



## **INSTITUT TEKNOLOGI DEL**

### **IMPLEMENTASI ALGORITMA *BACKTRACKING* CSP (*CONSTRAINT SATISFACTION PROBLEM*) DAN DIJKSTRA PADA PENJADWALAN MATA KULIAH (STUDI KASUS INSTITUT TEKNOLOGI DEL)**

#### **TUGAS AKHIR**

**Disampaikan Sebagai Bagian Dari Persyaratan Kelulusan Diploma 3  
Program Studi Teknik Informatika**

<b>11316003</b>	<b>Krisnomi Nainggolan</b>
<b>11316015</b>	<b>Pratiwi Lasniate Pandiangan</b>
<b>11316020</b>	<b>Sehat Maruli Tua Samosir</b>


**FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO  
PROGRAM STUDI DIII TEKNIK INFORMATIKA**


**LAGUBOTI  
AGUSTUS 2019**

### HALAMAN PERNYATAAN ORISINALITAS

Tugas Akhir ini adalah hasil karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : Krisnami Nuinggolan  
NIM : 11316003  
Tanda Tangan :   
Tanggal : 09 Agustus 2019

Nama : Pratiwi Lasniate Pandiangan  
NIM : 11316015  
Tanda Tangan :   
Tanggal : 09 Agustus 2019

Nama : Sehat Maruli Tua Samosir  
NIM : 11316020  
Tanda Tangan :   
Tanggal : 09 Agustus 2019

## HALAMAN PENGESAHAN

Tugas Akhir ini diajukan oleh :

1. Nama : Krisnomi Nainggolan  
NIM : 11316003  
Program Studi : DIII Teknik Informatika
  2. Nama : Pratiwi Lasniate Pandiangan  
NIM : 11316015  
Program Studi : DIII Teknik Informatika
  3. Nama : Sehat Maruli Tua Samosir  
NIM : 11316020  
Program Studi : DIII Teknik Informatika
- Judul dokumen tugas akhir : Implementasi Algoritma Backtracking CSP (*Constraint Satisfaction Problem*) dan Dijkstra pada Penjadwalan Mata Kuliah (Studi Kasus: Institut Teknologi DEL)

Telah berhasil dipertahankan dihadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Diploma III, pada program studi Diploma III Teknik Informatika Fakultas Informatika dan Teknik Elektro, Institut Teknologi Del.

## DEWAN PENGUJI

Pembimbing : Ike Fitriyaningsih, S.Si., M.Si

Penguji : Anthon Roberto Tampubolon,  
S.Kom., M. T

Penguji : Teamsar Muliadi Panggabean,  
S.Kom, PGCert



Ditetapkan di : Laguboti  
Tanggal : 09 Agustus 2019

## KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa, karena atas kasih karuniaNya, penulis dapat menyelesaikan Tugas Akhir 2 (TA 2) dengan judul ***“Implementasi Algoritma Backtracking CSP (Constraint Satisfaction Problem) dan Dijkstra pada Penjadwalan Mata Kuliah”*** ini. Penulisan Tugas Akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Diploma Tiga Teknik Informatika Fakultas Teknik Informatika dan Elektro Institut Teknologi Del. Penulis menyadari bahwa, tanpa bantuan, dukungan, dan bimbingan dari berbagai pihak, dimulai dari masa perkuliahan sampai pada penyusunan Tugas Akhir ini, sangatlah sulit bagi penulis untuk menyelesaikan Tugas Akhir ini.

Oleh karena itu, penulis menyampaikan rasa terimakasih kepada:

- 1) Ike Fitriyaningsih, S.Si., M.Si., selaku dosen pembimbing yang telah menyediakan waktu, tenaga dan pikiran untuk membantu penulis dalam mengarahkan dalam penyusunan Tugas Akhir ini;
- 2) Bapak/Ibu, selaku penguji yang telah menyediakan waktu, tenaga dan pikiran untuk memberikan kritikan dan masukan yang membangun dalam penyusunan Tugas Akhir ini.
- 3) Bapak Togu Novriansyah Turnip, S.S.T., M.IM., selaku koordinator TA yang telah menyediakan waktu, tenaga dan pikiran untuk membantu penulis dalam mengkoordinir penyusunan Tugas Akhir ini;
- 4) Orang Tua dan keluarga yang telah memberikan dukungan dan bantuan material dan moral;
- 5) Sahabat dan teman yang telah membantu dan mendukung penulis dalam menyelesaikan Tugas akhir ini.

Akhir kata, penulis berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah ikut serta membantu penulis dalam menyusun Tugas Akhir ini. Semoga Tugas Akhir ini bermanfaat bagi pengembangan ilmu pengetahuan dimasa yang akan datang.

Situluama, 09 Agustus 2019

11316003 Krisnomi Nainggolan

11316015 Pratiwi Lasniate Pandiangan

11316020 Sehat Maruli Tua Samosir

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS  
AKHIR UNTUK KEPENTINGAN AKADEMIS**

---

Sebagai sivitas akademik Institut Teknologi Del, kami yang bertanda tangan dibawah ini :

- |    |               |   |                                |
|----|---------------|---|--------------------------------|
| 1. | Nama          | : | Krisnomi Nainggolan            |
|    | NIM           | : | 11316003                       |
|    | Program Studi | : | DIII Teknik Informatika        |
| 2. | Nama          | : | Pratiwi Lasniate Pandiangan    |
|    | NIM           | : | 11316015                       |
|    | Program Studi | : | DIII Teknik Informatika        |
| 3. | Nama          | : | Sehat Maruli Tua Samosir       |
|    | NIM           | : | 11316020                       |
|    | Program Studi | : | DIII Teknik Informatika        |
|    | Fakultas      | : | Informatika dan Teknik Elektro |
|    | Jenis Karya   | : | Tugas Akhir                    |

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Institut Teknologi Del **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah penulis yang berjudul:

Implementasi Algoritma *Backtracking CSP (Constraint Satisfaction Problem)* dan Dijkstra pada Penjadwalan Mata Kuliah. Dengan Hak Bebas Royalti Noneksklusif ini Institut Teknologi Del berhak menyimpan, mengalih/media-format dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir penulis selama tetap mencantumkan nama penulis sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini penulis buat dengan sebenarnya.


Dibuat di : Laguboti

Pada tanggal : 10 September 2019

Yang menyatakan



(Krisnomi Nainggolan)



(Pratiwi L. Pandiangan)



(Sehat M T Samosir)



Sampul dan  
Cover Skripsi

## ABSTRAK

Nama : Krisnomi Nainggolan  
Program Studi : DIII Teknik Informatika  
Nama : Pratiwi Lasniate Pandiangan  
Program Studi : DIII Teknik Informatika  
Nama : Sehat Maruli Tua Samosir  
Program Studi : DIII Teknik Informatika  
Judul : Implementasi Algoritma *Backtracking CSP*  
(*Constraint Satisfaction Problem*) dan  
Dijkstra pada Penjadwalan Mata Kuliah

Penjadwalan merupakan proses yang digunakan untuk mengatur berbagai tugas dengan menggunakan sumber-sumber yang tersedia pada waktu yang telah ditetapkan. Dalam kegiatan belajar mengajar pada instansi pendidikan, diperlukan sistem penjadwalan matakuliah. Sistem penjadwalan ini merupakan hal yang kompleks dikarenakan perlunya pertimbangan terkait keterbatasan dan syarat yang dimiliki. Keterbatasan dan syarat dalam hal ini berupa dosen yang mengampuh matakuliah, dan muatan kelas yang disesuaikan dengan jumlah mahasiswa yang mengambil matakuliah tersebut. Penjadwalan juga harus memperhatikan jarak lokasi antar kelas untuk meminimalisasi penggunaan waktu yang berlebih akibat jarak perpindahan antar kelas pada jadwal dosen dan mahasiswa telah ditentukan. Kasus penjadwalan kuliah yang diambil untuk penelitian tugas akhir ini adalah penjadwalan kuliah di Institut Teknologi Del.

Oleh karena itu untuk mengatasi permasalahan penjadwalan di Institut Teknologi Del, maka pada penelitian ini dilakukan pembangunan aplikasi yang menerapkan penggabungan Algoritma *Backtracking* dengan metode CSP (*Constraint Satisfaction Problem*) dan Algoritma Dijkstra. Algoritma *Backtracking* akan mencoba semua kemungkinan solusi dalam bentuk pohon solusi (*tree*) berdasarkan *Constraint* yang didapatkan dengan menggunakan pendekatan CSP. Sedangkan Algoritma Dijkstra merupakan algoritma yang digunakan untuk penyelesaian masalah jalur terpendek.

Pada tugas akhir ini dilakukan pengujian *constraint-constraint* dan iterasi terhadap menu aplikasi, sehingga tidak ada *constraint* yang dilanggar saat jadwal di-*generate*. Dari hasil pengujian tersebut diketahui bahwa penggabungan Algoritma *Backtracking* dengan metode CSP dan Dijkstra terhadap penugasan setiap sumber daya penjadwalan kuliah dan jarak lokasi kelas lebih efektif dan menghasilkan penjadwalan matakuliah yang optimal.

Kata kunci:

Penjadwalan matakuliah, Algoritma *backtracking*, metode *Constraint Satisfaction Problem*, Algoritma Dijkstra,

## **ABSTRACT**

Name : Krisnomi Nainggolan  
Study Program : DIII Teknik Informatika  
Name : Pratiwi Lasniate Pandiangan  
Study Program : DIII Teknik Informatika  
Name : Sehat Maruli Tua Samosir  
Study Program : DIII Teknik Informatika  
Judul : Implementasi Algoritma *Backtracking CSP*  
(*Constraint Satisfaction Problem*) dan  
Dijkstra pada Penjadwalan Mata Kuliah

*Scheduling is a process that is used to manage various tasks by using available resources at a predetermined time. In teaching and learning activities in educational institutions, a scheduling system is needed. This scheduling system is complex because of the need for consideration regarding limitations and conditions. The limitations and conditions, in this case, are lecturers who apply the subject, and class content that is adjusted to the number of students taking the course. Scheduling must also pay attention to the distance between classes to minimize the excessive use of time due to the distance between classes in the lecturer and student schedule has been determined. The scheduling cases of lectures taken for this final assignment research are scheduling lectures at the Del Institute of Technology.*

*Therefore to overcome the scheduling problem at the Del Institute of Technology, in this study an application development was carried out which applied a combination of Backtracking Algorithms with CSP (Constraint Satisfaction Problem) and Dijkstra Algorithms. The Backtracking algorithm will try all possible solutions in the form of a tree based on constraints obtained using the CSP approach. While the Dijkstra algorithm is an algorithm used to solve the shortest path problem.*

*In this final project, constraints and iterations are tested on the application menu, so that no constraints are violated when the schedule is generated. From the results of the test it is known that combining the Backtracking Algorithm with CSP and Dijkstra's method towards assigning each lecture scheduling resource and class location distance is more effective and results in the scheduling of optimal courses.*

### **Keywords:**

*Course scheduling, backtracking algorithm, Constraint Satisfaction Problem approach, dijkstra algorithm*

## DAFTAR ISI

KATA PENGANTAR .....	iv
ABSTRAK .....	v
ABSTRACT .....	vii
<b>BAB 1 Pendahuluan .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Tujuan .....	2
1.3 Lingkup .....	3
1.4 Metodologi Penelitian .....	3
1.5 Sistematika Penyajian .....	4
<b>BAB 2 Tinjauan Pustaka .....</b>	<b>6</b>
2.1 Penjadwalan .....	6
2.1.1 Penjadwalan Mata Kuliah .....	7
2.1.2 Batasan-Batasan Masalah pada Penjadwalan Mata Kuliah .....	7
2.2 Constraint Satisfaction Problem .....	8
2.3 Algoritma Backtracking .....	9
2.4 Lintasan Terpendek .....	11
2.5 Algoritma Dijkstra .....	11
2.6 Penelitian Terdahulu .....	14
2.7 Kesimpulan .....	20
<b>BAB 3 Analisis Permasalahan dan Penggunaan Agoritma .....</b>	<b>21</b>
3.1 Analisis Penjadwalan Kuliah .....	21
3.1.1 Penjadwalan Kuliah di Institut Teknologi Del .....	21
3.1.2 Batasan-Batasan Penjadwalan Mata Kuliah di Institut Teknologi Del .....	23
3.1.3 Data Penjadwalan Kuliah .....	24
3.2 Analisis CSP ( <i>Constraint Satisfaction Problem</i> ) .....	39
3.3 Analisis Algoritma Backtracking .....	51
3.4 Analisis Perhitungan Jarak .....	56
3.4.1 Perhitungan Jarak Tanpa Algoritma Dijkstra .....	56
3.4.2 Perhitungan Jarak dengan Algoritma Dijkstra .....	57
3.4.3 Kesimpulan .....	60
3.4.4 Graph Penjadwalan .....	60
3.5 Analisis Kebutuhan Sistem .....	68
3.6 Class Diagram .....	69
3.7 Entity Relationship Diagram (ERD) .....	70
3.8 Struktur Database Penjadwalan .....	71
3.9 Antarmuka Perangkat Lunak .....	74
3.9.1 Antarmuka Pengguna .....	74
3.9.2 Lingkungan Pengembangan .....	74
3.10 Deskripsi Fungsional .....	75
3.10.1 Use Case Diagram Penjadwalan Mata Kuliah .....	75
3.10.2 Use Case Scenario Penjadwalan Mata Kuliah .....	76
3.10.2.1 Mengelola Data .....	77
3.10.2.2 Membuat Jadwal .....	78
3.10.2.3 Mengunduh Jadwal .....	80
3.11 Desain .....	82
<b>BAB 4 Implementasi dan Testing .....</b>	<b>88</b>
4.1 Kebutuhan Implementasi .....	88
4.2 Hasil Aplikasi .....	88
4.3 Testing .....	99
4.3.1 Metode Test .....	99
4.3.2 Tujuan Pengujian .....	99
4.3.3 Test Plan .....	100
<b>BAB 5 Hasil dan Pembahasan .....</b>	<b>108</b>
5.1 Hasil Jadwal .....	108
5.1.1 Hasil Penjadwalan FTB (Fakultas Teknik Bioproses) .....	108
5.1.2 Hasil Penjadwalan FTI (Fakultas Teknik Industri) .....	110



5.1.3	Hasil Penjadwalan (FTIE)Fakultas Teknik Informatika dan Elektro Prodi Diploma 111	
5.1.4	Hasil Penjadwalan Sarjana FTIE.....	113
5.2.	Hasil Dijkstra.....	114
<b>BAB 6</b>	<b>Kesimpulan dan Saran .....</b>	<b>118</b>
6.1.	Kesimpulan .....	118
6.2.	Saran.....	118
	<b>Rujukan dan Daftar Pustaka .....</b>	<b>120</b>

## DAFTAR TABEL

Tabel 2.1 Related Work .....	15
Tabel 3.1 Komponen Waktu .....	24
Tabel 3.2 Data Rombongan Kelas .....	25
Tabel 3.3 Data Dosen .....	27
Tabel 3.4 Data Mata Kuliah Program Sarjana.....	29
Tabel 3.5 Data Mata Kuliah Program Diploma .....	35
Tabel 3.6 Kompenen Ruangan .....	38
Tabel 3.7 Tabel matriks dari dosen dengan mata kuliah .....	44
Tabel 3.8 Relasi Mata Kuliah dengan Kelas.....	48
Tabel 3.9 Matriks antar Ruangan .....	59
Tabel 3.10 Jalur Dijkstra .....	61
Tabel 3.11 Tipe data dosen.....	71
Tabel 3.12 Tipe Data Hari.....	71
Tabel 3.13 Tipe Data Sesi .....	71
Tabel 3.14 Tipe Data Mata Kuliah .....	72
Tabel 3.15 Tipe Data Mahasiswa .....	72
Tabel 3.16 Tipe Data Tabel Jadwal .....	72
Tabel 3.17 Tipe Data Kategori .....	72
Tabel 3.18 Tipe Data Ruangan .....	73
Tabel 3.19 Tipe Data Relasi Dosen dengan Matakuliah .....	73
Tabel 3.20 Tipe Data Relasi Matakuliah dengan Kelas .....	73
Tabel 3.21 Tipe Data Tabel Jarak Ruangan .....	73
Tabel 3.22 Kebutuhan User Interface .....	74
Tabel 3.23 Mengelola Data .....	77
Tabel 3.24 Use Case Scenario Membuat Jadwal .....	78
Tabel 3.25 Mengunduh Jadwal.....	80
Tabel 4.1 Spesifikasi Perangkat Keras.....	88
Tabel 4.2 Spesifikasi Perangkat Lunak .....	88
Tabel 4.3 Test Plan Pengujian Mengelola Data Kelas .....	100
Tabel 4.4 Test Plan Pengujian Mengelola Data Dosen .....	101
Tabel 4.5 Test Plan Pengujian Mengelola Data Mata Kuliah .....	103
Tabel 4.6 Test Plan Pengujian Mengelola Ruangan.....	104
Tabel 4.7 Test Plan Pengujian Generate Jadwal .....	105
Tabel 4.8 Pengujian Constraint.....	107

## DAFTAR GAMBAR

Gambar 2.1 Pseudocode Backtracking .....	10
Gambar 2.2 Pencarian Jalur Terpendek pada Dijkstra.....	13
Gambar 3.1 Pewarnaan Graph Constraint.....	51
Gambar 3.2 Tree Algoritma Backtracking Pertama .....	53
Gambar 3.3 Tree Algoritma Backtracking Kedua .....	54
Gambar 3.4 Perkuliahan Sesi Pertama .....	56
Gambar 3.5 Perkuliahan Sesi ketiga .....	56
Gambar 3.6 Hubungan antar Gedung dengan Algoritma Dijkstra .....	61
Gambar 3.7 Graph Penjadwalan Senin Jam Pertama .....	62
Gambar 3.8 Graph Penjadwalan Senin Jam Kedua .....	63
Gambar 3.9 Graph Penjadwalan Senin Jam Ketiga .....	63
Gambar 3.10 Graph Penjadwalan Senin Jam Keempat.....	64
Gambar 3.11 Graph Penjadwalan Selasa Jam Pertama .....	64
Gambar 3.12 Graph Penjadwalan Selasa Jam Kedua .....	65
Gambar 3.13 Graph Penjadwalan Selasa Jam Ketiga .....	65
Gambar 3.14 Graph Penjadwalan Selasa Jam Keempat .....	66
Gambar 3.15 Graph Penjadwalan Rabu Jam Pertama .....	66
Gambar 3.16 Graph Penjadwalan Rabu Jam Kedua.....	67
Gambar 3.17 Graph Penjadwalan Rabu Jam Ketiga .....	67
Gambar 3.18 Graph Penjadwalan Rabu Jam Keempat .....	68
Gambar 3.19 Class Diagram Penjadwalan Kuliah .....	69
Gambar 3.20 ERD Penjadwalan Kuliah .....	70
Gambar 3.21 Use Case Diagram Penjadwalan Mata Kuliah .....	75
Gambar 3.22 Desain Menu Utama Aplikasi .....	82
Gambar 3.23 Halaman Kelola Data Kelas .....	83
Gambar 3.24 Halaman Kelola Data Dosen .....	83
Gambar 3.25 Halaman Kelola Data Mata Kuliah .....	84
Gambar 3.26 Halaman Kelola Data Jadwal .....	84
Gambar 3.27 Desain Output Jadwal Backtracking .....	85
Gambar 3.28 Halaman Data Ruangan .....	86
Gambar 3.29 Halaman Generate Dijkstra .....	87
Gambar 3.30 Desain Output Jadwal Dijkstra.....	87
Gambar 4.1 Tampilan Menu Utama Aplikasi Penjadwalan Mata Kuliah.....	89
Gambar 4.2 Tampilan Menu Kelola Kelas Aplikasi Jadwal Perkuliahan.....	90
Gambar 4.3 Tampilan Menu Kelola Dosen Aplikasi Jadwal Perkuliahan .....	90
Gambar 4.4 Tampilan Menu Kelola Mata Kuliah Aplikasi Jadwal Perkuliahan .....	91
Gambar 4.5 Tampilan Menu Kelola Ruangan Aplikasi Jadwal Perkuliahan.....	92
Gambar 4.6 Tampilan Menu Generate Jadwal .....	93
Gambar 4.7 Kode Program Deklarasi Variabel .....	94
Gambar 4.8 Kode Program Ketetapan Nilai Pada Constraint .....	94
Gambar 4.9 Kode Program Relasi Domain .....	94
Gambar 4.10 Kode Program Constraint Pertama .....	95
Gambar 4.11 Kode Program Constraint Kedua .....	96
Gambar 4.12 Kode Program Constraint Ketiga .....	96
Gambar 4.13 Kode Program Problem Solver .....	97
Gambar 4.14 Kode Program Penyimpanan Jadwal .....	97
Gambar 4.15 Kode Program Dijkstra .....	98
Gambar 4.16 Kode Program Dijkstra .....	99
Gambar 5.1 Hasil Generate Jadwal.....	108

Gambar 5.2 Hasil Export Jadwal TB pada File Excel .....	109
Gambar 5.3 Tampilan Hasil Generate Jadwal FTI .....	110
Gambar 5.4 Hasil Export Jadwal FTI pada File Excel .....	111
Gambar 5.5 Tampilan Hasil Generate Jadwal Teknik Komputer .....	112
Gambar 5.6 Hasil Export Jadwal Teknik Komputer pada File Excel.....	113
Gambar 5.7 Tampilan Menu Generate Dijkstra.....	114
Gambar 5.8 Tampilan Set Ruangan Manual.....	115
Gambar 5.9 Generate Dijkstra.....	116
Gambar 5.10 Hasil Export Hasil Generate Jadwal .....	117

# **BAB 1**

## **Pendahuluan**

Bab I berisi penjelasan mengenai latar belakang pemilihan topik, tujuan pelaksanaan TA, lingkup penelitian yang mendasari TA, pendekatan yang dilakukan selama melakukan TA, serta sistematika penyajian materi yang disediakan dalam laporan.

