi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

### 5. *Polimorfisme*

*Polimorfisme* merupakan suatu fungsionalitas yang diimplikasikan dengan berbagai cara yang berbeda. Pada program berorientasi objek, pembuat program dapat memiliki berbagai implementasi untuk sebagian fungsi tertentu.

### 6. Inheritas (Pewarisan)

Konsep inheritas mempunyai fungsi mengatur *polimorfisme* dan enkapsulasi dengan mengijinkan objek disefinisikan dan diciptakan dengan jenis khusus dari objek yang sudah ada. Objek-objek ini dapat membagi dan memperluas perilaku mereka tanpa mengimplementasikan perilaku tersebut.

## 2.2.10 Java

Java dikembangkan oleh perusahaan Sun Microsystem. Java menurut definisi Sun Microsystem adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan. Java 2 adalah generasi kedua dari java platform. (Rosa A.S dan M.Shalahddin, 2014:103).

Java merupakan bahasa pemrograman berorientasi objek untuk pengembangan aplikasi mandiri, aplikasi berbasis internet atau jaringan komunikasi. Melalui teknologi java, dimungkinkan perangkat audio stereo dirumah terhubung jaringan komputer. Java tidak lagi hanya untuk jaringan applet yang memerintah halaman web, tapi java telah menjadi bahasa untuk pengembangan aplikasi skala enterprise berbasis jaringan besar. (Bambang Haryanto, 2011:3).

Dari pengertian diatas dapat disimpulkan bahwa Java merupakan bahasa pemrograman berorientasi objek yang digunakan untuk membuat dan menjalankan perangkat lunak pada komputer berbagai platform.

Berikut adalah kelebihan dan kekurangan Java.

Kelebihan:

- Mudah untuk dikembangkan
- Sifatnya multi Platform
- Memiliki kemudahan dalam menyusun script
- Bahasa pemograman yang berorientasi objek
- Berorientasi pada usability
- Dinamis

Kekurangan:

- Penggunaan Memori yang banyak
- Mudah di dekompilasi, maksudnya adalah aplikasi yang sudah jadi dapat dengan mudah dikembalikan menjadi seperti kode awal atau kode sumber.

### 2.2.11 Java Development Kit (JDK)

*Java Development Kit* adalah paket fungsi API untuk bahasa pemrograman java, meliputi *Java Runtime Environment (JRE)* dan *Java Virtual Machine (JVM)*. JDK adalah perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode java ke bytecode yang dapat dimengerti dan dapat dijalankan oleh JRE (*Java Runtime Environment*). JDK wajib terinstal dikomputer yang akan melakukan proses pembuatan aplikasi berbasis java, namun tidak wajib terinstal dikomputer yang akan menjalankan aplikasi yang dibangun dengan java. Jika tidak ada JDK, maka kode-kode java yang sudah dibuat tidak akan bisa dijalankan. Sedangkan JRE merupakan komponen utama yang bertugas untuk menjalankan aplikasi berbasis java. Aplikasi java dapat berjalan disuatu perangkat komputer apabila perangkat komputer sudah terinstal JRE.

### 2.2.12 NetBeans

NetBeans merupakan platform framework dan IDE (*Integrated Development Environment*) berbasiskan Java dari Sun Microsystems yang berjalan diatas swing. Swing merupakan sebuah teknologi Java untuk pengembangan aplikasi desktop yang dapat berjalan pada berbagai macam platform seperti windows, linux, Mac OS X dan Solaris. Sebuah IDE merupakan lingkup pemrograman yang di integrasikan ke dalam suatu aplikasi perangkat lunak yang menyediakan *Grafihc User Interface (GUI)*, suatu kode editor atau text, suatu compiler dan suatu *debugger*.

Netbeans juga digunakan oleh sang programmer untuk menulis, meng-compile, mencari kesalahan dan menyebarkan program netbeans yang ditulis dalam bahasa pemrograman java namun selain itu dapat juga mendukung bahasa pemrograman lainnya dan program ini pun bebas untuk digunakan dan untuk membuat professional *desktop, enterprise, web, and mobile applications* dengan Java language, C/C++, dan bahkan *dynamic languages* seperti PHP, JavaScript, Groovy, dan Ruby.

Platform NetBeans memperbolehkan pembangunan aplikasi dengan menggunakan modul-modul. Aplikasi yang dibangun menggunakan NetBeans dapat dikembangkan oleh pihak ketiga. Platform NetBeans adalah platform yang dapat digunakan ulang (*reusable*) untuk mempermudah pembangunan program menggunakan bahasa Java.

### 2.2.13 *Unified Modeling Language* (UML)

UML adalah singkatan dari “*Unified Modelling Language*” adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, *Object Modeling Technique* (OMT) dan *Object Oriented Software Engineering* (OOSE). Metode Booch dari Grady Booch sangat terkenal dengan nama metode *Design Object Oriented*. Metode ini menjadikan proses analisis dan *design* kedalam empat tahap iteratif, yaitu: identifikasi kelas-kelas dan obyek-obyek, identifikasi semantik dari hubungan obyek dan kelas tersebut, perincian *interface* dan implementasi. Keunggulan metode Booch adalah pada detail dan kayanya dengan notasi dan elemen. Pemodelan OMT yang dikembangkan dalam Rumbaugh didasarkan pada analisis terstruktur dan pemodelan *entity-relationship*. Tahapan utama dalam metodologi ini adalah analisis, *designsystem*, *design* obyek dan implementasi.

Inilah beberapa tujuan atau fungsi dari penggunaan UML diantaranya:

- Dapat memberikan bahasa permodelan visual kepada pengguna dari berbagai macam pemrograman maupun proses rekayasa.
- Dapat menyatukan praktek-praktek terbaik yang ada dalam permodelan.

- Dapat memberikan model yang siap untuk digunakan, merupakan bahasa permodelan visual yang ekspresif untuk mengembangkan sistem dan untuk saling menukar model secara mudah.
- Dapat berguna sebagai blue print, sebab sangat lengkap dan detail dalam perancangannya yang nantinya akan diketahui informasi yang detail mengenai koding suatu program.
- Dapat memodelkan sistem yang berkonsep berorientasi objek, jadi tidak hanya digunakan untuk memodelkan perangkat lunak (*software*) saja.
- Dapat menciptakan suatu bahasa permodelan yang nantinya dapat dipergunakan oleh manusia maupun oleh mesin.

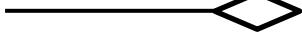
Dalam UML sendiri terdapat beberapa diagram, yaitu:

### 1. Struktur Diagram

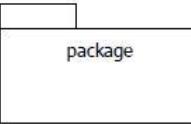
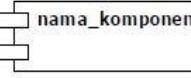
- a. *Class Diagram* diagram statis. Menggambarkan atribut, *operation* dan juga *constraint* yang terjadi pada sistem. *Class diagram* menunjukkan koleksi class, antarmuka, asosiasi, kolaborasi, dan *constraint*.

Tabel 2.1 Simbol-Simbol *Class Diagram*.

Simbol	Deskripsi
Asosiasi/association _____	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi berarah / directed association _____→	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Generalisasi _____→	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
kebergantungan/ dependency _____→	Relasi antar kelas dengan makna kebergantungan antar kelas

Agregasi/aggregation 	Relasi antar kelas dengan makna semua-bagian ( <i>whole-part</i> )
---	--

- b. *Component Diagram* adalah bagian fisik dari sebuah sistem, karena menetap dikomputer, bukan dibenak para analisis. Komponen bisa berupa tabel, file data, file exe dan dokumen.

Simbol	Deskripsi
<b>Package</b>  package	package merupakan sebuah bungkus dari satu atau lebih komponen
<b>Komponen</b>  nama_komponen	Komponen sistem
<b>Kebergantungan / dependency</b> 	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai
<b>Antarmuka / interface</b>	sama dengan konsep <i>interface</i> pada pemrograman berorientasi objek, yaitu sebagai antarmuka komponen

Gambar 2.5 Simbol-Simbol *Componen Diagram*.

- c. *Deployment Diagram* menunjukkan tata letak sebuah sistem secara fisik, menampilkan bagian-bagian software yang berjalan pada bagian-bagian hardware.
- d. *Object Diagram* adalah gambaran oyek-obyek secara ringkas di sebuah sistem pada suatu waktu, digunakan untuk menunjukan contoh konfigurasi dari oyek-obyek.

Simbol	Deskripsi
	objek dari kelas yang berjalan saat sistem dijalankan
	relasi antar objek

Gambar 2.6 Simbol-Simbol *Object Diagram*

## 2. Behavioral Diagram

- a. *Use Case Diagram* digunakan untuk menggambarkan analisis kebutuhan dari sistem dari level atas melalui fungsionalitas dari sistem dan interaksi diantara para *actor*. *Actor* adalah sesuatu yang berinteraksi dengan sistem.

Tabel 2.2 Simbol-Simbol *Use Case Diagram*.

Simbol	Deskripsi
	<i>Actor</i> : Mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem.
	<i>Use Case</i> : Adalah gambaran fungsionalitas dari suatu sistem, sehingga <i>customer</i> atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.
	<i>Association</i> : Menghubungkan link antar element.
	<<Includes>> : Yaitu kelakuan yang harus terpenuhi agar sebuah <i>event</i> dapat terjadi, dimana pada kondisi ini sebuah <i>use case</i> adalah bagian dari <i>use case</i> lainnya

- b. *Activity Diagram* adalah diagram yang menggabarkan aspek dinamis dari sistem. Menangkap tingkah laku dinamis dari sistem dengan cara menunjukkan aliran pesan dari suatu aktifitas ke aktifitas lainnya.

Tabel 2.3 Simbol-Simbol *Activity Diagram*.

No	Simbol	Nama	Deskripsi
1.		Action State	Menggambarkan keadaan dari suatu elemen dalam suatu aliran aktifitas
2.		State	Menggambarkan kondisi suatu
3.		Flow Control	Menggambarkan aliran aktifitas dari suatu elemen ke elemen lain
4.		Initial State	Menggambarkan titik awal siklus hidup suatu elemen
5.		Final State	Menggambarkan titik akhir yang menjadi kondisi akhir suatu elemen

- c. *State Machine Diagram* menelusuri individu-individu objek melalui keseluruhan daur hidupnya, menspesifikasikan semua urutan yang mungkin dari pesan-pesan yang akan diterima obyek tersebut, bersama-sama dengan tanggapan atas pesan-pesan tersebut.

Notasi	Penjelasan
	State, digambarkan berbentuk segi empat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu
	awal (start), digunakan untuk menggambarkan awal dari kejadian dalam suatu diagram statechart
	Titik akhir (end), digunakan untuk menggambarkan akhir dari kejadian dalam suatu diagram statechart
[guard]	Guard, yang merupakan syarat terjadinya transisi yang bersangkutan
	Point, digunakan untuk menggambarkan apakah akan masuk (entry point) ke dalam state atau akan keluar (exit point)
<i>event</i>	Event, digunakan untuk mendeskripsikan kondisi yang menyebabkan sesuatu pada state

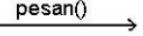
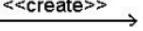
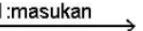
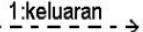
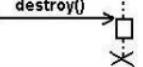
Gambar 2.7 Simbol-Simbol *State Machine Diagram*.

#### d. *Interaction Diagram*

- 1). *Sequence Diagram* digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukkan sejumlah contoh obyek dan pesan yang diletakan diantara obyek-obyek ini didalam *use case*.

Tabel 2.4 Simbol-Simbol *Sequenci Diagram*.

Simbol	Nama	Keterangan
	Aktor / Actor	<ul style="list-style-type: none"> <li>• Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi dan mendapat manfaat dari sistem.</li> <li>• Berpartisipasi secara berurutan dengan mengirimkan atau mengirimkan pesan.</li> <li>• Di tempatkan di bagian atas diagram.</li> </ul>
	Objek / Object	<ul style="list-style-type: none"> <li>• Berpartisipasi secara berurutan dengan mengirimkan atau menerima pesan.</li> <li>• Di tempatkan di bagian atas diagram</li> </ul>
	Garis Hidup / Lifeline	<ul style="list-style-type: none"> <li>• Menandakan kehidupan objek selama urutan.</li> <li>• Diakhiri tanda X pada titik di mana kelas tidak lagi berinteraksi.</li> </ul>

	Objek yang berinteraksi / <i>Activation</i>	<ul style="list-style-type: none"> <li>• Fokus kontrol adalah persegi panjang yang sempit panjang ditempatkan di atas sebuah garis hidup.</li> <li>• Menandakan ketika suatu objek mengirim atau menerima pesan.</li> </ul>
	Pesan / <i>Message</i>	<ul style="list-style-type: none"> <li>• Objek mengirim satu pesan ke objek lainnya.</li> </ul>
	Membuat / <i>Create</i>	<ul style="list-style-type: none"> <li>• Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</li> </ul>
	Masukkan / <i>Message Send</i>	<ul style="list-style-type: none"> <li>• Menyatakan bahwa suatu objek mengirim masukkan ke objek lainnya arah panah megarah pada objek yang dikirim.</li> </ul>
	Keluaran / <i>Message Return</i>	<ul style="list-style-type: none"> <li>• Objek atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</li> </ul>
	Hasil Akhir / <i>Destroy</i>	<ul style="list-style-type: none"> <li>• Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy.</li> </ul>

- 2). *Communication Diagram* adalah perluasan dari obyek diagram yang menunjukkan message-message obyek-obyek yang dikirimkan satu sama lain.
- 3). *Interaction Overview Diagram* adalah pencakokan secara bersama antara *activity* dengan *sequence diagram*.
- 4). *Timing Diagram* bisa digunakan untuk menampilkan perubahan status atau nilai dari satu atau lebih elemen dari waktu ke waktu, juga dapat digunakan untuk menunjukkan interaksi dari aktu ke waktu pada suatu durasi tertentu.

### 2.2.14 Javafx Scene Builder

Javafx Scene Builder atau lebih singkatnya Screen Builder adalah Tools yang digunakan untuk membuat desain *User Interface* dari sebuah aplikasi. JavaFX adalah penerus dari java swing, digunakan untuk membuat aplikasi java berbasis *Grafical User Interface* (GUI). Scene builder dapat membuat file fxml secara otomatis. Seperti yang kita ketahui, fxml adalah file yang berformat xml, yang isinya adalah layout dari aplikasi javafx. (Sigit Kurniawan, 2014, JavaFX Scene Builder, <https://sigitkurniawanisme.wordpress.com/2015/01/02/javafx-scene-builder/>)

JavaFX Scene Builder adalah alat tata letak visual yang memungkinkan pengguna dengan cepat merancang antarmuka pengguna aplikasi JavaFX, tanpa coding. Pengguna dapat menarik dan melepas komponen UI ke area kerja, memodifikasi propertinya, menerapkan *style sheet*, dan kode FXML untuk tata letak yang mereka buat secara otomatis dihasilkan di latar belakang. Hasilnya adalah file FXML yang kemudian dapat dikombinasikan dengan proyek Java dengan mengikat UI ke logika aplikasi.

### 2.2.15 Model-Driven Development (MDD)

*Model-Driven Development* (MDD) adalah sebuah paradigma untuk menulis dan mengimplementasikan program komputer secara tepat, efektif dan berbiaya minimum. Pendekatan MDD untuk pengembangan perangkat lunak memungkinkan orang untuk bekerja sama dalam sebuah proyek bahkan dengan tingkat pengalaman masing-masing yang sangat bervariasi. Hal ini memungkinkan perusahaan untuk memaksimalkan kerja yang efektif pada sebuah proyek dan meminimalkan overhead yang diperlukan untuk menghasilkan perangkat lunak yang dapat divalidasi oleh pengguna akhir dalam waktu sesingkat mungkin. MDD merupakan metodologi tangkas (*agile*), terus berkembang untuk memenuhi kebutuhan bisnis.

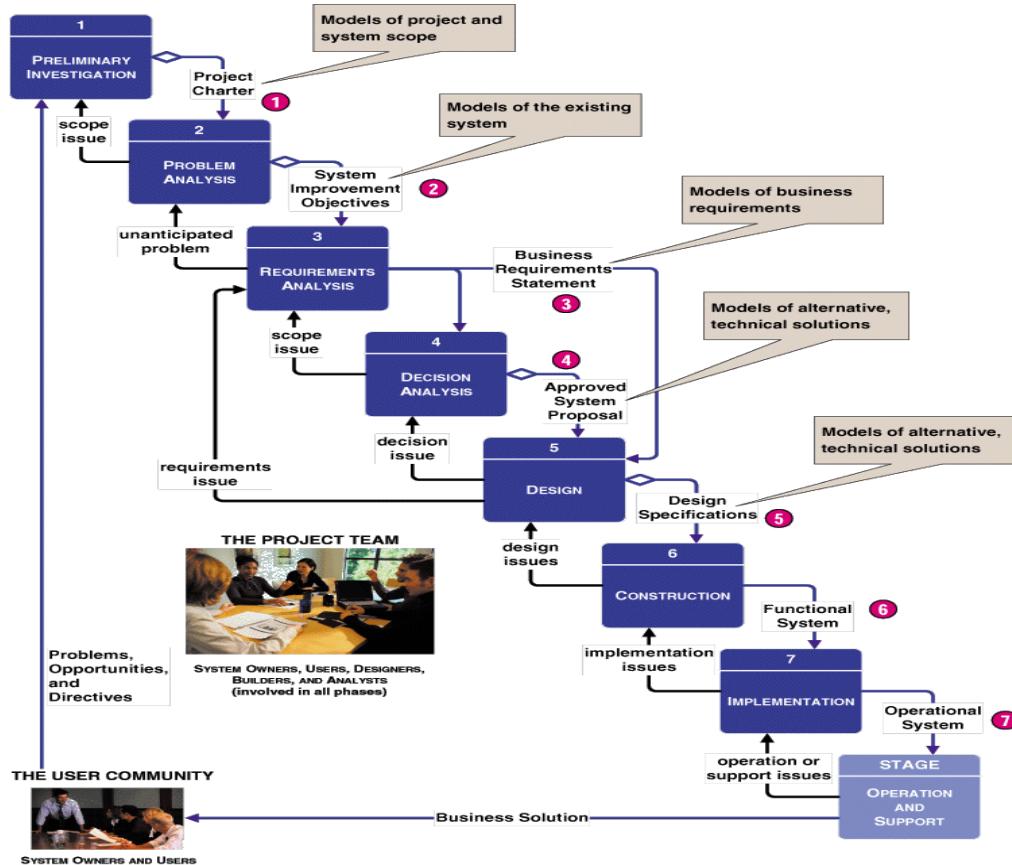
MDD merupakan pemikiran bagaimana membawa pengembangan perangkat lunak sebagai sebuah rangkaian dari transformasi model yang berawal

dari kebutuhan prasyarat (*requirements*) menuju sebuah model yang independen dan spesifik, dan kemudian membangkitkan kode sumber (*code*) yang dapat dikompilasi dalam sistem komputer. Oleh karena itu pengembangannya berorientasi *model-centric* dan banyak aktifitas, meliputi analisis perancangan awal dan pembangkitan kasus uji (*test case*), yang semuanya akan didasarkan pada model menggunakan UML.

Pengembangan sistem komputer menggunakan pendekatan MDD membutuhkan beberapa perkakas pengembangan sesuai tahapan. Pada tahap analisis dan perancangan, dibutuhkan perkakas seperti Visual Paradigm yang mampu menggambar model dalam notasi UML. Pada tahap implementasi (pemograman) dibutuhkan perkakas pengembangan seperti *Play Framework*, *Yii Framework*. Untuk lebih menyempurnakan pengembangan perangkat lunak dibutuhkan perkakas *Integrated Development Environment* (IDE) seperti *Eclipse*.

Pembuat model akan dengan cepat mengetahui apakah model yang dikembangkannya layak diimplementasikan melalui konsep *Forward Engineering*, arah maju dari model menuju kode sumber, dan sebaliknya pekerjaan pembuat kode sumber yaitu programmer akan dapat diperiksa dengan cepat oleh pembuat model apakah sesuai atau tidak dengan model yang dikembangkan melalui *reverse engineering*, arah balik balik dari kode sumber menuju model. Programmer bekerja dengan mudah karena sudah ada penuntun pengembangan program berupa nama fungsi/prosedur dalam kesatuan model yang memuat hubungan model beserta spesifikasinya. Pekerjaan programer mendetailkan dari spesifikasi menjadi kode sumber yang yang dapat dibaca oleh komputer.

Pengembangan system komputer menggunakan MDD dan dibantu oleh perkakas memberikan kemudahan proses *forward* dan *reverse engineering*. Kebutuhan baru pengembangan system mudah untuk diterapkan, perubahan kode sumber dapat ditelusuri melalui model. Semua langkah dilaksanakan terkomputerisasi dan tidak ada keadaan magic untuk menciptakan kode sumber, berbeda halnya dengan pengembangan pendekatan konvensional seperti pengembangan berorientasi fungsi (Grady Booch, 1998).



Gambar 2.8 *Model-Driven Development (MDD)*.

### 1. *Preliminary Investigation*

Pada tahap ini, penulis memulai pekerjaan dengan mendeskripsikan dan mengumpulkan semua data-data yang dibutuhkan dalam merancang suatu sistem yang dibutuhkan di tempat penelitian.

### 2. *Problem Analysis*

Pada tahap ini, penulis menganalisa masalah – masalah yang terdapat di lapangan dengan melakukan analisi terhadap sistem yang sedang berjalan.

### 3. *Requirements Analysis*

Pada tahap ini akan dilakukan pengurutan prioritas pada kebutuhan-kebutuhan yang ada. Tujuannya adalah mengidentifikasi data, proses dan antarmuka yang diinginkan pengguna dari sistem yang baru.

#### ***4. Design***

Pada tahap ini, dilakukan perancangan sistem yang bertujuan untuk memberikan gambaran apa yang seharusnya di kerjakan serta bagaimana tampilanya. Design tersebut meliputi rancangan output, rancangan input, rancangan struktur data yang digunakan, rancangan model dan rancangan lainnya. Tahapan ini membantu dalam menspesifikasikan kebutuhan dan arsitektur sistem secara keseluruhan.

#### ***5. Construction***

Pada tahapan ini akan dimulai untuk mengkonstruksi dan melakukan tahap uji coba terhadap sistem yang memenuhi kebutuhan-kebutuhan dan spesifikasi desain. Basis data, program aplikasi, dan antarmuka akan mulai dibangun pada tahap ini. Setelah dilakukan uji coba terhadap keseluruhan sistem, maka sistem siap untuk diimplementasikan.

#### ***6. Implementation***

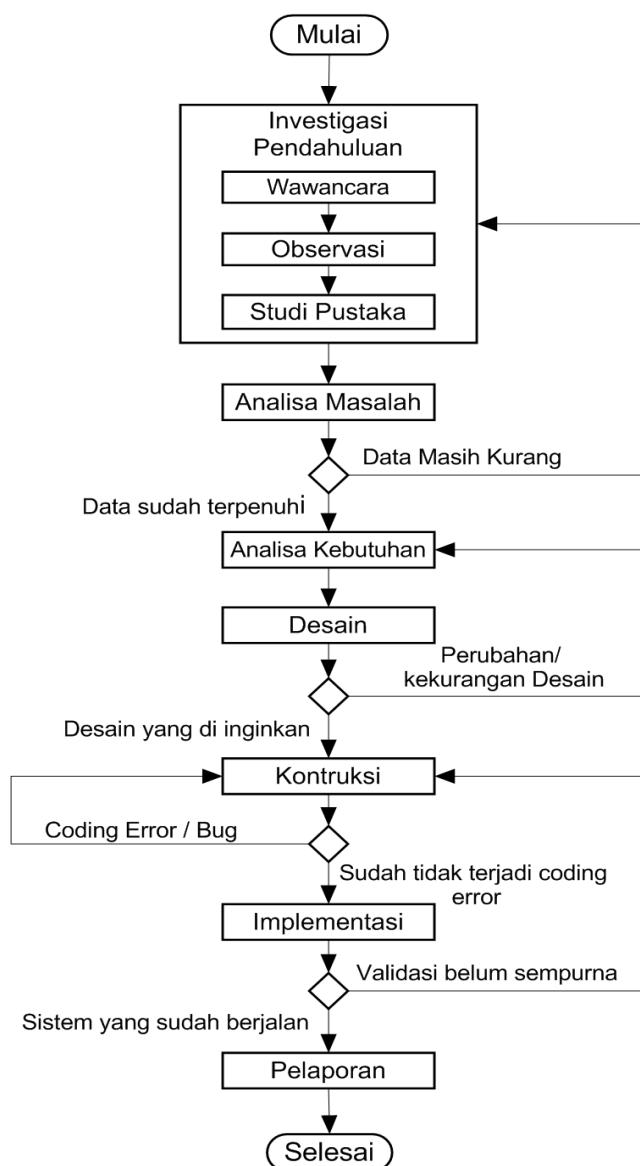
Tahap ini menerapkan hasil rancangan yang telah disusun sedemikian rupa ke dalam sistem untuk mendapatkan kondisi yang sesuai dengan kebutuhan, input dari tahap ini adalah sistem fungsional dari tahap kontruksi.

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Kerangka Berfikir

Kerangka berfikir pada perancangan aplikasi penjadwalan kuliah sebagai pedoman penulis untuk mencapai tujuan dari penelitian yang telah ditentukan sebelumnya adalah sebagai berikut.



Gambar 3.1 Kerangka Berfikir.

## 3.2 Penjelasan Skema Penelitian

### 3.2.1 Investigasi Pendahuluan

Penulis memulai pekerjaan dengan mengumpulkan semua data-data yang dibutuhkan untuk penelitian ini dengan mengunjungi Fakultas Teknologi Informasi Universitas Bale Bandung (UNIBBA) pada hari Sabtu tanggal 25 Mei 2019 untuk menemui staff TU yaitu Bapak Yusuf Muharom, S.Kom. Adapun langkah –langkah dalam mengumpulkan data adalah sebagai berikut:

#### 1. Wawancara

Wawancara dilakukan langsung dengan pihak-pihak terkait dalam pembuatan aplikasi penjadwalan kuliah yaitu Bapak Yusuf Muharom, S.Kom, sebagai staff TU di Fakultas Teknologi Informasi yang mengatur jadwal perkuliahan di Program Studi Teknik Informatika dan Sistem informasi. Dalam melakukan wawancara penulis mengajukan beberapa pertanyaan diantaranya sebagai berikut:

- a. Bagaimana sistem penjadwalan kuliah yang sedang berjalan di Fakultas Teknologi Informasi?

Jawaban :

Penjadwalan perkuliahan di Fakultas Teknologi Informasi masih manual dengan menggunakan *Microsoft Excel*.

- b. Apakah sistem penjadwalan di Fakultas Teknologi Informasi terintegrasi dengan perwalian?

Jawaban:

Penjadwalan perkuliahan di Fakultas Teknologi Informasi yang saat ini sedang berjalan hanya sebatas penyusunan jadwal mata kuliah dan dosen pengajar.

- c. Apa masalah-masalah yang sering muncul dalam proses menyusun jadwal perkuliahan?

Jawaban:

Masalah yang sering kali muncul adalah setiap jadwal yang dihasilkan sering kali terjadinya bentrok, baik dari dosen pengajar dan waktu mengajar mata kuliah.

- d. Sistem penjadwalan yang ada dan sedang berjalan sudah cukup efektif?

Jawaban:

Belum, dikarenakan kesulitan dalam menyusun dan permasalahan yang sering terjadi, sehingga jadwal yang dihasilkan tidak sesuai dengan keinginan dosen.

## 2. Obsevasi

Obsevasi dilakukan langsung ke Fakultas Teknologi Informasi Universitas Bale Bandung (UNIBBA)divisi akademik guna mengetahui bagaimana jalannya penjadwalan yang sedang berjalan saat ini. Kemudian penulis mencari masalah-masalah pada sistem yang berjalan, sehingga dapat dianalisis untuk pengembangan sistem yang akan dibangun guna membantu dalam proses penjadwalan kuliah yang di harapkan.

## 3. Studi Pustaka

Mempelajari teori-teori dan hasil penelitian sebelumnya yang mendukung pemecahan masalah bagi penelitian ini yang terdiri dari buku-buku digital (*ebook*), jurnal dan beberapa referensi dari website yang berkaitan dengan masalah-masalah yang ada pada sistem penjadwalan yang sedang berjalan di Fakultas Teknologi Informasi. Berikut adalah judul dari referensi:

- a. Penerapan Metode Algoritma *Hill Climbing* Pada Sistem Informasi Geografis Untuk Mencari Lintasan Terpendek (Eka Vickraien, Vincencius, Kusworo Adi,2015,Jurnal Sistem Informasi Bisnis).
- b. Pengembangan Sistem Penjadwalan Kuliah Menggunakan Algoritma Steepest Ascen Hill Climbing (Soffan Saifullah, Arief Hermawan,2016,Jurnal Sistem Komputer-Voll 6, No.2).
- c. Penjadwalan Kuliah dengan menggunakan Algoritma Hill Climbing (Aida Putri Deswanti, Aries Saifudin,2016,Prosiding Seminar Nasional Informatika dan Sistem Informasi, Vol-1, No.3).

- d. Buku dengan judul “Penyelesaian Masalah Optimasi Dengan Teknik-Teknik Heuristik” karangan Sri Kusumadewi dan Hari Purnomo.

### **3.2.2 Analisis Masalah**

Hasil dari analisis masalah yang ada pada penyusunan jadwal kuliah di Fakultas Teknologi Informasi. Fakta-fakta yang sudah ditemukan diantaranya:

- 1). Staff TU mengalami kesulitan saat menyusun jadwal perkuliahan, sehingga menyebabkan keterlambatan dan kekeliruan.
- 2). Bentrok jadwal yang terjadi pada saat menyusun jadwal perkuliahan.

Berdasarkan permasalahan tersebut, penulis melakukan perbandingan dengan data yang telah didapat pada proses pengumpulan data dan melakukan studi banding dengan penelitian-penelitian sebelumnya, maka dapat disimpulkan bahwa dalam melakukan peroses penjadwalan dibutuhkan algoritma pencarian dalam mencari bentrok pada data jadwal perkuliahan, untuk itu penulis memilih menerapkan algoritma pencarian *steepest-ascent hill climbing* sebagai solusi untuk permasalahan yang ada pada sistem yang sedang berjalan.

### **3.2.3 Analisa Kebutuhan**

Adapun analisis kebutuhan yang diperlukan untuk menjawab pertanyaan dan permasalahan yang ada terdiri dari 4, yaitu:

#### **1. Kebutuhan Prosedur**

Prosedur yang dibutuhkan yaitu mengumpulkan data-data yang akan digunakan. Berikut data-data yang dibutuhkan dalam perancangan aplikasi jadwal perkuliahan:

- a. Data dosen
- b. Data mahasiswa aktif
- c. Data matakuliah

- d. Data ruang kelas
  - e. Data kesediaan dosen mengajar.
2. Aplikasi Utama
    - a. Membuat aplikasi penjadwalan kuliah berbasis dekstop
    - b. Pada aplikasi yang dibuat akan menerapkan algoritma *steepest-ascent hill climbing* untuk mencari data jadwal mengajar yang bentrok.
  3. Kebutuhan Perangkat Lunak

Adapun perangkat lunak (*Software*) yang dibutuhkan pada penelitian ini sebagai berikut:

- a. MySQL, digunakan untuk menyimpan *database* dari setiap data yang dibutuhkan.
  - b. Java, sebagai bahasa pemrograman yang akan digunakan untuk membuat aplikasi.
  - c. XAMPP, digunakan sebagai control panel.
  - d. NetBeans IDE, digunakan untuk menjalankan Java sebagai pengembangan aplikasi.
  - e. Scene Buider, tool yang digunakan untuk perancangan antar muka aplikasi.
  - f. Star UML, digunakan untuk membuat diagram-diagram UML.
  - g. Balsamiq Mockups, digunakan untuk mendesain antar muka aplikasi.
  - h. Microsoft Word, digunakan untuk membuat laporan penelitian.
4. Kebutuhan Perangkat Keras

Adapun kebutuhan minimum perangkat keras (*hardware*) dalam pembuatan aplikasi penjadwalan dengan spesifikasi minimal pada perangkat keras adalah sebagai berikut:

- a. Processor Pentium(R) Dual-Core CPU T4200 @ 2.00GHz (2 CPUs),  
~ 2.0GHz
- b. RAM 4 GB
- c. HDD 250 GB
- d. *Printer* dan *Paper*

### 3.2.4 Desain

Pada tahap ini penulis membuat desain aplikasi penjadwalan yang akan dibangun dari hasil analisis yang telah dilakukan. Ada beberapa proses tahapan yang akan dilakukan diantaranya sebagai berikut :

1. Perancangan Model

Pada perancangan model dilakukan untuk mengetahui alur yang akan dibuat untuk membangun aplikasi penjadwalan kuliah. Perancangan akan dibuat dengan *Unified Modeling Language* (UML) agar dengan mudah dalam proses pengembangan dan visualisasinya. Diagram UML yang digunakan antara lain:

- a. *Use Case Diagram* digunakan untuk menggambarkan keterhubungan *actor* dan *use case* dalam aplikasi yang akan dibuat. *Actor* dan *use case* yang terlibat pada penelitian ini adalah:

- 1) *Actor*

Staff TU, Dosen dan mahasiswa

- 2) *Use case*

Kesediaan mengajar, penjadwalan kuliah, pengelolaan kelas, pengelolaan data dosen, pengelolaan mahasiswa program studi semester, pengelolaan matakuliah, pengelolaan ruang, pengelolaan waku, *view jadwal*.

- b. *Class Diagram* digunakan untuk menggambarkan atribut, operation dan juga constraint pada sistem yang akan dibuat. *Class diagram* yang akan dibuat adalah sebagai berikut:

- 1). *Class diagram* dosen

- 2). *Class diagram* matakuliah

- 3). *Class diagram* mahasiswa

- 4). *Class diagram* Kesediaan Mengajar

- 5). *Class diagram* Kelas

- 6). *Class diagram* penjadwalan

- c. *Activity Diagram* digunakan untuk memodelkan alur kerja dari sistem dan aktivitas dari actor dalam aplikasi. Adapun diagram alur yang akan dibuat diantaranya:
  - 1). Proses *login*
  - 2). Proses input data
  - 3). Proses simpan data
  - 4). Proses hapus data
  - 5). Proses edit data
  - 6). Proses cetak data
  - 7). Proses optimasi jadwal
- d. *Sequence Diagram* digunakan untuk menunjukkan rangkaian pesan yang dikirim antara objek dengan objek lain. Berikut *sequensi diagram* yang akan dibuat:
  - 1). *Sequensi diagram login*
  - 2). *Sequensi diagram data dosen*
  - 3). *Sequensi diagram data mahasiswa program studi (MPS)*
  - 4). *Sequensi diagram data mata kuliah*
  - 5). *Sequensi diagram kesediaan mengajar*
  - 6). *Sequensi diagram data kelas*
  - 7). *Sequensi diagram data ruangan*
  - 8). *Sequensi diagram data waktu*
  - 9). *Sequensi diagram penjadwalan*

## 2. *Database*

Data adalah salah satu bagian yang paling penting dalam membangun aplikasi jadwal perkuliahan. Pengelolaan data juga menjadi bagian paling utama dalam penelitian ini. Dalam perancangan *database*, data yang dibutuhkan didapatkan dari pengumpulan data sebelumnya. Beberapa data yang digunakan dalam penelitian ini adalah:

- 1) Data dosen
- 2) Data matakuliah
- 3) Data mahasiswa program studi
- 4) Data kesediaan mengajar dosen

- 5) Data ruangan/kelas
- 6) Data waktu
- 7) Data kelas
- 8) Data jadwal
- 9) Data user

### 3. Perancangan Antar Muka (*User Interface*)

Perancangan antar muka adalah perancangan tahap akhir yang akan dibuat. Adapun perancangan antar muka yang akan dibuat sebagai berikut:

- a. *Form* login
- b. *Form* halaman menu utama
- c. *Form* halaman dosen
- d. *Form* halaman mahasiswa
- e. *Form* halaman mata kuliah
- f. *Form* halaman kesediaan mengajar
- g. *Form* halaman ruang
- h. *Form* halaman waktu
- i. *Form* halaman kelas
- j. *Form* halaman jadwal.
- k. *Form* halaman *user*.

#### 3.2.5 Kontruksi

Tahap ini akan dimulai untuk membangun dan melakukan pengujian pada program yang akan dibuat dengan kebutuhan-kebutuhan dan spesifikasi desain. Basis data, program aplikasi dan antar muka akan dibangun pada tahap ini. Ada beberapa langkah pada pembuatan aplikasi penjadwalan kuliah ini yaitu:

- a. Membuat *database* di MySQL untuk membuat tabel yang dibutuhkan dan relasi antar tabel.
- b. Pada pembuatan program aplikasi dengan Java ada beberapa tahapan yang harus dilakukan:

- c. Pada program Java dibuat Java Package dengan nama “koneksi” untuk menghubungkan Java dengan database MySQL menggunakan XAMPP sebagai control panel.
- d. MySQL JDBC Driver digunakan sebagai jembatan penghubung Java dengan MySQL.
- e. Jfoeniq digunakan sebagai jembatan penghubung java dengan Scene Builder
- f. Visual interface pada aplikasi penjadwalan kuliah dibuat menggunakan Scene Builder
- g. Menampilkan database dengan menggunakan TabelGrid di Scene Bulder. Database dipanggil dengan kode program pada Java class koneksi.
- h. Dalam menerapkan algoritma *steepest-ascent hill climbing* di buat class pada Java.
- i. Menguji program yang sudah dibuat, jika masih terdapat error atau bug, maka dilakukan perbaikan pada *source code*.

### 3.2.6 Implementasi

Pada tahap implementasi ini dilakukan pengujian dan penerapan pada aplikasi penjadwalan baru yang sudah dirancang dan berhasil dibangun, yang nantinya akan mengganti sistem yang sedang berjalan saat ini. Pengujian dilakukan dengan menggunakan metode *black box* untuk menguji fungsionalitas aplikasi. Jika pada aplikasi yang telah dibuat masih terdapat kekurangan dan terdapat beberapa fungsi sistem yang masih kurang berjalan dengan baik, maka akan dilakukan kembali ke kontruksi sampai mendapat hasil yang sesuai dengan yang diharapkan, kemudian baru dilanjutkan ke tahapan selanjutnya.

### 3.2.7 Pembuatan Laporan

Pada tahap ini merupakan tahapan terakhir dengan melakukan penulisan dalam bentuk laporan untuk mempertanggung jawabkan hasil dari penelitian yang

telah dilakukan di Fakultas Teknologi informasi Universitas Bale Bandung (UNIBBA). Laporan yang dibuat dituangkan ke dalam bentuk skripsi yang nantinya akan di uji dalam sidang skripsi sebagai syarat memperoleh gelar sarjana Strata 1 (S1) Teknik Informatika.

## BAB I PENDAHULUAN

Pada bab ini dijelaskan beberapa hal yang mendasari atau yang melatar belakangi penulisan tugas akhir ini. Di dalam bab ini dideskripsikan tentang latar belakang, rumusan masalah, batasan, tujuan, metodologi penelitian, dan sistematika penulisan.

## BAB II TINJAUAN PUSTAKA

Pada bab ini berisi tentang landasan teori yang mendasari pembahasan secara detail dan digunakan sebagai dasar untuk menganalisis, turunan hasil-hasil penelitian terdahulu, dan dasar teori yang menjelaskan teori-teori yang relevan dengan kegiatan obsevasi/dilakukan atau masalah yang diteliti.

## BAB III METODOLOGI

Bab ini menjelaskan secara singkat metode yang digunakan untuk memperoleh data yang diperlukan atas permasalahan yang ada terdiri dari kerangka pemikiran, sumber data dan jenis data serta metode analisis data.

## BAB IV ANALISIS DAN PERANCANGAN

Pada bab ini akan dibahas mengenai analisis dan perancangan sistem aplikasi yang akan dibuat. Pokok bahasan yang terdapat dalam bab ini adalah deskripsi produk, kebutuhan khusus, kebutuhan fungsionalitas, dan perancangan arsitektur dari aplikasi penjadwalan perkuliahan.

## BAB V IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan dilakukan implementasi dan pengujian terhadap sistem. Tahapan ini dilakukan setelah perancangan selesai dilakukan dan selanjutnya akan di implementasikan pada bahasa pemrograman. Setelah di implementasikan maka

dilakukan pengujian terhadap sistem dan dilihat kekurangan-kekurangan pada aplikasi untuk pengembangan sistem selanjutnya.

## BAB VI KESIMPULAN DAN SARAN

Bab ini merupakan bagian akhir dari laporan penelitian yang berisi mengenai kesimpulan dan saran yang dibutuhkan terhadap hasil akhir dari penelitian yang telah dibuat.

## **BAB IV**

### **ANALISIS DAN PERANCANGAN**

#### **4.1 Analisis**

##### **4.1.1 Analisis Masalah**

Pada analisis masalah penulis melakukan analisis pada data jadwal perkuliahan semester ganjil pada tahun 2015 2017 dan 2018. Berdasarkan data yang ada kemudian jadwal disusun ulang berdasarkan tabel berikut ini:

- a. Data jadwal Program Studi Teknik Informatika dan Sistem Informasi semester ganjil pada tahun 2015 dapat di lihat pada Tabel 4.1.

Tabel 4.1 Tabel Jadwal Semester Ganjil 2015

HARI	KR	PS	SEM	KDMK	SKS	MULAI	SELESAI	NIDN
SENIN	R04	IF	1	FTI212	2	11	13	0429848950
SENIN	R01	SI	1	FTI212	2	13	15	0429848950
SENIN	R13	IF	3	IFPRK09	2	13	15	0418459074
SENIN	R01	IF	1	TIF202	2	09	11	0498347997
SENIN	R01	IF	3	FTI309	2	09	11	0459023522
SENIN	R14	IF	5	IFPRK11	2	13	15	0489657845
SENIN	R03	IF	5	FTI325	3	10	13	0489657845
SELASA	R03	SI	3	TIF324	3	13	16	0401988974
SELASA	R04	IF	5	FTI323	3	09	12	0401988974
SELASA	R03	IF	1	FTI210	3	14	17	0420988987
SELASA	R05	SI	1	FTI210	3	09	12	0420988987
SELASA	R02	IF	3	FTI324	3	09	12	0417768389
SELASA	R09	SI	7	SIF336	3	13	16	0419045901
SELASA	R06	IF	7	FTI337	3	09	12	0419045901
SELASA	R03	SI	3	TIF304	3	09	12	0416118901
SELASA	R02	SI	5	TIF317	3	09	12	0416118901
SELASA	R05	IF	5	FTI316	3	13	16	0412342356
SELASA	R12	SI	5	SIF341	3	09	12	0425038203
SELASA	R14	IF	1	IFPRK02	2	12	14	0478658342
RABU	R01	IF	1	TIF301	3	09	12	0402108204
RABU	R09	SI	7	UBB103	2	14	16	0420984578
RABU	R02	IF	7	UBB103	2	10	12	0420984578

RABU	R01	SI	5	TIF323	3	12	15	0401237635
RABU	R05	IF	1	UBB102	2	12	14	0416003733
RABU	R11	SI	1	UBB102	3	9	12	0416003733
RABU	R06	IF	5	FTI311	3	13	16	0429085487
RABU	R06	SI	3	SIF346	3	9	12	0429085487
RABU	R10	SI	5	SIF343	3	9	12	0412081963
RABU	R13	IF	5	IFPRK07	2	9	11	0459023522
RABU	R14	SI	3	IFPRK07	2	15	17	0459023522
RABU	R02	SI	5	TIF320	3	12	15	0478657346
RABU	R14	IF	3	IFPRK10	2	13	15	0489657845
RABU	R04	IF	3	FTI305	3	9	12	0489657845
RABU	R14	SI	1	IFPRK02	2	12	14	0478658342
RABU	R03	IF	7	FTI330	3	13	16	0422037002
KAMIS	R05	SI	7	SIF330	3	13	16	0419834800
KAMIS	R08	SI	3	TIF327	3	9	12	0420988987
KAMIS	R03	IF	3	FTI327	3	13	16	0402108204
KAMIS	R13	SI	5	IFPRK03	2	15	17	0401237635
KAMIS	R14	IF	7	IFPRK03	2	13	15	0401237635
KAMIS	R05	IF	7	TIF323	3	9	12	0401237635
KAMIS	R14	SI	3	IFPRK06	2	13	15	0409107503
KAMIS	R07	SI	3	AKU315	2	11	13	0409107503
KAMIS	R01	IF	3	FTI308	2	10	12	0498347997
KAMIS	R04	SI	7	SIF331	3	9	12	0412081963
KAMIS	R02	IF	5	FTI328	2	10	12	0428027501
JUMAT	R03	SI	1	TIF301	3	9	12	0402108204
JUMAT	R13	SI	3	IFPRK05	2	15	17	0401089734
JUMAT	R10	SI	3	FTI207	2	13	15	0401089734
JUMAT	R06	IF	3	FTI207	2	9	11	0401089734
JUMAT	R14	IF	5	IFPRK08	2	13	15	0419045901
JUMAT	R02	IF	5	FTI317	3	9	12	0419045901
JUMAT	R08	SI	1	SIF303	2	14	16	0428978922
JUMAT	R05	IF	3	FTI304	3	13	16	0498347997
JUMAT	R03	SI	7	FTI214	2	14	16	0408927573
JUMAT	R01	IF	7	FTI214	2	10	12	0408927573
JUMAT	R12	SI	7	UBB104	2	9	11	0489798754
JUMAT	R03	IF	7	UBB104	2	9	11	0428027501
SABTU	R04	SI	3	FTI307	3	13	16	0426028503
SABTU	R07	SI	5	SIF335	3	10	13	0426028503
SABTU	R02	IF	3	FTI307	2	9	11	0426028503
JUMAT	R13	SI	5	IFPRK04	2	10	12	0401057504
JUMAT	R07	SI	5	SIF326	3	13	16	0401057504
SABTU	R13	IF	1	IFPRK01	2	11	13	0401057504
SABTU	R01	IF	1	FTI204	2	9	11	0401057504

SABTU	R14	SI	1	IFPRK01	2	15	17	0401057504
SABTU	R04	SI	1	FTI204	2	13	15	0401057504
SABTU	R07	SI	1	FTI202	2	9	11	0489798754
SABTU	R06	IF	5	FTI321	3	9	12	0478657346
SABTU	R05	IF	7	FTI315	3	13	16	0478657346
SABTU	R04	IF	1	UBB105	2	15	17	0411354599
SABTU	R05	SI	1	UBB105	2	11	13	0411354599

b. Data jadwal Program Studi Teknik Informatika dan Sistem Informasi semester ganjil pada tahun 2017 dapat di lihat pada Tabel 4.2.

Tabel 4.2 Data Jadwal Semester Ganjil 2017

HARI	KR	PS	SEM	KDMK	SKS	MULAI	SELESAI	NIDN
SENIN	R05	SI	1	TIF301	3	9	12	0402108204
SENIN	R07	IF	1	TIF301	2	9	11	0402108204
SENIN	R06	SI	1	FTI212	2	13	15	0429848950
SENIN	R08	IF	1	FTI212	2	13	15	0429848950
SENIN	R08	SI	3	TIF304	3	14	17	0416118901
SENIN	R07	SI	5	TIF317	3	11	14	0416118901
SENIN	R12	IF	3	FTI308	2	16	18	0412081963
SENIN	R10	IF	5	FTI317	3	9	12	0402035566
SENIN	R13	IF	3	IFPRK10	2	11	13	0489657845
SENIN	R11	IF	3	FTI305	3	9	12	0489657845
SENIN	R09	IF	7	FTI214	2	9	11	0428027501
SELASA	R04	SI	3	FTI307	3	11	14	0426028503
SELASA	R13	SI	3	IFPRK05	2	13	15	0401089734
SELASA	R02	SI	3	FTI207	2	13	15	0401089734
SELASA	R08	IF	1	FTI210	3	13	16	0420988987
SELASA	R10	SI	1	FTI210	3	13	16	0420988987
SELASA	R07	IF	3	FTI307	2	11	13	0401237635
SELASA	R10	SI	5	TIF323	3	11	12	0401237635
SELASA	R07	SI	5	IFPRK03	2	11	13	0426655999
SELASA	R11	IF	1	TIF202	3	11	14	0498347997
SELASA	R08	SI	5	SIF343	3	12	15	0412081963
SELASA	R12	SI	1	FTI202	2	11	13	0489798754
SELASA	R10	IF	7	UBB104	2	13	15	0489798754
SELASA	R03	SI	7	UBB104	2	13	15	0489798754
SELASA	R12	IF	3	FTI207	2	13	15	0401263645
SELASA	R13	IF	5	IFPRK11	2	11	13	0489657845
SELASA	R09	IF	5	FTI325	3	9	12	0489657845

RABU	R09	IF	1	UBB102	2	13	15	0416003733
RABU	R01	SI	1	UBB102	3	13	16	0416003733
RABU	R13	IF	7	IFPRK03	2	13	15	0426655999
RABU	R10	IF	7	TIF323	3	13	16	0426655999
RABU	R12	IF	7	FTI315	3	10	13	0426655999
RABU	R06	SI	7	SIF330	3	13	16	0428978922
RABU	R11	IF	3	FTI304	3	11	14	0412081963
RABU	R04	SI	7	FTI214	2	8	10	0428027501
RABU	R12	IF	1	IFPRK02	2	11	13	0478658342
RABU	R14	SI	1	IFPRK02	2	11	13	0478658342
KAMIS	R07	IF	7	UBB103	2	11	13	0420984578
KAMIS	R11	SI	7	UBB103	2	11	13	0420984578
KAMIS	R14	SI	3	IFPRK06	2	14	16	0409107503
KAMIS	R11	IF	5	FTI323	3	13	16	0498347997
KAMIS	R12	SI	7	SIF331	3	14	17	0412081963
KAMIS	R13	IF	5	IFPRK08	2	9	11	0402035566
KAMIS	R08	SI	5	SIF341	3	13	16	0425038203
KAMIS	R06	IF	3	FTI327	3	13	16	0422457888
KAMIS	R04	SI	3	FTI327	3	13	16	0422457888
KAMIS	R08	IF	3	FTI324	3	9	12	0422037002
KAMIS	R05	SI	3	FTI324	3	9	12	0422037002
KAMIS	R09	SI	5	TIF320	3	9	12	0422037002
KAMIS	R09	IF	7	FTI330	3	13	16	0422037002
JUMAT	R02	SI	7	SIF336	3	14	17	0414722561
JUMAT	R14	IF	5	IFPRK12	2	13	15	0419045901
JUMAT	R03	SI	3	AKU315	2	13	15	0409107503
JUMAT	R01	SI	1	SIF303	2	15	17	0428978922
JUMAT	R04	SI	5	SIF335	3	13	16	0428978922
JUMAT	R10	IF	7	FTI337	3	14	17	0412081963
JUMAT	R07	IF	5	FTI328	2	14	16	0428027501
JUMAT	R09	IF	5	FTI311	3	9	12	0428027501
SABTU	R13	IF	1	IFPRK01	2	10	12	0401057504
SABTU	R14	SI	1	IFPRK01	2	14	16	0401057504
SABTU	R06	SI	1	FTI204	2	9	11	0401057504
SABTU	R07	IF	1	FTI204	2	9	11	0401057504
SABTU	R14	SI	5	IFPRK04	2	13	15	0401057504
SABTU	R12	SI	5	SIF326	3	10	13	0401057504
SABTU	R13	IF	3	IFPRK09	2	14	16	0419045901
SABTU	R08	IF	3	FTI309	2	13	15	0419045901
SABTU	R13	SI	3	IFPRK07	2	9	11	0459023522
SABTU	R11	SI	3	SIF346	3	9	12	0459023522
SABTU	R11	IF	5	FTI321	3	9	12	0478657346
SABTU	R12	IF	5	FTI316	3	16	19	0478657346

SABTU	R07	SI	1	UBB105	2	13	15	0411354599
SABTU	R09	IF	1	UBB105	2	13	15	0411354599

c. Data jadwal Program Studi Teknik Informatika dan Sistem Informasi semester ganjil pada tahun 2018 dapat di lihat pada Tabel 4.3.

Tabel 4.3 Data Jadwal Semester Ganjil 2018

HARI	KR	PS	SEM	KDMK	SKS	MULAI	SELESAI	KDDSN
SENIN	R09	SI	1	TIF301	3	11	14	0402108204
SENIN	R06	IF	1	TIF301	2	11	13	0402108204
SENIN	R10	SI	1	FTI212	2	13	15	0429848950
SENIN	R01	IF	1	FTI212	2	13	15	0429848950
SENIN	R09	SI	3	TIF304	3	14	17	0416118901
SENIN	R11	SI	5	TIF317	3	11	14	0416118901
SENIN	R04	IF	3	FTI308	2	16	18	0412081963
SENIN	R06	IF	7	UBB104	2	9	11	0489798754
SENIN	R03	IF	5	FTI317	3	9	12	0402035566
SENIN	R03	SI	5	SIF341	3	13	16	0425038203
SENIN	R02	IF	7	FTI214	2	9	11	0428027501
SELASA	R01	SI	3	FTI307	3	11	14	0426028503
SELASA	R02	SI	1	FTI210	3	13	16	0429848950
SELASA	R05	IF	3	FTI307	2	11	13	0401237635
SELASA	R01	IF	1	TIF202	3	13	16	0498347997
SELASA	R12	SI	1	FTI202	2	9	11	0489798754
SELASA	R03	SI	7	UBB104	2	13	15	0489798754
SELASA	R03	IF	3	FTI207	2	13	15	0401263645
SELASA	R11	SI	3	FTI207	2	13	15	0401263645
SELASA	R02	IF	5	FTI311	3	9	12	0428027501
RABU	R03	IF	1	FTI210	3	11	14	0420988987
RABU	R02	SI	5	TIF323	3	13	16	0401237635
RABU	R04	IF	1	UBB102	2	9	11	0416003733
RABU	R07	SI	1	UBB102	2	9	11	0416003733
RABU	R05	IF	7	FTI315	3	13	16	0426655999
RABU	R06	IF	7	TIF323	3	10	13	0426655999
RABU	R09	SI	1	SIF303	2	13	15	0428978922
RABU	R07	IF	3	FTI304	3	11	14	0412081963
RABU	R04	SI	5	SIF343	3	10	13	0412081963
RABU	R01	IF	3	FTI327	3	12	15	0422457888
RABU	R02	IF	5	FTI328	2	13	15	0428027501
RABU	R01	SI	7	FTI214	2	11	13	0428027501

KAMIS	R08	SI	5	SIF335	3	13	16	0426028503
KAMIS	R12	IF	7	UBB103	2	11	13	0420984578
KAMIS	R02	SI	7	UBB103	2	11	13	0420984578
KAMIS	R01	IF	5	FTI323	3	13	16	0498347997
KAMIS	R08	SI	3	FTI327	3	13	16	0422457888
KAMIS	R10	IF	3	FTI324	3	9	12	0422037002
KAMIS	R09	IF	7	FTI330	3	13	16	0422037002
KAMIS	R02	SI	3	FTI324	3	9	12	0422037002
KAMIS	R05	SI	5	TIF320	3	15	18	0422037002
JUMAT	R05	SI	7	SIF336	3	9	12	0414722561
JUMAT	R04	SI	3	AKU315	2	13	15	0409107503
JUMAT	R03	SI	7	SIF330	3	13	16	0428978922
JUMAT	R12	IF	7	FTI337	3	9	12	0412081963
JUMAT	R11	IF	3	FTI305	3	8	11	0489657845
SABTU	R12	SI	1	FTI204	2	9	11	0402108204
SABTU	R09	IF	1	FTI204	2	9	11	0401057504
SABTU	R03	SI	5	SIF326	3	9	12	0401057504
SABTU	R01	SI	7	SIF331	3	13	16	0412081963
SABTU	R02	SI	3	SIF346	3	11	14	0459023522
SABTU	R01	IF	3	FTI309	2	13	15	0459023522
SABTU	R04	IF	5	FTI316	3	13	16	0478657346
SABTU	R05	IF	5	FTI321	3	11	14	0478657346
SABTU	R02	IF	5	FTI325	3	9	12	0489657845
SABTU	R11	SI	1	UBB105	2	16	18	0411354599
SABTU	R03	IF	1	UBB105	2	16	18	0411354599

Pada jadwal matakuliah semester ganjil telah ditemukan bentrok yang diberi tanda warna merah muda pada tabel. Pada saat terjadi bentrok, maka dosen melakukan pemetaan jadwal ulang dan sering terjadi perkuliahan dengan tidak menggunakan model tatap muka.

#### 4.1.2 Instrumen penelitian

Instrumen penelitian untuk membantu mengumpulkan data pada sebuah penelitian dilakukan dengan observasi, wawancara, studi pustaka dan pembuatan laporan diperlukan sebuah alat bantu sebagai instrument. Adapun instrumen penelitian yang digunakan dalam penelitian ini adalah:

## 1. Perangkat Keras (*Hardware*)

Tabel 4.4 Perangkat Keras yang digunakan

Perangkat	Kegunaan
Smartphone	Digunakan sebagai alat untuk merekam wawancara.
PC dengan spesifikasi: • Intel(R) Core(TM) Duo CPU E8400 @ 3.00GHz (2 CPUs), ~3.0GHz • RAM 4GB • HDD 500 GB	Digunakan sebagai alat untuk pembuatan laporan dan pengembangan aplikasi.
Monitor	Sebagai Output dari memori komputer atau central processing unit berupa biner.
Printer Cannon MP287	Mencetak Laporan
Flashdisk sandisk 16GB	Digunakan untuk menyimpan file-file terkait dokumen skripsi.
Bulpoint Buku Catatan Kertas	Di gunakan untuk mencatat hasil observasi dan wawancara dan pelaporan hasil penelitian

## 2. Perangkat Lunak (*Software*)

Berikut kebutuhan *software* yang penulis gunakan dalam pengembangan Aplikasi Jadwal Perkuliahan dengan Menggunakan Metode Algoritma *Steepest-Ascent Hill Climbing*:

Tabel 4.5 Perangkat Lunak yang digunakan.

No	Software yang digunakan	Fungsi
1	Windows 7 64 Bit	Sistem Operasi
2	MySQL	Database

3	NetBeans IDE	<i>Code Editor</i>
4	XAMPP	<i>Control Panel</i>
5	Google Crome	Web Browser
6	StarUML V2.8.0	Pembuatan Diagram UML
7	Balsamic Mockups 3.5.17	Pembuatan Desain User Interface
8	Scene Builder 8.5.0	Pembuatan User Interface

#### 4.1.3 Analisis Pengguna

Pada pengelolaan jadwal perkuliahan, pengguna dibagi dalam 3, yaitu sebagai berikut:

1. Staff TU

Staff TU dapat mengelola dan mencetak penjadwalan yang meliputi data dosen, mahasiswa, matakuliah, ruang, kelas mengajar, kesediaan mengajar dan jadwal kuliah.