### **1.1 Latar Belakang**

Penjadwalan merupakan proses untuk melakukan tugas dengan menggunakan sumber-sumber yang tersedia pada waktu yang telah ditetapkan. Dalam kegiatan belajar mengajar pada instansi pendidikan diperlukan sistem penjadwalan mata kuliah. Sistem penjadwalan ini merupakan hal yang kompleks dikarenakan perlunya pertimbangan terkait keterbatasan dan syarat yang dimiliki. Keterbatasan dan syarat dalam hal ini berupa dosen yang mengampu mata kuliah, dan muatan kelas yang disesuaikan dengan jumlah mahasiswa yang mengambil mata kuliah tersebut. Penjadwalan juga harus memperhatikan jarak lokasi antar kelas untuk meminimalisasi penggunaan waktu yang berlebih, akibat jarak perpindahan antar kelas pada jadwal dosen dan mahasiswa telah ditentukan. Jarak antara gedung yang terlalu jauh harus ditempuh dengan selang waktu sepuluh menit dengan waktu efektif satu jam pelajaran 50 menit, apabila perkuliahan melebihi ketetapan waktu, maka akan berdampak untuk jam perkuliahan yang selanjutnya. Sehingga dalam hal ini, diperlukan penyelesaian masalah jalur terpendek yang akan dikaitkan dengan jarak antar kelas pada penjadwalan. Penjadwalan yang dibuat harus mampu menyelaraskan sumber daya yang tersedia. Adapun sumber daya yang dimaksud meliputi dosen pengampu dan atau asisten dosen, mahasiswa serta ruang kelas.

Penjadwalan kuliah di beberapa kampus saat ini masih menggunakan penjadwalan dengan menggunakan *Microsoft Excel* (Fahrurrozi, 2011). Penjadwalan yang menggunakan cara seperti ini, akan sulit untuk mendapatkan hasil yang optimal. Sehingga dapat mengakibatkan bentrokan waktu pada sumber daya dan penempatan kelas yang memiliki jarak yang terlalu jauh pada waktu yang beruntun,

serta adanya ketidaksesuaian antara fasilitas ruangan dengan jumlah mahasiswa yang mengambil mata kuliah tersebut. Namun pada pelaksanaannya, jadwal yang dibuat pada setiap awal semester akan mengalami perubahan beberapa kali dikarenakan jadwal yang ada tidak sesuai. Sehingga hal ini akan membuat perkuliahan berjalan tidak efektif.

Oleh karena itu diperlukan solusi untuk permasalahan penjadwalan di kampus saat ini. Pada penelitian ini akan dilakukan penerapan Algoritma *Backtracking* dengan *Constraint Satisfaction Problem* dan Dijkstra, dimana Algoritma *Backtracking* akan mencoba semua kemungkinan solusi dalam bentuk pohon solusi (*tree*) berdasarkan *Constraint* yang didapatkan dengan menggunakan pendekatan CSP. Pohon solusi akan ditelusuri secara DFS (*depth first search*) untuk mendapatkan solusi terbaik yang diinginkan. Sedangkan Algoritma Dijkstra merupakan algoritma yang digunakan untuk penyelesaian masalah jalur terpendek. Gabungan kedua algoritma ini diharapkan dapat mengoptimalkan penugasan setiap sumber daya dan memperoleh jarak lokasi kelas yang lebih efektif dalam waktu penjadwalan yang beruntun. Sehingga dapat menghasilkan penjadwalan kampus yang optimal.

## 1.2 Tujuan

Adapun tujuan dilaksanaannya Tugas Akhir ini adalah untuk:

1. Menggabungkan dan menerapkan penggabungan dari algoritma *Backtracking* menggunakan pendekatan *Constraint Satisfaction Problem (CSP)* dan Dijkstra pada penjadwalan mata kuliah .
2. Membuat *prototype* penjadwalan kampus yang menerapkan hasil penggabungan algoritma *Backtracking* menggunakan pendekatan *Constraint Satisfaction Problem (CSP)* dan Dijkstra pada penjadwalan mata kuliah.

Dalam mencapai tujuan ini terdapat langkah-langkah yang diperlukan, mulai dari studi literatur mengenai algoritma penjadwalan, melakukan analisis algoritma terhadap pembuatan penjadwalan kampus dengan menggunakan Algoritma Dijkstra dan *Backtracking* yang akan diterapkan dalam pembuatan penjadwalan tersebut.

### 1.3 Lingkup

Adapun ruang lingkup pada penulisan TA yang dibahas adalah:

1. Pada saat ini penjadwalan dibuat dengan data *dummy*.
2. Pada saat validasi menggunakan kasus yang terjadi di Institut Teknologi Del.
3. Implementasi dilakukan dengan menggunakan algoritma Dijkstra dan Backtracking

### 1.4 Metodologi Penelitian

Metodologi penelitian yang akan dilakukan dalam pelaksanaan Tugas Akhir ini meliputi:

#### 1. Jenis Penelitian

Jenis penelitian yang dilakukan yaitu penelitian kuantitatif. Metode penelitian ini digunakan untuk mengumpulkan data, menafsirkan data dan memperoleh hasil dari penafsiran dalam bentuk tabel.

#### 2. Metode pengumpulan data

Metode pengumpulan data yang digunakan antara lain:

##### a. Studi Literatur

Studi literatur dilakukan untuk mencari referensi atau teori yang relevan yang digunakan untuk penjadwalan kuliah.

##### b. Pengamatan (Observasi)

Pengamatan dilakukan untuk mengumpulkan data dengan cara mengamati secara langsung objek yang diteliti tanpa melakukan komunikasi *verbal* atau lisan.

#### 3. Tahapan Penelitian

Tahapan penelitian yang akan dilakukan adalah sebagai berikut:

- Menentukan masalah penelitian, dalam tahap ini penelitian mengadakan studi pendahuluan.
- Pengumpulan data, pada tahap ini peneliti mulai dengan menentukan sumber data dan pengumpulan dasar teori dalam membantu pemahaman terkait algoritma yang digunakan yaitu bersumber dari *paper* atau jurnal,

buku, dan lainnya yang berkaitan dengan penelitian pengumpulan data dengan menggunakan metode observasi dan dokumentasi.

- Analisis permasalahan dan pengamatan data yang ada serta menarik kesimpulan. Data yang dibutuhkan berupa data *resources* (*timeslot*, dosen, mahasiswa, mata kuliah, gedung, ruangan, program studi) dan data *soft constraint* (Oates, 2014).
- Pengembangan *prototype*, dibangun dengan menggunakan paradigma pengembangan sistem perangkat lunak, *Software Development Life Cycle* (SDLC) antara lain (Maulana, 2017):
  1. Perencanaan, merupakan sebuah proses dasar untuk memahami mengapa sebuah sistem itu harus dibangun, dan pada fase ini memang diperlukan analisa kelayakan dengan mencari data atau melakukan proses *information gathering* kepada pengguna.
  2. Analisa, merupakan sebuah proses investigasi terhadap sistem yang sedang berjalan itu sendiri dengan tujuan untuk mendapatkan jawaban mengenai pengguna sistem, cara kerjanya yaitu sistem dan waktu penggunaan sistem.
  3. Rancangan, merupakan proses penentuan cara kerja sistem dalam hal *architecture design*, *interface design*, *database* dan spesifikasi file, dan *program design*. Hasil dari proses perancangan ini akan didapatkan spesifikasi sistem.
  4. Implementasi, merupakan proses pembangunan dan pengujian sistem, instalasi sistem, dan rencana dukungan sistem.
  5. Pengujian, merupakan pengujian terhadap simulator yang telah dibangun, apakah sudah bekerja dengan benar.

### 1.5 Sistematika Penyajian

Secara garis besar laporan TA ini dibagi dalam beberapa bab. Bab I yaitu Pendahuluan. Bab ini dijelaskan mengenai gambaran tentang isi dari materi tugas akhir yang akan dikerjakan. Bab II yaitu Tinjauan Pustaka, bab ini berisi rangkuman informasi yang berhubungan dengan topik tugas akhir yang akan dikerjakan. Bab III Analisis yaitu Permasalahan dan Analisis Algoritma, bab ini dijelaskan



mengenai permasalahan pada penjadwalan kuliah serta penjelasan singkat mengenai analisis penggunaan algoritma penjadwalan kuliah. Bab IV Implementasi dan Pengujian, bab ini dijelaskan mengenai implementasi dan pengujian terhadap *prototype* penjadwalan kuliah yang telah dirancang bab ini dijelaskan mengenai bagaimana kegiatan-kegiatan yang akan dilakukan untuk merancang pembangunan aplikasi penjadwalan kuliah. Pada bagian Pengujian terdapat pengujian direncanakan, metode pengujian yang digunakan, desain dari pengujian, serta pelaksanaan dan hasilnya. Bab V yaitu Hasil dan Pembahasan, bab ini dijelaskan mengenai hasil serta pembahasan yang diperoleh pada saat implementasi terhadap sistem penjadwalan kuliah dilakukan. Bab VI yaitu Kesimpulan dan Saran, bab ini dijelaskan mengenai kesimpulan dan saran sesudah tugas akhir dilakukan

## **BAB 2**

### **Tinjauan Pustaka**

Bab 2 dijelaskan mengenai landasan teori yang didapatkan dari pustaka yang berhubungan dengan objek penelitian TA.

#### **2.1. Penjadwalan**

Penjadwalan merupakan salah satu alternatif untuk pengalokasian sumber daya atau mesin-mesin yang dapat digunakan untuk mengatur sekumpulan kegiatan dalam jangka waktu tertentu (Baker & Kenneth, 1974).

Adapun pembagian dari penjadwalan akademik (*academic timetables*), adalah sebagai berikut (Arviani, 2013):

##### **1. Penjadwalan Sekolah**

Penjadwalan sekolah merupakan salah satu penjadwalan akademik yang diatur oleh kurikulum atau sering disebut dengan penjadwalan secara nasional. Jadwal dibuat sekali dalam seminggu dimana setiap kelas terdiri dari seorang pengajar yang bertanggung jawab terhadap kelas dan siswa tersebut dalam periode waktu yang telah ditentukan.

##### **2. Penjadwalan Mata Kuliah**

Penjadwalan mata kuliah merupakan salah satu penjadwalan akademik berisi sekumpulan jadwal dari proses perkuliahan. Adapun karakteristik mata kuliah yang terdiri dari komponen ruangan dan periode waktu adalah sebagai berikut:

- a. Penjadwalan mata kuliah memiliki syarat bahwa setiap mahasiswa dapat mengampuh jumlah mata kuliah yang berbeda.
- b. Ketersediaan ruangan pada penjadwalan mata kuliah sangatlah berperan penting.
- c. Penjadwalan mata kuliah harus memenuhi syarat bahwa apabila terdapat dua ruangan dengan mahasiswa yang sama, maka ruangan tersebut tidak dapat dijadwalkan pada waktu bersamaan.
- d. Penjadwalan Ujian

Penjadwalan ujian merupakan salah satu penjadwalan akademik dengan karakteristik sebagai berikut:

- e. Adapun pada penjadwalan ujian yaitu terdapat satu ujian untuk tiap mata kuliah.
- Mahasiswa dapat memiliki jadwal ujian lebih dari satu pada waktu tertentu maupun tidak.
- Mahasiswa yang mengikuti satu ujian dapat memiliki lebih dari satu ruangan.

### **2.1.1 Penjadwalan Mata Kuliah**

Sistem penjadwalan kuliah merupakan masalah dengan tingkat kompleksitas begitu tinggi (Burke & Varley, 1997) serta multidimensi yang menyangkut sejumlah mahasiswa, dosen dan atau asisten dosen yang ditugaskan pada ruangan dan waktu untuk satu acara perkuliahan (S.Jat & S.Yang, 2009). Penjadwalan pada perguruan tinggi fokus pada pengelolaan data pada sumber daya dan dengan mempertimbangkan tantangan *constraint*. Proses penjadwalan berhubungan dengan variabel seperti ruangan, dosen dan atau asisten dosen, waktu, mata pelajaran, serta jumlah semester.

### **2.1.2 Batasan-Batasan Masalah pada Penjadwalan Mata Kuliah**

Untuk pembuatan jadwal perkuliahan manajemen akan selalu mengalami kesulitan dalam mengatur sumber daya yang ada dan *constraints*, sebagai akibatnya hasil yang diperoleh tidak efisien. Jadwal perkuliahan terdiri dari satu *set* program yang akan dijadwalkan pada periode hari dan jam serta memperhatikan *constraints* yang ada. Adapun *hard constraints* dan *soft constraints* dalam proses pembuatan jadwal kuliah (Shiau, 2011; Mansur, 2014) sebagai berikut:

#### **1. *Hard Constraints***

- a. Dalam satu waktu kuliah, instruktur hanya dapat mengajar mata kuliah sebanyak satu kali.
- b. Seorang dosen atau instruktur pada setiap waktu perkuliahan hanya dapat mengajar satu mata kuliah.
- c. Mahasiswa hanya dapat menghadiri satu waktu perkuliahan.

- d. Setiap ruangan hanya dapat digunakan pada satu perkuliahan yang dilakukan pada waktu yang sama.
- e. Tidak memperhatikan periode waktu untuk kegiatan ekstrakurikuler seperti waktu makan siang dan olahraga.
- f. Laboratorium komputer akan digunakan untuk jadwal mata kuliah tertentu.

## 2. *Soft Constraints*

- a. Dosen atau instruktur dan mahasiswa dapat memilih hari dan jam yang disukai berdasarkan waktu yang tersedia.
- b. Dosen atau instruktur dapat menentukan jumlah jam perkuliahan untuk periode waktu tertentu.
- c. Meminimalkan perpindahan mahasiswa untuk melakukan pergantian ruangan serta jadwal dan hari oleh dosen.

### 2.2. **Constraint Satisfaction Problem**

*Constraint Satisfaction Problem* (CSP) adalah sebuah teknik untuk mendapatkan suatu penyelesaian dari sebuah persoalan melalui pencarian objek atau kondisi yang memenuhi satu atau lebih kriteria (Bartak & Roman, 2003).

*Constraint Satisfaction Problem* memiliki tiga komponen yang perlu diperhatikan dalam pendekatannya, antara lain (Russell & Norvig, 2010) :

1. *Constraint*, merupakan suatu aturan yang ditentukan untuk mengatur nilai yang boleh diisikan ke dalam variabel atau kombinasi variabel. Terdapat beberapa jenis *constraint*, diantaranya *unary* (menyatakan persyaratan sepasang variabel), *binary* (menyatakan persyaratan sepasang variabel), *n-ary* (menyatakan persyaratan tiga atau lebih variabel), dan *preference* (syarat yang sebaiknya dipenuhi, tetapi tidak harus).
2. *Domain*, merupakan kumpulan nilai legal diisi ke dalam variabel. Dengan kata lain, sebuah domain akan membatasi nilai suatu variabel.
3. *Variabel*, merupakan suatu penampung yang dapat diisi dengan berbagai nilai. Biasanya persoalan dimulai di sini, yaitu ketika variabel harus diisi oleh domain yang telah memenuhi *constraint*.

Untuk menentukan isi sebuah variabel yang hendak diisi, dapat dilakukan melalui dua cara yaitu (Gunawan & Toba, 2016):

1. *Most Constrained Variable* Penentuan variabel yang pertama diisi dan berikutnya dimulai dari variabel yang paling banyak mengandung *constraint*.
2. *Least Constrained Variable* Penentuan variabel yang pertama diisi dan berikutnya dimulai dari variabel yang paling sedikit mengandung *constraint*.

Menurut (Muhyi, 2017) pemodelan penjadwalan kuliah dimulai dengan penelaahan data-data (atau dimensi-dimensi) yang umum dan mendasar. Arti dari dimensi di sini adalah himpunan objek yang sama, dan terlibat dalam penjadwalan kuliah. Dalam perhitungan, dimensi-dimensi ini menggunakan indeks angka integer, yang merepresentasikan isi anggotanya. Jadi, dimensi ini lebih tepat jika dimodelkan sebagai array. Misalkan dimensi D adalah himpunan semua huruf kapital, maka  $D = \{A, B, C, D, \dots, Z\}$ .

### 2.3. Algoritma Backtracking

Algoritma *Backtracking* adalah pendekatan *brute-force* yang sudah diperbaiki, yang secara sistematis mencari sebuah solusi atas sebuah masalah di antara semua pilihan yang ada (Gurari, 1999).

Skema umum untuk langkah-langkah pencarian sesuai algoritma *Backtracking* adalah seperti di bawah ini yang dibuat dalam bentuk *pseudocode* [M & M, 2013].

```

procedure Backtracking (input
n:integer)

Deklarasi
k : Integer

Algoritma
k ← 1
while k > 0 do
  if (x[k] belum dicoba sedemikian
  sehingga
    x[k] ← T(k) and (B(x[1],
x[2], ...,x[k]) =
    true) then
  If ((x[1], x[2], ... x[k] adalah
  lintasan dari
    Akar ke daun) then
    cetakSolusi(x)
  endif
  k ← k+1 {indeks anggota
  tuple berikutnya}
  else{x[1], x[2], ... x[k] tidak
  mengarah ke
    simpul solusi}
    k ← k-1 {runut balik ke
    anggota tuple
      sebelumnya}
  endif
endwhile

```

Gambar 2.1 Pseudocode Backtracking

Langkah-langkah pencarian algoritma *Backtracking* [M & M, 2013]:

- Solusi dicari dengan membentuk lintasan dari akar ke daun. Simpul yang sudah dilahirkan dinamakan simpul hidup dan simpul hidup yang diperluas dinamakan simpul-E (Expand-node).
- Jika lintasan yang diperoleh dari perluasan simpul-E tidak mengarah ke solusi, maka simpul itu akan menjadi simpul mati dimana simpul itu tidak akan diperluas lagi.
- Jika posisi terakhir ada simpul mati, maka pencarian dilakukan dengan membangkitkan simpul anak yang lainnya dan jika tidak ada simpul *child* (anak) maka dilakukan *backtracking* ke simpul induk.

- Pencarian dihentikan jika kita telah menemukan solusi atau tidak ada simpul hidup yang diperlukan.

#### 2.4. Lintasan Terpendek

Lintasan atau jalur terpendek adalah jalur minimum yang diperlukan untuk mencapai suatu lokasi ke lokasi lainnya (Yusaputra, 2013). Sedangkan menurut (Hayati & Yohanes, 2014) lintasan terpendek adalah lintasan atau jalur yang dilalui dari satu titik atau *node* ke titik lainnya dengan bobot yang paling kecil. Tujuan dari pencarian lintasan atau jalur terpendek adalah untuk mendapatkan biaya atau jarak yang paling optimal dalam sebuah perjalanan dari lokasi awal ke lokasi tujuan (Hayati & Yohanes, 2014). Dalam pencarian lintasan atau jalur terpendek, terdapat metode atau algoritma yang dapat digunakan. Algoritma adalah urutan dari intruksi atau perintah yang digunakan untuk menyelesaikan suatu masalah (Levitin, 2012). Algoritma pencarian jalur atau *route* bertujuan untuk memilih *route* paling optimal dari awal ke tempat yang dituju. Algoritma yang digunakan penulis dalam penyelesain pencarian jalur atau lintasan terpendek pada pengerjaan tugas akhir ini yaitu Algoritma Dijkstra.

#### 2.5. Algoritma Dijkstra

Algoritma Dijkstra merupakan salah satu metode yang sering digunakan untuk menyelesaikan jarak terpendek atau *shortest path*. Algoritma Dijkstra digunakan pada *undirected* atau graf tidak berarah dan *directed Graphs* atau graf berarah dengan bobot-bobot sisi yang bernilai tidak negatif (Levitin, 2012). *Graphs* tak berarah merupakan *Graph* dengan sisi yang tidak memiliki orientasi arah. Urutan pasangan simpul yang dihubungkan oleh sisi tidak diperhatikan, sehingga  $(u,v) = (v,u)$  adalah sisi yang sama (Levitin, 2012). *Graph* berarah (*directed Graph*) adalah *Graph* yang sisinya diberikan orientasi arah, sehingga pada *Graph* berarah  $(u,v)$  dan  $(v, u)$  menyatakan dua busur yang berbeda. Busur dengan  $(u,v)$  simpul  $u$  dinamakan simpul asal dan simpul  $v$  dinamakan dengan simpul terminal (Levitin, 2012).

Algoritma Dijkstra ditemukan oleh seorang ilmuwan komputer bernama Edsger Wybe Dijkstra pada tahun 1930-2002. Berikut ini adalah *pseudocode* dari Algoritma Dijkstra (Cormen, 2009):

```

function Dijkstra(Graph, source):
  for each vertex v in Graph:
    dist[v] := infinity;

    previous[v] := undefined;
  end for

  dist[source] := 0;
  Q := the Set of all nodes in Graph;

  while Q is not empty:
    u := vertex in Q with smallest distance in dist[];
    remove u from Q;
    if dist[u] = infinity:
      break;
    end if
    for each neighbor v of u:

      alt := dist[u] + dist_between(u, v);
      if alt < dist[v]:
        dist[v] := alt;
        previous[v] := u;
        decrease-key v in Q;
      end if
    end for
  end while
  return dist;

```

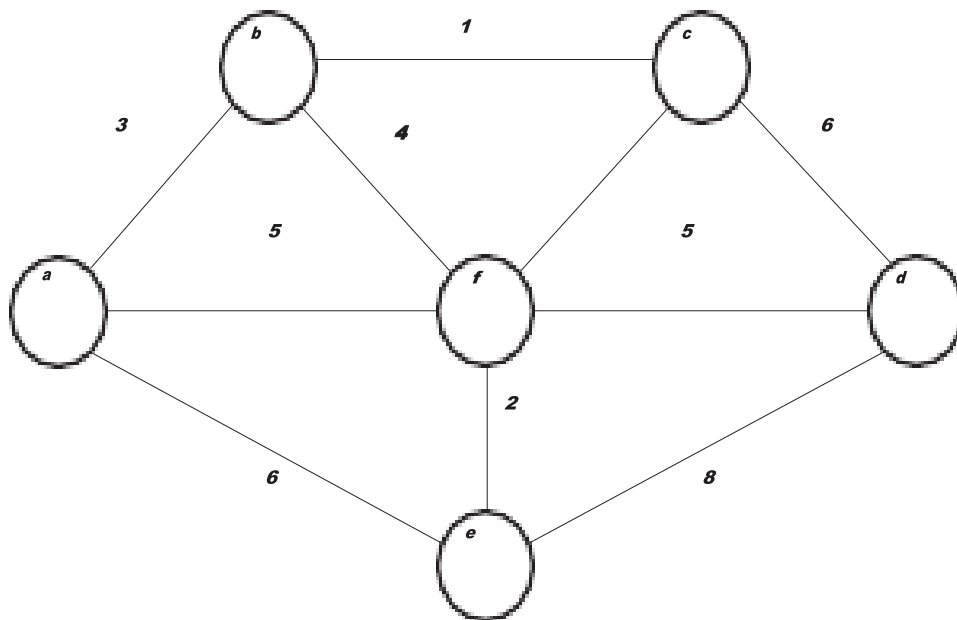
Keterangan dari *pseudocode* diatas adalah sebagai berikut (Cormen, 2009):

3. Pada bagian *pseudocode* yaitu baris pertama, melakukan inisialisasi yaitu penentuan *node* awal dan *node* yang akan dituju.
4. Pada bagian *pseudocode* yaitu baris kedua, ditetapkan *node* yang sudah pernah dikunjungi dengan status yaitu *empty*, dikarenakan tidak terjadi penelusuran *node*.
5. Pada bagian *pseudocode* yaitu baris ketiga dilakukan inisialisasi *queue* Q yang berisi semua *node* yang terhubung dengan simpul terdaftar pada *graph* akan dimasukkan pada variabel Q.
6. Pada bagian *pseudocode* yaitu baris keempat, akan dilakukan pengecekan kondisi selama Q tidak *empty* maka dijalankan bagian *pseudocode* yaitu baris kelima.