2. Dosen

Dosen dapat melihat dan mengelola data kesediaan mengajar.

3. Mahasiswa

Mahasiswa dapat melihat jadwal perkuliahan.

#### 4.1.4 Analisis User Interface

*User interface* bertujuan untuk mempermudah pengguna dalam menggunakan aplikasi ini dan telah disesuaikan dengan beberapa kebutuhan dalam penelitian. *User interface* akan dibuat begitu sederhana sehingga dalam penggunaanya aplikasi ini tidak membingungkan atau menyulitkan pengguna, seperti penggunaan aplikasi yang bersifat *responsive*, pemilihan huruf yang jelas, pemilihan warna yang tepat dan dibuat tampilan yang menarik.

#### 4.1.5 Fitur-Fitur

Berikut fitur-fitur yang disediakan oleh aplikasi ini :

1. Form login, digunakan untuk membatasi hak akses bagi user untuk melihat dan berinteraksi dengan data. Hanya user yang sudah terdaftar yang bisa mengakses.
2. Form halaman utama yang didalamnya terdapat menu form data,
3. Form data dosen dan input dosen untuk pengelolaan data dosen
4. Form data matakuliah dan input matakuliah untuk pengelolaan matakuliah
5. Form data mahasiswa untuk mengelolaan data mahasiswa
6. Form kesediaan mengajar dan input mengajar untuk pengelolaan data kesediaan mengajar dosen.
7. Form ruang dan waktu untuk mengelola data ruang dan waktu yang tersedia dan akan digunakan.
8. Form waktu
9. Form kelas dan input kelas untuk mengelola data kelas mengajar dosen.
10. Form penjadwalan untuk menampilkan jadwal yang sudah di optimasi.
11. Form user

#### 4.1.6 Analisa Data

##### 1. Analisis Bentrok Data Pada Jadwal

Penganalisaan data dilakukan dengan menggunakan data jadwal matakuliah Teknik Informatika dan Sistem Informasi semester ganjil 2015.

Tabel 4.6 Jenis Bentrok

	Ruang	Dosen	MPS	Ruang-Dosen	Dosen-MPS	Ruang-MPS
Ruang	V			V		V
Dosen		V		V	V	
MPS			V		V	V
Waktu	V	V	V	V	V	V

Pada tahap awal dalam melakukan pengujian, dibuat tabel kelas untuk setiap dosen pengajar dan matakuliah yang diajarkan berdasarkan dari data kesediaan dosen mengajar. Data kelas dapat dilihat pada tabel berikut.

Tabel 4.7 Data Kelas

KD_KELAS	KD_MTK	NIDN	KD_MPS	KTA	T/F
FTI3230401988974_IF05	FTI323	0401988974	IF05	1	T
TIF3240401988974_SI03	TIF324	0401988974	SI03	1	T
FTI3070426028503_IF03	FTI307	0426028503	IF03	1	T
FTI3070426028503_SI03	FTI307	0426028503	SI03	1	T
SIF3350426028503_SI05	SIF335	0426028503	SI05	1	T
TIF3010402108204_IF01	TIF301	0402108204	IF01	1	T
TIF3010402108204_SI01	TIF301	0402108204	SI01	1	T
FTI2070401089734_IF03	FTI207	0401089734	IF03	1	T
FTI2070401089734_SI03	FTI207	0401089734	SI03	1	T
IFPRK050401089734_SI03	IFPRK05	0401089734	SI03	1	T
SIF3300419834800_SI07	SIF330	0419834800	SI07	1	T
FTI2100420988987_IF01	FTI210	0420988987	IF01	1	T
FTI3270420988987_IF03	FTI327	0420988987	IF03	1	T
FTI2100420988987_SI01	FTI210	0420988987	SI01	1	T
TIF3270420988987_SI03	TIF327	0420988987	SI03	1	T
FTI3240417768389_IF03	FTI324	0417768389	IF03	1	T
FTI2040401057504_IF01	FTI204	0401057504	IF01	1	T
IFPRK010401057504_IF01	IFPRK01	0401057504	IF01	1	T
FTI2040401057504_SI01	FTI204	0401057504	SI01	1	T
IFPRK010401057504_SI01	IFPRK01	0401057504	SI01	1	T
SIF3260401057504_SI05	SIF326	0401057504	SI05	1	T
IFPRK040401057504_SI05	IFPRK04	0401057504	SI05	1	T
FTI2120429848950_IF01	FTI212	0429848950	IF01	1	T
FTI2120429848950_SI01	FTI212	0429848950	SI01	1	T
UBB1030420984578_IF07	UBB103	0420984578	IF07	1	T
UBB1030420984578_SI07	UBB103	0420984578	SI07	1	T
FTI3170420984578_IF05	FTI317	0420984578	IF05	1	T
IFPRK080419045901_IF05	IFPRK08	0419045901	IF05	1	T
FTI3370419045901_IF07	FTI337	0419045901	IF07	1	T
SIF3360419045901_SI07	SIF336	0419045901	SI07	1	T
TIF3230401237635_IF07	TIF323	0401237635	IF07	1	T
IFPRK030401237635_IF07	IFPRK03	0401237635	IF07	1	T
TIF3230401237635_SI05	TIF323	0401237635	SI05	1	T
IFPRK030401237635_SI05	IFPRK03	0401237635	SI05	1	T

UBB1020416003733_IF01	UBB102	0416003733	IF01	1	T
UBB1020416003733_SI01	UBB102	0416003733	SI01	1	T
TIF3040416118901_SI03	TIF304	0416118901	SI03	1	T
TIF3170416118901_SI05	TIF317	0416118901	SI05	1	T
FTI3110429085487_IF05	FTI311	0429085487	IF05	1	T
SIF3460429085487_SI03	SIF346	0429085487	SI03	1	T
IFPRK090418459074_IF03	IFPRK09	0418459074	IF03	1	T
IFPRK060409107503_SI03	IFPRK06	0409107503	SI03	1	T
AKU3150409107503_SI03	AKU315	0409107503	SI03	1	T
SIF3030428978922_SI01	SIF303	0428978922	SI01	1	T
TIF2020498347997_IF01	TIF202	0498347997	IF01	1	T
FTI3040498347997_IF03	FTI304	0498347997	IF03	1	T
FTI3080498347997_IF03	FTI308	0498347997	IF03	1	T
FTI2140408927573_IF07	FTI214	0408927573	IF07	1	T
FTI2140408927573_SI07	FTI214	0408927573	SI07	1	T
SIF3430412081963_SI05	SIF343	0412081963	SI05	1	T
SIF3310412081963_SI07	SIF331	0412081963	SI07	1	T
FTI2020489798754_SI01	FTI202	0489798754	SI01	1	T
UBB1040489798754_SI07	UBB104	0489798754	SI07	1	T
FTI3090459023522_IF03	FTI309	0459023522	IF03	1	T
IFPRK070459023522_IF05	IFPRK07	0459023522	IF05	1	T
IFPRK070459023522_SI03	IFPRK07	0459023522	SI03	1	T
FTI3160412342356_IF05	FTI316	0412342356	IF05	1	T
SIF3410425038203_SI05	SIF341	0425038203	SI05	1	T
FTI3210478657346_IF05	FTI321	0478657346	IF05	1	T
FTI3150478657346_IF07	FTI315	0478657346	IF07	1	T
TIF3200478657346_SI05	TIF320	0478657346	SI05	1	T
FTI3050489657845_IF03	FTI305	0489657845	IF03	1	T
IFPRK100489657845_IF03	IFPRK10	0489657845	IF03	1	T
FTI3250489657845_IF05	FTI325	0489657845	IF05	1	T
IFPRK100489657845_IF05	IFPRK10	0489657845	IF05	1	T
UBB1050411354599_IF01	UBB105	0411354599	IF01	1	T
UBB1050411354599_SI01	UBB105	0411354599	SI01	1	T
FTI3280428027501_IF05	FTI328	0428027501	IF05	1	T
UBB1040428027501_IF07	UBB104	0428027501	IF07	1	T
IFPRK020478658342_IF01	IFPRK02	0478658342	IF01	1	T
IFPRK020478658342_SI01	IFPRK02	0478658342	SI01	1	T
FTI3300478658342_IF07	FTI330	0422037002	IF07	1	T

Tabel 4.10 merupakan data kelas pada jadwal matakuliah yang digunakan untuk pengujian algoritma. Data kelas ini merupakan terdiri dari data dosen, matakuliah dan mahasiswa (program studi dan semester).

## 2. Analisis Algoritma *Steepest-Ascent Hill Climbing*

Dalam penelitian ini akan dilakukan pemodelan masalah mengenai penjadwalan kuliah dengan menerapkan algoritma *steepest-ascent hill climbing*.

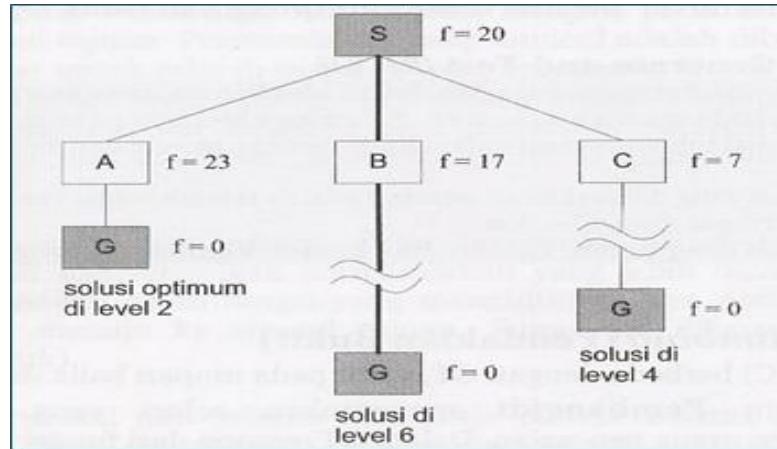
```

Masuk : Iter Maks, Permasalahan
Keluar : SolusiSekarang
SolusiSekarang ← SolusiRandom(Permasalahan)
Jika (SolusiSekarang == SolusiTujuan)
    Kembali SolusiSekarang
Bukan
    Ulang (Iter i=0 ∈ Iter Maks)
        SolusiKandidat ← DaftarSolusiPewaris(SolusiSekarang)
        Jika (Nilai(SolusiSekarang) > Nilai(SolusiKandidat))
            SolusiSekarang ← SolusiKandidat
            i = 0
            Jika ( SolusiSekarang == SolusiTujuan )
                Kembali SolusiSekarang
            Aahir_Jika
        Aahir_Jika
    Aahir_Ulang
    Kembali SolusiSekarang
Aahir_Bukan

```

Gambar 4.1 Pseudo Code Algoritma *Steepest-Ascent Hill Climbing*

Gambar 4.1 menggambarkan alur logika program algoritma *steepest-ascent hill climbing*. *Pseudo code* dibuat berdasarkan prosedur algoritma *steepest-ascent hill climbing*.



Gambar 4.2 Pencarian Solusi dengan Algoritma SAHC.

Gambar 4.2 Menjelaskan ilustrasi alur. Dari state S, algoritma akan mengevaluasi semua state yang menjadi next state atau suksesornya, yaitu A, B ,C. Dari ketiga suksesor tersebut dipilih suksesor dengan nilai f yang terkecil.State C akan dipilih sebagai suksesor S. Misalkan, hasil penelusuran menemukan solusi G di level 4, padahal ada solusi optimal di level 2, dalam hal ini *steepest-ascent hill climbing* dikatakan terjebak pada solusi lokal atau local minimum. Jadi *steepest-ascent hill climbing* juga tidak optimal.

Untuk membangkitkan jadwal baru atau proses manipulasi yaitu berdasarkan cek *constrain* pada setiap dosen pengajar. Jika dosen yang melanggar *constrain*, maka dosen akan dirandom untuk mendapatkan ruangan baru. Jika dosen tidak melanggar *constrain*, maka jam dan ruangan yang didapatkan tetap. Dari proses tersebut terbentuk jadwal baru yang merupakan turunan dari jadwal sebelumnya, dimana masih terdapat kesamaan dari jadwal sebelumnya. Selanjutnya evaluasi jadwal, jika jadwal baru lebih baik maka proses berikutnya menggunakan jadwal baru. Jika tidak maka proses selanjutnya menggunakan jadwal sebelumnya. Proses manipulasi jadwal akan dikerjakan sampai jadwal sudah sesuai dengan tujuan atau jadwal tidak berubah sampai batas yang sudah ditentukan.

Dalam penelitian yang dilakukan terdapat dua *constrain* yaitu *hard constrain* dan *soft constrain*. *Hard constrain* jadwal adalah dosen tidak boleh mengajar pada hari dan jam yang sama di ruangan yang berbeda. Ruangan pada

hari dan jam yang sama tidak boleh digunakan untuk lebih dari satu mata kuliah. Pada penelitian ini *hard constrain* dicek pada saat akan memulai membuat jadwal. Sedangkan *soft constrain* jadwal adalah pengajar tidak boleh melebihi batas beban sks pengajar. Dua *soft constrain* tersebut merupakan kriteria untuk menentukan kualitas jadwal.

Rumus untuk menghitung kualitas jadwal  $|f_{(ob)}|$  akan dijelaskan sebagai berikut :

$$\begin{aligned}
 f_{(ob)} &= \frac{C_1 \times f_{(sks)} + C_2 \times f_{(kelas)}}{C_1 + C_2} \\
 f_{(sks)} &= \frac{\sum_{i=1}^n sks_i}{n} \\
 sks_i &= \begin{cases} \text{jika } jsks_{(s+b)} < maks_{(sks)} = 0 \\ \quad \quad \quad \text{atau} \\ \quad \quad \quad jsks_{(s+b)} > maks_{(sks)} = jsks_{(s+b)} - maks_{(sks)} \end{cases} \\
 f_{(kelas)} &= \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}} \\
 x &= \frac{\sum_{i=1}^n x_i}{n}
 \end{aligned}$$

Keterangan :

$f_{(ob)}$	= Fungsi objektif jadwal
$f_{(sks)}$	= Fungsi beban sks mengajar
$f_{(kelas)}$	= Fungsi penyetaraan kelas
$C_1, C_2$	= Nilai penyetaraan
$sks_i$	= Fungsi pinalti beban sks tiap pengajar
$n$	= Jumlah pengajar
$jsks_{(s+b)}$	= Jumlah sks jadwal yang sudah dibuat dan akandibuat
$maks_{(sks)}$	= Batas maksimal sks
$x_i$	= Jumlah kelas pengajar
$x$	= Rata-rata kelas

Algoritma *steepest-ascent hill climbing* di implementasikan dengan menggunakan data jadwal kuliah semsester ganjil 2015.

Tabel 4.8 Jadwal

<b>KD_JDWL</b>	<b>KDK</b>	<b>KD_WKT</b>	<b>KDR</b>
1	FTI2120429848950_IF01	0111	R04
2	FTI2120429848950_IF01	0112	R04
3	FTI2120429848950_SI01	0113	R01
4	FTI2120429848950_SI01	0114	R01
5	TIF2020498347997_IF01	0109	R01
6	TIF2020498347997_IF01	0110	R01
7	IFPRK090418459074_IF03	0113	R13
8	IFPRK090418459074_IF03	0114	R13
9	FTI3090459023522_IF03	0109	R02
10	FTI3090459023522_IF03	0110	R02
11	FTI3250489657845_IF05	0110	R02
12	FTI3250489657845_IF05	0111	R02
13	FTI3250489657845_IF05	0112	R03
14	IFPRK100489657845_IF05	0113	R14
15	IFPRK100489657845_IF05	0114	R14
16	FTI3230401988974_IF05	0209	R04
17	FTI3230401988974_IF05	0210	R04
18	FTI3230401988974_IF05	0211	R04
19	TIF3240401988974_SI03	0213	R03
20	TIF3240401988974_SI03	0214	R03
21	TIF3240401988974_SI03	0215	R03
22	FTI2100420988987_SI01	0209	R05
23	FTI2100420988987_SI01	0210	R05
24	FTI2100420988987_SI01	0211	R05
25	FTI2100420988987_IF01	0214	R03
26	FTI2100420988987_IF01	0215	R03
27	FTI2100420988987_IF01	0216	R03
28	FTI3240417768389_IF03	0209	R02
29	FTI3240417768389_IF03	0210	R02
30	FTI3240417768389_IF03	0211	R02
31	FTI3370419045901_IF07	0209	R06
32	FTI3370419045901_IF07	0210	R06
33	FTI3370419045901_IF07	0211	R06
34	SIF3360419045901_SI07	0211	R09
35	SIF3360419045901_SI07	0212	R09
36	SIF3360419045901_SI07	0213	R09

Untuk membangkitkan jadwal baru atau proses manipulasi jadwal diperlukan nilai bobot pada setiap dosen pengajar. Bobot yang dihasilkan dengan penghitungan algoritma dan ada beberapa variabel yang menentukan nilai bobot. Adapun variabel penentu bobot adalah sebagai berikut:

Tabel 4.9 Variabel Penentu Nilai Bobot Dosen

<b>Variabel</b>	<b>Nilai Bobot</b>
Kaprodi	Nilai bobot mutlak
SKS	Semakin kecil SKS semakin besar nilai bobot. sebaliknya semakin besar SKS semakin kecil nilai bobotnya
Waktu	Semakin waktu banyak maka semakin kecil nilai bobotnya.sebaliknya semakin sedikit waktu tersedia semakin besar nilai bobotnya.

Tabel 4.10 Jadwal Random

<b>NIDN</b>	<b>JML_KLS</b>	<b>JML_SKS</b>	<b>TOTAL SKS</b>
0401988974	2	6	6
0426028503	3	8	8
0402108204	2	6	6
0401089734	3	6	6
0419834800	1	3	3
0420988987	4	12	12
0417768389	1	3	3
0401057504	6	13	13
0429848950	2	4	4
0420984578	2	4	4
0419045901	4	11	11
0401237635	4	10	10
0416003733	2	5	5
0416118901	2	6	6
0429085487	2	6	6
0418459074	1	2	2
0409107503	2	4	4
0428978922	1	2	2

0498347997	3	7	7
0408927573	2	4	4
0412081963	2	6	6
0489798754	2	4	4
0459023522	3	6	6
0412342356	1	3	3
0425038203	1	3	3
0478657346	3	9	9
0489657845	4	10	10
0411354599	2	4	4
0428027501	2	4	4
0478658342	2	4	4

Tabel 4.10 menjelaskan nilai yang didapatkan dari setiap pengajar seperti jumlah kelas, jumlah sks mengajar pada jadwal berdasarkan cek ketentuan, selanjutnya jadwal akan dinilai menggunakan rumus yang sudah dijelaskan sebelumnya. Perhitungan jadwal menggunakan rumus kesetaraan nilai memiliki notasi nilai  $C_1$  bernilai 3,5 dan  $C_2$  bernilai 16,5. Rumus pinalti memiliki notasi nilai  $Maks_{(skls)}$  bernilai 18 dan  $n$  bernilai 31. Contoh perhitungan akan dijelaskan sebagai berikut.

$$f_{(ob)} = \frac{(3,5 \times f_{(kelas)}) + (16,5 \times f_{(skls)})}{(3,5 + 16,5)} = 0,116$$

$$= \frac{(3,5 \times 0,666) + (16,5 \times 0,000)}{20} = 0,116$$

$$f_{(skls)} = \frac{(0 + 0 + 0 + 0 + 0 + \dots + 0)}{31} = 0,000$$

$$f_{(kelas)} = \sqrt{\frac{(1-2)^2 + (2-2)^2 + \dots + (2-2)^2}{(31 - 1)}} = 0,666$$

$$\bar{X} = \frac{(2+3+2+3+1+4+1+6+\dots+1)}{31} = 2$$

Perhitungan nilai kualitas jadwal didapat berdasarkan Tabel 4.10. Berdasarkan perhitungan tersebut, nilai kualitas jadwal dari hasil random adalah 0,116 karena tidak terjadi pelanggaran *soft constrain*. Nilai beban sks jadwal  $|f_{(sk)}|$  adalah 0,000 yang berarti bahwa tidak ada pengajar yang melanggar batas 20 sks seperti yang dijelaskan pada Table 4.10.

```

Fungsi Frata2_ngajar() : real
Begin
    Buka database tabel kelas
    Jumlah_dosen ← ø;
    T.NIDN = " "
    Jumlah_mengajar (I) ← ø
    I = ø
    Ulang tabel_kelas ≠ EOF (belum habis)
        Baca kelas;
        If kelas.NIDN ≠ T.NIDN then
            Jumlah_dosen ← Jumlah_dosen + 1
            Jumlah_mengajar (I) = 1
        Else
            Jumlah_mengajar (I) = Jumlah_mengajar (I) + 1
        Endif
    Akhir_Ulangi

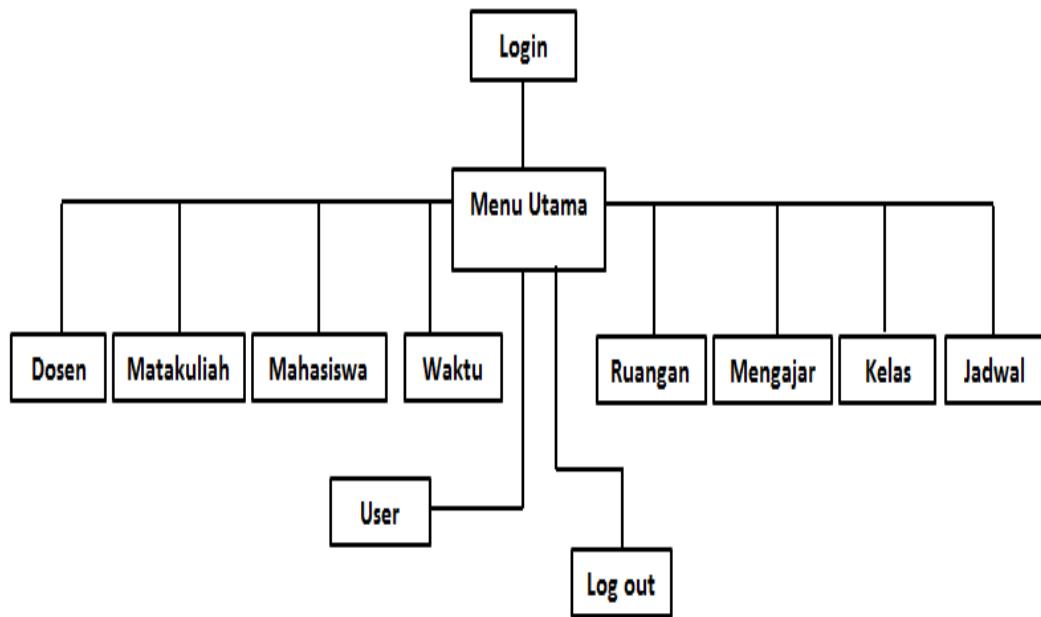
```

Gambar 4.3 Pseudocode Fungsi Rata-Rata Mengajar Dosen.

## 4.2 Perancangan

### 4.2.1 Struktur Menu

Struktur menu disusun secara tahapan untuk memudahkan dalam pengoperasian aplikasi. Berikut adalah struktur menu yang dibuat.



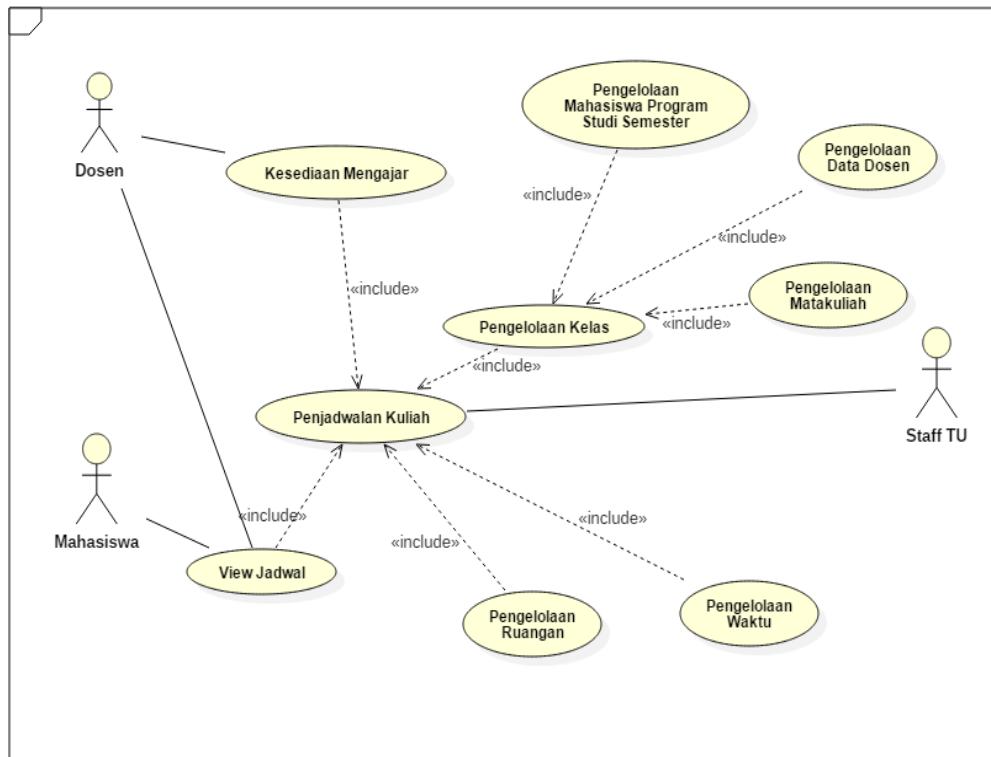
Gambar 4.4 Struktur Menu

Berikut ini adalah penjelasan dari struktur menu:

1. Staff TU melakukan login aplikasi, kemudian masuk ke halaman menu utama.
2. Pada halaman menu utama terdapat beberapa menu pengelolaan data master yaitu, data dosen, mata kuliah, mahasiswa, ruangan, dan waktu. Selain itu terdapat menu kesedian mengajar dosen mengajar, kelas mengajar dosen dan jadwal kuliah.
3. Pada menu user, pengguna dapat menambah akun user dan mengatur aktif/non aktif user.
4. Untuk menutup aplikasi pengguna dapat memilih menu log out.

#### 4.2.2 Use Case Diagram

*Use Case Diagram* dibuat dengan tujuan untuk memberikan gambaran interaksi antara sistem dan aktor pada aplikasi penjadwalan yang dibuat. Berikut adalah *use case diagram* yang digambarkan untuk penelitian ini.



Gambar 4.5 Use Case Diagram Aplikasi Penjadwalan

### 1. Skenario *Use Case*

Berikut adalah deskripsi *use case diagram*.

Tabel 4.11 Aksi Dosen

Aksi Aktor	Reaksi Sistem
a. Login Sistem	
b. Memilih menu form kesediaan dosen	b. Menampilkan halaman menu
c. Menambah, mengedit, menghapus kesediaan mengajar.	d. Menampilkan halaman data kesediaan dosen
d. Data kesediaan mengajar di simpan.	

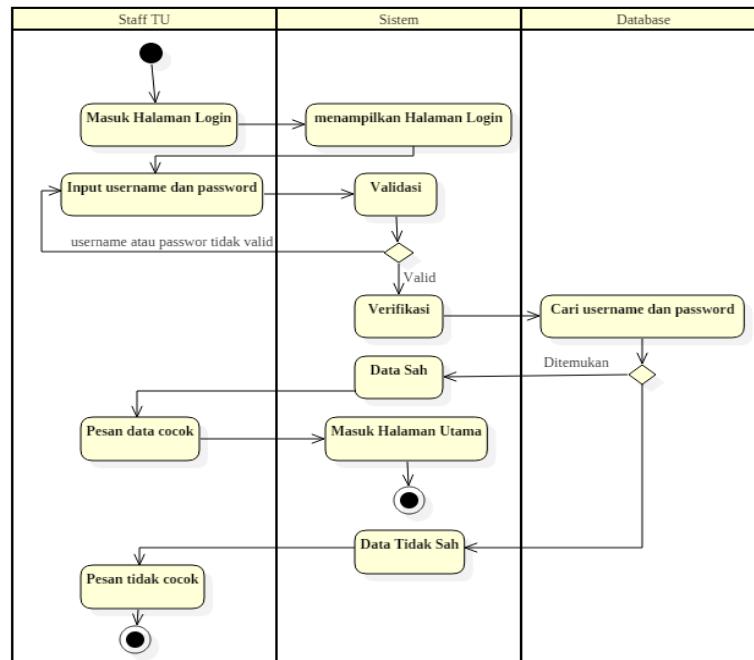
Tabel 4.12 Aksi Staff TU

<b>Use case</b>	<b>Reaksi Sistem</b>
a. Login Sistem	
	b. Menampilkan halaman menu
c. Memilih menu data (dosen, mata kuliah, mahasiswa, ruangan, waktu).	
	d. Menampilkan halaman data yang di pilih
e. Menambah, mengedit, menghapus data yang dipilih	
	f. Data data yang dipilih akan disimpan, di update atau dihapus.
g. Memilih menu optimasi penjadwalan SAHC	
	h. Memeriksa jadwal hasil dari optimasi

#### 4.2.3 *Activity Diagram*

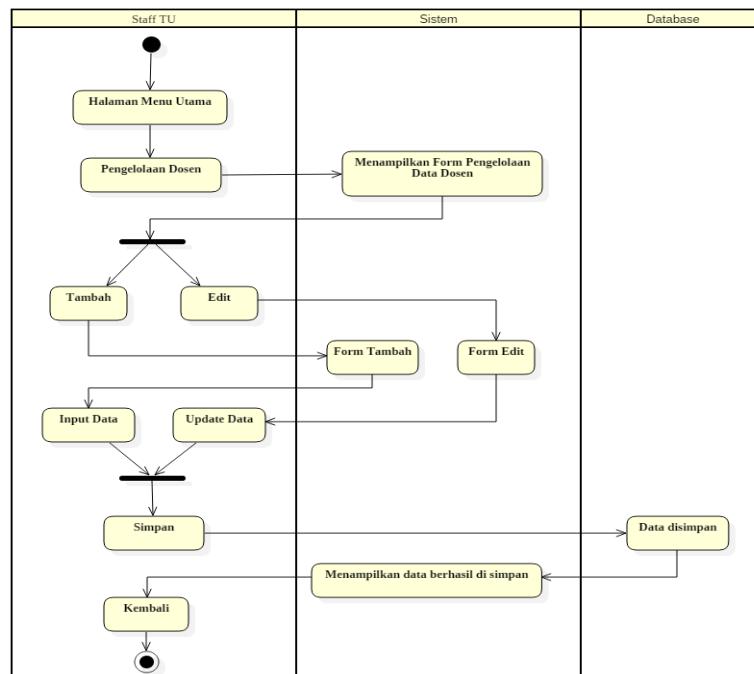
Berikut *activity diagram* dalam aplikasi penjadwalan yang menunjukkan aktivitas yang dilakukan *user* dan timbal balik dari sistem terhadap user secara sistematis. Adapun *activity diagram* yang dibuat diantaranya proses login, proses input dan update data, proses hapus data, proses cari data, proses proses optimasi penjadwalan kuliah.

### 1. Activity Diagram Login



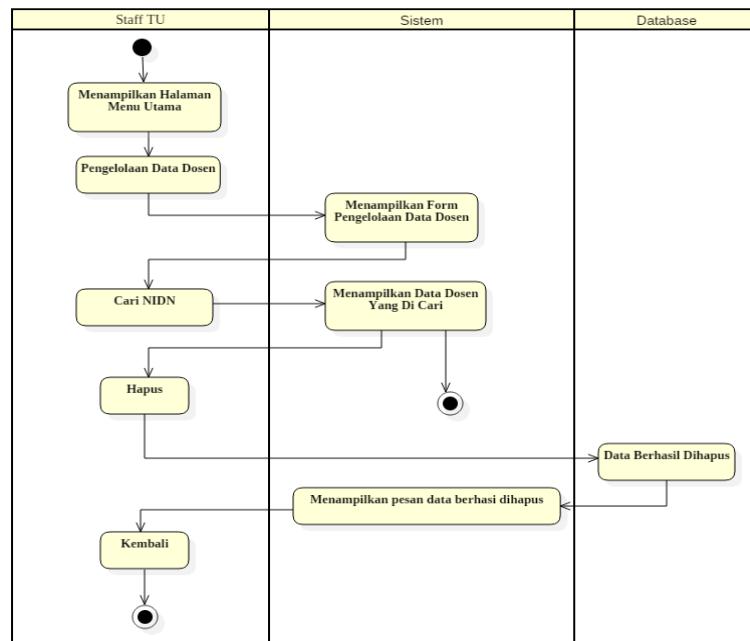
Gambar 4.6 Activity Diagram Proses Login

### 2. Activity Diagram Input dan Edit Data Dosen



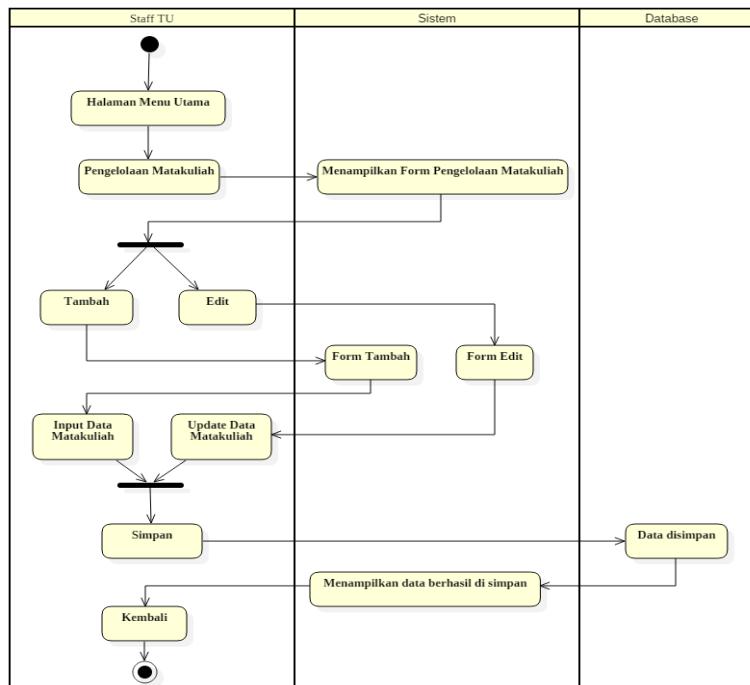
Gambar 4.7 Activity Diagram Proses Input dan Hapus Data Dosen

### 3. Activity Diagram Pencarian Data dan Hapus Data Dosen



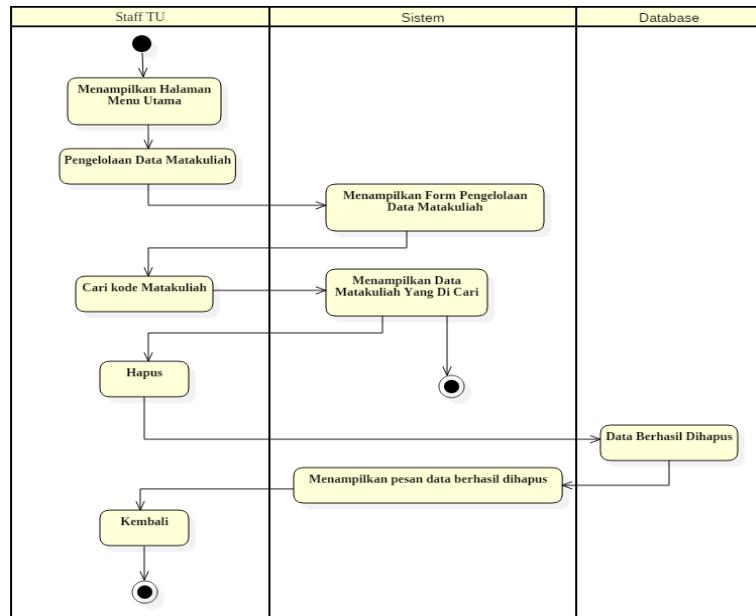
Gambar 4.8 Activity Diagram Pencarian dan Hapus Data Dosen

### 4. Activity Diagram Tambah dan Edit Data Matakuliah



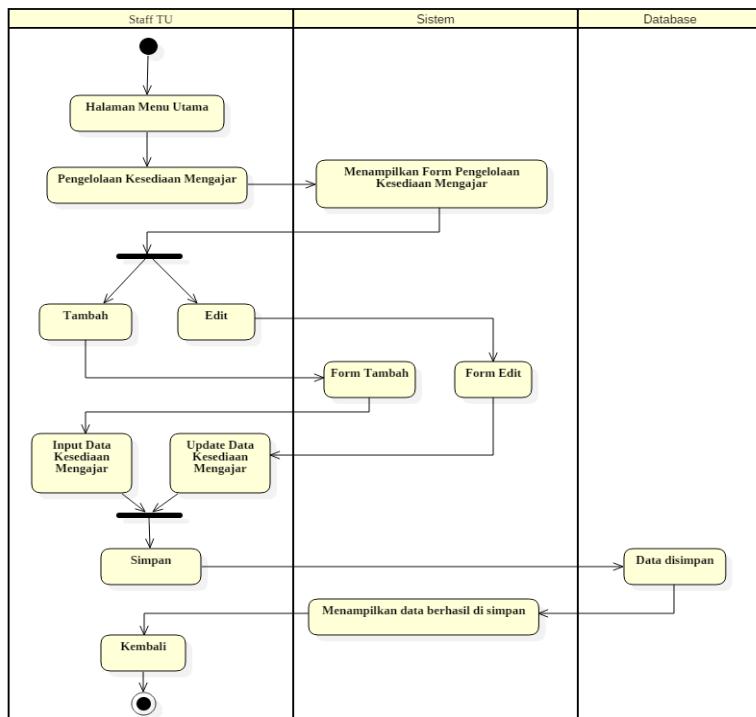
Gambar 4.9 Activity Diagram Tambah Dan Edit Data Matakuliah

## 5. Activity Diagram Pencarian dan Hapus Data Matakuliah



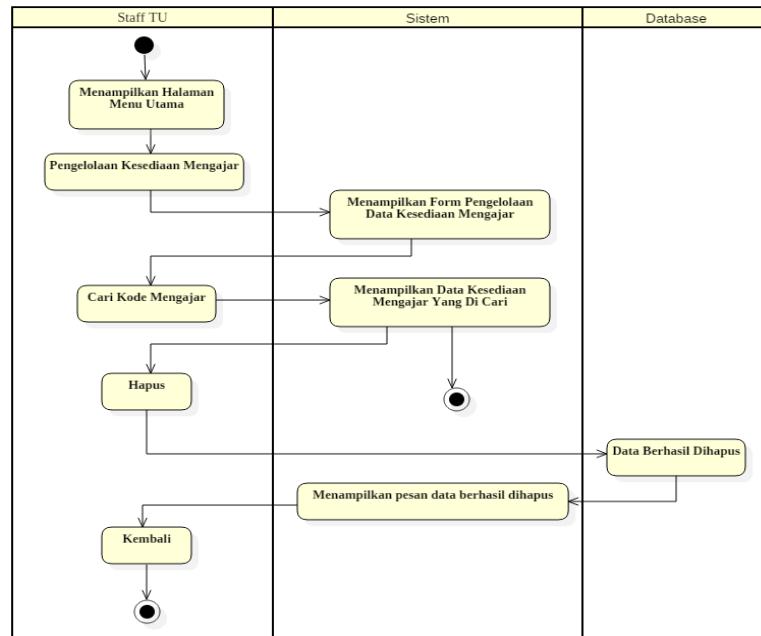
Gambar 4.10 Activity Diagram Pencarian dan Hapus Data Matakuliah

## 6. Activity Diagram Input dan Edit Data Kesediaan Mengajar



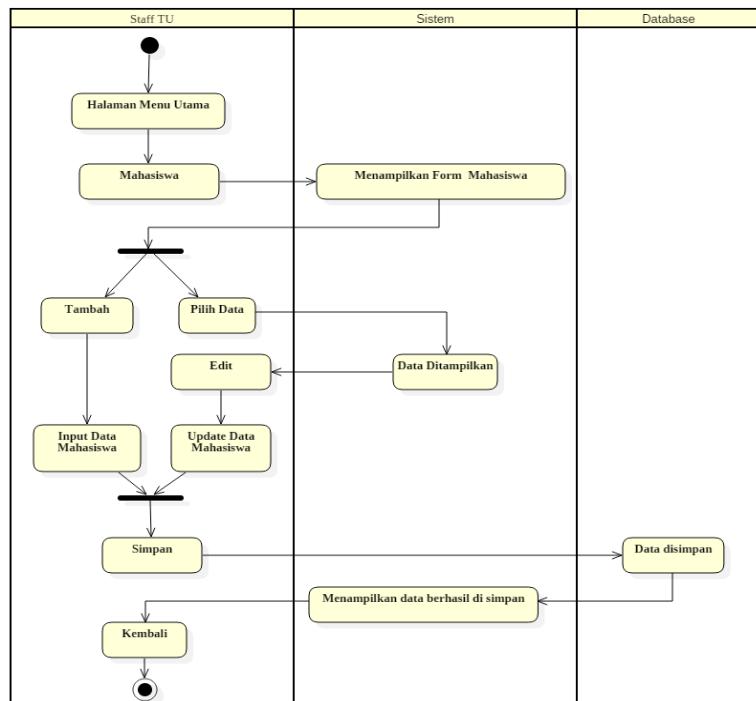
Gambar 4.11 Activity Diagram Input dan Edit Kesediaan Mengajar

## 7. Activity Diagram Pencarian dan Hapus Data Kesediaan Mengajar



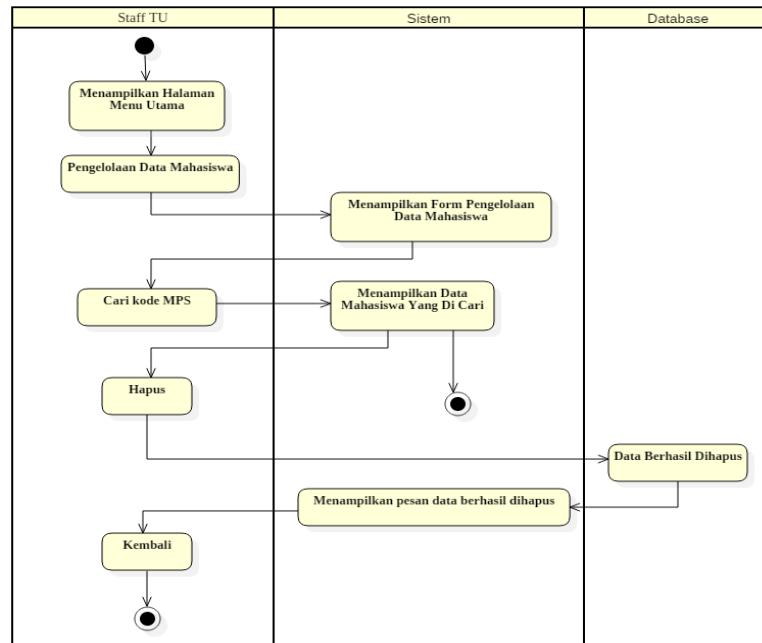
Gambar 4.12 Activity Diagram Pencarian dan Hapus Data Mengajar

## 8. Activity Diagram Tambah dan Edit Data Mahasiswa



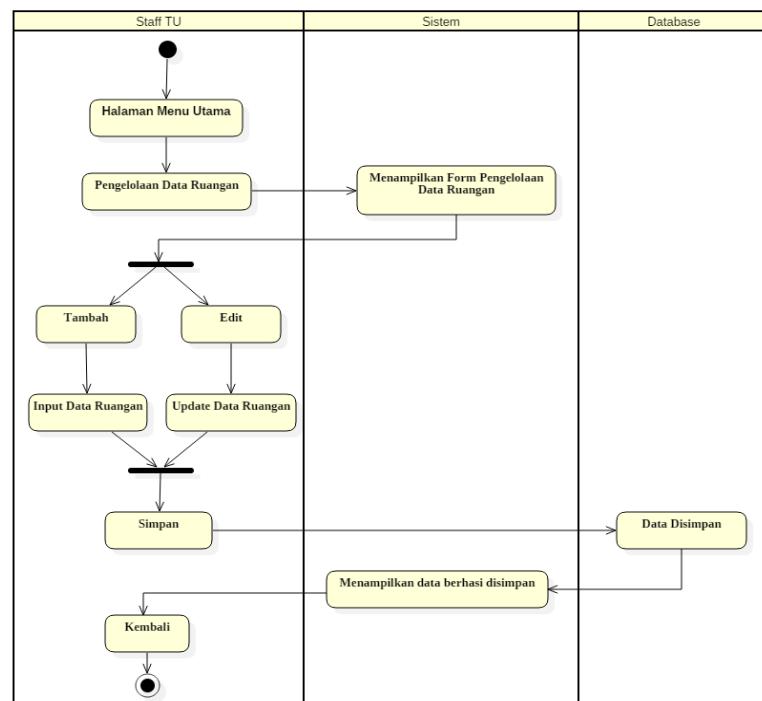
Gambar 4.13 Activity Diagram Input dan Edit Mahasiswa

### 9. Activity Diagram Pencarian dan Hapus Data Mahasiswa



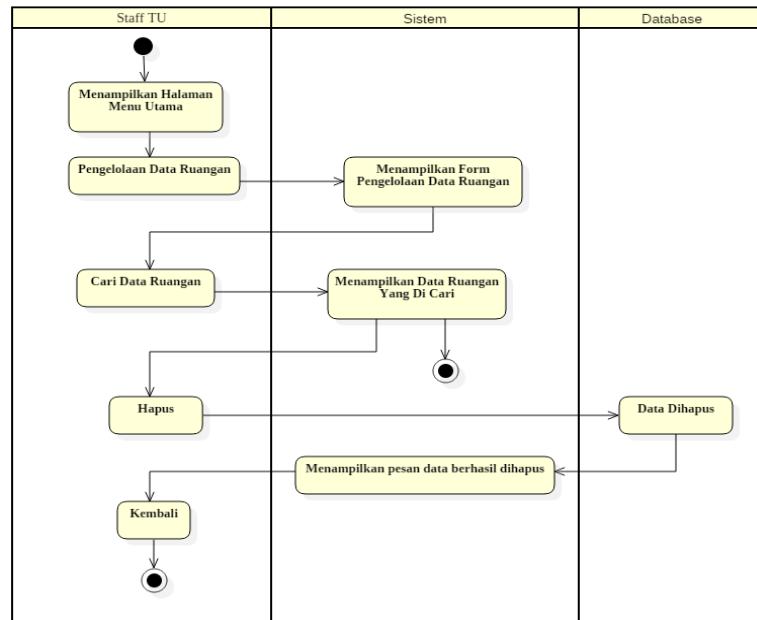
Gambar 4.14 Activity Diagram Pencarian dan Hapus Data Mahasiswa

### 10. Activity Diagram Tambah dan Edit Data Ruangan



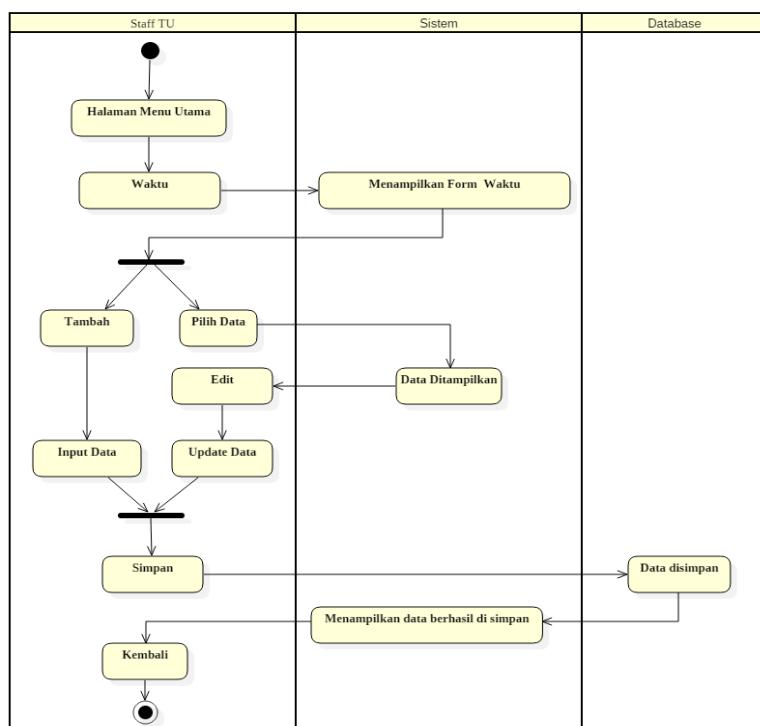
Gambar 4.15 Activity Diagram Input dan Edit Ruangan

### 11. Activity Diagram Pencarian dan Hapus Data Ruangan



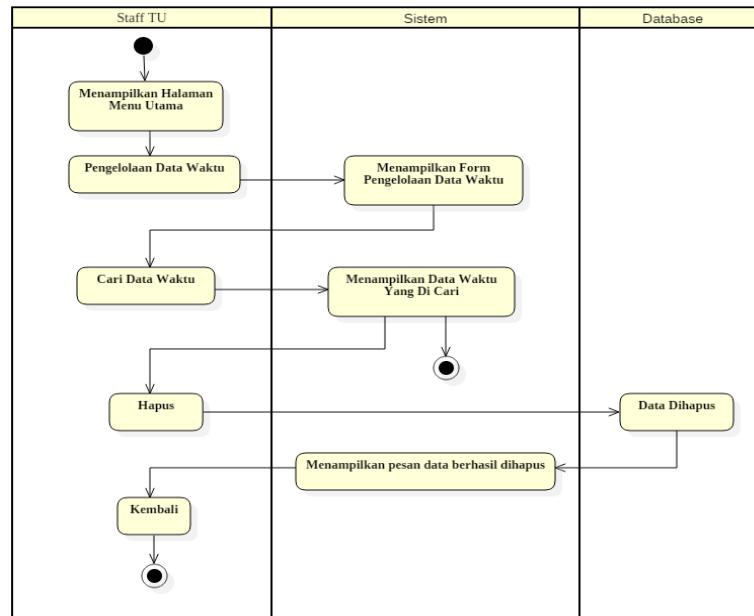
Gambar 4.16 Activity Diagram Pencarian dan Hapus Ruangan

### 12. Activity Diagram Tambah dan Edit Data Waktu



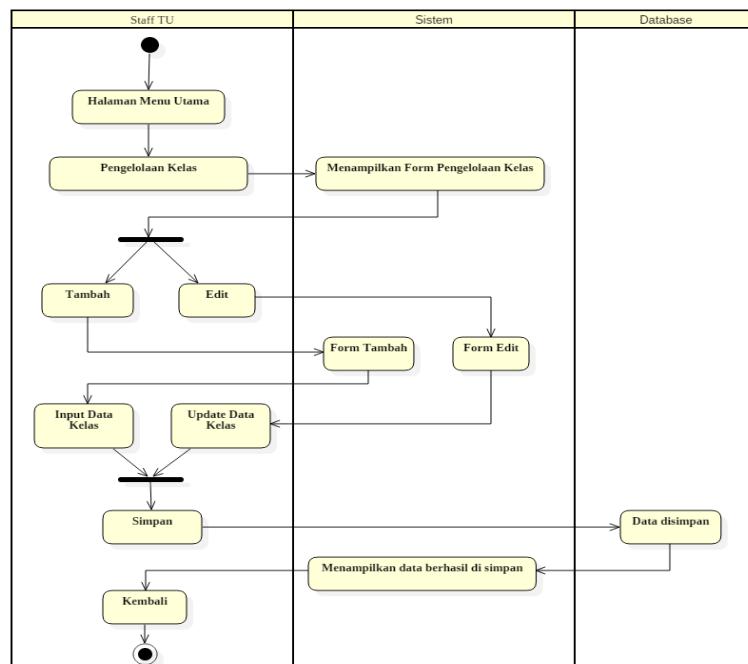
Gambar 4.17 Activity Diagram Tambah dan Edit Data Waktu

### 13. Activity Diagram Pencarian dan Hapus Data Waktu



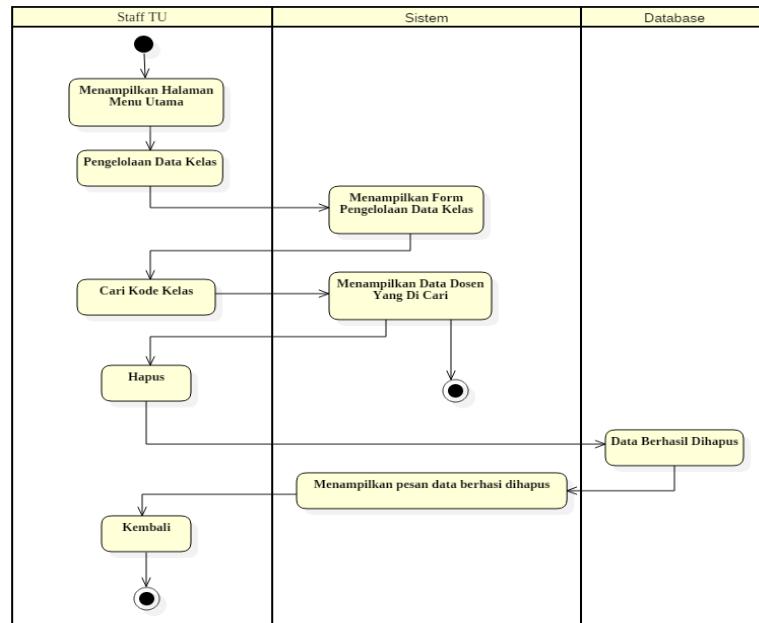
Gambar 4.18 Activity Diagram Pencarian dan Hapus Data Waktu

### 14. Activity Diagram Tambah dan Edit Data Kelas



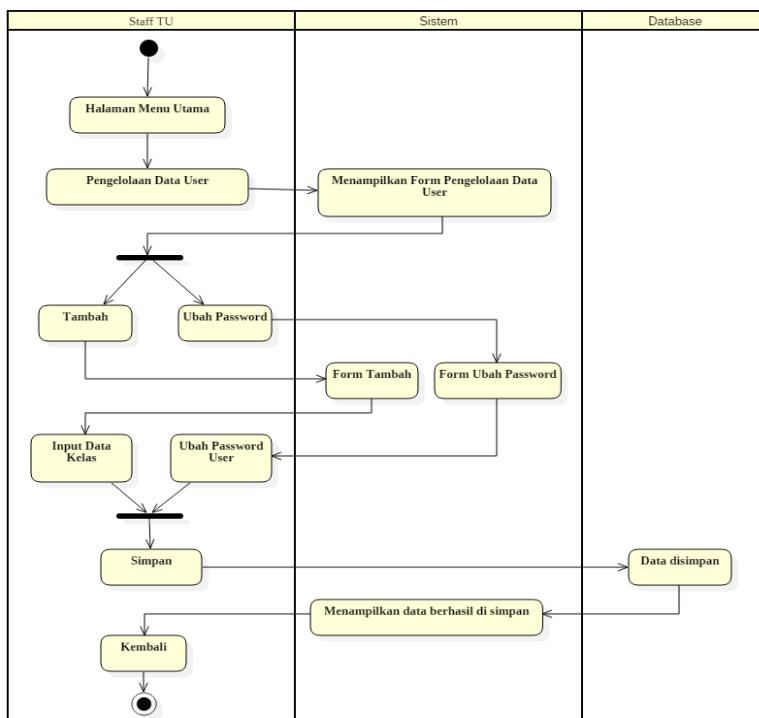
Gambar 4.19 Activity Diagram Tambah dan Edit Data Kelas

15. *Activity Diagram Pencarian Dan Hapus Data Kelas*



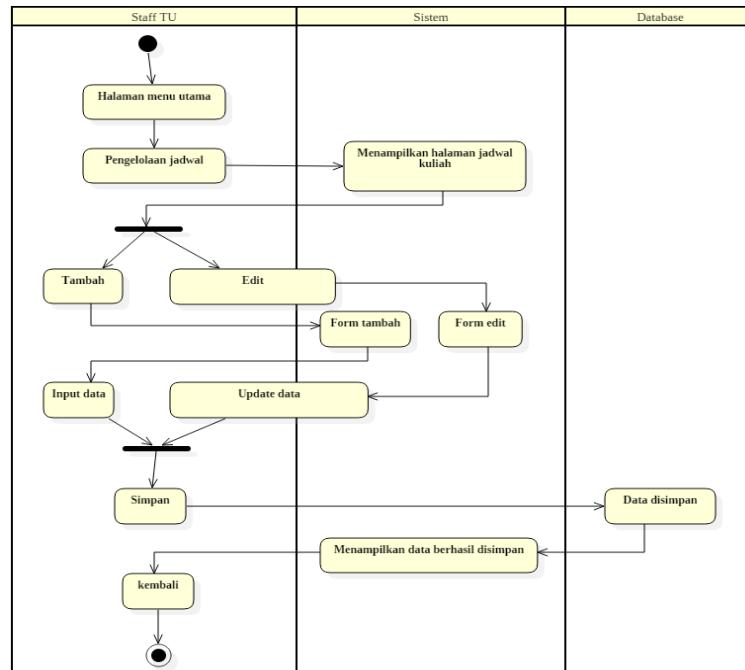
Gambar 4.20 Activity Diagram Pencarian Dan Hapus Data Kelas

16. *Activity Diagram Tambah dan Ubah Password User.*



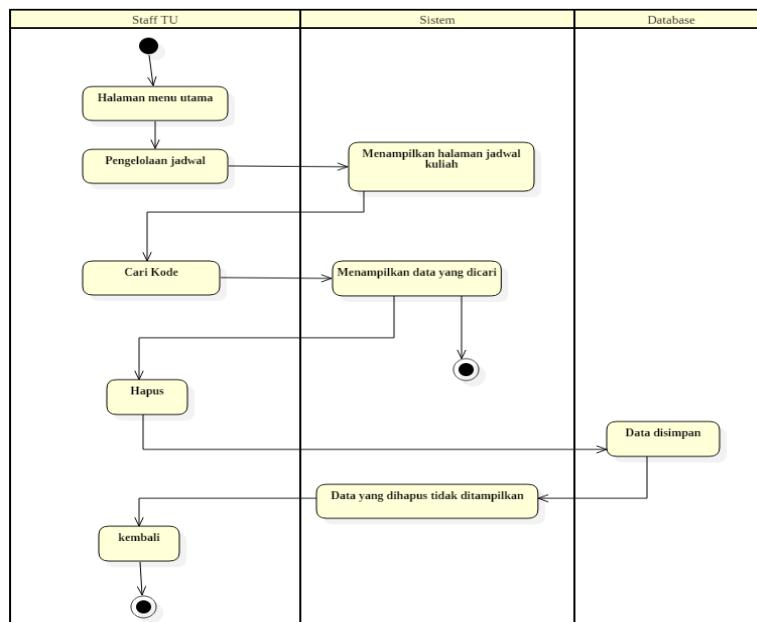
Gambar 4.21 Activity Diagram Tambah dan Ubah Password