7. Pada bagian *pseudocode* yaitu baris kelima, jika Q memenuhi persyaratan pada baris keempat maka dilakukan pemilihan *node* yang berada dalam *queue* Q yang bernilai paling kecil atau paling minimum.
8. Pada bagian *pseudocode* yaitu baris keenam, merupakan hasil dari bagian *pseudocode* baris kelima yang ditandai sebagai *node* yang sudah pernah dikunjungi.
9. Pada bagian *pseudocode* yaitu baris ketujuh sampai dengan kedelapan mencari *node* baru yang pernah dikunjungi lalu kembali pada baris ketiga samapai semua *node* yang terdaftar pada baris pertama telah dikunjungi.

Pencarian jalur terpendek pada Dijkstra dapat disimulasikan dengan contoh sebagai berikut (Levitin, 2012)



**Gambar 2.2 Pencarian Jalur Terpendek pada Dijkstra**

Tree vertices	Remaining vertices	Illustration
$a(-, 0)$	$b(a, 3) \quad c(-, \infty) \quad d(a, 7) \quad e(-, \infty)$	
$b(a, 3)$	$c(b, 3 + 4) \quad d(b, 3 + 2) \quad e(-, \infty)$	
$d(b, 5)$	$c(b, 7) \quad e(d, 5 + 4)$	
$c(b, 7)$	$e(d, 9)$	
$e(d, 9)$		

Melalui Pencarian beberapa jalur, didapatkan bahwa jarak terpendek:

- Jarak dari a ke b melalui a, b adalah 3
- Jarak dari a ke d melalui a, b, d adalah 5
- Jarak dari a ke c melalui a, b, c adalah 7
- Jarak dari a ke e melalui a, b, d, e adalah 9

## 2.6. Penelitian Terdahulu

Pada bagian ini terdapat daftar dari *related work* atau penelitian yang dilakukan sebelumnya.

Tabel 2.1 Related Work

No	Penulis	Tahun	Tujuan	Algoritma dan Dataset	Hasilnya	Perbedaan dengan TA1-D3TI-08	Saran
1.	Rajasa Nagara E.R Silitonga, dan Yan Ramadaniel Christoper Pane	2016	Menggabungkan dan menerapkan penggabungan Algoritma Genetika dan PSO. Selanjutnya membuat aplikasi penjadwalan yang menerapkan hasil penggabungan algoritma Genetika dan PSO.	Implementasi menggunakan metode penggabungan algoritma genetika dengan algoritma PSO dengan menggunakan bahasa pemrograman java.	Penerapan algoritma penjadwalan pada aplikasi yang didasarkan dengan Algoritma Genetika dan Algoritma PSO. Kedua algoritma ini mampu menghasilkan solusi yang mendekati optimal dengan memperhatikan sumber daya yang ada dan kedua algoritma mampu memberikan proses terbaik mereka untuk mengurangi atau meningkatkan kinerja antar algoritma.	Membuat simulator penjadwalan kampus yang menerapkan hasil penggabungan algoritma <i>Backtracking</i> pada penjadwalan mata kuliah dengan Dijkstra pada <i>shortest path</i> .	<ol style="list-style-type: none"> <li>1. Diperlukan optimasi lebih lanjut pada Algoritma PSOM yang telah dihasilkan, seperti dengan melakukan pengurutan atau pengkategorian pada data ruangan pada saat proses inisialisasi partikel.</li> <li>2. Modifikasi Algoritma PSOM pada fungsi mutasi untuk meningkatkan akurasi yang lebih baik.</li> <li>3. diperlukan mekanisme penerima batasan secara dinamis sehingga dapat menyesuaikan dengan kebutuhan</li> </ol>

							pengguna serta penambahan batasan terhadap kombinasi kelas Setiap partikel.
	Goklas Henry Agus Panjaitan, dan Irmandes Roy Mangapul Tambunan	2015	Memodifikasi algoritma yang telah ada untuk digunakan dalam penjadwalan kuliah. Selanjutnya membangun aplikasi untuk penjadwalan kuliah.	Genetika	Merancang penjadwalan kuliah dengan menggunakan Algoritma Genetika. Algoritma Genetika akan digunakan dalam pembangunan aplikasi, sehingga untuk melakukan penjadwalan kuliah tidak manual tetapi menggunakan sistem aplikasi yang <i>ter-generate</i> otomatis dengan menggunakan algoritma genetik.		
2.	Antony Siagian, Jaka Putra Lesmana Tambunan,	2009	Tujuan pelaksanaan Tugas Akhir ini adalah studi dan implementasi	Aplikasi yang mengimplementasikan Algoritma Genetik untuk menyusun jadwal	Pada Tugas Akhir ini dilakukan kajian untuk studi dan implementasi Algoritma Genetik		1. Pengembangan lebih lanjut disarankan fleksibel menggunakan

	Aslon Damanik, dan Grace Dona Harlita Tarihoran		Algoritma Genetik untuk penjadwalan kuliah studi kasus Politeknik PIDel. Tujuan rinci Tugas Akhir ini adalah menghasilkan aplikasi yang menerapkan algoritma genetik untuk <i>generate</i> jadwal kuliah di PI-Del, serta melakukan pengujian dan evaluasi hasil pengujian dari aplikasi yang dihasilkan.	kuliah dibangun dengan menggunakan bahasa pemrograman C# dan basis data SQL Server 2005	pada penjadwalan kuliah dengan mengambil studi kasus yaitu penjadwalan kuliah di PI-Del.		batasan-batasan pada jadwal kuliah manapun. 2. Dilakukan modifikasi pada proses persilangan yang diimplementasikan pada aplikasi untuk menyusun jadwal kuliah yang dibangun untuk meningkatkan akurasi jadwal kuliah yang dihasilkan.
3.	Rosni Lumbantoruan, Yosua N. Simatupang, Miria N Siahaan, Mei H Pardede,	2012	Membangun sebuah aplikasi yang mampu mengotomatisasi penyusunan jadwal kuliah dengan	<i>Backtracking</i>	Dalam penelitian ini diperoleh bahwa <i>constraint</i> merupakan hal terpenting yang harus dipenuhi dalam pembuatan jadwal kuliah.		1. Jika hendak <i>generate</i> sebuah jadwal kuliah dengan <i>constraint</i> yang berbeda, maka Aplikasi Penjadwalan perlu dikustomasi.

	Jontri Pakpahan		menerapkan sebuah algoritma.		Maka untuk menyelesaikan masalah pemenuhan <i>Constraint</i> digunakan <i>Constraint Satisfaction Problem</i> dengan algoritma <i>Backtracking</i> .		<ol style="list-style-type: none"> <li>2. <i>Constraint</i> yang belum diaplikasikan, sebaiknya diterapkan pada pengembangan selanjutnya.</li> <li>3. Sebaiknya <i>path cost</i> menjadi bahan pertimbangan dalam penyusunan jadwal kuliah.</li> </ol>
4.	Chandra Ari Gunawan, Hapnes Toba	2016	Penelitian ini bertujuan untuk melakukan penyusunan jadwal otomatis pada pengolaan staf dengan menggunakan pendekatan CSP untuk menghasilkan pembuatan jadwal yang fleksibel.	Pendekatan CSP yaitu pendekatan yang menerapkan sistem pencarian suatu solusi dari masalah dengan cara mencari objek yang dapat memenuhi kriteria dalam penjadwalan tersebut.	<ol style="list-style-type: none"> <li>1. Menghasilkan penanaman bentrok jadwal dengan memberikan prioritas sesuai jadwal itu sendiri.</li> <li>2. Memberikan alternatif untuk menghindari penggunaan teknik optimasi. Implementasi CSP</li> </ol>		<ol style="list-style-type: none"> <li>1. Memperbaiki implementasi kelas-kelas dalam aplikasi agar generalisasi pembentukan <i>Constraint</i> agar dapat digunakan pada kasus lain.</li> <li>2. Tampilan dirancang agar lebih menarik, sederhana, dan konsisten.</li> <li>3. Aplikasi diharapkan mampu terhubung langsung dengan sistem akademik</li> </ol>

					menghasilkn penjadwalan yang alternatif dengan jumlah <i>staff</i> sekitar 20.30 orang.		
5.	Anisah Fitri, Inggih Permana, Arif Marsal	2016	Penelitian ini bertujuan untuk membangun penjadwalan khutbah IKMI (Ikatan Mesjid Indonesia) secara otomatis	Penelitian ini menggunakan penerapan CSP pada metode <i>priority schedulling</i>	3. Penjadwalan otomatis dengan 5 kali iterasi dengan keakuratan 93.7 % pada data sebanyak 50 orang dan 50 masjid		<ol style="list-style-type: none"> <li>1. Diperlukan pengujian performa teknik pada data yang lebih besar.</li> <li>2. Modifikasi algoritma</li> </ol>

## 2.7. Kesimpulan

Penggunaan Algoritma *Backtracking* dalam pencarian solusi terbaik, dengan cara membentuk lintasan dari akar ke daun. *Expand node* yang tidak mengarah ke solusi akan menjadi simpul mati, dimana simpul tersebut tidak akan diperluas lagi. Jika tidak ada simpul *child* maka dilakukan *backtrack* ke simpul induk. Pencarian akan berhenti, apabila telah menemukan solusi atau tidak ada simpul hidup yang diperlukan, sehingga algoritma ini dapat melakukan pencarian solusi penjadwalan dengan seluruh kemungkinan solusi. Pendekatan CSP digunakan untuk mencari solusi dengan mencari objek yang memenuhi persyaratan atau *constraint*. Algoritma Dijkstra digunakan untuk memperoleh solusi optimal dengan penyelesaian masalah jalur terpendek terkait jarak kelas pada jadwal yang beruntun.



## **BAB 3**

### **Analisis Permasalahan dan Penggunaan Algoritma**

#### **3.1. Analisis Penjadwalan Kuliah**

Pada bab ini dijelaskan analisis dan pengamatan data pada pembuatan jadwal kuliah di Institut Teknologi Del.

##### **3.1.1 Penjadwalan Kuliah di Institut Teknologi Del**

Penjadwalan mata kuliah yang ada di Institut Teknologi Del harus disusun berdasarkan ketentuan program studi. Program studi sarjana memiliki program dengan sistem kredit, yang berarti mahasiswa diharuskan mengambil jumlah SKS yang telah ditentukan dan dapat mengambil mata kuliah diluar mata kuliah semester yang ada. Pada program sarjana juga diberlakukan pengambilan mata kuliah pada semester pendek bagi mahasiswa yang berkeinginan untuk melakukan perbaikan nilai mata kuliah, maupun pengambilan mata kuliah yang seharusnya diambil disemester yang akan datang. Dalam hal ini, untuk mengatur perkuliahan dari masing-masing kedua program studi yang dimiliki di Institut Teknologi Del, terdapat beberapa sumber daya yang harus diperhatikan, antara lain:

##### **1. Mata kuliah**

Mata kuliah merupakan pelajaran yang harus diambil oleh mahasiswa berdasarkan jurusan yang dimiliki. Mata kuliah yang ada memiliki ketetapan jumlah SKS yang digunakan untuk menentukan jumlah sesi teori dan sesi praktikum. Jumlah sesi teori dan praktikum setiap mata kuliah berdasarkan SKS tersebut memiliki perbedaan antara program Diploma dan Sarjana. SKS Setiap mata kuliah pada program Diploma dan Sarjana dijelaskan sebagai berikut:

1. SKS mata kuliah untuk program Diploma adalah sebagai berikut:
  - a. Mata kuliah dengan dua SKS memiliki satu sesi teori dan satu praktikum atau dua sesi teori.
  - b. Mata kuliah dengan tiga SKS memiliki satu sesi teori dan satu praktikum atau dua sesi teori.
  - c. Mata kuliah dengan empat SKS memiliki satu sesi teori dan dua sesi praktikum atau satu sesi teori dan tiga sesi praktikum.

Catatan: Program diploma, satu sesi (satu jam) teori sama dengan satu SKS teori dan dua sesi (dua jam) praktikum sama dengan satu SKS praktikum.

2. SKS mata kuliah untuk program Sarjana adalah sebagai berikut:

- a. Mata kuliah dengan satu SKS memiliki satu sesi teori dan satu praktikum.
- b. Mata kuliah dengan dua SKS memiliki satu sesi teori dan satu praktikum atau dua sesi teori.
- c. Mata kuliah dengan tiga SKS memiliki dua sesi teori dan satu praktikum atau tiga sesi teori.
- d. Mata kuliah dengan empat SKS memiliki dua sesi teori dan dua sesi praktikum atau empat sesi teori.

Catatan: Program sarjana, satu SKS teori sama dengan satu atau dua sesi dan satu SKS praktikum sama dengan dua atau tiga sesi praktikum.

2. Kelas (Mahasiswa)

Kelas merupakan pengelompokan mahasiswa berdasarkan program studi yang dipilih dan tahun masuk mahasiswa. Apabila program studi yang diterima melebihi kapasitas satu kelas, maka mahasiswa dalam satu program studi tersebut akan dibagi menjadi beberapa kelas.

3. Dosen

Dosen merupakan tenaga ahli pengajar yang akan dialokasikan pada satu atau lebih penugasan mata kuliah yang harus diajarkan kepada mahasiswa yang mengambil mata kuliah tersebut. Batasan yang berlaku untuk dosen adalah setiap dosen tidak dapat mengajar dalam waktu yang sama untuk mata kuliah yang berbeda.

4. Ruangan

Ruangan merupakan pengalokasian tempat untuk melaksanakan perkuliahan. Ruangan yang digunakan harus memperhatikan kapasitas ruangan dan jumlah mahasiswa yang menggunakannya.

5. Slot Waktu

*Slot* waktu merupakan durasi yang dimiliki dalam proses perkuliahan. Waktu efektif perkuliahan di IT Del terdiri dari lima hari setiap minggunya, terhitung dari

hari senin hingga hari jumat. Jam akademik memiliki delapan jam sesi perharinya yang terdiri dari sesi teori dan praktikum.

Masalah yang sering terjadi pada penjadwalan mata kuliah Institut Teknologi Del ialah sulitnya mengkombinasikan setiap sumber daya yang tersedia sehingga menimbulkan konflik atau bentrok ketika mengalokasikannya. Hal ini disebabkan karena penjadwalan kuliah saat ini masih menggunakan *microsoft excel*. Penjadwalan yang menggunakan cara seperti ini, akan sulit untuk mendapatkan hasil yang optimal karena membutuhkan banyak waktu dan energi dalam mengkombinasikan sumber daya yang telah ada.

### **3.1.2 Batasan-Batasan Penjadwalan Mata Kuliah di Institut Teknologi Del**

Pada bagian ini dijelaskan *hard constraint* dan *soft constraint* pada penjadwalan matakuliah di Institut Teknologi Del.

#### **1. *Hard Constraints***

- Setiap matakuliah yang berbeda tidak boleh berada pada kelas dan waktu yang sama. Contohnya, matakuliah DASPRO kelas 31TI1 dengan matakuliah PRPL untuk 41TI pada hari senin pukul 09.00
- Dosen tidak boleh mengajar satu atau lebih matakuliah pada satu atau lebih kelas.
- Dosen tidak boleh mengajar pada kelas yang bersamaan pada waktu yang bersamaan. Contohnya, mata kuliah PSW 1 dan PAP yang diajarkan oleh dosen yang sama yaitu TNT pada hari selasa pukul 13.00.
- Kapasitas ruangan tidak diperbolehkan lebih kecil dibandingkan jumlah mahasiswa yang mengambil mata kuliah tersebut. Misalnya, 33TI 1 dengan 43TI mengambil mata kuliah KEPAL dan masuk di kelas 711.
- Setiap mata kuliah harus dijadwalkan sesuai dengan banyaknya jumlah SKS. Misalnya, mata kuliah PUPL berjumlah tiga sks, sehingga jam yang dimiliki satu jam sesi teori dan dua jam untuk dua sesi praktikum.
- Sesi praktikum dapat dilakukan apabila sudah terlebih dahulu melakukan sesi teori.

- Jumlah kelas yang masuk mata kuliah tidak boleh melebihi jumlah ruangan yang tersedia.

### 3.1.3 Data Penjadwalan Kuliah

Data yang digunakan dalam pembuatan jadwal mata kuliah di Institut Teknologi Del adalah mata kuliah, dosen, *slot* waktu, ruangan, dan kelas. Tugas Akhir ini menggunakan data semester ganjil sebagai sumber data *valid* dalam membangun sebuah *prototype* penjadwalan mata kuliah. Selanjutnya akan dilakukan pengujian terhadap sumber daya, sehingga menghasilkan jadwal mata kuliah yang optimal.

Data Jadwal Institut Teknologi Del Semester Ganjil 2018/2019

#### 1. Data Waktu

Tabel 3.1 Komponen Waktu

Slot Waktu	Sesi	Waktu
1	Sesi 1 Senin	08.00 - 08.50
2	Sesi 2 Senin	09.00 - 09.50
3	Sesi 3 Senin	10.00 - 10.50
4	Sesi 4 Senin	11.00 - 11.50
5	Sesi 5 Senin	13.00 - 13.50
6	Sesi 6 Senin	14.00 - 14.50
7	Sesi 7 Senin	15.00 - 15.50
8	Sesi 8 Senin	16.00 - 16.50
9	Sesi 1 Selasa	08.00 - 08.50
10	Sesi 2 Selasa	09.00 - 09.50
11	Sesi 3 Selasa	10.00 - 10.50
12	Sesi 4 Selasa	11.00 - 11.50
13	Sesi 5 Selasa	13.00 - 13.50
14	Sesi 6 Selasa	14.00 - 14.50
15	Sesi 7 Selasa	15.00 - 15.50
16	Sesi 8 Selasa	16.00 - 16.50
17	Sesi 1 Rabu	08.00 - 08.50
18	Sesi 2 Rabu	09.00 - 09.50
19	Sesi 3 Rabu	10.00 - 10.50
20	Sesi 4 Rabu	11.00 - 11.50
21	Sesi 5 Rabu	13.00 - 13.50
22	Sesi 6 Rabu	14.00 - 14.50
23	Sesi 7 Rabu	15.00 - 15.50
24	Sesi 8 Rabu	16.00 - 16.50
25	Sesi 1 Kamis	08.00 - 08.50

Slot Waktu	Sesi	Waktu
26	Sesi 2 Kamis	09.00 - 09.50
27	Sesi 3 Kamis	10.00 - 10.50
28	Sesi 4 Kamis	11.00 - 11.50
29	Sesi 5 Kamis	13.00 - 13.50
30	Sesi 6 Kamis	14.00 - 14.50
31	Sesi 7 Kamis	15.00 - 15.50
32	Sesi 8 Kamis	16.00 - 16.50
33	Sesi 1 Jumat	08.00 - 08.50
34	Sesi 2 Jumat	09.00 - 09.50
35	Sesi 3 Jumat	10.00 - 10.50
36	Sesi 4 Jumat	11.00 - 11.50
37	Sesi 6 Jumat	14.30 – 15.50
38	Sesi 7 Jumat	15.00 - 15.50
39	Sesi 8 Jumat	16.00 - 16.50

Tabel diatas merupakan data waktu yang digunakan untuk perkuliahan yaitu 40 *slot* waktu. Jumlah *slot* waktu sebanyak 40 diperoleh dari delapan *slot* waktu dari hari senin sampai hari jumat.

Catatan: sesi lima yaitu 13.00- 13.50 dan sesi 6 yaitu 14.00 – 14.30 pada hari jumat tidak digunakan pada perkuliahan karena sesi tersebut digunakan untuk ibadah wajib mahasiswa Institut Teknologi Del.