17. *Activity Diagram Input dan Update Jadwal Kuliah*



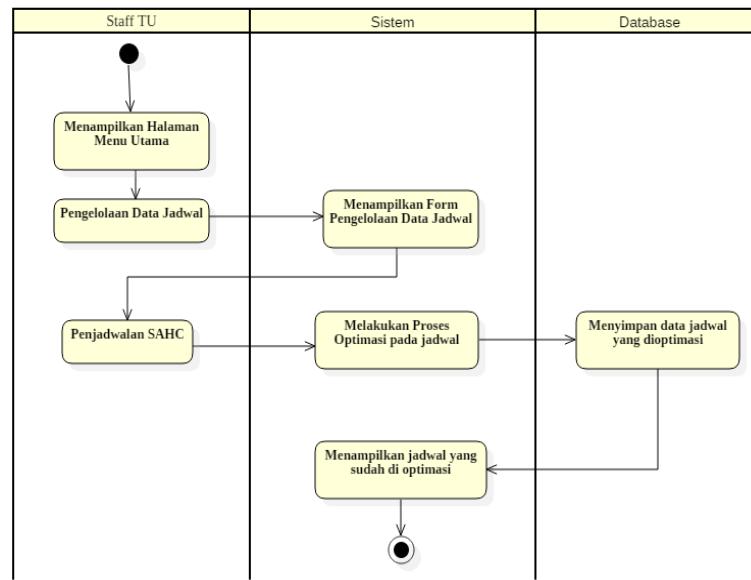
Gambar 4.22 Activity Diagram Input dan Update Jadwal Kuliah

18. *Activity Diagram Pencarian Dan Hapus Jadwal*



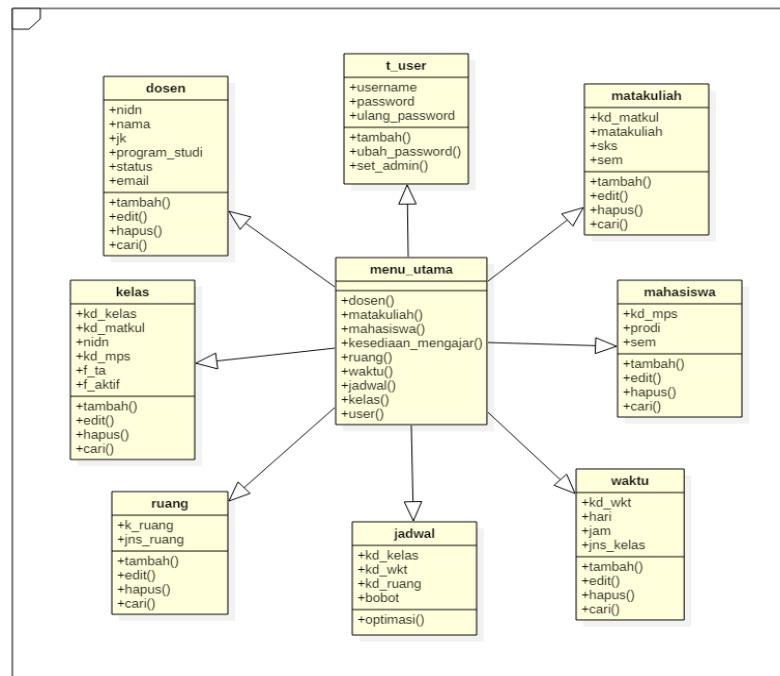
Gambar 4.23 Activity Diagram Pencarian Dan Hapus Jadwal

### 19. Activity Diagram Proses Optimasi Jadwal Kuliah



Gambar 4.24 Activity Diagram Proses Optimasi Jadwal Kuliah

#### 4.2.4 Class Diagram

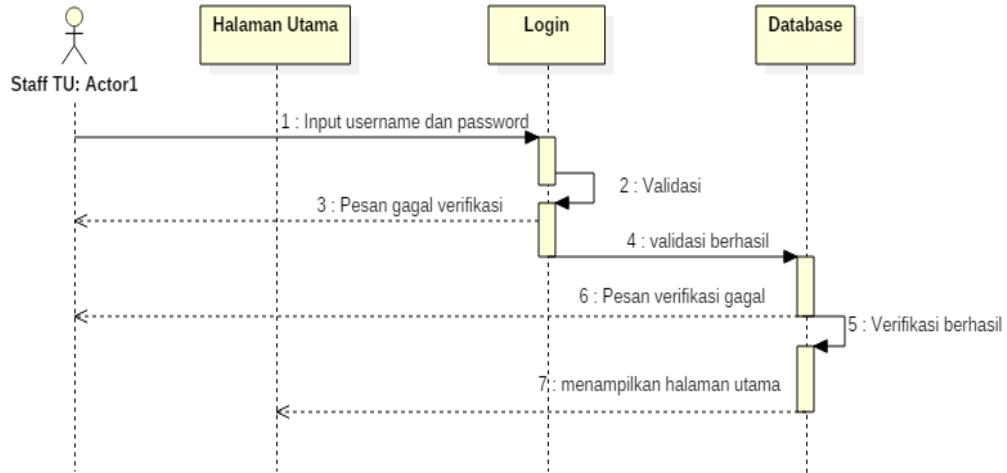


Gambar 4.25 Class Diagram Penjadwalan Yang Diusulkan

#### 4.2.5 Sequence Diagram

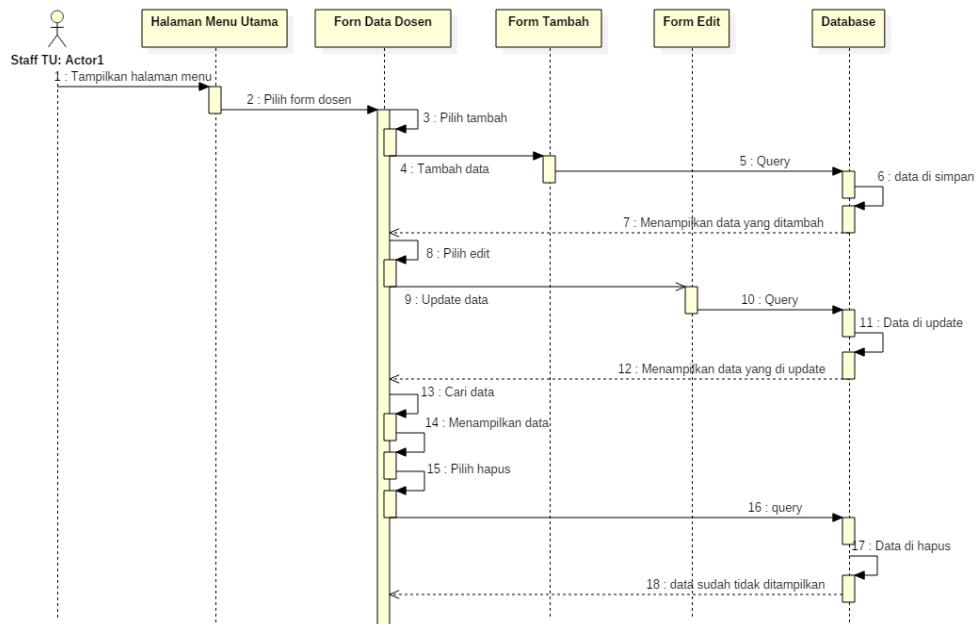
Berikut adalah *sequence diagram* untuk menunjukkan aliran pesan pada aplikasi penjadwalan.

##### 1. Sequence Diagram Login Sistem



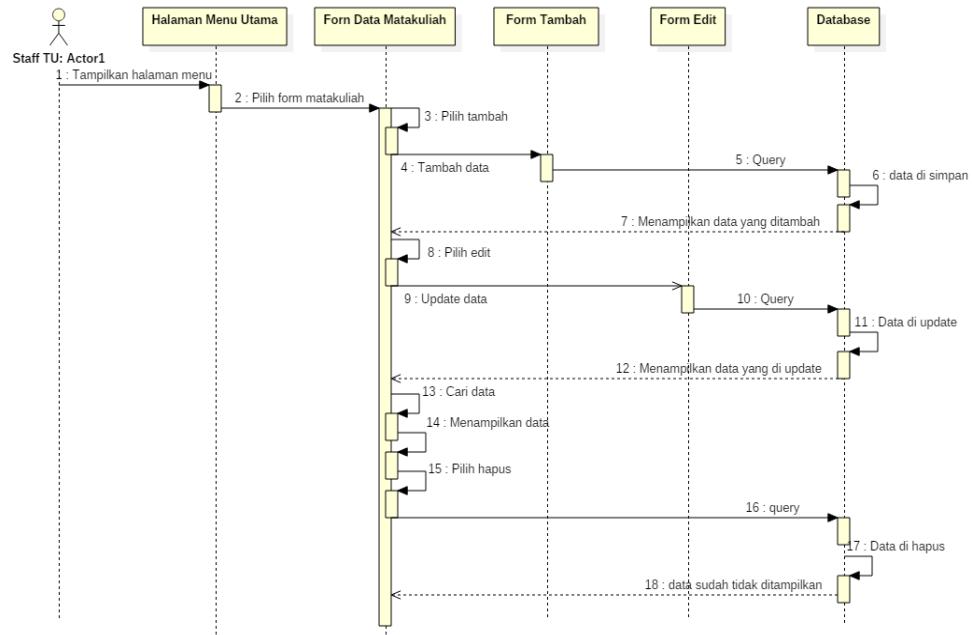
Gambar 4.26 Sequence Diagram Login

##### 2. Sequence Diagram Pengelolaan Data Dosen



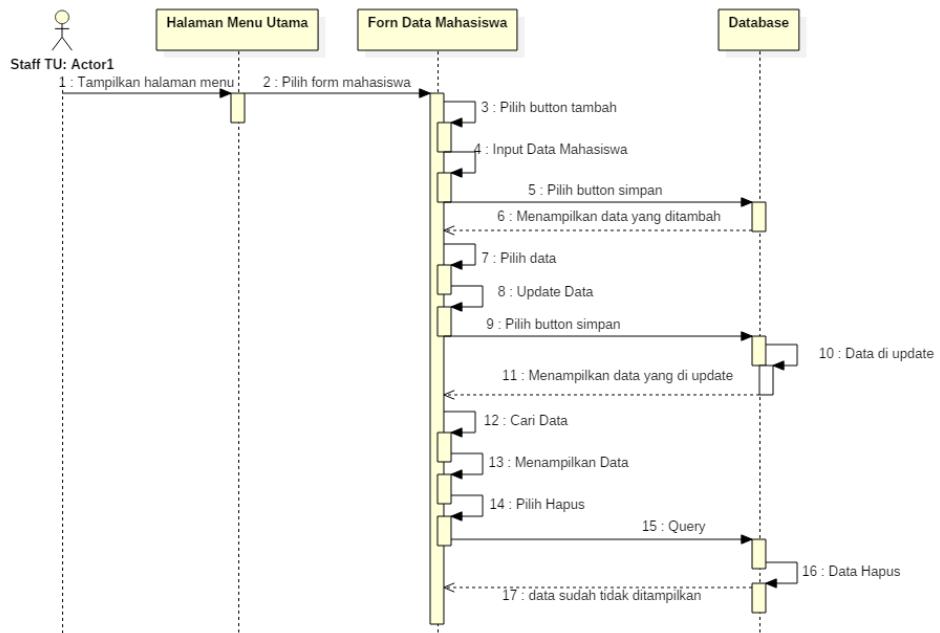
Gambar 4.27 Sequence Diagram Input Dosen

### 3. Sequence Diagram Pengelolaan Data Matakuliah



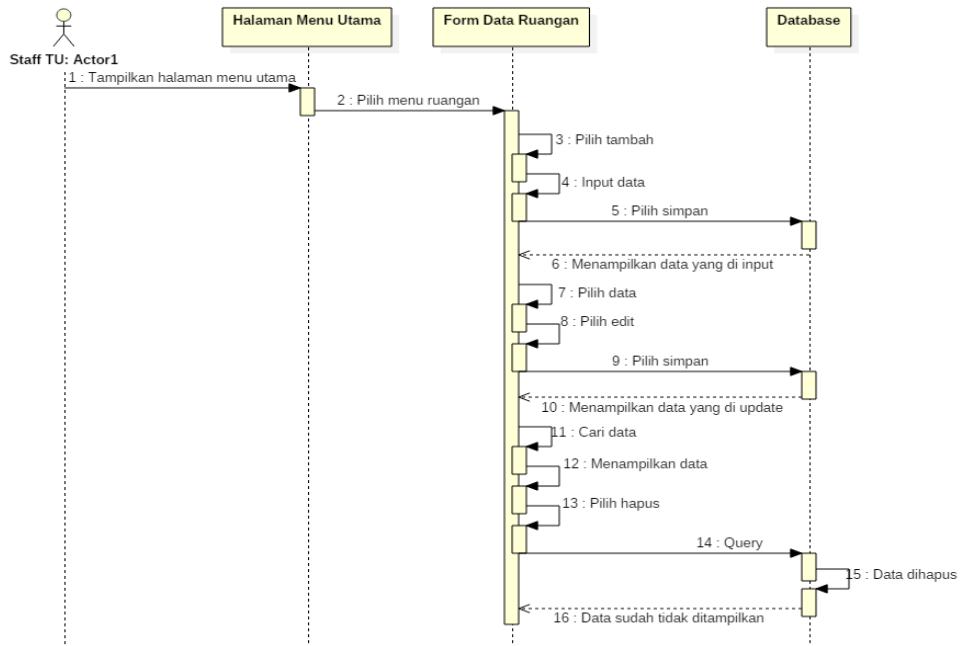
Gambar 4.28 Sequence Diagram Pengelolaan Matakuliah

### 4. Sequence Diagram Pengelolaan Data Mahasiswa



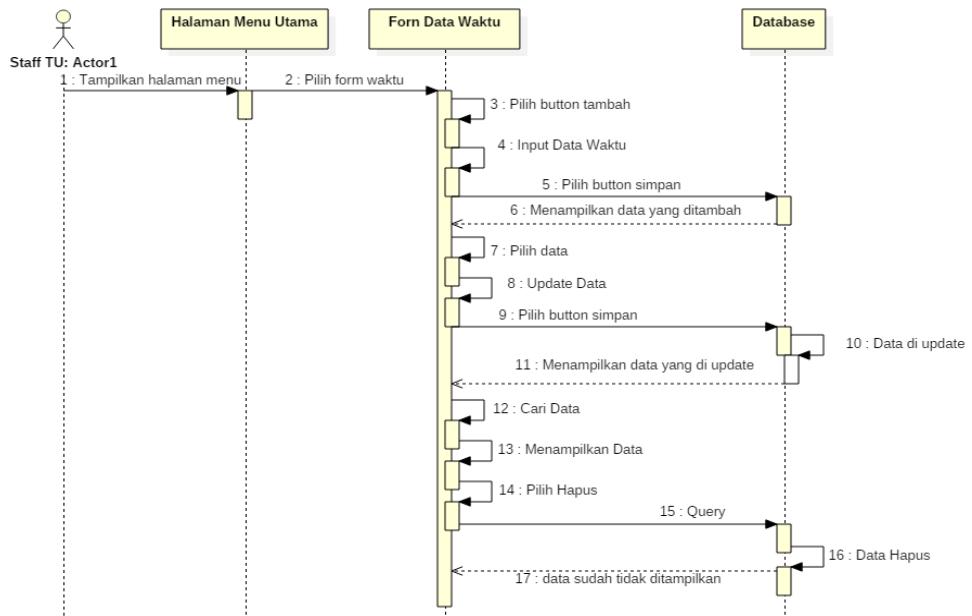
Gambar 4.29 Sequence Diagram Pengelolaan Mahasiswa

## 5. Sequence Diagram Pengelolaan Data Ruangan



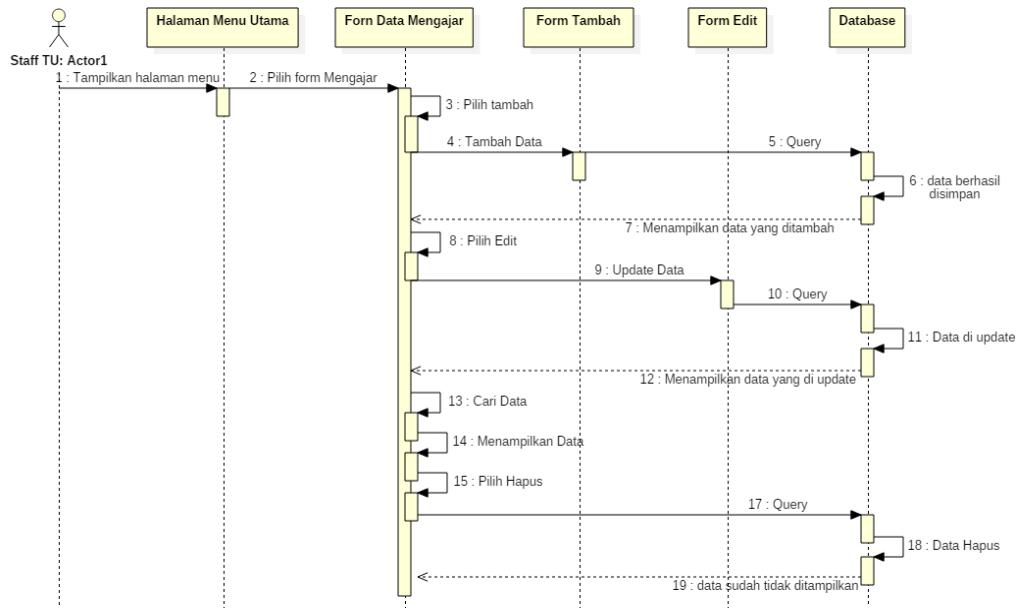
Gambar 4.30 Sequence Diagram Pengelolaan Ruangan

## 6. Sequence Diagram Pengelolaan Data Waktu



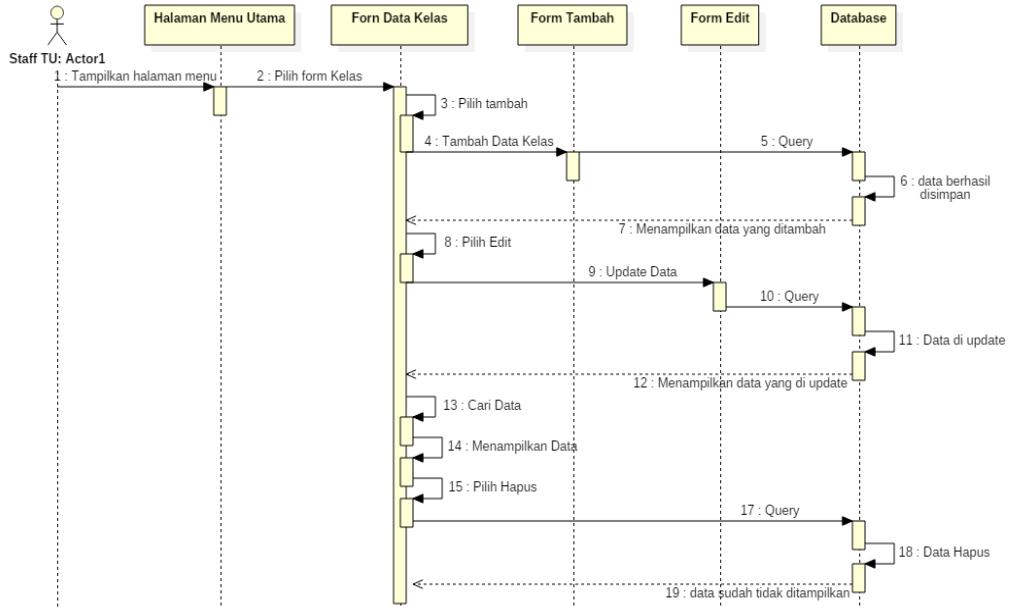
Gambar 4.31 Sequence Diagram Pengelolaan Waktu

## 7. Sequence Diagram Data Kesediaan mengajar



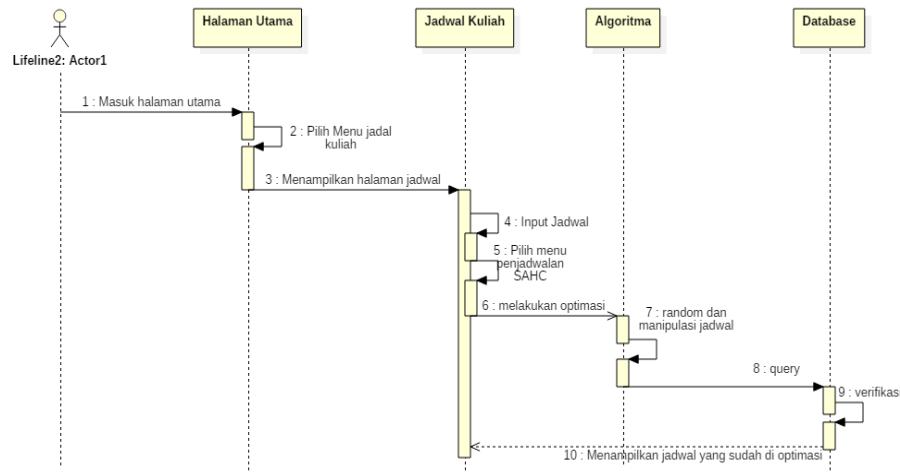
Gambar 4.32 Sequence Diagrm Data Kesediaan mengajar

## 8. Sequence Diagram Data Kelas



Gambar 4.33 Sequnce Diagram Pengelolaan Kelas.

## 9. Sequence Diagram Penjadwalan Steepest Ascent Hill Climbing



Gambar 4.34 Sequnce Diagram Penjadwalan SAHC.

### 4.2.6 Struktur Tabel

Struktur tabel yang digunakan pada pembangunan aplikasi ini adalah:

Tabel 4.13 Struktur Tabel User

Nama Field	Type	Size	Keterangan
id	int	11	Id User
username	varchar	25	Username
password	varchar	10	Password
status	varchar	10	Aktif/ Non aktif
last_login	datetime	3	Waktu Login Terakhir

Tabel 4.14 Struktur Tabel Dosen

Nama Field	Type	Size	Keterangan
nidn	varchar	10	Nomor Induk Dosen
nama_dosen	varchar	50	Nama dosen
jk	varchar	2	Jenis Kelamin

kd_prodi	varchar	8	Kode Program Studi
status	varchar	3	Aktif/Tidak
email	varchar	40	Alamat Email

Tabel 4.15 Struktur Tabel Matakuliah

Nama Field	Type	Size	Keterangan
kd_matkul	varchar	8	Kode Mata Kuliah
nama_matkul	varchar	40	Nama Mata Kuliah
sks	varchar	2	Satuan Kredit Semester
sem	varchar	6	Ganjil/ Genap

Tabel 4.16 Struktur Tabel Mahasiswa (MPS)

Nama Field	Type	Size	Keterangan
kd_mps	varchar	5	Kode Mahasiswa Program Studi
prodi	varchar	10	Program Studi
sem	varchar	40	Ganjil/ Genap

Tabel 4.17 Struktur Tabel Kesediaan Mengajar

Nama Field	Type	Size	Keterangan
kd_mengajar	varchar	8	Kode Kesediaan Mengajar
nidn	varchar	40	Nomor Induk Dosen
kd_matkul	varchar	2	Kode Mata Kuliah
hari	varchar	6	Kesediaan Hari
jam	varchar	8	Kesediaan Waktu

Tabel 4.18 Struktur Tabel Ruangan

Nama Field	Type	Size	Keterangan
kd_ruang	varchar	3	Kode ruangan
jns_ruangan	varchar	30	Nama ruangan

Tabel 4.19 Struktur Tabel Hari

Nama Field	Type	Size	Keterangan
kd_jam	varchar	4	Kode Jam
hari	varchar	6	Hari
jam	Time	8	Jam

Tabel 4.20 Struktur Tabel Kelas

Nama Field	Type	Size	Keterangan
kd_kelas	varchar	25	Kode Kelas
kd_matkul	varchar	8	Kode Matakuliah
nidn	varchar	10	Nomor Induk Dosen
kd_mps	varchar	4	Kode Mahasiswa Program Studi
sem	varchar	6	Ganjil/ Genap
f_aktif	varchar	1	T/F

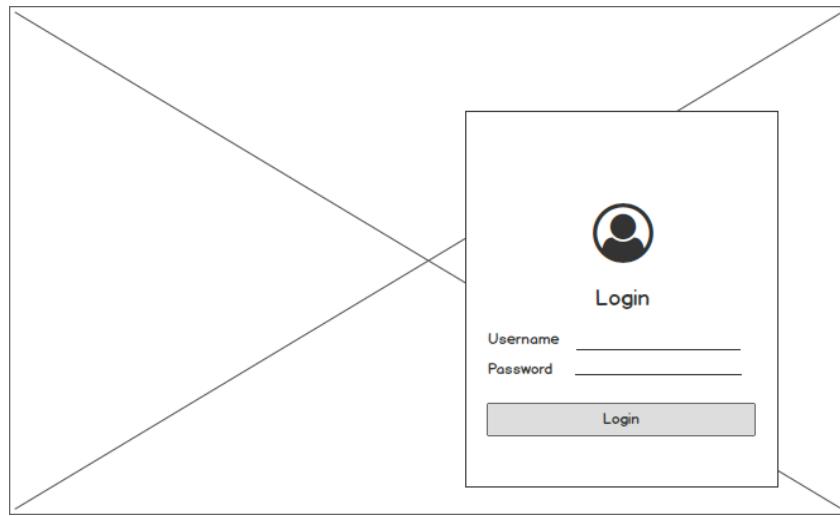
Tabel 4.21 Struktur Tabel Jadwal

Nama Field	Type	Size	Keterangan
kd_jadwal	int	11	Kode Jadwal
kd_kelas	varchar	25	Kode Kelas
kd_waktu	varchar	4	Kode Waktu
kd_ruang	varchar	4	Kode Ruangan
bobot	varchar	3	Nilai Bobot Dosen

#### 4.2.7 Perancangan User Interface

##### 1. Desain Form Login

Pada halaman ini Staff TU harus melakukan proses login dengan memasukan username dan password yang sudah terdaftar sebelumnya pada *database* agar dapat masuk dan mengakses aplikasi.



Gambar 4.35 Desain Form Login

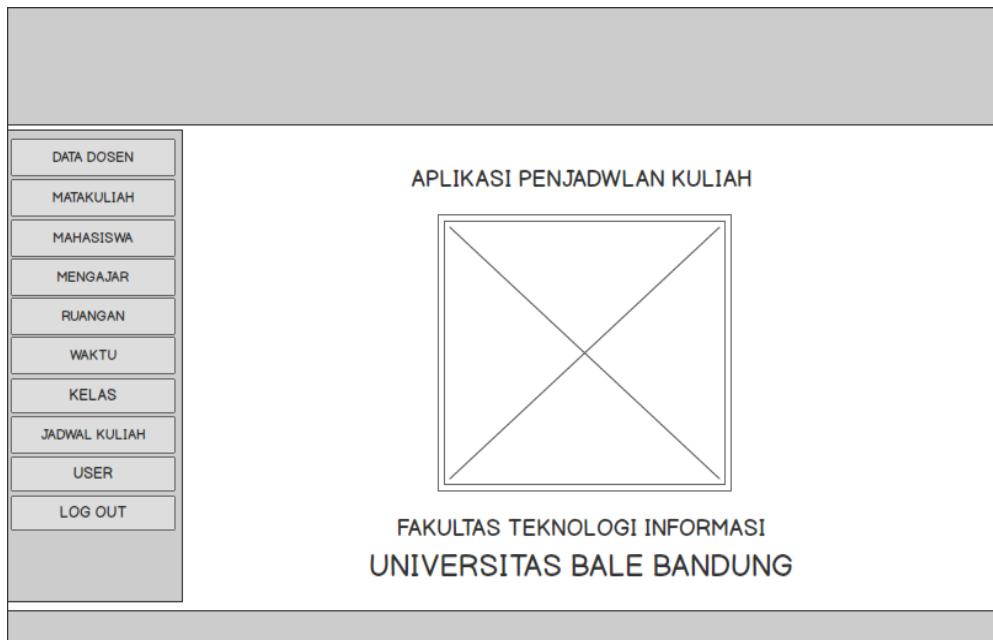
Keterangan fungsi *button*:

- *Button* Login berfungsi untuk mengakses aplikasi dan memanggil halaman menu utama.

## 2. Desain Form Halaman Menu Utama

Pada perancangan *interface* halaman menu utama dapat dilihat pada Gambar 4.36. Halaman utama ini akan ditampilkan setelah admin berhasil melakukan login. Pada halaman ini terdapat menu yang disediakan diantaranya:

- 1). Menu Data Dosen
- 2). Menu Data Matakuliah
- 3). Menu Data Mahasiswa
- 4). Menu Data Kesediaan Dosen Mengajar
- 5). Menu Data Ruangan
- 6). Menu Data Waktu
- 7). Menu Data Jadwal
- 8). Menu Data Kelas
- 9). Menu Data User
- 10). Menu Log Out



Gambar 4.36 Desain Form Menu Utama

Keterangan fungsi *button*:

- Menu Data Dosen, menampilkan halaman pengelolaan data dosen
- Menu Matakuliah, menampilkan halaman pengelolaan data matakuliah.
- Menu Mahasiswa, menampilkan halaman pengelolaan data mahasiswa.
- Menu Kesediaan Mengajar, menampilkan halaman pengelolaan data kesediaan dosen mengajar.
- Menu Ruangan, menampilkan halaman pengelolaan data ruangan yang tersedia.
- Menu Waktu, menampilkan halaman pengelolaan data waktu yang ditentukan.
- Menu Kelas, menampilkan halaman pengelolaan kelas mengajar.
- Menu Jadwal Kuliah, menampilkan halaman pengelolaan data jadwal perkuliahan.
- Menu *User*, menampilkan halaman pengelolaan data user
- Menu *Log Out*, digunakan untuk keluar dari aplikasi.