## 2. Data Rombongan Kelas

Tabel 3.2 Data Rombongan Kelas

No	Kelas	Keterangan	Jumlah Mahasiswa
1	11MR1	S1 Manajemen Rekayasa 01 Angkatan 2018	33
2	11MR2	S1 Manajemen Rekayasa 02 Angkatan 2018	33
3	11SI1	S1 Sistem Informasi 01 Angkatan 2018	30
4	11SI2	S1 Sistem Informasi 02 Angkatan 2018	29
5	11TB	S1 Teknik Bioproses Angkatan 2018	21
6	11TE1	S1 Teknik Elektro 01 Angkatan 2018	26
7	11TE2	S1 Teknik Elektro 02 Angkatan 2018	25
8	11TI1	S1 Teknik Informatika 01 Angkatan 2018	32
9	11TI2	S1 Teknik Informatika 02 Angkatan 2018	31
10	12MR1	S1 Manajemen Rekayasa 01 Angkatan 2017	30

No	Kelas	Keterangan	Jumlah Mahasiswa
11	12MR2	S1 Manajemen Rekayasa 02 Angkatan 2017	30
12	12SI1	S1 Sistem Informasi 01 Angkatan 2017	30
13	12SI2	S1 Sistem Informasi 02 Angkatan 2017	30
14	12TB	S1 Teknik Bioproses Angkatan 2017	30
15	12TE1	S1 Teknik Elektro 01 Angkatan 2017	29
16	12TE2	S1 Teknik Elektro 02 Angkatan 2017	26
17	12TI1	S1 Teknik Informatika 01 Angkatan 2017	30
18	12TI2	S1 Teknik Informatika 02 Angkatan 2017	30
19	13MR1	S1 Manajemen Rekayasa 01 Angkatan 2016	30
20	13MR2	S1 Manajemen Rekayasa 02 Angkatan 2016	30
21	13SI1	S1 Sistem Informasi 01 Angkatan 2016	28
22	13SI2	S1 Sistem Informasi 02 Angkatan 2016	26
23	13TB	S1 Teknik Bioproses Angkatan 2016	30
24	13TE1	S1 Teknik Elektro 01 Angkatan 2016	28
25	13TE2	S1 Teknik Elektro 02 Angkatan 2016	27
26	13TI1	S1 Teknik Informatika 01 Angkatan 2016	29
27	13TI2	S1 Teknik Informatika 02 Angkatan 2016	26
28	14MR1	S1 Manajemen Rekayasa 01 Angkatan 2015	30
29	14MR2	S1 Manajemen Rekayasa 02 Angkatan 2015	27
30	14SI1	S1 Sistem Informasi 01 Angkatan 2015	25
31	14SI2	S1 Sistem Informasi 02 Angkatan 2015	28
32	14TB	S1 Teknik Bioproses Angkatan 2015	30
33	14TE1	S1 Teknik Elektro 01 Angkatan 2015	26
34	14TE2	S1 Teknik Elektro 02 Angkatan 2015	24
35	14TI1	S1 Teknik Informatika 01 Angkatan 2015	30
36	14TI2	S1 Teknik Informatika 02 Angkatan 2015	27
37	31TI1	D3 Teknik Informatika 01 Angkatan 2018	32
38	31TI2	D3 Teknik Informatika 02 Angkatan 2018	33
39	31TK1	D3 Teknik Komputer 01 Angkatan 2018	28
40	31TK2	D3 Teknik Komputer 02 Angkatan 2018	30
41	32TI1	D3 Teknik Informatika 01 Angkatan 2017	30
42	32TI2	D3 Teknik Informatika 02 Angkatan 2017	30
43	32TK	D3 Teknik Komputer Angkatan 2017	30
44	33TI1	D3 Teknik Informatika 01 Angkatan 2016	28
45	33TI2	D3 Teknik Informatika 02 Angkatan 2016	16
46	33TK	D3 Teknik Komputer Angkatan 2016	27
47	41TI	D4 Teknik Informatika Angkatan 2018	29
48	42TI	D4 Teknik Informatika Angkatan 2017	30

No	Kelas	Keterangan	Jumlah Mahasiswa
49	43TI	D4 Teknik Informatika Angkatan 2016	26
50	44TI	D4 Teknik Informatika Angkatan 2015	30

Tabel diatas merupakan data rombongan kelas yang digunakan dalam pembuatan *prototype* di Institut Teknologi Del. Data rombongan kelas terdiri dari 50 kelas sampai tahun ajaran 2018/2019.

### 3. Data Dosen

Tabel 3.3 Data Dosen

No	Inisial Dosen	Mata Kuliah yang Diajarkan
1	ABS	RE, PCD, PSDF, STE, PSD
2	ACB	PUPL, KUPEL
3	AAD	UPL
4	AFS	ELEKTRO II, FISDAS I
5	AHF	DSI, BAMI
6	AMS	RPLSD, MANPRO, KUPEL, METPEN, PRPL
7	ANA	DEL CHA, OMI, PPBO, KEBIS, KP
8	ANM	KST, PRAK MIKRUM
9	ARR	PRAK PIP, TRB
10	ART	CERTAN, APPEL, MANPRO, BPR, PSW
11	ASD	PTI, PSW, PSW I, PABI, PENGPRO
12	BLT	ATI
13	BSA	PCD
14	CJS	AKBI, KKB, EKOTEK
15	DDA	APKP II, SPP
16	DWS	TECHNO, EKOTEK, AKBI, PENGPROD, WIRA, CAPSTONE
17	EAN	MIKRUM, PRAK MIKRUM, APKP II, AITB
18	EBN	KOMIN
19	ESS	PAP, PJK, TA1, ANJAR I, JARKOM, ANJAR II
20	GDE	RE, SISDIG
21	GFP	SISDIG
22	HSS	SIMIS, REMU, TPE
23	ICB	ALSRUDAT, PENGPRO, DSI
24	IFY	MADAS I, PROBSTAT, METPEN, PM
25	IHT	ELEKTRO II, SKD, MECRO, STE
26	IL	RPSLD
27	IPM	DACOM , AOK, PTI
28	MPR	ING I, ENG III

No	Inisialisasi Dosen	Mata Kuliah yang Diajarkan
29	IJK	PROBSTAT
30	LMG	PTI, LOGIN, PG, DETING
31	MMM	STA, PP
32	MMS	BASDATLAN, MMPE, ADJAR, KAJAR. SISOP
33	MNP	FENOPOR, OFP II, PEN I
34	MSL	MADAS I, ALJALI, MATEK I, MATDIS, CRIPTO, SGS
35	MSS	PABWE, AUDTI, SISENT
36	MVL	JARKOM, DACOM, TA I, SISTAN, SISOP
37	NMA	FISDAS I, MATDIS, KP
38	NSS	VGT, SIX SIGMA
39	PAT	BASDAT, BAMI
40	PDS	ELEKTRO II, PRAK ELEKTRO, ANJAR I, PIC, DE
41	PAT	TA I, AUDTI
42	PN	PC
43	RCS	PMR, PMB, PPP, KEBIS, KOMIN
44	RDT	TAI, PSW I, OOSD
45	RFK	PRAK KIMOR, BIOKIM, PEN I
46	RIR	FENOPOR, PP, UPL
47	RIS	TECHNO, PAP, KREN, DETING, WIRA, MAIN
48	RML	MADAS I, PROBSTAT, DSI, MATDIS
49	RMM	ENG I, ING III, ING IV
50	RMS	ENG I
51	RZS	FISDAS I, TERMO, TPE, CAPSTONE
52	SAM	ENG I, ENG III, ING V, CAPSTONE, PPBO, MANSTRA
53	SFA	KIMDAS, MTE, TA I, MECRO
54	SGS	DSI, SISENT, BAMI, KP, BASDATLAN, CERTAN
55	SML	MATDIS
56	THS	ALSRUDAT, BASDATLAN, TA I, KESLAN
57	TLG	TTKI, IND
58	TLS	SISOP, DACOM, ADJAR, PAP
59	TMP	DASPRO, PM
60	TMS	REMU
61	TNT	PABWE, PAP, KEPAL, BPR, PSW I, CRIPTO
62	VES	SBD, PRPL, MAIN
63	WSA	PSI
64	YBN	ALNUM, PDO
65	YMA	EKOTEK, TPE, VGT, KKB, ASTEK
66	YOP	PPI
67	YUL	KIMFIS, PRAK KIMFIS, AKRB, PP, SPP
68	YBN	ALU, MATDIS
69	YUS	MODSIM, PENGPRO, ALU, APPEL
70	YYS	MATDIS, TBFA, PROG I
71	YHP	CERTAN, ALU, ICB, DSI



Tabel diatas merupakan data dosen pengajar mata kuliah yang digunakan dalam pembuatan *prototype* di Institut Teknologi Del. Data dosen berdasarkan inisial terdiri dari 71 Dosen sampai ajaran 2018/2019.

#### 4. Data Mata Kuliah

Tabel 3.4 Data Mata Kuliah Program Sarjana

No	Kode Mata Kuliah	Mata Kuliah	Singkatan	SK S	Komposisi	Kelas
1	MAS1101	Matematika Dasar I	MADAS I	4	2T,2T	11TI1, 11TI2
2	FIS1101	Fisika Dasar I (+P)	FISDAS I	4	2T,2T,3P	11TI1, 11TI2
3	ISS1101	Sains Teknologi dan Seni di Masyarakat	STA	2	2T	11TI1, 11TI2
4	KUS1102	Bahasa Inggris I	ENG I	2	2T	11TI1, 11TI2
5	IFS1101	Pengantar Teknologi Informasi	PTI	2	2T	11TI1, 11TI2
6	KUS1101	Pembentukan Karakter Del	DEL CHA	2	2T	11TI1, 11TI2
7	ISS1102	Programming I	PROG I	2	2T,2P	11TI1, 11TI2
8	ELS2180	Sistem Digital	SISDIG	3	2T,1T	12TI1, 12TI2
9	MAS2102	Matematika Diskrit	MATDIS	3	2T,1T	12TI1, 12TI2
10	ISS2101	Basis Data	BASDAT	3	2T,2P	12TI1, 12TI2
11	IFS2102	Logika Informatika	LOGIN	3	2T,1T	12TI1, 12TI2
12	MAS2001	Probabilitas dan Statistika	PROBSTAT	3	2T,1T	12TI1, 12TI2
13	IFS2101	Algoritma dan Struktur Data	ALSRUDAT	3	2T,2P,1T	12TI1, 12TI2
14	NWS3101	Jaringan Komputer	JARKOM	2	2T,2P	13TI1, 13TI2
15	IFS3101	Kecerdasan Buatan	CERTAN	3	2T,2P,1T	13TI1, 13TI2

No	Kode Mata Kuliah	Mata Kuliah	Singkatan	SK S	Komposisi	Kelas
16	IFS3102	Teori Bahasa Formal dan Automata	TBFA	3	2T,2P,1T	13TI1, 13TI2
17	IFS3103	Pengembangan Aplikasi Berbasis Web	PABWE	3	2T,2P,2T	13TI1, 13TI2
18	KUS2001	Bahasa Inggris III	ENG III	2	2T	13TI1, 13TI2
19	MRS3108	Technopreneurship	TECHNO	2	1T, 2P	13TI1, 13TI2
20	ISS1001	Dasar Sistem Informasi	DSI	3	2T,2P	13TI1, 13TI2
21	MAS1101	Matematika Dasar I	MADAS I	4	2T,2T	11SI1, 11SI2
22	FIS1101	Fisika Dasar I (+P)	FISDAS I	4	2T,2T,3P	11SI1, 11SI2
23	ISS1101	Sains Teknologi dan Seni di Masyarakat	STA	2	2T	11SI1, 11SI2
24	KUS1102	Bahasa Inggris I	ENG I	2	2T	11SI1, 11SI2
25	IFS1101	Pengantar Teknologi Informasi	PTI	2	2T	11SI1, 11SI2
26	KUS1101	Pembentukan Karakter Del	DEL CHA	2	2T	11SI1, 11SI2
27	ISS1102	Programming I	PROG I	2	2T,2P	11SI1, 11SI2
28	ELS2180	Sistem Digital	SISDIG	3	2T,1T	12SI1, 12SI2
29	MAS2102	Matematika Diskrit	MATDIS	3	2T,1T	12SI1, 12SI2
30	ISS2101	Basis Data	BASDAT	3	2T,2P	12SI1, 12SI2
31	ISS2102	Organisasi dan Manajemen Industri	OMI	3	2T,1T	12SI1, 12SI2
32	MAS2001	Probabilitas dan Statistika	PROBSTAT	3	2T,1T	12SI1, 12SI2
33	IFS2101	Algoritma dan Struktur Data	ALSRUDAT	3	2T,3P,1T	12SI1, 12SI2

No	Kode Mata Kuliah	Mata Kuliah	Singkatan	SK S	Komposisi	Kelas
34	ISS3104	Pengantar Jaringan Komputer	PJK	3	2T,2P,1T	13SI1, 13SI2
35	ISS3102	Kecerdasan Buatan	CERTAN	4	2T,2P,2T	13SI1, 13SI2
36	ISS3101	Basis Data Lanjut	BASDATLAN	3	2T,3P,1T	13SI1, 13SI2
37	ISS3103	Pengembangan Aplikasi Berbasis Web	PABWE	4	2T,2P,2T, 2P	13SI1, 13SI2
38	KUS2001	Bahasa Inggris III	ENG III	2	2T	13SI1, 13SI2
39	MRS3180	Technopreneurship	TECHNO	2	1T, 2P	13SI1, 13SI2
40	MAS1101	Matematika Dasar I	MADAS I	4	2T, 2T	11TB
41	FIS1101	Fisika Dasar I (+P)	FISDAS I	4	2T, 3P, 2T	11TB
42	ISS1101 /IFS1103 /ELS1101 /BPS1101 /MRS1101	Sains Teknologi dan Seni di Masyarakat	STA	2	2T	11TB
43	KUS1101	Bahasa Inggris I	ENG I	2	2T	11TB
44	IFS1101	Pengantar Teknologi Informasi	PTI	2	2T	11TB
45	KUS1102	Pembentukan Karakter Del	DEL CHA	2	2T	11TB
46	BP			4	2T, 4P, 2T	11TB
47	BIS1201	Kimia Dasar (+P)	KIMDAS	4	2T, 2T	11TB
48	KIS2101	Praktikum Kimia Organik	PRAK KIMOR	1	4P	12TB
49	MAS2103	Aljabar Linear	ALJALI	3	2T, 1T	12TB
50	MAS2104	Analisis Numerik	ALNUM	3	2T, 1T	12TB

No	Kode Mata Kuliah	Mata Kuliah	Singkatan	SK S	Komposisi	Kelas
51	MAS2105	Persamaan Diferensial Ordiner	PDO	3	2T, 1T	12TB
52	BIS2101	Mikrobiologi Umum	MIKRUM	2	2T	12TB
53	BIS2102	Praktikum Mikrobiologi Umum	PRAK MIKRUM	1	4P	12TB
54	KIS2102	Kimia Fisik	KIMFIS	3	2T, 1T	12TB
55	KIS2103	Praktikum Kimia Fisik	PRAK KIMFIS	1	4P	12TB
56	KIS2104	Biokimia	BIOKIM	3	2T, 1T	12TB
57	MAS2001	Probabilitas dan Statistika	PROBSTAT	3	2T, 1T	13TB
58	BPS3105	Analisis Keekonomian Rekayasa Bioproses	AKRB	3	2T, 1T	13TB
59	BPS3101	Fenomena Transpor	FENOPOR	3	2T, 1T	13TB
60	BPS3102	Analisis Pemodelan dan Komputasi Proses II	APKP II	3	2T, 1T	13TB
61	BPS3103	Operasi Fisik Pendukung II (*)	OFP II	3	2T, 1T	13TB
62	BPS3104	Proses Pemisahan	PP	3	2T, 1T	13TB
63	BPS3107	Konstruksi Sosial Teknologi	KST	2	2T	13TB
64	BPS4101	Pengendalian dan Instrumentasi Proses	PIP	2	2T	14TB
65	BPS4102	Praktikum Pengendalian dan Instrumentasi Proses	PRAK PIP	1	4P	14TB
66	BPS4103	Sintesa dan Perancangan Proses	SPP	3	2T,1T	14TB

No	Kode Mata Kuliah	Mata Kuliah	Singkatan	SK S	Komposisi	Kelas
67	BPS4104	Utilitas dan Pengolahan Limbah	UPL	3	2T,1T	14TB
68	BPS4105	Teknik Reaktor Bioproses	TRB	3	2T,1T	14TB
69	BPS4190	Proyek Akhir I (Penelitian)	PEN I	2	2T	14TB
70	BPS4107	Aplikasi dan Industri Teknik Bioproses (Pilihan)	AITB	2	2T	14TB
71	KUS4101	Pancasila dan Kewarganegaraan	PC	2	2T	14TB
72		MATLAB	MATLAB	2	2P	14TB
73	MAS1101	Matematika Dasar I	MADAS I	3	2T, 1T	12MR1, 12MR2
74	FIS1101	Fisika Dasar I (+P)	FISDAS I	2	2T	12MR1, 12MR2
75	MRS1101	Sains Teknologi dan Seni di Masyarakat	STA	4	2T, 2T	12MR1, 12MR2
76	KUS1102	Bahasa Inggris I	ENG I	2	2T, 2P	12MR1, 12MR2
77	IFS1101	Pengantar Teknologi Informasi	PTI	3	2T, 1T	12MR1, 12MR2
78	KUS1101	Pembentukan Karakter Del	DEL CHA	3	2T, 1T	12MR1, 12MR2
79	MRS1102	Pengantar Manajemen Rekayasa	PMR	2	2T	12MR1, 12MR2

No	Kode Mata Kuliah	Mata Kuliah	Singkatan	SK S	Komposisi	Kelas
80	MRS3101	Perancangan Proses Bisnis dan Organisasi	PPBO		2T, 1T	13MR1, 13MR2
81	MRS3102	Akutansi Biaya	AKBI	3	2T, 1T	13MR1, 13MR2
82	MRS3104	Perancangan Model Bisnis	PMB	3	2T, 1T	13MR1, 13MR2
83	MRS3106	Perancangan dan Pengembangan Produk	PENGPROD	3	2T	13MR1, 13MR2
84	MRS3103	Penelitian Operasional II	OR II	5	2T, 1T	13MR1, 13MR2
85	MRS3107	Metodelogi Penelitian	METPEN	3	2T	13MR1, 13MR2
86	MRS3105	Penelitian Pasar dan Pemasaran	PPP	3	2T	13MR1, 13MR2
87	KUS2001	Bahasa Inggris III	ENG III	3	2T	13MR1
88	MAS1101	Matematika Dasar I	MADAS I	4	2T, 2T	11TE1, 11TE2
89	FIS1101	Fisika Dasar I (+P)	FISDAS I	4	2T, 2T, 3P	11TE1, 11TE2
90	ELS1101	Sains Teknologi dan Seni di Masyarakat	STA	2	2T	11TE1, 11TE2
91	KUS1102	Bahasa Inggris I	ENG I	2	2T	11TE1, 11TE2
92	IFS1101	Pengantar Teknologi Informasi	PTI	2	2T	11TE1, 11TE2
93	KUS1101	Pembentukan Karakter Del	DEL CHA	2	2T	11TE1, 11TE2

No	Kode Mata Kuliah	Mata Kuliah	Singkatan	SK S	Komposisi	Kelas
94	KUS1001	Tata Tulis Karya Ilmiah	TTKI	2	2T	11TE1, 11TE2
95	ELS2102	Praktikum Rangkaian Elektrik	PRAK RE	1	4P	12TE1, 12TE2
96	ELS2104	Praktikum Sistem Digital	PRAK SISDIG	1	4P	12TE1, 12TE2
97	ELS2101	Rangkaian Elektrik	RE	4	2T, 2T	12TE1, 12TE2
98	ELS2103	Sistem Digital	SISDIG	4	2T, 2T	12TE1, 12TE2
99	IFS2103	Pengantar Pemrograman	PENGPRO	2	2T, 2P	12TE1, 12TE2
100	MAS2102	Matematika Diskrit	MATDIS	3	2T, 1T	12TE1, 12TE2
101	MAS2101	Matematika Teknik I	MATEK I	3	2T, 1T	12TE1, 12TE2
102	NWS3102	Arsitektur Sistem Komputer	ASK	3	2T, 1T	13TE1
103	NWS3103	Praktikum Arsitektur Sistem Komputer	PRAK ASK	1	4P	13TE1
104	ELS3101	Elektronika II	ELEKTRO II	3	2T, 1T	13TE1
105	ELS3105	Material Teknik Elektro	MTE	3	2T, 1T	13TE1
106	ELS3103	Pengolahan Sinyal Digital	PSD	3	2T, 1T	13TE1
107	ELS3102	Praktikum Elektronika II	PRAK ELEKTRO II	1	2P	13TE1
108	ELS3104	Praktikum Pengolahan Sinyal Digital	PRAK PSD	1	2P	13TE1
109	ELS3106	Sistem Tenaga Elektrik	STE	3	2T, 1T	13TE1

Tabel 3.5 Data Mata Kuliah Program Diploma

No	Kode Mata Kuliah	Mata Kuliah	Singkatan	SKS	Komposisi	Kelas
1	CE311111	Bahasa Inggris I	ING I	2	1T, 2P	31TK

No	Kode Mata Kuliah	Mata Kuliah	Singkatan	SKS	Komposisi	Kelas
2	CE311212	Matematika Diskrit	MATDIS	3	1T, 2P, 1T	31TK
3	CE311213	Perancangan Antar Muka Pengguna	PAP	2	1T, 2P	31TK
4	CE311214	Organisasi dan Arsitektur Komputer	OAK	3	1T, 2P, 1T	31TK
5	CE311316	Dasar Pemrograman	DASPRO	3	1T, 2P, 2P	31TK
6	CE311317	Pengembangan Situs Web	PSW	3	1T, 2P, 2P	31TK
7	CE311315	Pengantar Teknologi Informasi	PTI	2	1T, 2P	31TK
8	IF321111	Del Character	DEL CHA	2	2T	31TK
9	IF321112	Bahasa Inggris III	ING III	2	1T, 2P	32TK
10	IF321314	Basis Data	BASDAT	3	1T, 2P, 2P	32TK
11	IF321313	Pengembangan Aplikasi Berbasis Internet	PABI	4	1T, 2P, 1T, 2P	32TK
12	IF321316	Komunikasi Data	DACOM	2	1T, 2P	32TK
13	IF321317	Antar Jaringan I	ANJAR I	3	1T, 2P, 1T	32TK
14	IF321111	Manajemen Media Penyimpanan Enterprise	MMPE	3	1T, 2P, 2P	32TK
15	IF321112	Probabilitas dan Statistika	PROBSTAT	3	1T, 2P, 1T	32TK
16	IF331111	Bahasa Inggris V	ING V	2	1T, 2P	33TK
17	CE331313	Tugas Akhir I	TA I	4	1T, 2P, 2P	33TK
18	CE331312	Administrasi Jaringan	ADJAR	3	1T, 2P, 2P	33TK
19	CE331314	Penerapan Infrastruktur Cloud	PIC	4	1T, 2P, 2P	33TK
20	CE331317	Sistem Tertanam	SISTAN	3	1T, 2P, 2P	33TK
21	IF321316	Antar Jaringan II	ANJAR II	3	1T, 2P, 1T	33TK



No	Kode Mata Kuliah	Mata Kuliah	Singkatan	SKS	Komposisi	Kelas
22	IF311211	Matematika Diskrit	MATDIS	3	1T, 2P,1T	41TI1, 31TI1, 31TI2
23	IF311112	Bahasa Inggris I	ING I	2	1T, 2P	41TI1, 31TI1, 31TI2
24	IF311313	Pengantar Teknologi Informasi	PTI	2	1T, 2P	41TI1, 31TI1, 31TI2
25	IF311314	Pengembangan Situs Web I	PSW I	3	1T, 2P,2P	41TI1, 31TI1, 31TI2
26	IF311215	Dasar Pemrograman	DASPRO	3	1T, 2P,2P	41TI1, 31TI1, 31TI2
27	IF311316	Sistem Operasi	SISOP	2	1T, 2P	41TI1, 31TI1, 31TI2
28	IF311317	Pengenalan Rekayasa Perangkat Lunak	PRPL	3	1T, 2P,1T	41TI1, 31TI1, 31TI2
29	KUS1101	Pembentukan Karakter Del	DEL CHA	2	2T	41TI1, 31TI1, 31TI2
30	IF421111	Bahasa Inggris III	ING III	2	1T, 2P	42TI, 32TI1, 32TI2
31	IF421312	Object-Oriented Software Development	OOSD	4	1T, 2P,1TP,2P	42TI, 32TI1, 32TI2
32	IF421313	Sistem Basis Data	SBD	3	1T, 2P, 2P	42TI, 32TI1, 32TI2
33	IF421314	Perancangan Antarmuka Pengguna	PAP	3	1T, 2P, 2P	42TI, 32TI1, 32TI2
34	IF421115	Kreativitas dan Inovasi	KREN	3	1T, 2P, 2P	42TI, 32TI1, 32TI2
35	IF421316	Arsitektur dan Organisasi Komputer	AOK	3	1T, 2P, 1T	42TI, 32TI1, 32TI2

No	Kode Mata Kuliah	Mata Kuliah	Singkatan	SKS	Komposisi	Kelas
36	IF431111	Bahasa Inggris V	ING V	2	1T, 2P	43TI
37	IF431312	Pengembangan Aplikasi Game	PG	3	1T, 2P, 2P	43TI
38	IF431313	Algoritma Lanjut	ALU	3	1T, 2P, 1T	43TI
39	IF431315	Keamanan Perangkat Lunak	KEPAL	3	2T,3P	43TI
40		Bahasa Indonesia	IND	2	1T, 2P	43TI, 33TI
41	IF431117	Metode Penelitian	METPEN	3	1T,2P,1T	43TI
42	IF331111/IF431111	Bahasa Inggris V	ING V	2	1T, 2P	33TI
43	IE331315/IF431315	Pengembangan Aplikasi Game	PG	3	1T, 2P, 2P	33TI
44	IF331417/IF431417	Algoritma Lanjut	ALU	3	1T, 2P, 1T	33TI
45	IE331312/IF431312	Keamanan Perangkat Lunak	KEPAL	3	2T,3P	33TI
46	IF431316	Tugas Akhir I	TA I		1T, 2P,2P	33TI

Tabel diatas merupakan data mata kuliah program sarjana dan diploma di Institut Teknologi Del. Data berisi jumlah SKS yang akan menentukan jumlah sesi teori dan praktikum. Satu sesi teori sama dengan satu jam perkuliahan sedangkan satu sesi praktikum sama dengan dua jam perkuliahan.

## 5. Data Ruangan

Tabel 3.6 Komponen Ruangan

No	Ruangan	Jenis Ruangan	Kapasitas Maksimal	Utilisasi (Jam/minggu)
1	AUD	Ruang	10 kelas	40
2	GD516	Ruang	2 kelas	40
3	GD711	LAB	1 kelas	40
4	GD712	LAB	1 kelas	40
5	GD714	LAB	2 kelas	40
6	GD721	Ruang	2 kelas	40
7	GD722	Ruang	2 kelas	40

No	Ruangan	Jenis Ruangan	Kapasitas Maksimal	Utilisasi (Jam/minggu)
8	GD723	LAB	2 kelas	40
9	GD912	Ruang	2 kelas	40
10	GD913	Ruang	2 kelas	40
11	GD914	Ruang	2 kelas	40
12	GD916	Ruang	2 kelas	40
13	GD917	Ruang	2 kelas	40
14	GD923	Ruang	2 kelas	40
15	GD924	Ruang	2 kelas	40
16	GD925	Ruang	2 kelas	40
17	GD927	Ruang	2 kelas	40
18	GD928	Ruang	2 kelas	40
19	GD929	Ruang	2 kelas	40
20	GD933	Ruang	2 kelas	40
21	GD934	Ruang	2 kelas	40
22	GD935	Ruang	2 kelas	40
23	GD937	Ruang	2 kelas	40
24	GD938	Ruang	2 kelas	40
25	GD942	Ruang	2 kelas	40
26	GD943	Ruang	2 kelas	40
27	GD944	Ruang	2 ke las	40
28	GD513	LAB	1 kelas	40
29	GD514	LAB	1 kelas	40
30	GD515	LAB	1 kelas	40
31	GD525	LAB	1 kelas	40
32	GD713	LAB	1 kelas	40
33	GD724	LAB	1 kelas	40
34	GD725	LAB	1 kelas	40
35	GD726	LAB	1 kelas	40

Tabel diatas merupakan data ruangan yang dapat digunakan pada proses perkuliahan di Institut Teknologi Del yaitu sebanyak 35 ruangan, yang terdiri dari 12 ruangan laboratorium dan 23 ruangan digunakan untuk perkuliahan sesi teori. Jumlah kapasitas ruangan yang dapat ditempati mahasiswa dalam satu ruangan dibedakan menjadi tiga kategori antara lain 30, 60 dan 300 mahasiswa.

### 3.2. Analisis CSP (*Constraint Satisfaction Problem*)

CSP merupakan metode yang digunakan dalam pembuatan solusi-solusi yang memungkinkan dan tidak melanggar *constraint* yang ada, agar tidak ada *constraint* yang dilanggar dalam pembuatan jadwal. CSP menggunakan sistem penggolongan

*constraint* berdasarkan aturan, bahwa *constraint* yang ada, akan dipisahkan antara *constraint* yang sejenis. *constraint* yang dibuat, bertujuan untuk menghindari kemungkinan bentrok. Berikut dibahas mengenai *constraint* yang telah dibentuk menggunakan CSP,

- Dosen dan jam mata kuliah akan dipisahkan berdasarkan *constraint* yang telah ditentukan. Misalnya, satu dosen yang mengajar dua mata kuliah, tidak dapat mengajar dua kelas dan dua mata kuliah dalam waktu yang sama.
- Memisahkan dosen berdasarkan tugas mata kuliah yang diajarkan dan kelas yang diajar.
- Sesi praktikum hanya dapat dilakukan setelah sesi teori selesai dilakukan.
- Membuat batas jumlah ruangan yang dapat digunakan, agar saat pengalokasian ruangan menggunakan Dijkstra tidak mengalami kekurangan kelas.
- Membuat batas jam sesi dalam sehari, sejumlah delapan jam.

### **Pemodelan Variabel dan *Constraint***

Pemodelan variabel pada penjadwalan kuliah merupakan penampung yang terdiri domain (*value*) yang sesuai dengan variabel itu sendiri. Variabel ini akan diinput oleh *user* untuk menentukan jumlah domain setiap variabel. Domain akan disimpan dalam bentuk *array*. Berikut merupakan pemodelan variabel dan domain sederhana yang mengacu pada gambar 6. *tree backtracking*,

#### 1. Dosen

$D = \{SML, RZS, MM, MPR, ASD, ANA, SFA, RFK, MSL, YBN, EAN, YUL, IFY, MNP, DDA, ANM, ARR, DDA, AAD, PN\}$

$D[0] = SML$

$D[6] = SFA$

$D[1] = RZS$

$D[7] = RFK$

$D[2] = MM$

$D[8] = MSL$

$D[3] = MPR$

$D[9] = YBN$

$D[4] = ASD$

$D[10] = EAN$

$D[5] = ANA$

$D[11] = YUL$

$$D[12] = IFY$$

$$D[13] = MNP$$

$$D[14] = DDA$$

$$D[15] = ANM$$

$$D[16] = ARR$$

$$D[17] = DDA$$

$$D[18] = P$$

## 2. Mahasiswa

$$M = \{11TB, 12TB, 13TB\}$$

$$M[0] = 11TB$$

$$M[1] = 12TB$$

$$M[2] = 13TB$$

$$M[3] = 14TB$$

## 3. Mata kuliah

MK={MADAS I, FISDAS I, STA, ENG 1, PTI, DEL CHA, KIMDAS, PRAK KIMOR, ALJALI, ALNUM, PDO, MIKRUM, PRAK MIKRUM, KIMFIS, PRAK KIMFIS, BIODIKIM, PROBSTAT, AKRB, FENOPOR, APKP II, OFP II, PP, KST, PIP, PRAK PIP, SPP, UPL, TRB, PEN I, AITB, PC, MATLAB}

$$MK[0]=MADAS I$$

$$MK[1]= FISDAS I$$

$$MK[2]= STA$$

$$MK[3]= ENG I$$

$$MK[4]= PTI$$

$$MK[5]= DEL CHA$$

$$MK[6]= KIMDAS$$

$$MK[7]= PRAK KIMOR$$

$$MK[8]= ALJALI$$

$$MK[9]= ALNUM$$

$$MK[10]= PDO$$

$$MK[11]= MIKRUM$$

$$MK[12]= PRAK MIKRUM$$

$$MK[13]= KIMFIS$$

$$MK[14]= PRAK KIMFIS$$

$$MK[15]= BIODIKIM$$

$$MK[16]= PROBSTAT$$

$$MK[17]= AKRB$$

$$MK[18]= FENOPOR$$

$$MK[19]= APKP II$$

$$MK[20]= OFP II$$

$$MK[21]= PP$$

$$MK[22]= KST$$

$$MK[23]= PIP$$

$$MK[24]= PRAK PIP$$

$$MK[25]= SPP$$

$$MK[26]= UPL$$

$$MK[27]= TRB$$

$$MK[28]= PEN I,$$

$$MK[29]= AITB$$

MK[30]= PC

MK[31]= MATLAB

### 1. Hari

H={senin, selasa, rabu, Kamis, jumat}

H[0]=jumat

H[3]=selasa

H[1]=Kamis

H[4]=senin

H[2]=rabu

### 2. Sesi

S= { 1, 2, 3, 4, 5, 6, 7, 8 }

S[0]=8

S[1]=7

S[2]=6

S[3]=5

S[4]=4

S[5]=3

S[6]=2

S[7]=1

### 3. Ruangan

R = { 711, 712, 713. 511, 512, 513, 514 }

R[0]=711

R[1]=712

R[2]= 713

Variabel Tambahan:

KR (Kapasitas Ruangan) : 30, 60, 200

C (Kredit) : SKS

MT (Mata kuliah teori)

MK (Mata kuliah praktikum)

*Constraint* pada penjadwalan terbentuk dari keterkaitan antar domain yang terdiri dari batasan-batasan yang terletak pada domain. Relasi domain akan bernilai satu jika memiliki keterkaitan, dan akan bernilai nol jika tidak memiliki keterkaitan.

#### 1. Dosen-Matakuliah (DM)

Pada bagian ini relasi Dosen dengan Mata kuliah dijelaskan menggunakan tabel matriks yang berisi array masing-masing dosen dan mata kuliah yang diajarkan. Berikut merupakan contoh relasi domain yang bernilai satu dan bernilai nol:

- Relasi domain yang bernilai satu :  
dengan inisial SML pada array ke 0 dimodelkan sebagai  $D[0] = \text{SML}$ , dan Mata Kuliah MADAS I pada array ke 0 dimodelkan  $MK[0] = \text{MADAS I}$ . Sehingga hasil pemodelan yang diperoleh antara dosen dengan mata kuliah yaitu  $DM[0][0] = 1 = DM[\text{SML}][\text{MADAS I}]$ , artinya relasi domain bernilai satu karena memiliki keterkaitan yaitu dosen SML mengajar mata kuliah MADAS I.
- Relasi domain yang bernilai nol:  
Dosen dengan inisial SML pada array ke 0 dimodelkan sebagai  $D[0] = \text{SML}$ , dan mata kuliah dengan nama FISDAS I dimodelkan sebagai  $MK[1] = \text{FISDAS I}$  pada array ke 1 dimodelkan sebagai  $MK[1] = \text{FISDAS I}$ . Sehingga hasil pemodelan yang diperoleh antara dosen dengan mata kuliah yaitu  $DM[0][1] = 0 = DM[\text{SML}][\text{FISDAS I}]$ , artinya relasi domain bernilai nol karena tidak memiliki keterkaitan yaitu dosen SML tidak mengajar mata kuliah FISDAS I.

Berikut merupakan relasi secara keseluruhan antara dosen yang mengajar mata kuliah pada fakultas Teknik Bioproses:

Tabel 3.7 Tabel matriks dari dosen dengan mata kuliah

Mata Kuliah	Dosen																			
	SM L [0]	RZ S [1]	MM M [2]	MP R [3]	AS D [4]	AN A [5]	SF A [6]	RF K [7]	MS L [8]	YB N [9]	EA N [10]	YU L [11]	IF Y [12]	MN P [13]	DD A [14]	MN P [15]	AN M [16]	AR R [17]	AA D [18]	PN [19]
MADAS I [0]	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FISDAS I [1]	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
STA [2]	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ENG I [3]	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PTI [4]	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DEL CHA [5]	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KIMDAS [6]	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
PRAK KIMOR [7]	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
ALJALI [8]	0	0	0	0	0	0	0	0	1		0	0	0	0	0	0	0	0	0	0
ALNUM [9]	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
PDO [10]	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
MIKRUM [11]	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0



<b>PRAK MIKRUM [12]</b>	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
<b>KIMFIS [13]</b>	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
<b>PRAK KIMFIS [14]</b>	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
<b>BIOKIM [15]</b>	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
<b>PROBSTA T [16]</b>	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
<b>AKRB [17]</b>	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
<b>FENOPOR [18]</b>	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
<b>APKP II [19]</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
<b>OFP II [20]</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
<b>PP [21]</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
<b>KST [22]</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
<b>PIP [23]</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
<b>PRAK PIP [24]</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
<b>SPP [25]</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
<b>UPL [26]</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
<b>TRB [27]</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

<b>PEN I</b> <b>[28]</b>	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
<b>AITB</b> <b>[29]</b>	0	0	0	0	0	0	0	0	0		1	0	0	0	0	0	0	0	0	0
<b>PC</b> <b>[30]</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
<b>MATLAB</b> <b>[31]</b>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Berdasarkan tabel matriks antara dosen dengan matakuliah diatas, maka Berikut merupakan memiliki keterkaitan atau bernilai satu:

$DM[0][0] = 1 = DM[SML] [MADAS I]$   
 $DM[1][1] = 1 = DM[RZS] [FISDAS I]$   
 $DM[2][2] = 1 = DM[MMM] [STA]$   
 $DM[3][3] = 1 = DM[MPR] [ENG I]$   
 $DM[4][4] = 1 = DM[ASD] [PTI]$   
 $DM[5][5] = 1 = DM[ANA] [DELCHA]$   
 $DM[6][6] = 1 = DM[SFA] [KIMDAS]$   
 $DM[7][7] = 1 = DM[RFK] [PRAK KIMOR]$   
 $DM[8][8] = 1 = DM[MSL] [ALJALI]$   
 $DM[9][9] = 1 = DM[YBN] [ALNUM]$   
 $DM[9][10] = 1 = DM[YBN] [PDO]$   
 $DM[10][11] = 1 = DM[EAN] [MIKRUM]$   
 $DM[10][12] = 1 = DM[EAN] [PRAK MIKRUM]$   
 $DM[11][13] = 1 = DM[YUL] [KIMFIS]$   
 $DM[11][14] = 1 = DM[YUL] [PRAK KIMFIS]$   
 $DM[7][15] = 1 = DM[RFK] [BIOKIM]$   
 $DM[12][16] = 1 = DM[IFY] [PROBSTAT]$   
 $DM[11][17] = 1 = DM[YUL] [AKRB]$   
 $DM[12][18] = 1 = DM[MNP] [FENOPOR]$   
 $DM[12][19] = 1 = DM[MNP] [APKP II]$   
 $DM[13][20] = 1 = DM[DDA] [OFP II]$   
 $DM[14][21] = 1 = DM[MNP] [PP]$   
 $DM[15][22] = 1 = DM[ANM] [KST]$   
 $DM[16][23] = 1 = DM[ARR] [PIP]$   
 $DM[16][24] = 1 = DM[ARR] [PRAK PIP]$   
 $DM[13][24] = 1 = DM[DDA] [SPP]$   
 $DM[17][25] = 1 = DM[AAD] [UPL]$   
 $DM[16][26] = 1 = DM[ARR] [TRB]$   
 $DM[7][27] = 1 = DM[RFK] [PEN I]$   
 $DM[10][28] = 1 = DM[EAN] [AITB]$   
 $DM[18][29] = 1 = DM[PN] [PC]$   
 $DM[16][30] = 1 = DM[ARR] [MATLAB]$

## 2. Mata kuliah–Mahasiswa (MK-M)

Pada bagian ini relasi antara mata kuliah dan kelas dijelaskan menggunakan tabel matriks yang berisi array masing-masing mata kuliah yang diampuh oleh setiap kelas. Berikut merupakan contoh relasi mata kuliah dengan kelas dengan domain bernilai satu dan bernilai nol:

- Relasi domain yang bernilai satu :

Mata kuliah dengan inisial MADAS I pada array ke 0 dimodelkan sebagai  $MK[0] = MADAS\ I$ , dan mahasiswa 11TB dengan array ke 0 dimodelkan sebagai  $M[0] = 11TB$  Sehingga hasil pemodelan yang diperoleh antara dosen dengan matakuliah yaitu  $MK - M[0][0] = 1 = MK - M[MADAS\ I][11TB]$ , artinya relasi domain bernilai satu karena memiliki keterkaitan yaitu mata kuliah MADAS I yang diampuh oleh mahasiswa 11TB.

- Relasi domain yang bernilai nol:

Mata kuliah dengan inisial MADAS I pada array ke 0 dimodelkan sebagai  $MK[0] = MADAS\ I$ , dan mahasiswa 12TB dengan array ke 1 dimodelkan sebagai  $M[1] = 12TB$  Sehingga hasil pemodelan yang diperoleh antara dosen dengan mata kuliah yaitu  $MK - M[0][1] = 0 = MK - M[MADAS\ I][12TB]$ , artinya relasi domain bernilai nol karena memiliki keterkaitan yaitu mata kuliah MADAS I tidak diampuh oleh mahasiswa 11TB.

Berikut merupakan relasi secara keseluruhan antara kelas dengan mata kuliah yang diampuh pada fakultas Teknik Bioproses:

Tabel 3.8 Relasi Mata Kuliah dengan Kelas

Mata kuliah	Kelas			
	11 TB [0]	12 TB [1]	13 TB [2]	14 TB [3]
MADAS I [0]	0	1	2	3
FISDAS I [1]	1	0	0	0
STA [2]	1	0	0	0
ENG I [3]	1	0	0	0
PTI [4]	1	0	0	0
DEL CHA [5]	1	0	0	0
KIMDAS [6]	1	0	0	0
PRAK KIMOR [7]	1	0	0	0

ALJALI	[8]	0	1	0	0
ALNUM	[9]	0	1	0	0
PDO	[10]	0	1	0	0
MIKRUM	[11]	0	1	0	0
PRAK MIKRUM	[12]	0	1	0	0
KIMFIS	[13]	0	1	0	0
PRAK KIMFIS	[14]	0	1	0	0
BIOKIM	[15]	0	1	0	0
PROBSTAT	[16]	0	1	1	0
AKRB	[17]	0	0	1	0
FENOPOR	[18]	0	0	1	0
APKP II	[19]	0	0	1	0
OFP II	[20]	0	0	1	0
PP	[21]	0	0	1	0
KST	[22]	0	0	1	0
PIP	[23]	0	0	1	0
PRAK PIP	[24]	0	0	0	1
SPP	[25]	0	0	0	1
UPL	[26]	0	0	0	1
TRB	[27]	0	0	0	1
PEN I	[28]	0	0	0	1
AITB	[29]	0	0	0	1
PC	[30]	0	0	0	1
MATLAB	[31]	0	0	0	1

Berikut merupakan pemodelan *constraint* dengan memperhatikan keterkaitan antar domain, variabel dituliskan dengan notasi array. Variabel  $X$  yang terbentuk akan memiliki nilai antara 0 dan 1. Variabel yang bernilai 0 adalah variabel yang tidak berelasi, sedangkan variabel yang bernilai 1 adalah variabel yang berelasi. Variabel yang bernilai 1, merupakan solusi pada penjadwalan yang akan dibentuk variabel  $X$ . Variabel tersebut dibentuk dalam array, yaitu  $X[D][Mm][K][S][H]$ . Domain yang berelasi merupakan batasan yang digunakan pada variabel. Sehingga, variabel  $X$  bernilai lebih kecil atau sama dengan variabel yang berelasi. Hal ini digunakan untuk menjamin konsistennya variabel  $X$  dengan *constraint-constraint* yang ada.