### 3. Desain Form Pengelolaan Data Dosen

Pada halaman ini Staff TU dapat mengelola data dosen diantaranya menambah, mengubah, menghapus serta mencari data dosen. Perancangan halaman pengelolaan data dosen dapat dilihat pada Gambar 4.37.

NIDN	Nama	Jk	Prodi	Status	Email

Gambar 4.37 Desain Form Pengelolaan Dosen

Keterangan fungsi *button*:

- *Button Halaman Menu Utama*, kembali ke halaman utama.
- *Button Tambah*, menampilkan form input data untuk menambah data dosen.
- *Button Edit*, menampilkan form input data untuk mengubah data dosen.
- *Button Hapus*, menghapus data dosen yang telah dipilih.
- *Button Refresh*, untuk memperbarui data dosen pada tabel setelah di input dan update.
- *Button Cari*, mencari data dosen yang telah di input pada textfield.
- *Button Print*. Digunakan untuk mencetak data.

#### 4. Desain Form Input Data Dosen

Perancangan form input data dosen dapat di lihat pada Gambar 4.38. Form input ini akan ditampilkan dengan mengklik *button* tambah pada halaman pengelolaan dosen, digunakan untuk menginput, mengedit dan menyimpan data dosen yang ditambahkan atau data yang di edit.

The image shows a user interface for 'Input Dosen' (Input Teacher). It consists of several input fields and buttons. At the top is a title 'Input Dosen'. Below it are fields for 'NIDN' (with a text input box), 'Nama' (with a text input box), and 'Jenis Kelamin' (with two radio button options: 'Laki\_laki' and 'Perempuan'). There is also a dropdown menu for 'Program Studi' with 'PILIH' as the default value. Below these are fields for 'Status' (text input box) and 'Email' (text input box). At the bottom are two buttons: 'Simpan' (Save) and 'Kembali' (Back).

Gambar 4.38 Desain Form Input Dosen

Keterangan fungsi *button*:

- *Button Simpan*, untuk menyimpan data dosen yang di input atau di update ke dalam *database*.
- *Button Kembali*, untuk menutup form input dosen.

#### 5. Desain Form Pengelolaan Data Matakuliah

Pada halaman ini Staff TU dapat mengelola data matakuliah diantaranya menambah, mengubah, menghapus serta mencari data matakuliah.

Perancangan halaman pengelolaan data matakuliah dapat dilihat pada Gambar 4.39.

Kode	Mata Kuliah	SKS	Semester

Gambar 4.39 Desain Form Matakuliah

Keterangan fungsi *button*:

- *Button Halaman Menu Utama*, kembali ke halaman utama.
- *Button Tambah*, menampilkan form input data untuk menambah data matakuliah.
- *Button Edit*, Menampilkan form input data untuk mengubah data matakuliah.
- *Button Hapus*, menghapus data matakuliah yang telah dipilih.
- *Button Refresh*, untuk memperbaharui data matakuliah pada tabel setelah di input dan update.
- *Button Cari*, mencari data matakuliah yang telah di input pada textfield.
- *Button Print*, untuk mencetak data.

## 6. Desain Form Input Matakuliah

Perancangan form input data matakuliah dapat di lihat pada Gambar 4.40. Form input ini akan ditampilkan dengan mengklik button tambah pada halaman pengelolaan matakuliah, digunakan untuk menginput, mengedit dan menyimpan data matakuliah yang ditambahkan atau di edit.



The image shows a window titled "Input Matakuliah". It contains four text input fields labeled "Kode", "Matakuliah", "SKS", and "Semester". Below the input fields are two buttons: "Simpan" and "Kembali".

Input Matakuliah		
Kode	:	<input type="text"/>
Matakuliah	:	<input type="text"/>
SKS	:	<input type="text"/>
Semester	:	<input type="text"/>
		<input type="button" value="Simpan"/> <input type="button" value="Kembali"/>

Gambar 4.40 Desain Form Input Matakuliah

Keterangan fungsi *button*:

- *Button Simpan*, untuk menyimpan data dosen yang di input dan di update ke dalam *database*.
- *Button Kembali* , untuk menutup form input matakuliah

## 7. Desain Form Pengelolaan Data Mahasiswa Program Studi (MPS)

Perancangan halaman pengelolaan data mahasiswa dapat di lihat pada Gambar 4.41. Data ini berisi mahasiswa program studi (MPS), pada halaman ini Staff TU dapat mengelola data mahasiswa diantaranya menambah, mengubah, menghapus serta mencari data.

Gambar 4.41 Desain Form Mahasiswa

Keterangan fungsi *button*:

- *Button* Halaman Menu Utama, kembali ke halaman utama.
- *Button* Tambah, untuk menambah data mahasiswa.
- *Button* Edit, untuk mengubah data ruangan yang sudah ada.
- *Button* Simpan, digunakan untuk menyimpan data mahasiswa yang telah di input dan di update ke dalam *database*.
- *Button* Hapus, menghapus data ruangan yang dipilih.
- *Button* Cari, mencari data ruangan yang telah di input pada *textfield*.

#### 8. Desain Form Pengelolaan Data Kesediaaan Mengajar Dosen

Perancangan halaman pengelolaan data kesediaan mengajar dosen dapat di lihat pada Gambar 4.42. Pada halaman ini Staff TU dapat mengelola data kesediaan mengajar yang diusulkan oleh setiap dosen pengajar, pengelolaan pada halaman ini diantaranya menambah, mengubah, menghapus serta mencari data.

The screenshot shows a user interface for managing teaching availability. At the top, there's a header with the application name 'APLIKASI PENJADWLAN KULIAH' and 'PENGELOLAAN KESEDIAAN MENGAJAR'. To the left of the header is the 'Logo FTI'. Below the header, a search bar has the placeholder 'Halaman Menu Utama'. The main content area is titled 'DATA KESEDIAAN MENGAJAR DOSEN'. It features a table with five columns: 'Kode', 'NIDN', 'Kode Matakuliah', 'Hari', and 'Jam'. There are ten rows in the table. Below the table are several buttons: 'Print', 'Refresh', 'Tambah', 'Edit', and 'Hapus'. Above the 'Edit' and 'Hapus' buttons is a search field with a 'Cari' button next to it.

Gambar 4.42 Desain Form Kesediaan Mengajar Dosen

Keterangan fungsi *button*:

- *Button* Halaman Menu Utama, kembali ke halaman utama.
- *Button* Tambah, menampilkan form input data untuk menambah data kesediaan mengajar.
- *Button* Edit, Menampilkan form input data untuk mengubah data kesediaan mengajar.
- *Button* Hapus, menghapus data kesediaan mengajar yang telah dipilih.
- *Button* Refresh, untuk memperbaharui data kesediaan mengajar pada tabel setelah di input dan update..
- *Button* Cari, mencari data yang telah di input pada textfield.
- *Button* Print, untuk mencetak data.

## 9. Desain Form Input Mengajar

Perancangan form input data kesediaan mengajar dosen dapat di lihat pada Gambar 4.43. Form input ini akan ditampilkan dengan mengklik *button* tambah pada halaman pengelolaan kesediaan mengajar dosen, digunakan untuk menginput mengedit dan menyimpan data kesediaan mengajar yang ditambahkan.

**Input Mengajar Dosen**

KMD : \_\_\_\_\_

NIDN : Pilih ▾

Kode Matakuliah : Pilih ▾

Hari : Pilih ▾

Jam : Pilih ▾

**Simpan** **Kembali**

Gambar 4.43 Desain Form Input Mengajar

Keterangan fungsi *button* dan *combo box* :

- *Button Simpan*, untuk menyimpan data mengajar yang di input dan di update ke dalam *database*.
- *Button Kembali* , untuk menutup form input kesediaan mengajar
- *Combo Box* pada Kode Matakuliah, untuk memilih kode matakuliah
- *Combo Box* NIDN, untuk memilih Nomer Induk Dosen
- *Combo Box* Hari, untuk memilih dan menginput hari
- *Combo Box* Kode MPS, untuk memilih dan menginput jam.

#### 10. Desain Form Pengelolaan Data Ruangan

Perancangan halaman pengelolaan data ruangan dapat di lihat pada Gambar 4.44. Pada halaman ini Staff TU dapat mengelola data ruangan yang yang tersedia, pengelolaan pada halaman ini diantaranya menambah, mengubah, menghapus serta mencari data.

The screenshot shows a Java Swing application window. In the top left corner is the 'Logo FTI'. The top center contains the text 'APLIKASI PENJADWALAN KULIAH', 'FAKULTAS TEKNOLOGI INFORMASI', and 'PENGELOLAAN RUANGAN'. In the top right corner is a button labeled 'Halaman Menu Utama'. Below the header, the title 'DATA RUANGAN' is displayed. On the left is a table with two columns: 'KODE' and 'Ruangan'. On the right is a form panel containing fields for 'Kode' and 'Ruangan', and buttons for 'Cari', 'Tambah', 'Simpan', 'Edit', and 'Hapus'.

Gambar 4.44 Desain Form Pengelolaan Data Ruangan

Keterangan fungsi *button*:

- *Button* Halaman Menu Utama, kembali ke halaman utama.
- *Button* Tambah, untuk menambah data.
- *Button* Edit, untuk mengubah data yang sudah ada.
- *Button* Simpan, digunakan untuk menyimpan data yang telah di input dan diubah ke dalam database.
- *Button* Hapus, menghapus data ruangan yang dipilih.
- *Button* Cari, mencari data yang telah di input pada textfield.

#### 11. Desain Form Pengelolaan Data Waktu

Perancangan halaman pengelolaan data waktu dapat di lihat pada Gambar 4.45. Pada halaman ini Staff TU dapat mengelola data waktu yang sudah ditetapkan, pengelolaan pada halaman ini diantaranya menambah, mengubah, menghapus serta mencari data.

The screenshot shows a Java Swing application window. At the top left is the 'Logo FTI'. The title bar reads 'APLIKASI PENJADWALAN KULIAH' and 'FAKULTAS TEKNOLOGI INFORMASI' with 'PENGELOLAAN WAKTU' below it. On the right side of the title bar is a button labeled 'Halaman Menu Utama'. Below the title bar, the main area has a section titled 'DATA WAKTU' containing a table with three columns: 'KWKT', 'Hari', and 'Jam'. To the right of the table is a vertical panel for managing data. It includes a search field with a 'Cari' button, and input fields for 'Kode', 'Hari', and 'Jam'. Below these are buttons for 'Tambah', 'Simpan', 'Edit', and 'Hapus'.

Gambar 4.45 Desain Form Waktu

Keterangan fungsi *button*:

- *Button* Halaman Menu Utama, kembali ke halaman utama.
- *Button* Tambah, untuk menambah data.
- *Button* Edit, untuk mengubah data yang sudah ada.
- *Button* Simpan, digunakan untuk menyimpan data yang telah di input dan diubah ke dalam database.
- *Button* Hapus, menghapus data yang dipilih.
- *Button* Cari, mencari data yang telah di input pada textfield.

## 12. Desain Form Pengelolaan Data Kelas

Perancangan halaman pengelolaan data kelas mengajar dapat di lihat pada Gambar 4.46. Pada halaman ini Staff TU dapat mengelola data kelas yang akan disusun berdasaran kesediaan mengajar dosen, pengelolaaan pada halaman ini diantaranya menambah, mengubah, menghapus serta mencari data.

Kode Kelas	Kode Matakuliah	NIDN	Kode MPS	KTA	Aktif/ Non Aktif

Gambar 4.46 Desain Form Pengelolaan Data Kelas

Keterangan fungsi *button*:

- *Button Halaman Menu Utama*, kembali ke halaman utama.
- *Button Tambah*, untuk menambah data kelas.
- *Button Edit*, untuk mengubah data waktu yang sudah ada.
- *Button Simpan*, digunakan untuk menyimpan data yang telah di input dan diubah ke dalam database.
- *Button Hapus*, menghapus data kelas yang dipilih.
- *Button Refresh*, untuk memperbaharui data pada tabel setelah di input dan di update.
- *Button Cari*, mencari data.

### 13. Desain Form Input Kelas

Perancangan form input data kelas dapat di lihat pada Gambar 4.47. Form input ini akan ditampilkan dengan mengklik *button tambah* pada halaman pengelolaan kelas yang berisi data kelas mengajar masing-masing dosen, digunakan untuk menginput dan menyimpan data kelas yang ditambahkan dan di update.

The image shows a user interface titled "Input Kelas". It contains five input fields: "Kode Kelas" (text input), "Kode Matakuliah" (dropdown menu with "Pilih" option), "NIDN" (dropdown menu with "Pilih" option), "Kode MPS" (dropdown menu with "Pilih" option), and "KTA" (text input). Below these fields are two buttons: "Simpan" and "Kembali".

Gambar 4.47 Desain Form Input Kelas

Keterangan fungsi *button* dan *combo box*:

- *Button Simpan*, untuk menyimpan data kelas yang di input dan di update ke dalam database.
- *Button Kembali*, untuk menutup form input kelas.
- *Combo Box Kode Matakuliah*, untuk memilih kode matakuliah.
- *Combo Box NIDN*, untuk memilih Nomor Induk Dosen.
- *Combo Box Kode MPS*, untuk memilih kode MPS.

#### 14. Desain Form Pengelolaan Data User

Perancangan halaman pengelolaan data user dapat di lihat pada Gambar 4.48. Pada halaman ini Staff TU dapat mengelola data user yang sudah diatur, Staff TU dapat mengatur user mana saja yang akan di aktifkan/ di non aktifkan, pengelolaan pada halaman ini diantaranya menambah, mengubah, menghapus serta mencari data user.

The screenshot displays a user management application window. At the top left is the Logo FTI. The title bar reads "APLIKASI PENJADWALAN KULIAH", "FAKULTAS TEKNOLOGI INFORMASI", and "PENGELOLAAN USER". A button "Halaman Menu Utama" is located in the top right corner. The main content area is titled "DATA USER" and contains a table with four columns: "ID", "Username", "Status Aktif", and "Last Login". Below the table are several buttons: "Refresh", "Tambah", "Ubah Password", "Set Aktif/ Non Aktif", and "Cari".

Gambar 4.48 Desain Form Pengelolaan Data User

Keterangan fungsi *button* dan *combo box*:

- *Button Halaman Menu Utama*, kembali ke halaman utama.
- *Button Tambah*, untuk menambah data user.
- *Button Ubah Password*, untuk mengganti password user.
- *Button Refresh*, untuk memperbaharui data user pada tabel setelah di input, update password dan dihapus.
- *Button Set Aktif/ Non Aktif*, untuk mengatur user yang akan diaktifkan atau di non aktifkan.
- *Button Cari*, mencari data user yang telah di input pada textfield.

## 15. Desain Form Input Data User

Perancangan form input data matakuliah dapat di lihat pada Gambar 4.49. Form input ini akan ditampilkan dengan mengklik button tambah pada halaman pengelolaan matakuliah , digunakan untuk menginput dan menyimpan data yang ditambahkan.

The image shows a user input form titled "Form User". It contains three text input fields: "Username : \_\_\_\_\_", "Password : \_\_\_\_\_", and "Ulangi Password : \_\_\_\_\_". Below the fields are two buttons: "Simpan" and "Kembali".

Gambar 4.49 Desain Form Input Data User

Keterangan fungsi *button*:

- *Button Simpan*, untuk menyimpan data mengajar yang di input dan di update ke dalam *database*.
- *Button Kembali* , untuk menutup form input kesediaan mengajar

#### 16. Desain Form Input Jadwal Kuliah

Form input jadwal kuliah dapat di lihat pada Gambar 4.50. Form input ini akan ditampilkan dengan mengklik *button* tambah pada halaman pengelolaan jadwal kuliah digunakan untuk menginput dan menyimpan data yang ditambahkan.

The image shows a form titled "Input Jadwal Kuliah". It contains five input fields: "Kode Jadwal : \_\_\_\_\_", "Kode Kelas : Pilih (dropdown)", "Waktu : Pilih (dropdown)", "Ruang : Pilih (dropdown)", and "Bobot : \_\_\_\_\_". Below the fields are two buttons: "Simpan" and "Kembali".

Gambar 4.50 Desain Form Input Jadwal Kuliah

Keterangan fungsi *button*:

- *Button Simpan*, untuk menyimpan data jadwal kuliah yang di input dan di update ke dalam *database*.
- *Button Kembali*, untuk menutup form input jadwal kuliah.
- *Combo Box* Kode kelas, untuk memilih kode kelas.
- *Combo Box* Waktu, untuk memilih kode waktu.
- *Combo Box* Ruang, untuk memilih kode ruang.

## 17. Form Pengelolaan Jadwal Kuliah

Pada perancangan halaman pengelolaan jadwal kuliah dapat di lihat pada Gambar 4.51. Halaman ini untuk menginput jadwal secara manual dan acak yang selanjutnya dilakukan proses optimasi penjadwalan dengan *algoritma steepest ascent hill climbing* untuk menghitung nilai bobot pada masing-masing pengajar.

KODE JADWAL	KODE KELAS	KODE WAKTU	KODE RUANG	BOBOT

Gambar 4.51 Desain Form Pengelolaan Jadwal Kuliah

Keterangan fungsi *button*:

- *Button Halaman Menu Utama*, kembali ke halaman utama.
- *Button Tambah*, untuk menambah data jadwal.

- *Button Edit*, untuk mengubah data jadwal yang sudah ada.
- *Button Simpan*, digunakan untuk menyimpan data yang telah di input dan diubah ke dalam database.
- *Button Hapus*, menghapus data kelas yang dipilih.
- *Button Refresh*, untuk memperbaharui data pada tabel setelah di input, update dan dihapus.
- *Button Penjadwalan*, untuk melakukan proses pebnjadwalan dengan algoritma.
- *Button Print*, mencetak jadwal.
- *Button Cari*, mencari data jadwal.

## **BAB V**

### **IMPLEMENTASI DAN PENGUJIAN**

#### **5.1 Implementasi**

Implementasi ini dilakukan untuk pengujian dari aplikasi penjadwalan kuliah yang telah dibuat dengan menggunakan metode *blackbox*. Tempat Implementasi di Fakultas Teknologi Informasi di Universitas Bale Bandung Jl. R.A.A Wiranatakusumah No. 7 Baleendah.

##### **5.1.1 Spesifikasi Sistem**

Spesifikasi system yang digunakan untuk membangun serta menjalankan aplikasi penjadwalan kuliah di Fakultas Teknologi Informasi adalah sebagai berikut:

i. Minimum Kebutuhan Perangkat Keras

Minimum perangkat keras yang harus dipenuhi, agar dapat menjalankan aplikasi penjadwalan kuliah dengan perangkat keras yang sesuai. Semakin tinggi spesifikasi perangkat keras yang digunakan, maka akan semakin baik pada saat menjalankan aplikasi.

1. Komputer atau Laptop

Komputer atau laptop yang digunakan oleh peneliti dalam pembuatan aplikasi penjadwalan kuliah di Fakultas Teknologi Informasi adalah dengan spesifikasi sebagai berikut:

Tabel 5.1 Perangkat Keras Yang Digunakan

Komputer PC	: HP
Monitor	Samsung LED
Processor	: Intel(R) Core(TM)2 Duo CPU E8400 @ 3.00GHz (2 CPUs), ~3.0GHz
Memory	: 4096 MB RAM

Harddisk	: 500 GB
Name	: Intel(R) HD Graphics

Sedangkan untuk minimum spesifikasi perangkat keras dalam menjalankan aplikasi penjadwalan kuliah agar dapat berjalan adalah sebagai berikut:

Tabel 5.2 Minimum Perangkat Keras

OS	: Windows XP, Windows 7 ( <i>Recommended</i> )
Processor	: Intel Pentium 4 Celeron
Memory	: 2048 MB RAM
Harddisk	: 125 GB

#### j. Implementasi Perangkat Lunak

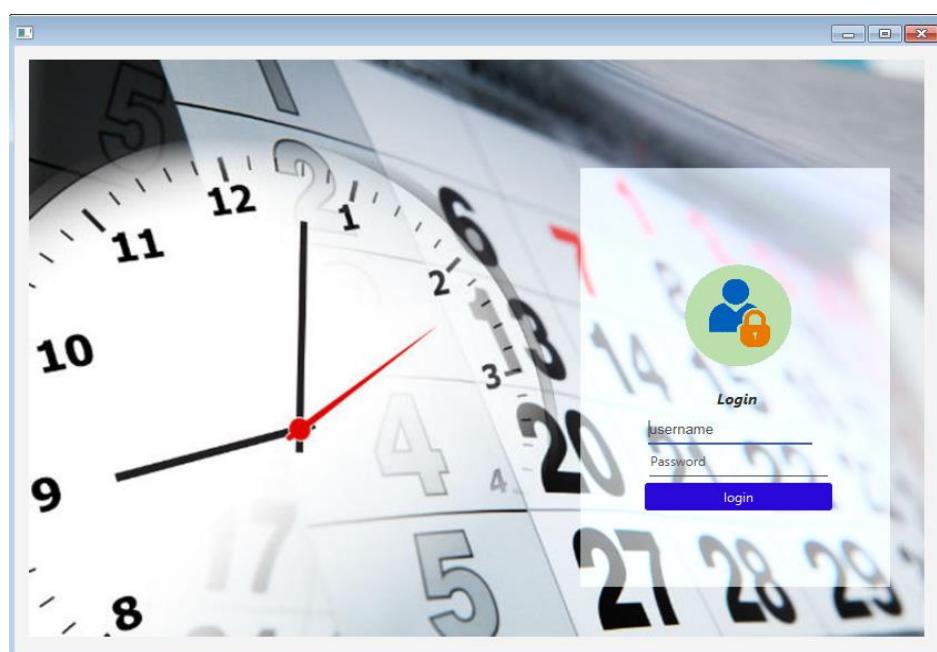
Perangkat lunak yang digunakan untuk membangun aplikasi penjadwalan kuliah di Fakultas Teknologi Informasi Universitas Bale Bandung adalah sebagai berikut:

1. Sistem Operasi Windows 7 Home Premium 64-bit
2. Bahasa pemrograman Java untuk
3. XAMPP v3.2.2 control panel untuk penyimpanan database MySQL
4. NetBeans, digunakan untuk membuat dan mengembangkan sebuah aplikasi pencarian ini dengan bahasa java.
5. Google Chrome, sebagai browser untuk *preview* penggerjaan aplikasi penjadwalan kuliah yang dibangun
6. StarUML V2.8.0 sebagai pemuat diagram UML
7. Balsamic Mockups 3.5.17 sebagai perancangan *user interface*
8. Scene Builder 8.5.0 sebagai pembuatan *user interface*

### 5.1.2 Implementasi Antar Muka

Antar Muka (*User Interface*) pada aplikasi penjadwalan kuliah dengan menggunakan algoritma *steepest-ascent hill climbing* di Fakultas Teknologi Informasi Universitas Bale Bandung terdiri dari beberapa tampilan diantaranya halaman login, menu utama, pengelolaan data dosen, matakuliah, mahasiswa program studi, kesediaan mengajar, ruangan, waktu, kelas, jadwal kuliah dan pengelolaan data user. Berikut ini adalah tampilan dari antar muka aplikasi:

1. Login



Gambar 5.1 Tampilan Struktur Menu Login

Merupakan Halaman awal untuk login staff TU pada saat membuka aplikasi penjadwalan kuliah. Pada form login ini staff TU diharuskan untuk memasukan *username* dan *password* yang sebelumnya telah tandaftar yang nantinya akan diarahkan ke form halaman menu utama untuk menjalankan aplikasi.

## 2. Menu Utama



Gambar 5.2 Tampilan Halaman Menu Utama

Halaman menu utama ini merupakan tampilan awal aplikasi penjadwalan kuliah dan di dalamnya terdapat tombol menu data.

## 3. Dosen

No	NIDN	Nama	JK	Prodi	Status
3	0401988974	Ani Amaliah	P	IF	DTT *****
4	0402108204	Ceccep Suwanda, S.Si	L	IF	DTT *****
5	0407047706	Yaya Suharyana, S.Kom., M.T.	L	IF	DT *****
6	0411057607	Asep Muhammad Indra, M.Kom.	L	SI	DTT *****
7	0412086301	Mochammad Ridwan, S.T., M.Kom.	L	IF	DT *****
8	0416017704	Rustiyana, S.T., M.T.	L	IF	DT *****
9	0417768389	Deni Supriadi, S.T. M.T	L	IF	DTT *****
10	0419834800	Dede Sulaeman, S.Kom. M.Kom.	L	SI	DTT *****
11	0420988987	Deni Sopiansyah, Dr.	L	IF	DT *****
12	0422037002	Zen Munawar, M.Kom	L	IF	DT *****
13	0428027501	Yudi Herdiana, S.T., M.Kom	L	IF	DT *****
14	0435038203	Rosmalina, S.T., M.Kom.	L	SI	DT *****

Gambar 5.3 Tampilan Pengelolaan Dosen

Halaman digunakan untuk mengelola data dosen yang mengajar di Fakultas Teknologi informasi, halaman ini dapat menampilkan seluruh data.

#### 4. Input Dosen

NIDN:

Nama:

Jenis Kelamin:

Laki-laki

Perempuan

Program Studi: Pilih

Status Dosen:

Email:

**Simpan** **Kembali**

Gambar 5.4 Tampilan Input Dosen

Halaman digunakan untuk menginput data dosen, pada halaman ini diharuskan mengisi semua data pada field dan menyimpan data dengan menekan *button* simpan.

#### 5. Matakuliah

No	Kode	Matakuliah	SKS	Semester
2	FTI204	Aplikasi Dasar Komputer	2	
3	FTI207	Statistik Dan Probabilitas	2	
4	FTI210	Kalkulus 1	3	
5	FTI212	Bahasa Indonesia	2	
6	FTI307	Interaksi Manusia Dan Komputer	3	
7	FTI317	Jaringan Komputer	3	
8	FTI323	Teknik Kompilasi	3	
9	FTI327	Aljabar Linear Dan Matriks	3	
10	FTI337	Topik Khusus	3	
11	IFPRK01	Praktikum Aplikasi Dasar Komputer	2	
12	IFPRK04	Praktikum Pro Model	2	
13	IFPRK05	Praktikum Statistik & Probabilitas	2	
14	SIF326	Sistem Penunjang Keputusan	3	

Print Refresh Tambah Edit Hapus

Gambar 5.5 Tampilan Pengelolaan Matakuliah

Halaman digunakan untuk mengelola data matakuliah, halaman ini dapat menampilkan seluruh data dosen yang sudah di input. Fungsi dari form ini yaitu untuk menambah data, update data, dan hapus data matakuliah.

## 6. Input Matakuliah

KODE	<input type="text" value=" "/>
Matakuliah	<input type="text"/>
SKS	<input type="text"/>
Semester	<input type="text"/>

**Simpan** **Kembali**

Gambar 5.6 Tampilan Input Matakuliah

Halaman digunakan untuk menginput dan mengupdate data matakuliah, pada halaman ini diharuskan mengisi semua data pada field dan menyimpan data dengan menekan *button* simpan.

## 7. Mahasiswa

Kode MPS	Program Studi	Semester
IF01	IF	01
IF02	IF	02
IF03	IF	03
IF04	IF	04
IF05	IF	05
IF06	IF	06
IF07	IF	07
IF08	IF	08
SI01	SI	01
SI02	SI	02
SI03	SI	03
SI04	SI	04
SI05	SI	05

**Cari**  
**Tambah** **Simpan**  
**Edit** **Hapus**

Gambar 5.7 Tampilan Pengelolaan Mahasiswa

Halaman digunakan untuk mengelola data mahasiswa, halaman ini dapat menampilkan seluruh data mahasiswa yang sudah di input yang terrelasi ke data kesediaan mengajar. Fungsi dari form ini yaitu untuk menambah data, update data, dan hapus data mahasiswa.

## 8. Ruangan



Gambar 5.8 Tampilan Pengelolaan Ruangan

Halaman pengelolaan ruangan digunakan untuk mengelola data ruangan, halaman ini dapat menampilkan seluruh data dosen yang sudah di input. Fungsi dari form ini yaitu untuk menambah data, update data, dan hapus data ruangan.

## 9. Waktu



Gambar 5.9 Tampilan Pengelolaan Waktu

Halaman digunakan untuk mengelola data waktu yang akan atau sudah ditentukan. Fungsi dari form ini yaitu untuk menambah data, update data, dan hapus data waktu.

## 10. Kesediaan Mengajar

Kode	NIDN	Kode Matakuliah	Hari	Jam
1	0401089734	FTI204	Senin	10:00:00
2	0417768389	FTI317	Selasa	15:00:00
3	0419834800	FTI337	Senin	11:00:00
4	0401988974	FTI323	Selasa	13:00:00
5	0401089734	IFPRK05	Senin	13:00:00
6	0401057504	IFPRK01	Senin	14:00:00
7	0428027501	TIF202	Selasa	11:00:00
8	0428027501	TIF301	Rabu	13:00:00

Print Refresh Tambah Edit Hapus

Gambar 5.10 Tampilan Kesediaan Mengajar

Halaman digunakan untuk mengelola kesediaan mengajar masing-masing dosen pengajar yang menjadi acuan untuk data kelas mengajar, halaman ini dapat menampilkan seluruh data. Fungsi dari form ini yaitu untuk menambah data, update data, dan hapus data waktu.