1. satu dosen yang mengajar dua mata kuliah, tidak dapat mengajar dua kelas dan dua mata kuliah dalam waktu yang sama.

$$\sum_{mkr} X[D_d][M][K][R][S_s][H_h] \leq 1$$

2. Memisahkan dosen berdasarkan tugas mata kuliah yang diajarkan dan kelas yang diajar

$$\sum_{sh} X[D_d][M_m][K_k][R][S][H] \leq 1$$

3. Setiap mata kuliah yang berbeda tidak boleh berada pada kelas dan waktu yang sama.

$$\sum_{dr} X[D][M_m][K_k][R][S_s][H_h] \leq 1$$

4. Kapasitas ruangan tidak diperbolehkan lebih kecil dibandingkan jumlah mahasiswa yang mengambil mata kuliah tersebut.

$$\sum_{dk} X[D][M_m][K][R_r][S_s][H_h] \leq KR[r]$$

5. Setiap mata kuliah harus dijadwalkan sesuai dengan banyaknya jumlah SKS

$$\sum_{dkrsh} X[D][M_m][K][R][S][H] = C[m]$$

6. Sesi praktikum dapat dilakukan apabila sudah terlebih dahulu melakukan sesi teori.

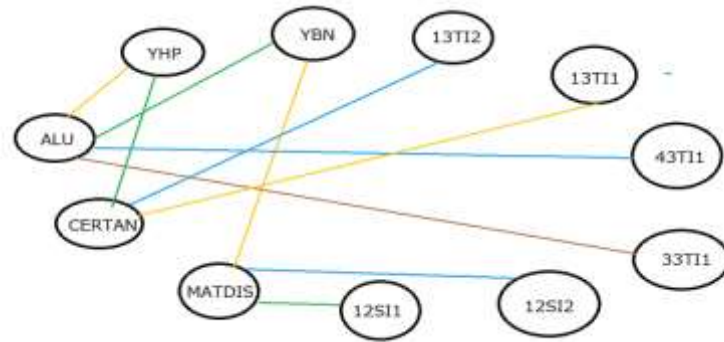
$$\sum_{dk} X[D][M_{tmt}][K][R][S_s][H_h] > M[mpsh]$$

7. Jumlah kelas yang masuk mata kuliah tidak boleh melebihi jumlah ruangan yang tersedia

$$\sum_{dm} X[D][M][K_k][R][S_s][H_h] \leq JR[R]$$

## Pewarnaan Graph

*Constraint* yang ada dapat dijelaskan menggunakan pewarnaan graph. Pewarnaan graph memiliki syarat yang harus diperhatikan. Setiap pewarnaan yang dimiliki harus merujuk pada *constraint* yang dimiliki. Untuk pewarnaan *constraint*, setiap node yang dihubungkan dengan node lain untuk tujuan yang sama harus memiliki warna yang berbeda. Graph tersebut menjelaskan antara keterkaitan antar sumber daya dan pembentukan *constraint* berdasarkan tugas yang dimiliki. Berikut merupakan pewarnaan graph untuk *constraint* yang telah dibentuk berdasarkan contoh pemodelan tree pada gambar 4 dan gambar 5,



Gambar 3.1 Pewarnaan Graph Constraint

### 3.3. Analisis Algoritma Backtracking

Algoritma *Backtracking* digunakan untuk menghasilkan jadwal yang tidak mengalami bentrok antara sumber daya. Algoritma *Backtracking* dibuat dalam bentuk pohon (*tree*) solusi. Untuk mendapatkan solusi terbaik, pohon solusi tersebut akan ditelusuri secara DFS (*Depth First Search*). Penggunaan Algoritma *Backtracking* dilakukan dengan tahapan logika sebagai berikut:

1. Penelusuran keseluruhan *node* dimulai dari akar hingga ke daun. Simpul yang lahir disebut simpul hidup dan dapat diperluas dengan diberi nama simpul-E. Penelusuran ini dilakukan berdasarkan *tree* yang telah tercipta dari solusi-solusi yang terkumpul dari metode CSP yang telah ada sebelumnya.
2. Apabila lintasan yang didapatkan dari perluasan simpul-E tidak mengarah ke solusi, maka simpul tersebut akan menjadi simpul mati dan tidak akan dilakukan perluasan lagi.
3. Apabila posisi saat ini berada pada *node* yang merupakan simpul mati, maka akan dilakukan pencarian selanjutnya dengan membangkitkan simpul anak yang lain. Namun, jika tidak ada simpul *child* maka akan dilakukan *Backtracking* ke simpul induk.
4. Proses ini akan berhenti apabila telah menemukan solusi terbaik atau tidak ada lagi simpul hidup yang diperlukan.

Untuk penjadwalan mata kuliah, akan dilakukan terlebih dahulu pencari solusi. Pencarian solusi dapat disimulasikan sebagai berikut:

- ACB, PRPL, 31TI1, 41TI

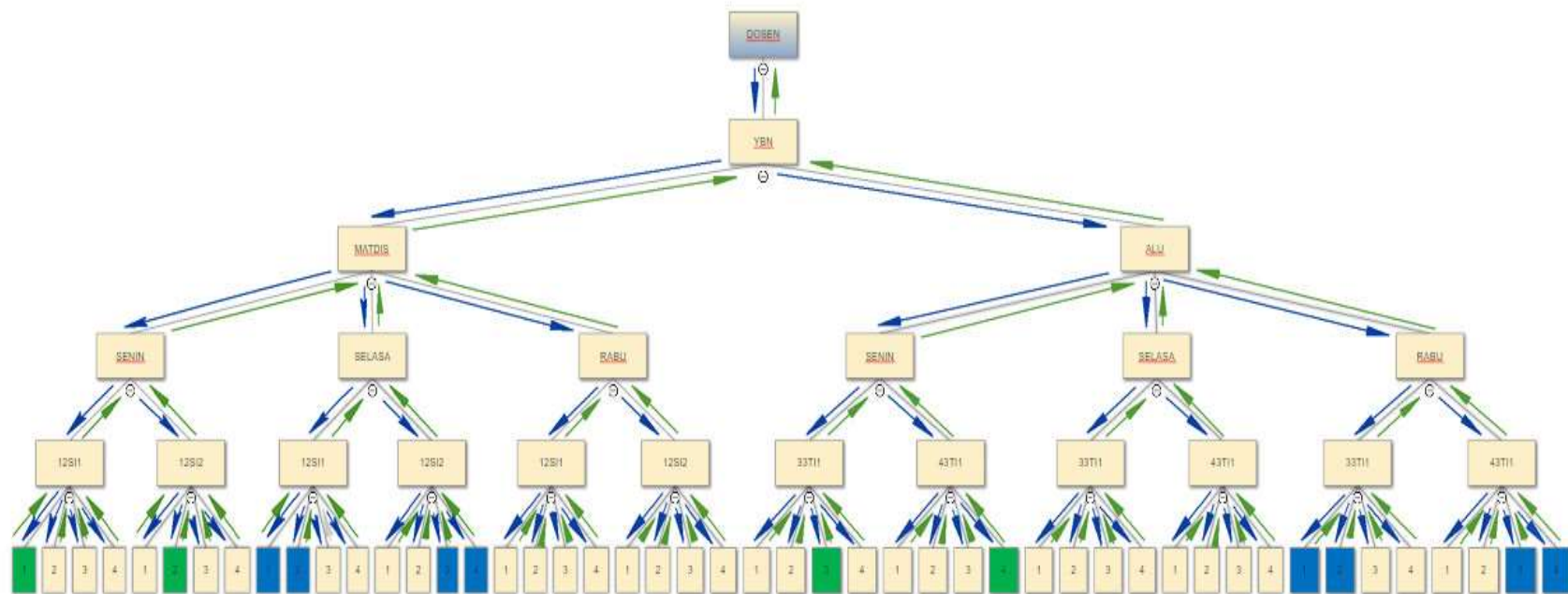
- ACB, PUPL, 32TI, 42TI
- RDT, PAM, 32TI, 42TI
- RDT, PBO, 32TI, 42TI
- TNT, PSW, 31TI, 41TI
- TNT, KEPAL, 33TI, 43TI

Dalam menghasilkan jadwal yang sesuai, maka Algoritma *Backtracking* harus memperhatikan *constraint* yang telah dibentuk menggunakan CSP. untuk simulasi yang dilakukan, perlu diperhatikan *constraint-constraint* berikut diantaranya,

- Hari akademik, menggunakan tiga hari yaitu hari senin, selasa dan rabu
- Jumlah jam sehari yaitu sejumlah empat jam.
- Ruangan maksimal yang dimiliki dalam sehari sejumlah sepuluh unit.
- Jam praktikum dapat dilakukan setelah jam teori telah selesai dilaksanakan.



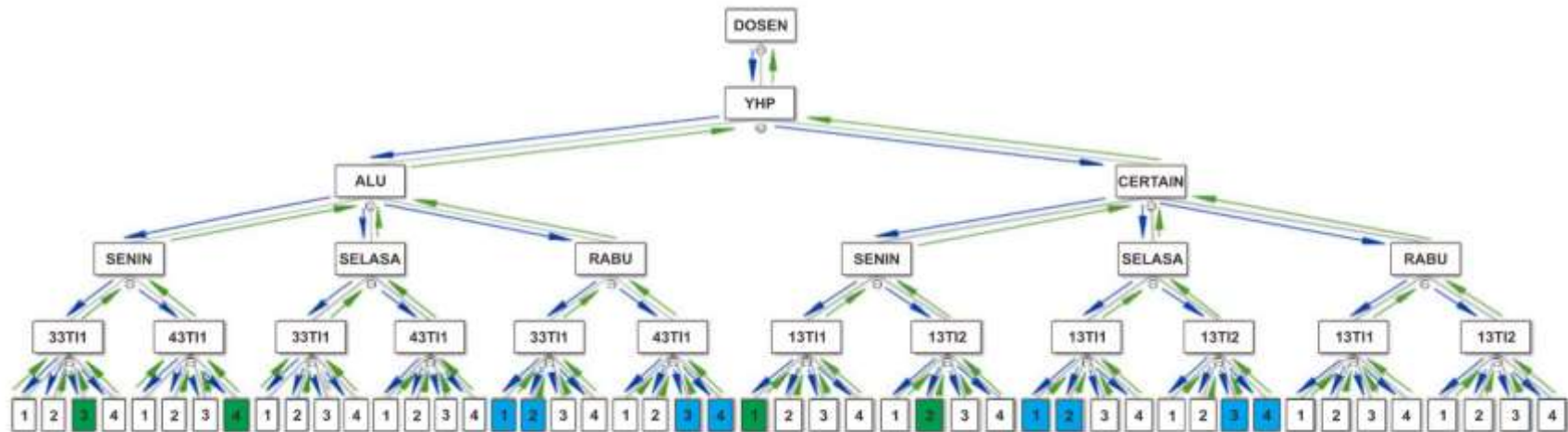
Berikut merupakan *tree* yang menggambarkan algoritma *Backtracking* dalam pengalokasian hari dan waktu:



**Gambar 3.2 Tree Algoritma Backtracking Pertama**

*Tree* diatas terdiri atas 6 level, diantaranya *root* merupakan dosen yang memiliki dua percabangan yaitu inisial dosen, *node* inisial dosen memiliki percabangan yaitu, mata kuliah yang diajarkan, percabangan dari *node* mata kuliah yaitu hari akademik. Percabangan selanjutnya yaitu kelas yang mengambil mata kuliah, dan percabangan terakhir yaitu jam akademik, seperti yang terlihat pada gambar mengenai pengalokasian waktu terhadap dosen, mata kuliah dan kelas yang dimasuki. dan jam akademik perharinya sejumlah empat jam. Dapat dijelaskan bahwa inisial dosen yaitu YBN mengajar dua mata kuliah yaitu MATDIS dan ALU. Proses *Backtracking* ini akan

berjalan dengan mengikuti *constraint-constraint* yang ada. Melalui *tree* ini akan ditelusuri solusi yang memungkinkan, seluruh *node* akan ditelusuri untuk memastikan keseluruhan *node* yang ada. Ketika pencarian tiba di *child* maka akan dilakukan proses *backtrack*.



Gambar 3.3 Tree Algoritma Backtracking Kedua

Keterangan:

- Sesi teori
- Sesi praktikum

Pada gambar diatas dijelaskan bahwa dosen dengan inisial YHP mengajar mata kuliah ALU dan CERTAN. Dosen YBN dan YHP berkolaborasi pada mata kuliah ALU, dalam proses perkuliahan. Sehingga, dalam hal ini harus dipastikan bahwa dosen YHP harus memiliki jadwal masuk yang sama dengan YBN dalam mata kuliah ALU. Sehingga dalam hal ini, *tree* akan ditelusuri dengan status jam mata kuliah ALU pada dosen YHP sudah terisi sama seperti dosen inisial YBN. Untuk penelusuran mata kuliah CERTAN tidak dapat mengisi jam yang telah diatur sebagai jadwal untuk ALU.

Berikut merupakan program sederhana dalam penggunaan *Backtracking* (Lumbantoruan, et al., 2012)

```
function BACKTRACKING-SEARCH(csp) returns a solution, or failure
    return BACKTRACK({},csp)
```

```
function BACKTRACK(assignment, csp) returns a solution, or failure
    if assignment is complete then return assignment
    var ← SELECT-UNASSIGNED-VARIABLE(csp)
    for each value in ORDER-DOMAIN-VALUES(var,assignment,csp) do
        if value is consistent with assignment then
            add {var = value } to assignment
            inferences ← INFERENCE(csp,var,value)
            if inferences ≠ failure then
                add inferences to assignment
                result ← BACKTRACK(assignment,csp)
                if result ≠ failure then
                    return result
            remove {var = value } and inferences from assignment
    return failure
```

### 3.4. Analisis Perhitungan Jarak

Pada bab ini dijelaskan analisis dan pengamatan terhadap penjadwalan mata kuliah dengan perhitungan jarak dengan menggunakan algoritma Dijkstra dengan perhitungan jarak tanpa Algoritma Dijkstra.

#### 3.4.1. Perhitungan Jarak Tanpa Algoritma Dijkstra

Pada bab ini dijelaskan penjadwalan kuliah di Institut Teknologi Del berupa tabel excel yang dibuat oleh BAAK, seperti gambar berikut :

JADWAL KULIAH SEMESTER GASAL T.A 2018/2019											
Senin											
Sesi	WAKTU	Kode MK	MK	D1	D2	D3	I1	I2	Grup	Ruang	
36	1	08.00 - 08.50									
37	1	08.00 - 08.50	ING V	T	SAM	*	*	*	33T11	GD923	
38	1	08.00 - 08.50	ING V	T	*	*	*	*	33T12	GD923	
39	1	08.00 - 08.50	PIC	T	MMS	*	*	*	33TK	GD924	
40	1	08.00 - 08.50				*	*	*	43T1		
41	1	08.00 - 08.50	TBFA	T	YYS	*	*	*	13T11	GD923	
42	1	08.00 - 08.50	TBFA	T	YYS	*	*	*	13T12	GD923	
43	1	08.00 - 08.50	PABWE	T	MSS	*	*	*	13S11	GD516	
44	1	08.00 - 08.50	PABWE	T	MSS	*	*	*	13S12	GD516	
45	1	08.00 - 08.50	MTE	T	SFA	*	*	*	13TE1	GD942	
46	1	08.00 - 08.50	MTE	T	SFA	*	*	*	13TE2	GD942	
47	1	08.00 - 08.50	OR II	T	HSS	*	*	*	13MR1	GD927	
48	1	08.00 - 08.50	OR II	T	HSS	*	*	*	13MR2	GD927	
49	1	08.00 - 08.50			*	*	*	*	13TB		
50	1	08.00 - 08.50									

Gambar 3.4 Perkuliahan Sesi Pertama

140	3	10.00 - 10.50			*	*	*	*	33T11		
141	3	10.00 - 10.50			*	*	*	*	33T12		
142	3	10.00 - 10.50			*	*	*	*	33TK		
143	3	10.00 - 10.50	IND	T	TLG	*	*	*	43T1	GD944	
144	3	10.00 - 10.50	CERTAN	T	ART	YHP	*	*	13T11	GD516	
145	3	10.00 - 10.50	CERTAN	T	ART	YHP	*	*	13T12	GD516	
146	3	10.00 - 10.50	PJK	T	ESS	*	*	*	13S11	GD934	
147	3	10.00 - 10.50	PJK	T	ESS	*	*	*	13S12	GD934	
148	3	10.00 - 10.50			*	*	*	*	13TE1		
149	3	10.00 - 10.50			*	*	*	*	13TE2		
150	3	10.00 - 10.50			*	*	*	*	13MR1		
151	3	10.00 - 10.50			*	*	*	*	13MR2		
152	3	10.00 - 10.50	PROBSTAT	T	IFY	JUN	*	*	13TB	GD924	
153	3	10.00 - 10.50									
154	3	10.00 - 10.50									
155	3	10.00 - 10.50									
156	3	10.00 - 10.50									
157	3	10.00 - 10.50									
158	3	10.00 - 10.50									
159	3	10.00 - 10.50									
160	3	10.00 - 10.50									

Gambar 3.5 Perkuliahan Sesi ketiga

Gambar diatas merupakan salah satu potongan excel dari penjadwalan pada hari senin. Jadwal tersebut berisi mata kuliah yang diampuh dengan komposisi yaitu teori, pengajar, dan gedung tempat terjadinya kegiatan kuliah. Selain berisi gedung tabel tersebut juga terdiri dari kelas-kelas yang akan memasuki setiap ruangan yang ada. Setiap gedung terdiri dari beberapa ruangan dan dapat digunakan oleh beberapa kelas, tergantung jumlah ruangan kosong yang dapat digunakan oleh

setiap kelas tersebut. Sebagai contoh kelas 13SI1 dan kelas 13SI2 mengampuh mata kuliah PABWE yaitu teori pada sesi pertama pukul 08:00– 08: 50 di gedung Lima (GD 5) tepatnya ruangan 516. Kemudian perkuliahan tersebut dilanjutkan kembali oleh kelas yang sama yaitu kelas 13SI1 dan kelas 13SI2 mengampuh mata kuliah PJK yaitu teori pada sesi ketiga pukul 10:00–10:50 di gedung sembilan (GD 9) tepatnya ruangan 934. Sehingga dari penjadwalan diatas tidak memperhatikan jarak lokasi antar kelas untuk meminimalisasi penggunaan waktu yang berlebih akibat jarak perpindahan antar kelas pada jadwal dosen dan mahasiswa telah ditentukan. Adapun jarak antara gedung yang ditempuh dari gedung lima menuju gedung sembilan dapat ditempuh dengan selang waktu 10 sampai dengan 15 menit.

### 3.4.2. Perhitungan Jarak dengan Algoritma Dijkstra

Algoritma Dijkstra digunakan untuk menyelesaikan jalur terpendek. Pada kasus penjadwalan, dibutuhkan Algoritma Dijkstra untuk penentuan ruangan kelas yang paling tepat, berdasarkan jarak antar kelas. Algoritma Dijkstra melakukan pencarian jalur terpendek dengan melalui satu titik ke titik lain dan bertahap ke titik lain. Lokasi mengajar (ruangan) akan dibuat menggunakan Algoritma Dijkstra agar dapat meminimalisasi penggunaan waktu saat menempuh ruangan dikarenakan jarak yang jauh. *Rute* perjalanan ke ruangan kelas diibaratkan sebagai *Graph*, titik mempresentasikan ruangan yang akan dijadikan tempat kegiatan belajar, dan bobot mempresentasikan jarak antara ruangan yang akan digunakan.

Berikut merupakan urutan logika algoritma Dijkstra dalam mencari jarak terpendek ruangan pada penjadwalan:

1. Menentukan titik awal, yaitu kelas pertama yang dimasuki mahasiswa dan beri nilai 0, dan *Set node* ini sebagai *node* keberangkatan.
2. Beri bobot untuk jarak antara *node* yang satu dengan lainnya, yaitu jarak antar gedung.
3. *Set* semua *node* yang belum dilalui, yaitu setiap gedung yang digunakan sebagai tempat perkuliahan.
4. Hitung jarak setiap percabangan dari *node* keberangkatan menuju *node* selanjutnya dan pilih jarak yang paling dekat.

5. Tandai *node* yang telah dilalui, sehingga *node* yang telah dilalui tidak dapat dilalui lagi.
6. Setelah jarak *node* terkecil telah ditemukan, dilanjutkan ke *node* selanjutnya, hingga mencapai *node* yang ingin dituju, dengan memilih setiap jalur dengan nilai yang paling minimum.

Berikut adalah tabel matriks yang terdiri dari ruangan dari setiap gedung yang dapat digunakan oleh setiap kelas. Gedung terdiri dari empat gedung yaitu GD 5 yang terdiri dari lima ruangan (516, 515, 514, 513, 525), GD 7 terdiri dari sepuluh ruangan (711, 712, 713, 714, 721, 722, 723, 724, 725, 726), GD 9 terdiri dari 20 ruangan (912, 913, 914, 915, 916, 917, 923, 924, 9124, 927, 926, 928, 929, 935, 937, 938, 942, 943, 944). Jarak antar gedung yang digunakan dalam matriks penjadwalan pada tabel masih menggunakan data *dummy*. Adapun jarak antar ruangan dalam gedung yang sama yaitu dianggap satu.

Tabel 3.9 Matriks antar Ruangan

Gedung	516	515	514	513	525	711	712	713	714	721	722	723	724	725	726	912	913	914	915	916	917	923	924	925	927	928	929	933	934	935	937	938	942	943	944
	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
516	0	1	1	1	1	2	2	2	2	2	2	2	2	2	2	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
515	1	0	1	1	1	2	2	2	2	2	2	2	2	2	2	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
514	1	1	0	1	1	2	2	2	2	2	2	2	2	2	2	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
513	1	1	1	0	1	2	2	2	2	2	2	2	2	2	2	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
525	1	1	1	1	0	2	2	2	2	2	2	2	2	2	2	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
711	2	2	2	2	2	0	1	1	1	1	1	1	1	1	1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
712	2	2	2	2	2	1	0	1	1	1	1	1	1	1	1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
713	2	2	2	2	2	1	1	0	1	1	1	1	1	1	1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
714	2	2	2	2	2	1	1	1	0	1	1	1	1	1	1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
721	2	2	2	2	2	1	1	1	1	0	1	1	1	1	1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
722	2	2	2	2	2	1	1	1	1	1	0	1	1	1	1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
723	2	2	2	2	2	1	1	1	1	1	1	0	1	1	1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
724	2	2	2	2	2	1	1	1	1	1	1	1	0	1	1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
725	2	2	2	2	2	1	1	1	1	1	1	1	1	0	1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
726	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
912	4	4	4	4	4	4	5	5	5	5	5	5	5	5	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
913	4	4	4	4	4	4	5	5	5	5	5	5	5	5	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
914	4	4	4	4	4	4	5	5	5	5	5	5	5	5	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
915	4	4	4	4	4	4	5	5	5	5	5	5	5	5	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
916	4	4	4	4	4	4	5	5	5	5	5	5	5	5	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
917	4	4	4	4	4	4	5	5	5	5	5	5	5	5	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
923	4	4	4	4	4	4	5	5	5	5	5	5	5	5	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
924	4	4	4	4	4	4	5	5	5	5	5	5	5	5	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
925	4	4	4	4	4	4	5	5	5	5	5	5	5	5	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
927	4	4	4	4	4	4	5	5	5	5	5	5	5	5	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
928	4	4	4	4	4	4	5	5	5	5	5	5	5	5	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
929	4	4	4	4	4	4	5	5	5	5	5	5	5	5	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
933	4	4	4	4	4	4	5	5	5	5	5	5	5	5	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
934	4	4	4	4	4	4	5	5	5	5	5	5	5	5	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
935	4	4	4	4	4	4	5	5	5	5	5	5	5	5	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
937	4	4	4	4	4	4	5	5	5	5	5	5	5	5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
938	4	4	4	4	4	4	5	5	5	5	5	5	5	5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
942	4	4	4	4	4	4	5	5	5	5	5	5	5	5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
943	4	4	4	4	4	4	5	5	5	5	5	5	5	5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
944	4	4	4	4	4	4	5	5	5	5	5	5	5	5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

### 3.4.3. Kesimpulan

Kesimpulan yang diperoleh dari penjadwalan dengan perhitungan tanpa Algoritma Dijkstra kurang efektif dan efisien karena tidak memperhatikan adanya jarak antar gedung pada saat pergantian jadwal kuliah. Hal tersebut mengakibatkan penggunaan waktu yang berlebih akibat jarak perpindahan antar kelas pada jadwal dosen dan mahasiswa telah ditentukan.

Sedangkan penjadwalan dengan perhitungan jarak menggunakan Algoritma Dijkstra merupakan algoritma yang digunakan untuk penyelesaian masalah jalur terpendek. Dari hasil perhitungan Algoritma Dijkstra akan tercipta penjadwalan yang akan mempermudah setiap kelas untuk pencarian ruangan yang kosong dengan menggunakan jarak terpendek sehingga proses perpindahan antar kelas juga dapat ditempuh dengan waktu yang efektif dan efisien.

### 3.4.4. Graph Penjadwalan

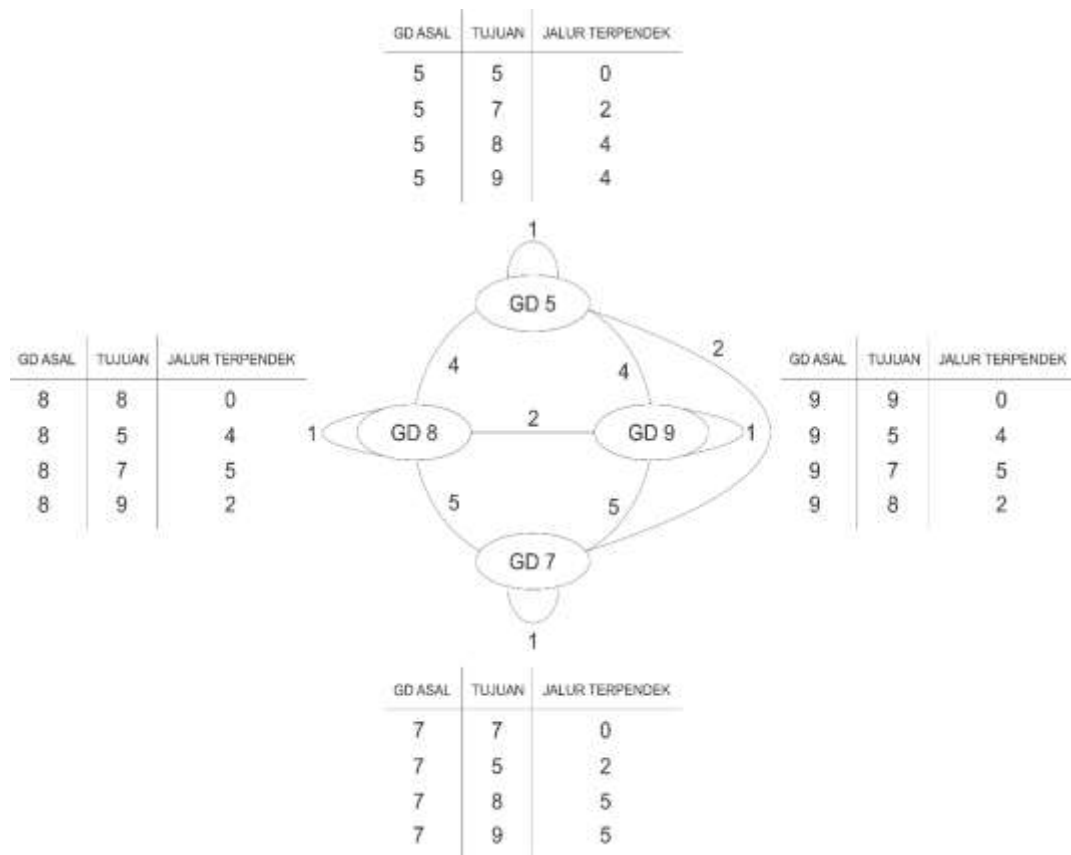
Pembangunan sistem penjadwalan dengan jarak terpendek di Institut Teknologi Del menggunakan *Graph*. *Graph* digunakan untuk menggambarkan *route* yang akan dilalui pada saat pergantian ruangan kelas yaitu dengan jarak terpendek. Adapun hasil *Graph* penjadwalan dengan jarak terpendek pada pengerjaan tugas akhir ini menggunakan Algoritma Dijkstra. Jarak antar gedung yang digunakan dalam *Graph* penjadwalan diatas masih menggunakan data *dummy*.

Gambar diatas merupakan *Graph* dari penjadwalan yang akan dilakukan pada hari senin sampai dengan rabu yaitu dimulai dari jam pertama hingga jam kedelapan. *Graph* tersebut terdiri dari lima simpul (beberapa GD atau gedung) yaitu GD 5, GD 7, GD 8, GD 9. Selain berisi gedung dan jarak antar gedung, *Graph* tersebut juga terdiri dari kelas-kelas yang akan memasuki setiap ruangan yang ada. Setiap gedung terdiri dari beberapa ruangan dan setiap ruangan tersebut dapat digunakan oleh beberapa kelas, tergantung jumlah ruangan kosong yang dapat digunakan oleh setiap kelas tersebut. Oleh karena itu dengan adanya *route* dari hasil *generate* algoritma Dijkstra tersebut, maka akan tercipta penjadwalan yang akan mempermudah setiap kelas untuk menghindari jadwal bentrok serta mempermudah pencarian ruangan yang kosong dengan menggunakan jarak terpendek sehingga



proses perpindahan antar kelas juga dapat ditempuh dengan waktu yang efektif dan efisien.

Berikut merupakan penerapan Algoritma Dijkstra yang menggambarkan hubungan dan jarak yang terdapat antar gedung:



**Gambar 3.6 Hubungan antar Gedung dengan Algoritma Dijkstra**

Bagian ini menjelaskan tabel hasil perhitungan untuk rute terpendek dengan titik awal dimulai dari titik GD5. Sehingga hasil yang diperoleh adalah sebagai berikut:

- Dari GD 5 ke GD 7 memiliki bobot jarak yaitu dua
- Dari GD 5 ke GD 8 memiliki bobot jarak yaitu empat
- Dari GD 5 ke GD 9 memiliki bobot jarak yaitu empat

**Tabel 3.10 Jalur Dijkstra**

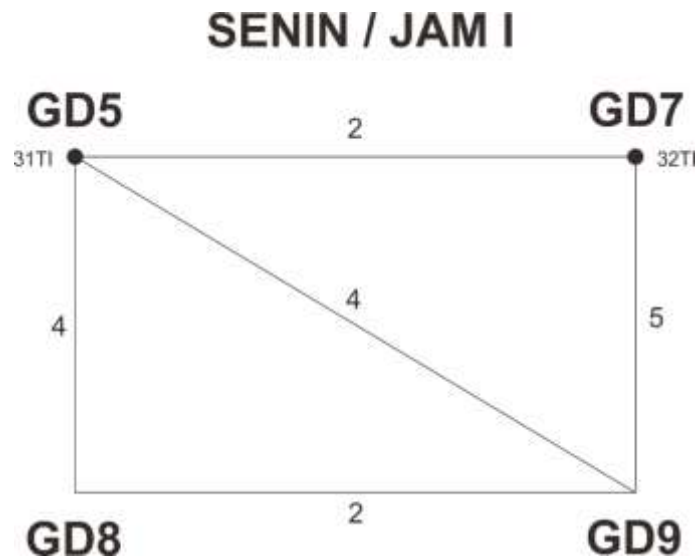
Vertex	Queue
GD 5(-,0)	GD7(-, ∞) GD8(-, ∞) GD9(-, ∞)

GD 5(-, 0)	GD 7(-, 2) GD8(-, 4) GD9(-, 4)
GD7(GD 5, 2)	GD8 (-, 4) GD9(-, 4)
GD8(GD 5, 4)	GD9(-, 4)
GD9 GD 5, 4)	-

Berikut merupakan *Graph* yang menggambarkan pengalokasian kelas menggunakan Algoritma Dijkstra.

#### 1. Graph Penjadwalan Senin Jam Pertama

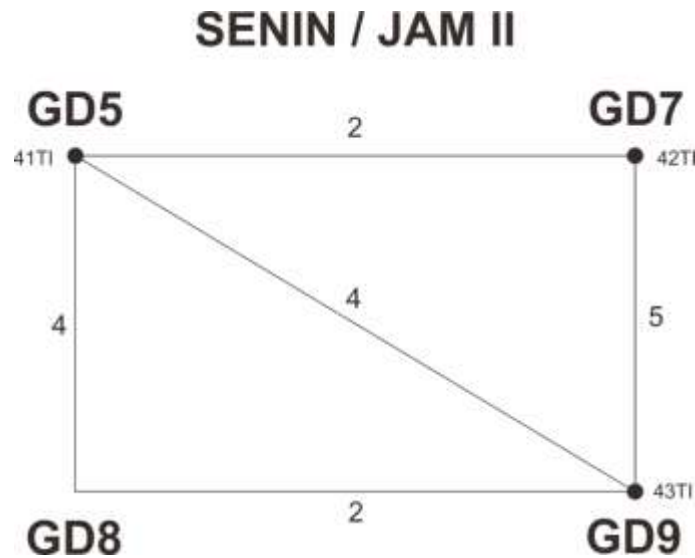
Gambar berikut merupakan *Graph* dari penjadwalan yang akan dilakukan pada hari senin pada jam pertama. *Graph* tersebut dimasuki oleh kelas 31TI dan 32TI dengan masing-masing gedung 5 dan gedung 7.



Gambar 3.7 Graph Penjadwalan Senin Jam Pertama

#### 2. Graph Penjadwalan Senin Jam kedua

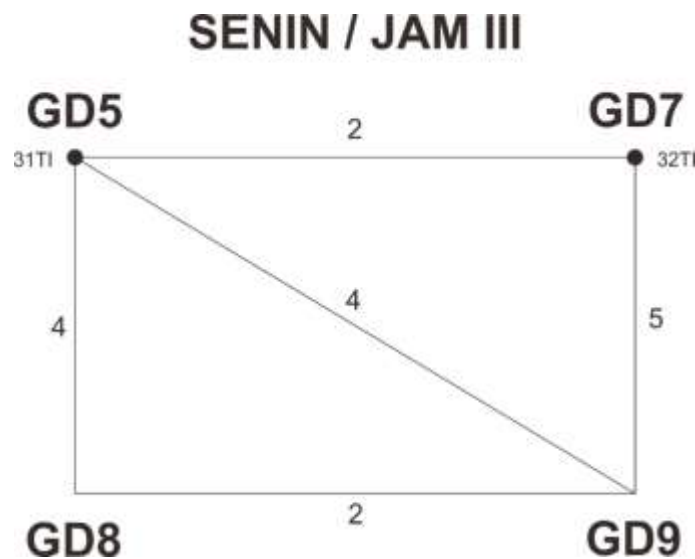
Gambar berikut merupakan *Graph* dari penjadwalan yang akan dilakukan pada hari senin pada jam kedua. *Graph* tersebut dimasuki oleh kelas 41TI dan 32TI, serta 43TI dengan masing-masing gedung 5 dan 7 juga 9.



**Gambar 3.8 Graph Penjadwalan Senin Jam Kedua**

3. *Graph* Penjadwalan Senin Jam ketiga

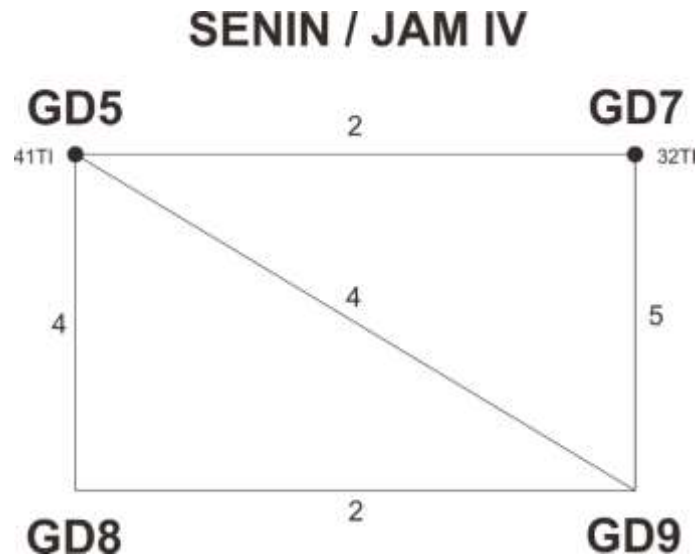
Gambar berikut merupakan *Graph* dari penjadwalan yang akan dilakukan pada hari senin pada jam ketiga. *Graph* tersebut dimasuki oleh kelas 31TI dan 32TI dengan masing-masing gedung 5 dan gedung 7.



**Gambar 3.9 Graph Penjadwalan Senin Jam Ketiga**

4. *Graph* Penjadwalan Senin Jam keempat

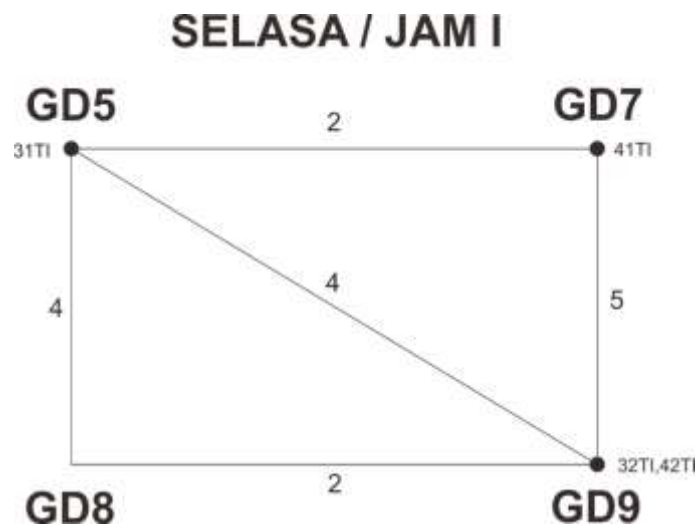
Gambar berikut merupakan *Graph* dari penjadwalan yang akan dilakukan pada hari senin pada jam pertama. *Graph* tersebut dimasuki oleh kelas 41TI dan 32TI dengan masing-masing gedung 5 dan gedung 7.



**Gambar 3.10 Graph Penjadwalan Senin Jam Keempat**

5. *Graph* Penjadwalan Selasa Jam Pertama

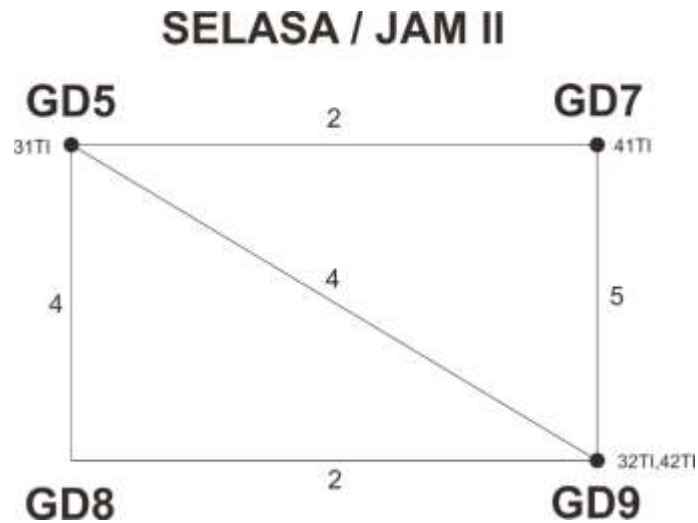
Gambar berikut merupakan *Graph* dari penjadwalan yang akan dilakukan pada hari Selasa pada jam pertama. *Graph* tersebut dimasuki oleh kelas 31TI dan 42TI serta 32TI dengan 42TI dengan masing-masing gedung 5 dan 7 serta 9.



**Gambar 3.11 Graph Penjadwalan Selasa Jam Pertama**

6. *Graph* Penjadwaan Selasa Jam Kedua

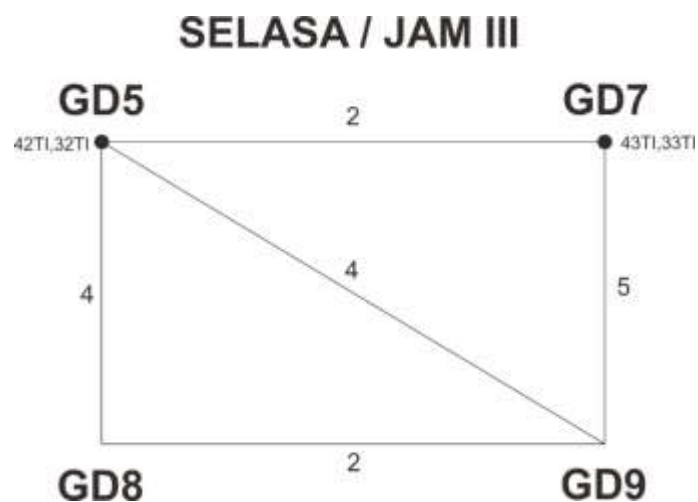
Gambar berikut merupakan *Graph* dari penjadwalan yang akan dilakukan pada hari Selasa pada jam kedua. *Graph* tersebut dimasuki oleh kelas 31TI dan 42TI serta 32TI dengan 42TI dengan masing-masing gedung 5 dan 7 serta 9.



**Gambar 3.12 Graph Penjadwalan Selasa Jam Kedua**

7. Graph Penjadwalan Selasa pada Jam Ketiga

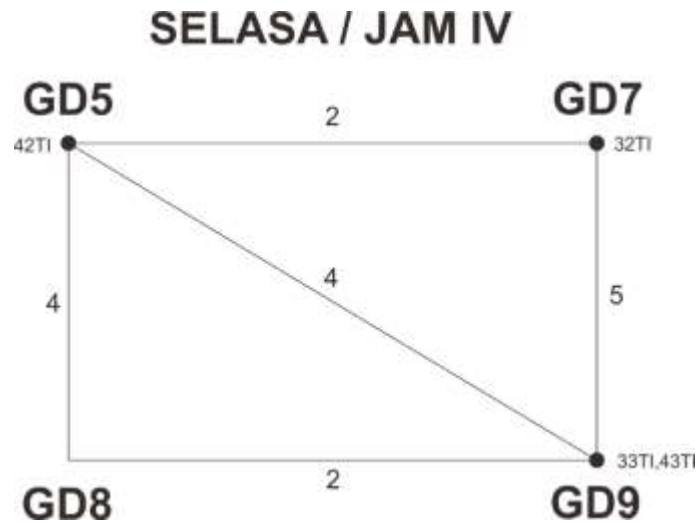
Gambar berikut merupakan *Graph* dari penjadwalan yang akan dilakukan pada hari selasa pada jam ketiga. *Graph* tersebut dimasuki oleh kelas 32TI dengan 42TI dan 33TI dengan 43TI dengan masing-masing gedung 5 dan 7.



**Gambar 3.13 Graph Penjadwalan Selasa Jam Ketiga**

8. Graph Penjadwalan Selasa Jam Keempat

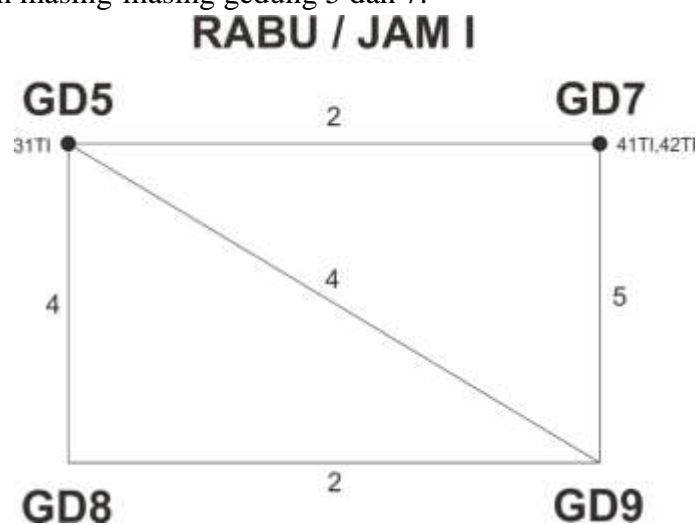
Gambar berikut merupakan *Graph* dari penjadwalan yang akan dilakukan pada hari selasa pada jam keempat. *Graph* tersebut dimasuki oleh kelas 42TI, 32TI dan 33TI dengan 43TI dengan masing-masing gedung 5, 7 dan 9.



**Gambar 3.14 Graph Penjadwalan Selasa Jam Keempat**

9. Graph Penjadwalan Rabu Jam Pertama

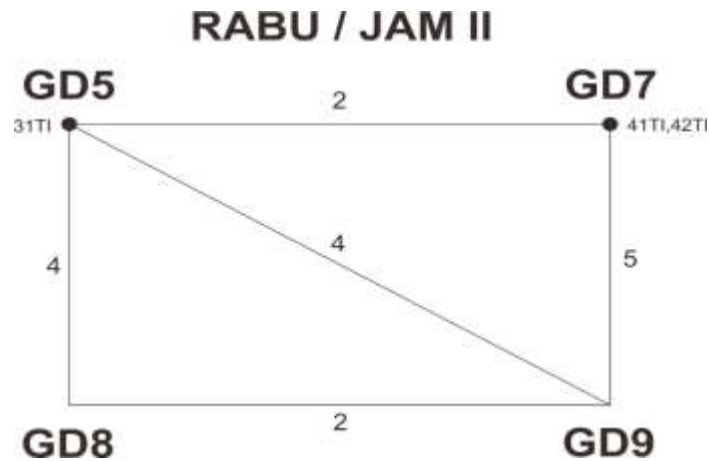
Gambar berikut merupakan *Graph* dari penjadwalan yang akan dilakukan pada hari rabu pada jam pertama. *Graph* tersebut dimasuki oleh kelas 31TI dan 41TI, 42TI dengan masing-masing gedung 5 dan 7.