## 11. Input Kesediaan Mengajar

**Input Mengajar Dosen**

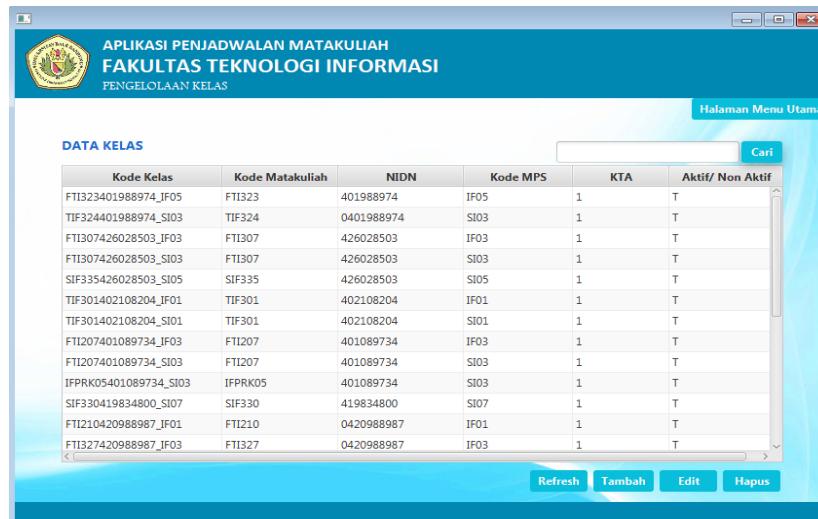
KMD	Pilih
NIDN	Pilih
Mata Kuliah	Pilih
Hari	Pilih
Waktu	Pilih

Simpan Kembali

Gambar 5.11 Input Kesediaan Mengajar

Form ini digunakan untuk menginput dan meng update data, pada halaman ini diharuskan mengisi semua data pada field dan menyimpan data dengan menekan *button* simpan.

## 12. Halaman Pengelolaan Kelas



The screenshot shows a Windows application window titled 'APLIKASI PENJADWALAN MATAKULIAH FAKULTAS TEKNOLOGI INFORMASI PENGELOLAAN KELAS'. The main title bar has the university logo and the text 'APLIKASI PENJADWALAN MATAKULIAH FAKULTAS TEKNOLOGI INFORMASI' above 'PENGELOLAAN KELAS'. Below the title bar, there is a sub-menu bar with the text 'Halaman Menu Utama'. The main content area is titled 'DATA KELAS' and contains a table with the following columns: 'Kode Kelas', 'Kode Matakuliah', 'NIDN', 'Kode MPS', 'KTA', and 'Aktif/ Non Aktif'. The table lists 18 rows of data. At the bottom of the table are four buttons: 'Refresh', 'Tambah', 'Edit', and 'Hapus'.

Kode Kelas	Kode Matakuliah	NIDN	Kode MPS	KTA	Aktif/ Non Aktif
FTI323401988974_JF05	FTI123	401988974	IF05	1	T
TIF324401988974_SI03	TIF324	0401988974	SI03	1	T
FTI307426028503_JF03	FTI307	426028503	IF03	1	T
FTI307426028503_SI03	FTI307	426028503	SI03	1	T
SIF335426028503_SI05	SIF335	426028503	SI05	1	T
TIF301402108204_JF01	TIF301	402108204	IF01	1	T
TIF301402108204_SI01	TIF301	402108204	SI01	1	T
FTI207401089734_JF03	FTI207	401089734	IF03	1	T
FTI207401089734_SI03	FTI207	401089734	SI03	1	T
IFPRK05401089734_SI03	IFPRK05	401089734	SI03	1	T
SIF330419834800_SI07	SIF330	419834800	SI07	1	T
FTI210420988987_JF01	FTI210	0420988987	IF01	1	T
FTI327420988987_JF03	FTI327	0420988987	IF03	1	T

Gambar 5.12 Halaman Pengelolaan Kelas

Halaman digunakan untuk mengelola data kelas mengajar, data pada halaman ini merupakan data yang menjadi acuan untuk jadwal kuliah masing-masing dosen pengajar. Fungsi dari form ini yaitu untuk menambah data, update data, dan hapus data waktu.

## 13. Form Input Kelas



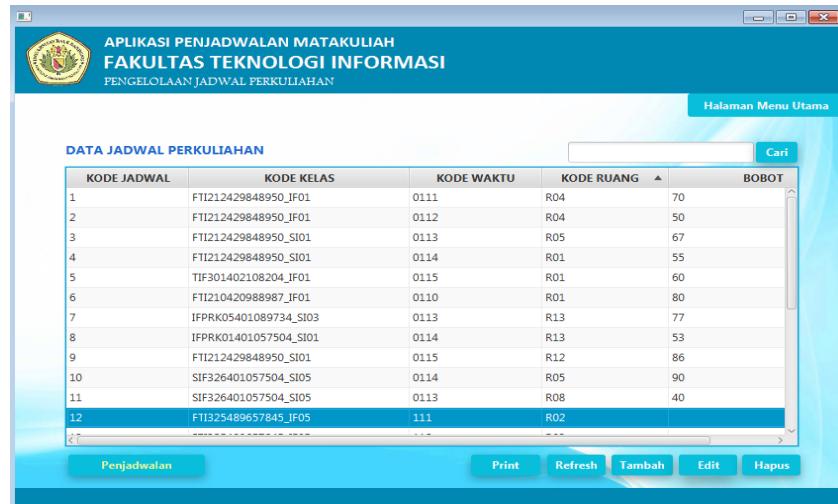
The screenshot shows a Windows application window titled 'Form Kelas'. The main title bar has the text 'Form Kelas'. Below the title bar, there is a sub-menu bar with the text 'Input Kelas Mengajar'. The main content area is titled 'Input Kelas Mengajar' and contains the following fields:

- KODE: dropdown menu labeled 'Pilih'
- Matakuliah: dropdown menu labeled 'Pilih'
- NIDN: dropdown menu labeled 'Pilih'
- MPS: dropdown menu labeled 'Pilih'
- Tahun Ajaran: dropdown menu labeled 'Pilih'
- Aktif/ Non Aktif: dropdown menu labeled 'Pilih'

At the bottom of the form are two buttons: 'Simpan' and 'Kembali'.

Gambar 5.13 Form Input Kelas

#### 14. Halaman Pengelolaan Jadwal



The screenshot shows a Windows application window titled 'APLIKASI PENJADWALAN MATAKULIAH FAKULTAS TEKNOLOGI INFORMASI PENGELOLAAN JADWAL PERKULIAHAN'. At the top right is a search bar labeled 'Cari' and a button labeled 'Halaman Menu Utama'. Below the title is a grid table with columns: KODE JADWAL, KODE KELAS, KODE WAKTU, KODE RUANG, and BOBOT. The table contains 12 rows of data. At the bottom of the grid are buttons for 'Penjadwalan', 'Print', 'Refresh', 'Tambah', 'Edit', and 'Hapus'. The last row of the table is highlighted.

KODE JADWAL	KODE KELAS	KODE WAKTU	KODE RUANG	BOBOT
1	FTI212429848950_IF01	0111	R04	70
2	FTI212429848950_IF01	0112	R04	50
3	FTI212429848950_SI01	0113	R05	67
4	FTI212429848950_SI01	0114	R01	55
5	TIF301402108204_IF01	0115	R01	60
6	FTI210420988987_IF01	0110	R01	80
7	IFPRK05401089734_SI03	0113	R13	77
8	IFPRK01401057504_SI01	0114	R13	53
9	FTI212429848950_SI01	0115	R12	86
10	SIF326401057504_SI05	0114	R05	90
11	SIF326401057504_SI05	0113	R08	40
12	FTI325489657845_IF05	111	R02	

Gambar 5.14 Halaman Pengelolaan Jadwal

Halaman digunakan untuk pengelolaan jadwal dan pada proses dimana optimasi jadwal dilakukan, halaman ini dapat menampilkan seluruh data jadwal yang sudah di input dan menampilkan jadwal hasil dari proses penjadwalan. Fungsi dari form ini yaitu untuk menambah data, update data, dan hapus jadwal.

#### 15. Form Input Jadwal



The screenshot shows a Windows application window titled 'Form Jadwal' with a sub-section titled 'Input Jadwa Kuliah'. It contains four input fields: 'Kode Jadwal' (text box), 'Kode Kelas' (dropdown menu with 'Pilih' option), 'Waktu' (dropdown menu with 'Pilih' option), and 'Ruang' (dropdown menu with 'Pilih' option). Below these fields are buttons for 'Simpan' (Save) and 'Kembali' (Back).

Gambar 5.15 Form Input Jadwal

## 16. Pengelolaan User



The screenshot shows a Windows application window titled 'APLIKASI PENJADWALAN MATAKULIAH FAKULTAS TEKNOLOGI INFORMASI PENGELOLAAN USER'. At the top right is a 'Halaman Menu Utama' button. Below it is a search bar with a 'Cari' button. The main area is titled 'DATA USER' and contains a table with the following data:

No	ID	Username	Status Aktif	Last Login
1	1	admin	AKTIF	
2	2	1	NONAKTIF	
3	3	2	AKTIF	
4	9	3	AKTIF	
5	13	4	NONAKTIF	
6	14	5	AKTIF	
7	15	budiman	AKTIF	
8	16	ridwan	NONAKTIF	
9	17	budi	AKTIF	

At the bottom are buttons for Refresh, Tambah, Ubah Password, and Set Aktif / Nonaktif.

Gambar 5.16 Tampilan Pengelolaan user

Halaman pengelolaan user ini digunakan sebagai penyimpanan data user, Staff TU dapat mengelola user, dan mengubah status user aktif atau non aktif.

## 17. Input User



The screenshot shows a Windows application window titled 'Form User'. It contains three text input fields labeled 'Username', 'Password', and 'Ulangi Password'. Below the fields are two buttons: 'Simpan' and 'Kembali'.

Gambar 5.17 Form Input User

## 5.2 Pengujian

Pengujian dilakukan untuk menjamin kualitas dan juga mengetahui kekurangan dan kelemahan dari aplikasi penjadwalan kuliah yang telah dibuat serta menjami bahwa aplikasi yang dibuat sesuai dengan yang diharapkan. Pengujian menggunakan metode *blackbox*. Pengujian *blackbox* digunakan untuk menguji fungsionalitas dari aplikasi penjadwalan yang dirancang. Berikut ini adalah pengujian dari beberapa form.

### 5.2.1 Rencana Pengujian

Rencana pengujian yang dilakukan adalah pengujian terhadap fungsionalitas yang terdapat di dalam sistem, apakah berjalan dengan baik sesuai dengan yang diharapkan atau tidak. Berikut tabel rencana pengujian pada aplikasi penjadwalan kuliah yang dibangun.

Tabel 5.3 Rencana Pengujian

No	Item	Detail Pengujian	Jenis Pengujian
1	Tampilan Login	Mengisi username dan password dan menekan <i>button</i> Login	<i>Black Box</i>
2	Halaman Menu Utama	Menekan <i>button</i> data dosen	<i>Black Box</i>
		Menekan <i>button</i> matakuliah	<i>Black Box</i>
		Menekan <i>button</i> mahasiswa	<i>Black Box</i>
		Menekan <i>button</i> mengajar	<i>Black Box</i>
		Menekan <i>button</i> ruangan	<i>Black Box</i>
		Menekan <i>button</i> kelas	<i>Black Box</i>
		Menekan <i>button</i> waktu	<i>Black Box</i>
		Menekan <i>button</i> jadwal kuliah	<i>Black Box</i>
		Menekan <i>button</i> user	<i>Black Box</i>
		Menekan <i>button</i> log out	<i>Black Box</i>
3	Data Dosen	Menekan <i>button</i> tambah	<i>Black Box</i>
		Menekan <i>button</i> edit	<i>Black Box</i>
		Menekan <i>button</i> hapus	<i>Black Box</i>

		Mengisi field cari dan menekan <i>button</i> cari	<i>Black Box</i>
		Menekan <i>button</i> refresh	<i>Black Box</i>
4	Form Tambah Dosen	Mengisi semua field inputan dan menyimpan data dengan menekan <i>button</i> simpan	<i>Black Box</i>
		Menekan <i>button</i> kembali	<i>Black Box</i>
5.	Matakuliah	Menekan <i>button</i> tambah	<i>Black Box</i>
		Menekan <i>button</i> edit	<i>Black Box</i>
		Menekan <i>button</i> hapus	<i>Black Box</i>
		Mengisi field cari dan menekan <i>button</i> cari	<i>Black Box</i>
		Menekan <i>button</i> refresh	<i>Black Box</i>
6	Form Input Matakuliah	Mengisi semua field inputan dan menyimpan data dengan menekan <i>button</i> simpan	<i>Black Box</i>
		Menekan <i>button</i> kembali	<i>Black Box</i>
7	Matahasiswa	Menekan <i>button</i> tambah	<i>Black Box</i>
		Mengisi semua field inputan	<i>Black Box</i>
		Menyimpan data inputan dengan menekan <i>button</i> simpan	<i>Black Box</i>
		Menekan <i>button</i> edit	<i>Black Box</i>
		Mengupdate data pada field inputan dan menekan <i>button</i> simpan	<i>Black Box</i>
		Input data pada field cari dan menekan tombol cari	<i>Black Box</i>
		Menghapus data dengan menekan <i>button</i> hapus	<i>Black Box</i>
		Menekan <i>button</i> refresh	<i>Black Box</i>
8	Mengajar	Menekan <i>button</i> tambah	<i>Black Box</i>
		Menekan <i>button</i> edit	<i>Black Box</i>
		Menekan <i>button</i> hapus	<i>Black Box</i>
		Mengisi field cari dan menekan <i>button</i> cari	<i>Black Box</i>
		Menekan <i>button</i> refresh	<i>Black Box</i>

	Tambah Mengajar	Mengisi semua field inputan dan menyimpan data dengan menekan <i>button simpan</i>	<i>Black Box</i>
		Menekan <i>button kembali</i>	<i>Black Box</i>
9	Ruangan	Menekan <i>button tambah</i>	<i>Black Box</i>
		Mengisi semua field inputan	<i>Black Box</i>
		Menyimpan data inputan dengan menekan <i>button simpan</i>	<i>Black Box</i>
		Menekan <i>button edit</i>	<i>Black Box</i>
		Mengupdate data pada field inputan dan menekan <i>button simpan</i>	<i>Black Box</i>
		Input data pada field cari dan menekan tombol cari	<i>Black Box</i>
		Menghapus data dengan menekan <i>button hapus</i>	<i>Black Box</i>
		Menekan <i>button refresh</i>	<i>Black Box</i>
10	Waktu	Menekan <i>button tambah</i>	<i>Black Box</i>
		Mengisi semua field inputan	<i>Black Box</i>
		Menyimpan data inputan dengan menekan <i>button simpan</i>	<i>Black Box</i>
		Menekan <i>button edit</i>	<i>Black Box</i>
		Mengupdate data pada field inputan dan menekan <i>button simpan</i>	<i>Black Box</i>
		Input data pada field cari dan menekan tombol cari	<i>Black Box</i>
		Menghapus data dengan menekan <i>button hapus</i>	<i>Black Box</i>
		Menekan <i>button refresh</i>	<i>Black Box</i>
11	Kelas	Menekan <i>button tambah</i>	<i>Black Box</i>
		Menekan <i>button edit</i>	<i>Black Box</i>
		Menekan <i>button hapus</i>	<i>Black Box</i>
		Menekan <i>button refresh</i>	<i>Black Box</i>
		Mengisi field cari dan menekan <i>button cari</i>	<i>Black Box</i>

12	Input Kelas	Mengisi semua field inputan dan menyimpan data dengan menekan <i>button simpan</i>	<i>Black Box</i>
		Menekan <i>button kembali</i>	
13	User	Menekan <i>button tambah</i>	<i>Black Box</i>
		dengan menekan <i>button ubah password</i>	<i>Black Box</i>
		Menekan <i>button set aktif/non aktif</i>	<i>Black Box</i>
		Mengupdate data pada field inputan dan menekan <i>button simpan</i>	<i>Black Box</i>
		Menghapus refresh	<i>Black Box</i>
14	Form Input User	Mengisi semua field inputan dan menyimpan data dengan menekan <i>button simpan</i>	<i>Black Box</i>
		Menekan <i>button kembali</i>	
15	Jadwal Kuliah	Menekan <i>button tambah</i>	<i>Black Box</i>
		Mengisi semua field inputan	<i>Black Box</i>
		Menyimpan data inputan dengan menekan <i>button simpan</i>	<i>Black Box</i>
		Menekan <i>button edit</i>	<i>Black Box</i>
		Mengupdate data pada field inputan dan menekan <i>button simpan</i>	<i>Black Box</i>
		Input data pada field cari dan menekan tombol cari	<i>Black Box</i>
		Menghapus data dengan menekan <i>button hapus</i>	<i>Black Box</i>
		Menekan <i>button penjadwalan SAHC</i>	<i>Black Box</i>
16	Form Input Jadwal	Mengisi semua field inputan dan menyimpan data dengan menekan <i>button simpan</i>	<i>Black Box</i>
		Menekan <i>button kembali</i>	<i>Black Box</i>
17	Log Out	Menekan menu log out	<i>Black Box</i>

### 5.2.2 Kasus dan Hasil Pengujian

Tabel 5.4 Pengujian Aplikasi

No	Item	Scenario Pengujian	Hasil yang diharapkan	Hasil Pengujian
1	Tampilan Login	Mengisi username dan password dan menekan <i>button</i> Login	Ketika inputan benar maka akan masuk ke halaman menu utama, namun jika inputan salah akan menampilkan pesan gagal.	+Berhasil
2	Halaman Menu Utama	Menekan menu Data Dosen	Ketika memilih menu data dosen maka akan menampilkan halaman pengelolaan dosen	+Berhasil
		Menekan menu matakuliah	Ketika memilih menu matakuliah maka akan menampilkan halaman pengelolaan matakuliah	+Berhasil
		Menekan menu mahasiswa	Ketika memilih menu menampilkan halaman pengelolaan mahasiswa	+Berhasil
		Menekan menu mengajar	Ketika memilih menu mengajar maka akan menampilkan halaman pengelolaan mengajar	+Berhasil
		Menekan menu ruangan	Ketika memilih menu ruangan maka akan menampilkan halaman pengelolaan ruangan	+Berhasil
		Menekan menu waktu	Ketika memilih menu waktu maka akan menampilkan halaman pengelolaan waktu	+Berhasil
		Menekan menu kelas	Ketika memilih menu waktu maka akan menampilkan halaman	+Berhasil

			pengelolaan kelas	
		Menekan menu jadwal kuliah	Ketika memilih menu jadwal kuliah maka akan menampilkan halaman jadwal kuliah	+Berhasil
		Menekan menu user	Ketika memilih menu waktu maka akan menampilkan halaman pengelolaan user	+Berhasil
		Menekan menu jadwal <i>log out</i>	Ketika memilih menu log out maka akan menampilkan menutup aplikasi	+Berhasil
3	Data Dosen	Menekan <i>button tambah</i>	Ketika memilih <i>button tambah</i> maka akan menampilkan form input	+Berhasil
		Menekan <i>button edit</i>	Ketika memilih <i>button edit</i> maka akan menampilkan form input	+Berhasil
		Menekan <i>button hapus</i>	Ketika memilih <i>button hapus</i> maka akan menghapus data yang dipilih	+Berhasil
		Mengisi field cari dan menekan <i>button cari</i>	Ketika mengisi nama dosen dan menekan <i>button cari</i> , maka akan menampilkan data yang dicari	+Berhasil
		Menekan <i>button refresh</i>	Ketika memilih <i>button refresh</i> maka akan menampilkan data yang d update dan di input pada tabel.	+Berhasil
4	Form Input Dosen	Mengisi semua field inputan	Mengisi field inputan sesuai data sebenarnya yang nantinya akan di simpan	+Berhasil

		Menyimpan data dengan menekan <i>button simpan</i>	Data yang di input disimpan di <i>database</i>	+Berhasil
		Menekan <i>button kembali</i>	Menutup form input dan kembali ke halaman dosen	+Berhasil
5.	Matakuliah	Menekan <i>button tambah</i>	Ketika memilih <i>button tambah</i> maka akan menampilkan form input	+Berhasil
		Menekan <i>button edit</i>	Ketika memilih <i>button edit</i> maka akan menampilkan form input	+Berhasil
		Menekan <i>button hapus</i>	Ketika memilih <i>button hapus</i> maka akan menghapus data yang dipilih.	+Berhasil
		Mengisi field cari dan menekan <i>button cari</i>	Ketika mengisi kode matakuliah dan menekan <i>button cari</i> maka akan muncul data yang dicari	+Berhasil
		Menekan <i>button refresh</i>	Ketika memilih <i>button refresh</i> maka akan menampilkan data yang d update dan di input pada tabel.	+Berhasil
6	Form Input Matakuliah	Mengisi semua field inputan	Mengisi field inputan sesuai data sebenarnya yang nantinya akan di simpan	+Berhasil
		Menyimpan data dengan menekan <i>button simpan</i>	Data yang di input disimpan di <i>database</i>	+Berhasil
		Menekan <i>button kembali</i>	Menutup form input dan kembali ke halaman matakuliah	+Berhasil
7	Matahasiswa	Menekan <i>button tambah</i>	Bersihkan field inputan	+Berhasil
		Mengisi semua	Mengisi field inputan	+Berhasil

		field inputan	sesuai data sebenarnya yang nantinya akan disimpan	
		Menyimpan data inputan dengan menekan button simpan	Data yang di input disimpan di <i>database</i>	+Berhasil
		Menekan <i>button edit</i>	Menampilkan data yang akan di update pada field	+Berhasil
		Mengupdate data pada field.	Mengubah data pada field	+Berhasil
		menekan <i>button simpan</i>	Menyimpan data yang di update.	+Berhasil
		Input data pada field cari dan menekan tombol cari	Input nim pada field dan menekan button cari untuk pencarian	+Berhasil
		Menghapus data dengan menekan <i>button hapus</i>	Menghapus data mahasiswa yang dipilih	+Berhasil
8	Mengajar	Menekan <i>button tambah</i>	Ketika memilih <i>button tambah</i> maka akan menampilkan form mengajar.	+Berhasil
		Menekan <i>button edit</i>	Ketika memilih <i>button edit</i> maka akan menampilkan form input.	+Berhasil
		Menekan <i>button hapus</i>	Ketika memilih <i>button hapus</i> maka akan menghapus data yang dipilih	+Berhasil
		Mengisi field cari dan menekan <i>button cari</i>	Ketika mengisi kode matakuliah dan menekan <i>button cari</i> maka akan muncul data yang dicari	+Berhasil
		Menekan <i>button</i>	Ketika memilih <i>button</i>	+Berhasil

		refresh	refresh maka akan menampilkan data yang d update dan di input pada tabel.	
	Input Mengajar	Mengisi semua field inputan	Mengisi semua field inputan dan menyimpan data dengan menekan button simpan	+Berhasil
		Menyimpan data dengan menekan button simpan	Data yang di input disimpan di <i>database</i>	+Berhasil
		Menekan <i>button kembali</i>	Menutup form input dan kembali ke halaman mengajar.	+Berhasil
9	Ruangan	Menekan <i>button tambah</i>	Bersihkan field inputan	+Berhasil
		Mengisi semua field inputan	Mengisi field inputan sesuai data sebenarnya yang nantinya akan disimpan	+Berhasil
		Menyimpan data inputan dengan menekan button simpan	Data yang di input disimpan di <i>database</i>	+Berhasil
		Menekan <i>button edit</i>	Menampilkan data yang akan di update pada field	+Berhasil
		Mengupdate data pada field inputan	Mengubah data pada field	+Berhasil
		menekan <i>button simpan</i>	Menyimpan data yang di update.	+Berhasil
		Input data pada field cari dan menekan tombol cari	Input nim pada field dan menekan <i>button cari</i> untuk pencarian	+Berhasil
		Menghapus data dengan	Menghapus data mahasiswa yang dipilih	+Berhasil

		menekan <i>button</i> hapus		
10	Waktu	Menekan <i>button</i> tambah	Bersihkan field inputan	+Berhasil
		Mengisi semua field inputan	Mengisi field inputan sesuai data sebenarnya yang nantinya akan disimpan	+Berhasil
		Menyimpan data inputan dengan menekan <i>button</i> simpan	Data yang di input disimpan di <i>database</i>	+Berhasil
		Menekan <i>button</i> edit	Menampilkan data yang akan di update pada field	+Berhasil
		Mengupdate data pada field inputan	Mengubah data pada field	+Berhasil
		menekan <i>button</i> simpan	Menyimpan data yang di update.	+Berhasil
		Menghapus data dengan menekan <i>button</i> hapus	Menghapus data waktu yang dipilih	+Berhasil
11	Kelas	Menekan <i>button</i> tambah	Ketika memilih <i>button</i> tambah maka akan muncul form input.	+Berhasil
		Menekan <i>button</i> edit	Ketika memilih <i>button</i> edit maka akan muncul form input.	+Berhasil
		Menekan <i>button</i> hapus	Ketika memilih <i>button</i> hapus maka akan menghapus data yang dipilih	+Berhasil
		Mengisi field cari dan menekan <i>button</i> cari	Ketika mengisi nama dosen dan menekan <i>button</i> cari maka akan muncul data yang dicari	+Berhasil

		Menekan <i>button</i> refresh	Ketika memilih <i>button</i> refresh maka akan menampilkan data yang d update dan di input pada tabel.	+Berhasil
12	Form Input Kelas	Mengisi semua field inputan	Mengisi field inputan sesuai data sebenarnya yang nantinya akan di simpan	+Berhasil
		Menyimpan data dengan menekan <i>button</i> simpan	Data yang di input disimpan di <i>database</i>	+Berhasil
		Menekan <i>button</i> kembali	Menutup form input dan kembali ke halaman kelas	+Berhasil
13	Jadwal kuliah	Menekan <i>button</i> tambah	Ketika memilih <i>button</i> tambah maka akan muncul form input.	+Berhasil
		Menekan <i>button</i> edit	Ketika memilih <i>button</i> edit maka akan muncul form input.	+Berhasil
		Menekan <i>button</i> hapus	Ketika memilih <i>button</i> hapus maka akan menghapus data yang dipilih	+Berhasil
		Mengisi field cari dan menekan <i>button</i> cari	Ketika mengisi nama dosen dan menekan <i>button</i> cari maka akan muncul data yang dicari	+Berhasil
		Menekan <i>button</i> refresh	Ketika memilih <i>button</i> refresh maka akan menampilkan data yang d update dan di input pada tabel.	+Berhasil
		Menekan <i>button</i> penjadwalan	Ketika menekan penjadwalan maka data yang sudah di input secara acak akan dioptimasi jan menampilkan jadwal yang sudah di optimasi	+Berhasil
14	Form Input Jadwal	Mengisi semua field inputan	Mengisi field inputan sesuai data sebenarnya	+Berhasil

			yang nantinya akan di simpan	
		Menyimpan data dengan menekan button simpan	Data yang di input disimpan di database	+Berhasil
		Menekan button kembali	Menutup form input dan kembali ke halaman jadwal	+Berhasil
15	User	Menekan button tambah	Ketika memilih button tambah maka akan memunculkan form input	+Berhasil
		Menekan button ubah password	Ketika memilih button ubah passwor maka akan menampilkan form input untuk update password	+Berhasil
		Menekan set admin	Ketika memilih button set maka akan merubah data admin menjadi aktif/non aktif.	+Berhasil
		Mengisi field cari dan menekan button cari	Ketika mengisi ID dan menekan button cari maka akan muncul data yang dicari	+Berhasil
		Menekan button refresh	Ketika memilih button refresh maka akan menampilkan data yang d update dan di input pada tabel.	+Berhasil
16	Form Input User	Mengisi semua field inputan	Mengisi field inputan sesuai data sebenarnya yang nantinya akan di simpan	+Berhasil
		Menyimpan data dengan menekan button simpan	Data yang di input disimpan di database	+Berhasil
		Menekan button kembali	Menutup form input dan kembali ke halaman user	+Berhasil
17	Log out	Menekan button log out	Menutup aplikasi	+Berhasil

## **BAB VI**

### **KESIMPULAN DAN SARAN**

#### **6.1 Kesimpulan**

Berdasarkan hasil penelitian dan analisa serta perancangan aplikasi yang telah dihasilkan, penulis menyelesaikan laporan dan Aplikasi Penjadwalan Kuliah Dengan Menggunakan Metode Algoritma *Steepest Ascent Hill Climbing* Di Fakultas Teknologi Informasi Universitas Bale Bandung, maka dapat ditarik kesimpulan bahwa:

1. Aplikasi penjadwalan kuliah ini dapat diterapkan dalam mempermudah proses penjadwalan perkuliahan oleh Staff TU di Fakultas Teknologi Informasi Universitas Bale Bandung.
2. Algoritma *steepest-ascent hill climbing* yang diterapkan belum berjalan dengan optimal.
3. Pada proses penjadwalan memerlukan waktu yang cukup lama untuk data yang banyak.

#### **6.2 Saran**

Aplikasi penjadwalan kuliah ini masih memiliki banyak kekurangan, oleh sebab itu penulis berharap aplikasi ini dapat dikembangkan lebih lanjut demi tercapainya hasil sesuai dengan yang diharapkan. Untuk itu penulis memberikan beberapa saran sebagai berikut:

1. Penulis sarankan untuk pengembangan selanjutnya diharapkan aplikasi ini dapat terintegrasi dengan perwalian dan kartu rencana studi dan dapat digunakan di lingkup Universitas.
2. Aplikasi dapat dikembangkan menjadi berbasis web dan mobile.

## DAFTAR PUSTAKA

- A.S, R., & M.Shalahuddin. (2014). *Rekayasa Perangkat Lunak Struktur Berorientasi Objek*. Bandung: Penerbit INFORMATIKA.
- Anisyah. (2000). *Analisa dan Desain Sistem Informasi*. Yogyakarta: Penerbit Andi.
- Deswanti, A. P., & Saifudin, A. (2016). Penjdwalan Kuliah Dengan Menggunakan Algoritma Hill Climbing. *Prosiding Seminar Nasional Informatika dan Sistem Informasi*, Vol-1, No.3.
- Dini. (2015, Desember 22). *8 Kelebihan dan Kekurangan Java*. Dipetik April 30, 2019, dari dosenit.com: <https://dosenit.com/kuliah-it/pemrograman/kelebihan-dan-kekurangan-java>
- Fahrur. (2013, Mei). *Pengertian NetBeans*. Dipetik Mei 04, 2019, dari <http://fahrur-it.blogspot.com/2013/05/pengertian-netbeans.html>
- Haryanto, B. (2011). *Esensi-esensi Bahasa Pemrograman Java*. Yogyakarta: Andi.
- Hermawan. (2019, April 27). *Pengertian Database Beserta Fungsi dan Jenis-Jenis Database yang Paling Umum Digunakan*. Dipetik Mei 06, 2019, dari www.nesabamedia.com: <https://www.nesabamedia.com/pengertian-database-dan-fungsinya/>
- Irawan, E. O. (2012, November 22). *Macam-Macam bahasa Pemprograman*. Dipetik Mei 25, 2019, dari <http://okta-nata.blogspot.com/2012/11/macam-macam-bahasa-pemprograman.html>
- Kamus Besar Bahasa Indonesia (KBBI)*. Dipetik April 05, 2019, dari <https://kbbi.web.id/>: <https://kbbi.web.id/jadwal>
- Kusumadewi, S., & Purnomo, H. (2005). *Penyelesaian Masalah Optimasi Dengan Teknik-Teknik Heuristik*. Yogyakarta: Penerbit GRAHA ILMU.
- Mujiono. (2015). *Pengertian MySQL beserta kelebihan dan kelemahannya*. Dipetik April 10, 2019, dari <http://www.teorikomputer.com/http://www.teorikomputer.com/2015/10/pengertian-mysql-beserta-kelebihan-dan.html>.

- Munawar. (2018). *Analisis Perancangan Sistem Berorientasi Objek dengan Unified Modeling Language*. Bandung: Penerbit INFORMATIKA.
- Nikko, S. (2015, Mei). *Pengertian DBMS Dan Contohnya Lengkap Serta Jelas*. Dipetik Mei 12, 2019, dari www.pengertianku.com:  
<http://www.pengertianku.net/2015/05/pengertian-dbms-dan-contohnya-lengkap.html>.
- Pengertian, Komponen Dan Fungsi XAMPP Lengkap Dengan Penjelasannya*. (2019, Februari 01). Dipetik April 28, 2019, dari www.dosenpendidikan.com:  
<https://www.dosenpendidikan.com/pengertian-komponen-dan-fungsi-xampp-lengkap-dengan-penjelasannya/>
- Prayogi, S. (2013, Agustus 29). *Teknik Pencarian Heuristik (Heuristic Search)*. Dipetik April 20, 2019, dari <https://shabri-prayogi.blogspot.com>:  
<https://shabri-prayogi.blogspot.com/2013/08/teknik-pencarian-heuristik-heuristic.html>
- Ramadhan, Y. (2017, Oktober 21). *Metode Pencarian Heuristik (Hill Climbing)*. Dipetik April 10, 2019, dari [yogaramadhans.blogspot.com](http://yogaramadhans.blogspot.com):  
<http://yogaramadhans.blogspot.com/2017/10/metode-pencarian-heuristic-hill.html>
- Riadi, M. (2016, Oktober 13). *Pemrograman Berorientasi Objek (OOP)*. Dipetik Juni 21, 2019, dari www.kajianpustaka.com:  
<https://www.kajianpustaka.com/2016/10/pemrograman-berorientasi-objek-oop.html>
- Saifullah, S., & Hermawan, A. (2016). Pengembangan Sistem Penjadwalan Kuliah Menggunakan Algoritma Steepest Ascent Hill Climbing. *Jurnal Sistem Komputer*, Vol-6, No.2.
- Saryanti, I. A., & Wijanegara, I. (2017). Penerapan Metode Algoritma Genetika Untuk Penjadwalan Mengajar. *Jurnal SIMETRIS*, Vol-8, No.1.
- Taryana, A., & Wijayanto, B. (2014). Model-Driven Development : Fase Awal Verifikasi Model Design Rekam Medis Elektronis Menggunakan Perumusan Graf Lengkap. *JMP*, Vol-6, No.2, hal 53-64.
- Vickraien, E., Kusworo, A., & Vincencius. (2015). Penerapan Metode Algoritma Hill Climbing Pada Sistem Informasi Geografis Untuk Mencari Lintasan Terpendek. *Jurnal Sistem Informasi Bisnis*.

## LAMPIRAN

### Lampiran 1 Listing Program

#### 1. DB\_Koneksi.java

```
public class DB_koneksi {
    public static Connection getConnection() throws SQLException {
        Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost/penjadwalan","root",
", "");
        return connection;
    }
    public static class getConnection {
        public getConnection() {
        }
    }
}
```

#### 2. Tampilan\_Awal.java

```
public class Tampilan_Awal extends Application {

    @Override
    public void start(Stage stage) throws Exception {
        Parent root =
FXMLLoader.load(getClass().getResource("login.fxml"));

        Scene scene = new Scene(root);

        stage.setScene(scene);
        stage.show();
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        launch(args);
    }
}
```

### 3. LoginController.Java

```

public class LoginController implements Initializable {

    @FXML
    private AnchorPane id_login;
    @FXML
    private JFXButton login;
    @FXML
    private JFXTextField id_user;
    @FXML
    private JFXPasswordField id_pass;

    ResultSet resultset;

    /**
     * Initializes the controller class.
     */
    @Override
    public void initialize(URL url, ResourceBundle rb) {
        // TODO
    }

    @FXML
    private void btn_login(ActionEvent event) {

        try {
            Connection con = DB_koneksi.getConnection();
            PreparedStatement ps;
            String sql = "SELECT id,username FROM login WHERE
username=? and password=md5(?) AND status=?";
            ps = con.prepareStatement(sql);
            ps.setString(1,id_user.getText());
            ps.setString(2,id_pass.getText());
            ps.setString(3,"AKTIF");
            resultset = ps.executeQuery();
            int baris = 0;
            while (resultset.next()){
                baris = resultset.getRow();
            }
            if (baris==1){
                sql = "UPDATE login SET last_login=NOW() WHERE
username=?;";
                ps = con.prepareStatement(sql);
                ps.setString(1,id_user.getText());
                ps.execute();
                id_login.getChildren().clear();
                Node [] nodes = new Node[15];
                for(int i = 0;i<10; i++){
                    try {
                        nodes[i] =
(Node)FXMLLoader.load(getClass().getResource("/Interface/menu_utama.fx
ml"));
                    id_login.getChildren().add(nodes[i]);
                }
            }
        }
    }
}

```

```

        } catch (IOException ex) {

Logger.getLogger(LoginController.class.getName()).log(Level.SEVERE,
null, ex);
        }
    }else{
        System.out.println("Login Gagal");
        //message validasi username dan password
        Alert alert = new Alert(Alert.AlertType.INFORMATION);
        alert = new Alert(Alert.AlertType.INFORMATION);

        alert.setTitle("Login");
        alert.setHeaderText(null);
        alert.setContentText("Login gagal\nSilahkan cek
username / password!!!");
        alert.showAndWait();
    }
} catch (SQLException ex) {

Logger.getLogger(MahasiswaController.class.getName()).log(Level.SEVERE
, null, ex);
}
}

@FXML
private void tf_user(ActionEvent event) {

}

@FXML
private void fx_pass(ActionEvent event) {
}
}

```

#### 4. Context.java

```

public class Context {

    private final static Context instance = new Context();

    public static Context getInstance() {
        return instance;
    }

    private final ModelTabelDosen dosen = new ModelTabelDosen();

    public ModelTabelDosen currentDosen() {
        return dosen;
    }

    private final ModelTabelMatkul matkul = new ModelTabelMatkul();
}

```

```
public ModelTabelMatkul currentMatkul() {
    return matkul;
}

private final ModelTabelUser user = new ModelTabelUser();

public ModelTabelUser currentUser() {
    return user;
}

private final ModelTabelRuang ruang = new ModelTabelRuang();

public ModelTabelRuang currentRuang() {
    return ruang;
}

private final ModelTabelJadwal jadwal = new ModelTabelJadwal();

public ModelTabelJadwal currentJadwal() {
    return jadwal;
}

private final ModelTabelMengajar mengajar = new
ModelTabelMengajar();

public ModelTabelMengajar currentMengajar() {
    return mengajar;
}

private final ModelTabelKelas kelas = new ModelTabelKelas();

public ModelTabelKelas currentKelas() {
    return kelas;
}

private final ModelTabelMPS mps = new ModelTabelMPS();

public ModelTabelMPS currentMps() {
    return mps;
}

private final ModelTabelWaktu hari = new ModelTabelWaktu();

public ModelTabelWaktu currentHari() {
    return hari;
}

}
```

## 5. MenuUtamaController.java

```
public class MenuUtamaController implements Initializable {

    @FXML
    private AnchorPane id_menu_utama;
    @FXML
    private JFXButton ID_datadosen;
    @FXML
    private JFXButton ID_datamatkul;
    @FXML
    private JFXButton ID_datamahasiswa;
    @FXML
    private JFXButton ID_ngajardosen;
    @FXML
    private JFXButton ID_dataruang;
    @FXML
    private JFXButton ID_datawaktu;
    @FXML
    private JFXButton ID_datajadwal;
    @FXML
    private JFXButton ID_kelas;
    @FXML
    private JFXButton ID_user;
    @FXML
    private JFXButton ID_logout;

    /**
     * Initializes the controller class.
     */
    @Override
    public void initialize(URL url, ResourceBundle rb) {
        // TODO
    }

    @FXML
    private void btn_dosen(ActionEvent event) {

        id_menu_utama.getChildren().clear();
        Node [] nodes = new Node[15];
        for(int i = 0; i<10; i++){
            try {
                nodes[i] =
(Node)FXMLLoader.load(getClass().getResource("/Interface/dosen.fxml"));
            id_menu_utama.getChildren().add(nodes[i]);
        } catch (IOException ex) {

Logger.getLogger(MenuUtamaController.class.getName()).log(Level.SEVERE
, null, ex);

            }
        }
    }
}
```

```

@FXML
private void btn_matkul(ActionEvent event) {
    id_menu_utama.getChildren().clear();
    Node [] nodes = new Node[15];
    for(int i = 0; i<10; i++){
        try {
            nodes[i] =
(Node)FXMLLoader.load(getClass().getResource("/Interface/matkul.fxml"));
        } catch (IOException ex) {

Logger.getLogger(MenuUtamaController.class.getName()).log(Level.SEVERE
, null, ex);

        }
    }
}

@FXML
private void btn_mahasiswa(ActionEvent event) {
    id_menu_utama.getChildren().clear();
    Node [] nodes = new Node[15];
    for(int i = 0; i<10; i++){
        try {
            nodes[i] =
(Node)FXMLLoader.load(getClass().getResource("/Interface/mahasiswa.fxml"));
        } catch (IOException ex) {

Logger.getLogger(MenuUtamaController.class.getName()).log(Level.SEVERE
, null, ex);

        }
    }
}

@FXML
private void btn_ngajar(ActionEvent event) {
    id_menu_utama.getChildren().clear();
    Node [] nodes = new Node[15];
    for(int i = 0; i<10; i++){
        try {
            nodes[i] =
(Node)FXMLLoader.load(getClass().getResource("/Interface/mengajar.fxml"));
        } catch (IOException ex) {

Logger.getLogger(MenuUtamaController.class.getName()).log(Level.SEVERE
, null, ex);

        }
    }
}

```

```
}

@FXML
private void btn_ruang(ActionEvent event) {
    id_menu_utama.getChildren().clear();
    Node [] nodes = new Node[15];
    for(int i = 0; i<10; i++){
        try {
            nodes[i] =
(FXMLLoader.load(getClass().getResource("/Interface/ruang.fxml")));
;
            id_menu_utama.getChildren().add(nodes[i]);
        } catch (IOException ex) {
Logger.getLogger(MenuUtamaController.class.getName()).log(Level.SEVERE
, null, ex);
    }
}

@FXML
private void btn_waktu(ActionEvent event) {
    id_menu_utama.getChildren().clear();
    Node [] nodes = new Node[15];
    for(int i = 0; i<10; i++){
        try {
            nodes[i] =
(FXMLLoader.load(getClass().getResource("/Interface/waktu.fxml")));
;
            id_menu_utama.getChildren().add(nodes[i]);
        } catch (IOException ex) {
Logger.getLogger(MenuUtamaController.class.getName()).log(Level.SEVERE
, null, ex);
    }
}

@FXML
private void btn_jadwal(ActionEvent event) {
    id_menu_utama.getChildren().clear();
    Node [] nodes = new Node[15];
    for(int i = 0; i<10; i++){
        try {
            nodes[i] =
(FXMLLoader.load(getClass().getResource("/Interface/jadwal.fxml")));
;
            id_menu_utama.getChildren().add(nodes[i]);
        } catch (IOException ex) {
Logger.getLogger(MenuUtamaController.class.getName()).log(Level.SEVERE
, null, ex);
}
```

```

        }
    }

@FXML
private void btn_kelas(ActionEvent event) {
    id_menu_utama.getChildren().clear();
    Node [] nodes = new Node[15];
    for(int i = 0; i<10; i++){
        try {
            nodes[i] =
(Node)FXMLLoader.load(getClass().getResource("/Interface/Kelas.fxml"));
;
            id_menu_utama.getChildren().add(nodes[i]);
        } catch (IOException ex) {

Logger.getLogger(MenuUtamaController.class.getName()).log(Level.SEVERE
, null, ex);

        }
    }
}

@FXML
private void btn_user(ActionEvent event) {
    id_menu_utama.getChildren().clear();
    Node [] nodes = new Node[15];
    for(int i = 0; i<10; i++){
        try {
            nodes[i] =
(Node)FXMLLoader.load(getClass().getResource("/Interface/user.fxml"));
;
            id_menu_utama.getChildren().add(nodes[i]);
        } catch (IOException ex) {

Logger.getLogger(MenuUtamaController.class.getName()).log(Level.SEVERE
, null, ex);
        }
    }
}

@FXML
private void btn_logout(ActionEvent event) throws IOException {
    System.exit(0);
}
}

```

## 6. ModelTabelDosen.java

```

public class ModelTabelDosen {
    String crud, nidn, nama_dosen, jk, kd_prodi, status, email;

    public ModelTabelDosen(){

```

```
    }
    public ModelTabelDosen(String nidn, String nama_dosen, String jk,
String kd_prodi,
                        String status, String email){
        this.nidn = nidn;
        this.nama_dosen = nama_dosen;
        this.jk = jk;
        this.kd_prodi = kd_prodi;
        this.status = status;
        this.email = email;
    }
    public String getCrud(){
        return crud;
    }
    public void setCrud(String crud){
        this.crud = crud;
    }
    public String getNidn(){
        return nidn;
    }
    public void setNidn(String nidn){
        this.nidn = nidn;
    }
    public String getNama_dosen(){
        return nama_dosen;
    }
    public void setNama_dosen(String nama_dosen){
        this.nama_dosen = nama_dosen;
    }
    public String getJk(){
        return jk;
    }
    public void setJk(String jk){
        this.jk = jk;
    }
    public String getKd_prodi(){
        return kd_prodi;
    }
    public void setKd_prodi(String kd_prodi){
        this.kd_prodi = kd_prodi;
    }
    public String getStatus(){
        return status;
    }
    public void setStatus(String status){
        this.status = status;
    }
    public String getEmail(){
        return email;
    }
    public void setEmail(String email){
        this.email = email;
    }
}
```

## 7. DosenController.java

```
public class DosenController implements Initializable {

    @FXML
    private AnchorPane id_dosen2;
    @FXML
    private JFXButton id_tambah;
    @FXML
    private JFXButton id_edit;
    @FXML
    private JFXButton id_hapus;
    @FXML
    private JFXButton id_utama;
    @FXML
    private TableView<ModelTabelDosen> t_dosen;
    @FXML
    private TextField id_cari;
    @FXML
    private JFXButton id_btncari;
    @FXML
    private JFXButton id_refresh;
    @FXML
    private TableColumn<ModelTabelDosen, String> col_nidn;
    @FXML
    private TableColumn<ModelTabelDosen, String> col_nama;
    @FXML
    private TableColumn<ModelTabelDosen, String> col_jk;
    @FXML
    private TableColumn<ModelTabelDosen, String> col_prodi;
    @FXML
    private TableColumn<ModelTabelDosen, String> col_status;
    @FXML
    private TableColumn<ModelTabelDosen, String> col_email;
    ObservableList<ModelTabelDosen> cblist;
    private Object table;
    @FXML
    private JFXButton id_kembali;

    public DosenController() {
        this.cblist = FXCollections.observableArrayList();
    }
    /**
     * Initializes the controller class.
     * @param url
     * @param rb
     */
    @Override
    public void initialize(URL url, ResourceBundle rb) {
        try {
            cblist.clear();
            Connection con = DB_koneksi.getConnection();
            ResultSet rs = con.createStatement().executeQuery("select * from dosen");
        }
    }
}
```

```

        while (rs.next()) {
            cblist.add(new ModelTabelDosen(rs.getString("nidn"),
                rs.getString("nama_dosen"), rs.getString("jk"),
                rs.getString("kd_prodi"), rs.getString("status"), rs.getString("email")));
        }
    }

    t_dosen.getColumns().add(0, NumberTableCellFactory.createNumberColumn("No",
        1));

} catch (SQLException ex) {

Logger.getLogger(DosenController.class.getName()).log(Level.SEVERE,
null, ex);
}

col_nidn.setCellValueFactory(new
PropertyValueFactory<>("nidn"));
col_nama.setCellValueFactory(new
PropertyValueFactory<>("nama_dosen"));
col_jk.setCellValueFactory(new PropertyValueFactory<>("jk"));
col_prodi.setCellValueFactory(new
PropertyValueFactory<>("kd_prodi"));
col_status.setCellValueFactory(new
PropertyValueFactory<>("status"));
col_email.setCellValueFactory(new
PropertyValueFactory<>("email"));

t_dosen.setItems(cblist);
}

@FXML
private void btn_utama(ActionEvent event) {
    id_dosen2.getChildren().clear();
    Node [] nodes = new Node[15];
    for(int i = 0; i<10; i++){
        try {
            nodes[i] =
(Node)FXMLLoader.load(getClass().getResource("/Interface/menu_utama.fxml"));
            id_dosen2.getChildren().add(nodes[i]);
        } catch (IOException ex) {

Logger.getLogger(DosenController.class.getName()).log(Level.SEVERE,
null, ex);
        }
    }
}

@FXML
public void pencarian(ActionEvent event) {
    cari();
}

```

```

@FXML
private void btn_tambah(ActionEvent event) throws IOException {
    Context.getInstance().currentDosen().setCrud("tambah");
    modaltambah(event);
    cari();
}

@FXML
private void btn_edit(ActionEvent event) throws IOException {
    Context.getInstance().currentDosen().setCrud("edit");
    ModelTabelDosen klik =
t_dosen.getSelectionModel().getSelectedItems().get(0);
    Context.getInstance().currentDosen().setNidn(klik.getNidn());

Context.getInstance().currentDosen().setNama_dosen(klik.getNama_dosen());
    Context.getInstance().currentDosen().setJk(klik.getJk());

Context.getInstance().currentDosen().setKd_prodi(klik.getKd_prodi());

Context.getInstance().currentDosen().setStatus(klik.getStatus());
    Context.getInstance().currentDosen().setEmail(klik.getEmail());
    modaltambah(event);
}

@FXML
private void btn_hapus(ActionEvent event) {
    hapus();
    cari();
}

@FXML
private void klik tabel(MouseEvent event) {

}

@FXML
private void btn_refresh(ActionEvent event) {
    cari();
}

public void cari(){
    try {
        cblist.clear();
        Connection conn = DB_koneksi.getConnection();
        PreparedStatement ps;
        ResultSet rs;
        String sql = "SELECT * FROM dosen WHERE nidn LIKE ? OR
nama_dosen LIKE ? OR jk LIKE ? OR kd_prodi LIKE ? OR status LIKE ? OR
email LIKE ?";
        ps = conn.prepareStatement(sql);
        ps.setString(1,"%"+id_cari.getText()+"%");
        ps.setString(2,"%"+id_cari.getText()+"%");
        ps.setString(3,"%"+id_cari.getText()+"%");
        ps.setString(4,"%"+id_cari.getText()+"%");
        ps.setString(5,"%"+id_cari.getText()+"%");
```

```

        ps.setString(6,"%"+id_cari.getText()+"%");
        rs = ps.executeQuery();
        while (rs.next()) {
            cblist.add(new ModelTabelDosen(rs.getString("nidn"),
rs.getString("nama_dosen"), rs.getString("jk"),

rs.getString("kd_prodi"),rs.getString("status"),rs.getString("email")))
;
        }
    } catch (SQLException ex) {

Logger.getLogger(DosenController.class.getName()).log(Level.SEVERE,
null, ex);
}

        col_nidn.setCellValueFactory(new
PropertyValueFactory<>("nidn"));
        col_nama.setCellValueFactory(new
PropertyValueFactory<>("nama_dosen"));
        col_jk.setCellValueFactory(new PropertyValueFactory<>("jk"));
        col_prodi.setCellValueFactory(new
PropertyValueFactory<>("kd_prodi"));
        col_status.setCellValueFactory(new
PropertyValueFactory<>("status"));
        col_email.setCellValueFactory(new
PropertyValueFactory<>("email"));

        t_dosen.setItems(cblist);
    }

    public void modaltambah(ActionEvent event) throws IOException {
        Stage stage = new Stage();
        Parent root = FXMLLoader.load(
DosenController.class.getResource("/Interface/t_dosen.fxml"));
        stage.setScene(new Scene(root));
        stage.setTitle("Form Dosen");
        stage.initModality(Modality.WINDOW_MODAL);
        stage.initOwner(
            ((Node)event.getSource()).getScene().getWindow() );
        stage.show();
    }

    public void hapus(){
        try {
            ModelTabelDosen klik =
t_dosen.getSelectionModel().getSelectedItems().get(0);
            Connection conn = DB_koneksi.getConnection();
            PreparedStatement ps;
            String sql = "DELETE FROM dosen WHERE nidn=?";
            ps = conn.prepareStatement(sql);
            ps.setString(1,klik.getNidn());
            ps.execute();
        } catch (Exception e) {

```

```

Logger.getLogger(DosenController.class.getName()).log(Level.SEVERE,
null, e);
}
}
public class report extends javax.swing.JFrame {
    JasperReport JasRep;
    JasperPrint Jaspri;
    Map param = (Map) new HashMap();
    JasperDesign JasDes;
    Connection connection;
    Statement stm;
    // Resultset result;
    String url, password, username, sql;

    public report () {
        // initComponent();

        // DB_koneksi();

    }

}

@FXML
private void btn_kembali(ActionEvent event) {

((Stage)((Button)event.getSource()).getScene().getWindow())).close();
}

@FXML
private void btn_print(ActionEvent event) {
    try {
        JasperPrint jp =
JasperFillManager.fillReport(getClass().getResourceAsStream("report_dos
en.jasper"), null, DB_koneksi.getConnection());
        JasperViewer.viewReport(jp, false);

    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Gagal Membuka Laporan", "Cetak
Laporan", JOptionPane.ERROR_MESSAGE);
    }
}
}
}

```

## 8. Ft\_DosenController.java

```

public class Ft_DosenController implements Initializable {

    @FXML
    private TextField status;
    @FXML
    private TextField nidn;
}

```

```
@FXML  
private TextField nama;  
@FXML  
private JFXComboBox prodi;  
@FXML  
private TextField email;  
@FXML  
  
private RadioButton pria;  
@FXML  
private RadioButton wanita;  
@FXML  
private JFXButton id_simpan;  
@FXML  
private AnchorPane id_tam_dosen;  
@FXML  
private Label statusD;  
@FXML  
  
  
private ToggleGroup gender;  
@FXML  
private JFXButton id_kembali;  
@FXML  
private JFXComboBox prodi_id;  
/**  
 * Initializes the controller class.  
 * @param url  
 */  
@Override  
public void initialize(URL url, ResourceBundle rb) {  
    // TODO  
    try {  
        Connection con = DB_koneksi.getConnection();  
        ResultSet rs = con.createStatement().executeQuery("select  
* from prodi");  
  
        while (rs.next()) {  
            prodi_id.getItems().addAll(rs.getString("kd_prodi"));  
            prodi.getItems().addAll(rs.getString("nama_prodi"));  
        }  
  
        if  
(Context.getInstance().currentDosen().getCrud().equals("edit")){  
            nidn.setEditable(false);  
}  
  
nidn.setText(Context.getInstance().currentDosen().getNidn());  
  
nama.setText(Context.getInstance().currentDosen().getNama_dosen());  
        if  
(Context.getInstance().currentDosen().getJk().equals("L")){  
            pria.setSelected(true);  
        }else{  
            wanita.setSelected(true);  
}
```

```

        }
prodi_id.getSelectionModel().select(Context.getInstance().currentDosen()
().getKd_prodi());

prodi.getSelectionModel().select(prodi_id.getSelectionModel().getSelectedIndex());

status.setText(Context.getInstance().currentDosen().getStatus());

email.setText(Context.getInstance().currentDosen().getEmail());
    }
} catch (SQLException ex) {

Logger.getLogger(DosenController.class.getName()).log(Level.SEVERE,
null, ex);
}
}

@FXML

private void btn_simpan(ActionEvent event) throws SQLException,
IOException {
    Connection conn = DB_koneksi.getConnection();
    PreparedStatement ps;
    String lp = "";
    try {
        if (pria.isSelected() == true){
            lp="L";
        }else if (wanita.isSelected() == true){
            lp="P";
        }
        String sql="";
        if
(Context.getInstance().currentDosen().getCrud().equals("tambah")){
            sql = "INSERT INTO dosen
(nama_dosen,jk,kd_prodi,status,email,nidn) VALUES (?,?,?,?,?,?)";
        }else if
(Context.getInstance().currentDosen().getCrud().equals("edit")){
            sql = "UPDATE dosen SET
nama_dosen=?,jk=?,kd_prodi=?,status=?,email=? WHERE nidn=?";
        }
        ps = conn.prepareStatement(sql);
        ps.setString(1,nama.getText());
        ps.setString(2,lp);

ps.setString(3,prodi_id.getSelectionModel().getSelectedItem().toString());
        ps.setString(4,status.getText());
        ps.setString(5,email.getText());
        ps.setString(6,nidn.getText());
        ps.execute();
    } catch (SQLException ex) {
        System.out.println("Error"+ex);
    }
}

```

```
Alert alert = new Alert(Alert.AlertType.INFORMATION);
alert = new Alert(Alert.AlertType.INFORMATION);

alert.setTitle("pesan");
alert.setHeaderText(null);
alert.setContentText("Data berhasil di simpan...\"");
alert.showAndWait();
btn_kembali(event);

System.out.println("context=>" + Context.getInstance().currentDosen().getCrud());
}

@FXML
private void btn_kembali(ActionEvent event) {

((Stage)((Button)event.getSource()).getScene().getWindow())).close();
}

@FXML
private void prodi_pilih(ActionEvent event) {

prodi_id.getSelectionModel().select(prodi.getSelectionModel().getSelectedIndex());
}

@FXML
private void prodi_id_pilih(ActionEvent event) {

prodi.getSelectionModel().select(prodi_id.getSelectionModel().getSelectedIndex());
}
}
```

## **Lampiran 2 Biodata Penulis**

### **BIODATA PENULIS**



RIDWAN BUDIMAN, Dilahirkan di Bandung pada hari Senin tanggal 06 Juni 1994. Anak ke sembilan dari dua belas bersaudara pasangan dari Entoy Suhendar dan Pupu Yuhanah. Peneliti menyelesaikan pendidikan di Sekolah Dasar di SDN Rancakole II di Kecamatan Arjasari Kabupaten Bandung pada tahun pada tahun 2006. Pada tahun itu juga peneliti melanjutkan Pendidikan di SMP Negeri 3 Ciparay bertempat di Kecamatan Ciparay dan tamat pada tahun 2009, kemudian melanjutkan Sekolah Menengah Kejuruan di SMK Wirakarya 1 Ciparay pada tahun 2009 dan selesai pada tahun 2012. Pada tahun 2015 peneliti melanjutkan pendidikan di perguruan tinggi, tepatnya di Universitas Bale Bandung (UNIBBA) Fakultas Teknologi Informasi pada Program Studi Teknik Informatika.

Demikian Riwayat hidup penulis untuk sekedar diketahui.

Terima Kasih

Penulis