**Gambar 3.15 Graph Penjadwalan Rabu Jam Pertama**

10. Graph Penjadwalan Rabu Jam Kedua

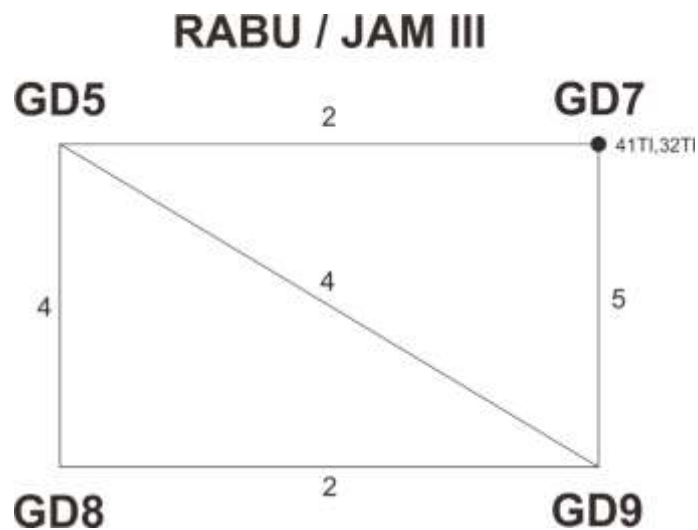
Gambar diatas merupakan *Graph* dari penjadwalan yang akan dilakukan pada hari rabu pada jam kedua. *Graph* tersebut dimasuki oleh kelas 31TI dan 41TI, 42TI dengan masing-masing gedung 5 dan 7.



**Gambar 3.16 Graph Penjadwalan Rabu Jam Kedua**

11. Graph Penjadwalan Rabu Jam Ketiga

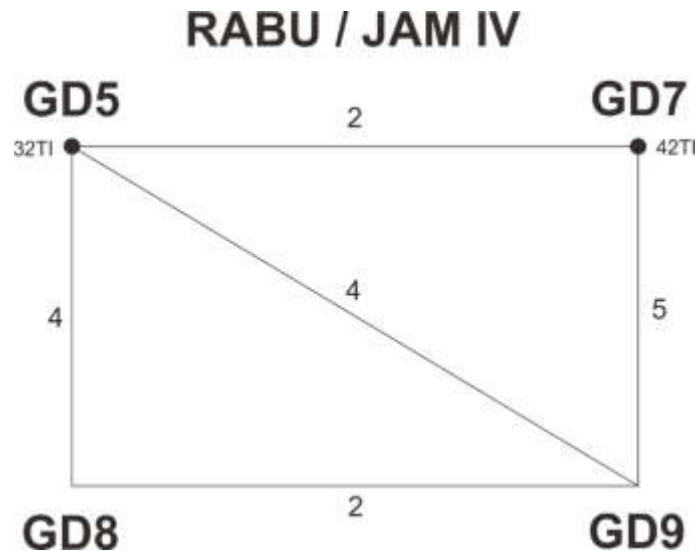
Gambar diatas merupakan *Graph* dari penjadwalan yang akan dilakukan pada hari rabu pada jam pertama. *Graph* tersebut dimasuki oleh kelas 41TI dengan 32TI di gedung 7.



**Gambar 3.17 Graph Penjadwalan Rabu Jam Ketiga**

12. Graph Penjadwalan Rabu Jam Keempat

Gambar berikut merupakan *Graph* dari penjadwalan yang akan dilakukan pada hari rabu pada jam pertama. *Graph* tersebut dimasuki oleh kelas 32TI dan 42TI dengan masing-masing gedung 5 dan 7.



**Gambar 3.18 Graph Penjadwalan Rabu Jam Keempat**

### 3.5. Analisis Kebutuhan Sistem

Pada bab ini dijelaskan kebutuhan dari sistem yang akan dibangun. Sistem yang akan dihasilkan diharapkan dapat menyelesaikan permasalahan pembuatan jadwal perkuliahan dengan jarak terpendek. Kebutuhan sistem yang akan dibangun secara garis besar adalah sebagai berikut:

#### 1. User

*User* yang akan mengelola sistem yang akan dibangun yaitu pihak BAAK di Institut Teknologi Del. Pengguna dapat melakukan akses terhadap modul yang ada pada sistem yang akan dibangun serta mengoperasikannya. Hal yang dapat dilakukan oleh pengguna sistem yaitu memberikan masukan berdasarkan batasan yang telah ada sebelumnya adalah sebagai berikut:

- a. Mengelola data
- b. Membuat jadwal
- c. Mengunduh jadwal

#### 2. Input

*Input*-an yang digunakan dalam pembuatan jadwal perkuliahan ialah berupa data penjadwalan seperti data dosen, mahasiswa, ruangan, waktu, dan mata kuliah berdasarkan *slot* waktu yang tersedia.

#### 3. Proses



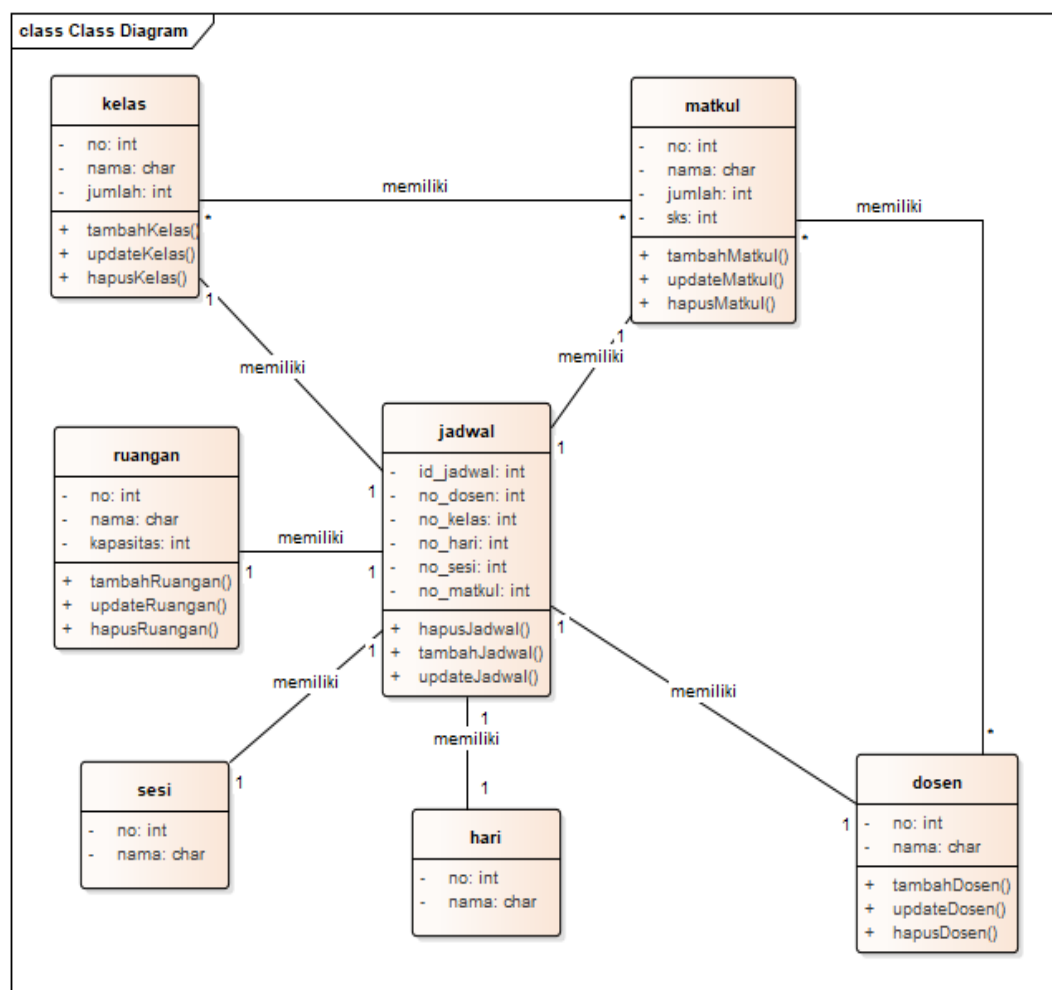
Bagian ini dijelaskan mengenai proses terhadap data yang telah di input pada sistem dalam pembuatan jadwal perkuliahan dengan pendekatan CSP menggunakan algoritma *Backtracking* dengan rute terpendek menggunakan algoritma Dijkstra.

#### 4. Output

*Output* dari sistem yang akan dibangun akan menghasilkan tampilan jadwal perkuliahan pada sistem serta dapat di *export* menjadi tabel excel.

### 3.6. Class Diagram

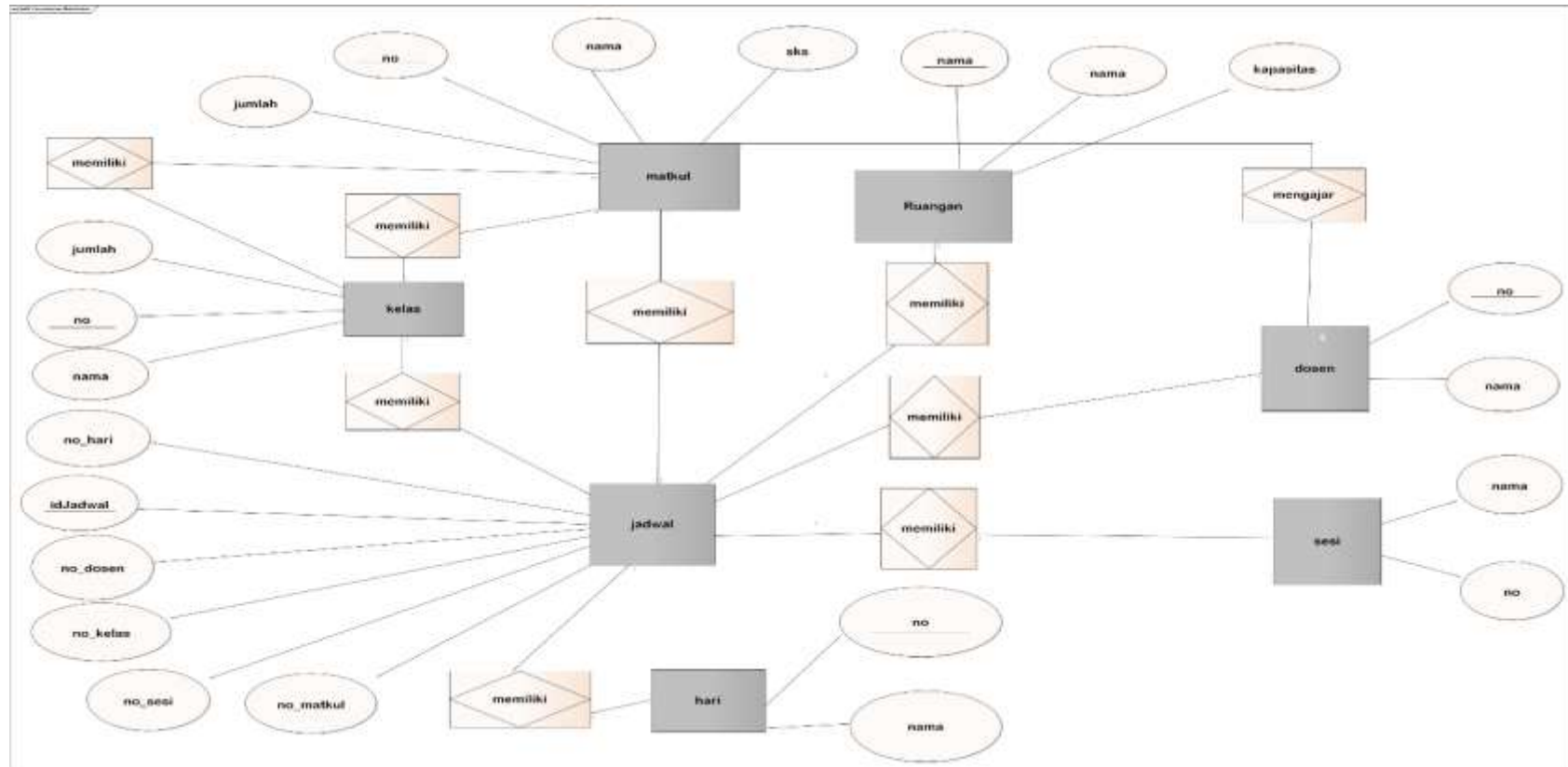
Berikut merupakan gambar class diagram dari aplikasi penjadwalan kuliah. Pada diagram terdapat tujuh kelas yaitu Jadwal, Dosen, Matkul, Kelas, Ruangan, Sesi, dan Hari.



Gambar 3.19 Class Diagram Penjadwalan Kuliah

### 3.7. Entity Relationship Diagram (ERD)

Pada bab ini penulis menggunakan *Entity Relationship Diagram* (ERD) dengan notasi Chen yaitu untuk menggambarkan relasi antar *entity* dalam basis data pada sistem penjadwalan mata kuliah seperti gambar berikut:



Gambar 3.20 ERD Penjadwalan Kuliah

### 3.8. Struktur Database Penjadwalan

#### 1. Tabel Dosen

Digunakan untuk menyimpan data pengajar atau dosen.

Nama Tabel : dosen

Primary Key : no

Tabel 3.11 Tipe data dosen

<b>Nama Field</b>	<b>Tipe</b>	<b>Keterangan</b>
No	Int	Id dari tabel dosen
Nama	varchar	Nama Dosen

#### 2. Tabel Hari

Digunakan untuk menyimpan data hari dari penjadwalan.

Nama Tabel : hari

Primary Key : no

Tabel 3.12 Tipe Data Hari

<b>Nama Field</b>	<b>Tipe</b>	<b>Keterangan</b>
No	int	Id dari tabel hari
Nama	varchar	Hari dalam penjadwalan

#### 3. Tabel Sesi

Digunakan untuk menyimpan data sesi dalam penjadwalan.

Nama Tabel : sesi

Primary Key : no

Tabel 3.13 Tipe Data Sesi

<b>Nama Field</b>	<b>Tipe</b>	<b>Keterangan</b>
No	Int	Id dari tabel sesi
Nama	varchar	Sesi dalam penjadwalan

#### 4. Tabel Mata Kuliah

Digunakan untuk menyimpan data mata kuliah.

Nama Tabel : matkul

Primary Key : no

Tabel 3.14 Tipe Data Mata Kuliah

<b>Nama Field</b>	<b>Tipe</b>	<b>Keterangan</b>
No	Int	Id dari tabel mata kuliah
Nama	Varchar	Nama mata kuliah
Sks	Int	Jumlah sks dari setiap mata kuliah
Jumlah	Int	Jumlah mahasiswa

## 5. Tabel Kelas atau Mahasiswa

Digunakan untuk menyimpan data mahasiswa.

Nama Tabel : kelas

Primary Key : no

Tabel 3.15 Tipe Data Mahasiswa

<b>Nama Field</b>	<b>Tipe</b>	<b>Keterangan</b>
No	Int	Id dari tabel Kelas
Nama	Char	Nama Kelas
Jumlah	Int	Jumlah mahasiswa

## 6. Tabel Jadwal

Digunakan untuk menyimpan data jadwal.

Nama Tabel : jadwal

Primary Key : id\_jadwal

Tabel 3.16 Tipe Data Tabel Jadwal

<b>Nama Field</b>	<b>Tipe</b>	<b>Keterangan</b>
id_jadwal	int	Id dari jadwal
no_dosen	int	Id dari dosen
no_hari	int	Id dari hari
no_sesi	int	Id dari sesi
no_matkul	int	Id dari mata kuliah
no_kelas	int	Id dari kelas

## 7. Kategori

Nama Tabel : kategori

Primary Key : no

Tabel 3.17 Tipe Data Kategori

<b>Nama Field</b>	<b>Tipe</b>	<b>Keterangan</b>
-------------------	-------------	-------------------

No	int	Id dari kategori
Nama	char	Nama kategori

#### 8. Tabel Ruangan

Digunakan untuk menyimpan data mata kuliah.

Nama Tabel : ruangan

Primary Key : no

Tabel 3.18 Tipe Data Ruangan

Nama Field	Tipe	Keterangan
No	int	Id dari tabel ruangan
Nama	varchar	Nama ruangan
Kapasitas	int	Kapasitas ruangan

#### 9. Dosen\_Matakuliah

Nama Tabel : dosen\_matkul

Tabel 3.19 Tipe Data Relasi Dosen dengan Matakuliah

Nama Field	Tipe	Keterangan
No_dosen	int	Id dari dosen
no_matkul	int	Id dari matkul
Nilai	int	Nilai dari relasi

#### 10. Matakuliah\_kelas

Nama Tabel : matkul\_dosen

Tabel 3.20 Tipe Data Relasi Matakuliah dengan Kelas

Nama Field	Tipe	Keterangan
no_matkul	int	Id dari matakuliah
no_kelas	int	Id dari kelas
Nilai	int	Nilai dari relasi

#### 11. Jarak Ruangan

Nama Tabel : jarak\_ruangan

Tabel 3.21 Tipe Data Tabel Jarak Ruangan

Nama Field	Tipe	Keterangan
No	int	Id dari dosen
Source	int	Ruangan asal
Destination	int	Ruangan tujuan
Weight	int	Ukuran jarak

Status	int	Status ruangan kosong atau berisi
--------	-----	-----------------------------------

### 3.9. Antarmuka Perangkat Lunak

Pada bagian ini penulis Antarmuka Perangkat Lunak (*software interface*) yang dibutuhkan dalam mendukung pembangunan sistem yaitu menggunakan Xampp karena mendukung banyak sistem operasi yang merupakan kompilasi dari beberapa program yang digunakan sebagai penerjemahan bahasa yang ditulis dengan bahasa pemrograman java.

#### 3.9.1. Antarmuka Pengguna

Pada bagian ini penulis menjelaskan kebutuhan atau gambaran pelayanan yang harus dipenuhi pada sistem yang akan dibangun. Kebutuhan antarmuka pengguna (*user interface*) untuk mengoperasikan sistem yang akan dibangun terdapat pada tabel berikut:

Tabel 3.22 Kebutuhan User Interface

No	Nama	Keterangan
1	<i>Monitor</i>	<i>Monitor</i> digunakan untuk menampilkan <i>output</i> dari proses yang dilakukan
2	<i>Keyboard</i>	Antarmuka yang digunakan untuk memasukkan data ke dalam sistem.
3	<i>Monitor</i>	Antarmuka yang memungkinkan pengguna untuk melihat perubahan yang terjadi pada sistem serta mempermudah pengguna untuk berinteraksi dengan sistem yang akan dibangun.

#### 3.9.2. Lingkungan Pengembangan

Pada bagian ini penulis menjelaskan *development environment* atau lingkungan pengembangan berisi spesifikasi teknis dari perangkat lunak yang diperlukan dalam pembangunan sistem. Daftar perangkat yang akan digunakan oleh *developer* dalam pembangunan adalah sebagai berikut:

1. *Client*
  - a. *Operating System*: Windows 10 Pro
  - b. *DBMS*: *MySQL*
  - c. *Tools*: *XAMPP*, *Google Chrome*, *Mozilla Firefox*, *SQLYog*

## 2. Hardware

Beberapa spesifikasi *hardware* yang digunakan antara lain:

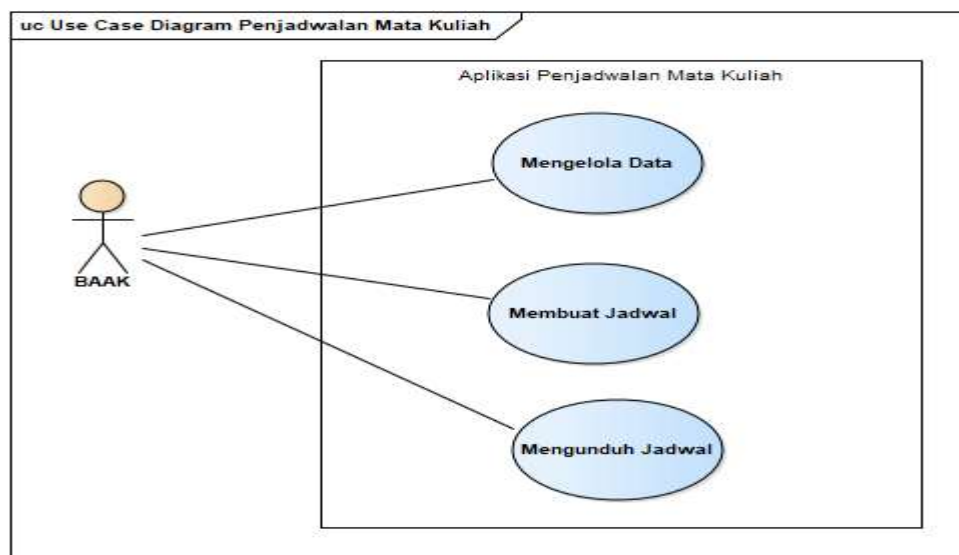
- a. *Processor*: Intel Core I5
- b. *Hard disk*: 500GB
- c. *Memory*: 8GB

### 3.10. Deskripsi Fungsional

Deskripsi fungsional dari sistem penjadwalan yang akan dibangun dapat digambarkan dalam bentuk *use case diagram* dan *use case scenario*.

#### 3.10.1. Use Case Diagram Penjadwalan Mata Kuliah

Pada bagian ini dijelaskan fungsi-fungsi dari sistem penjadwalan yang digambarkan dalam bentuk *use case diagram* seperti berikut:



Gambar 3.21 Use Case Diagram Penjadwalan Mata Kuliah

Penjelasan fungsi-fungsi pada *use case diagram* penjadwalan mata kuliah diatas adalah sebagai berikut:

#### 1. Mengelola Data

BAAK sebagai *user* pada sistem penjadwalan mata kuliah yang akan dibangun dapat melakukan hal-hal sebagai berikut:

- a. Menambahkan Data, BAAK dapat menambahkan data seperti dosen, mahasiswa, mata kuliah, dan ruangan pada jadwal.

- b. Mengedit Data, BAAK dapat mengedit data seperti dosen, mahasiswa, mata kuliah, dan ruangan pada jadwal.
- c. Menghapus Data, BAAK dapat menghapus data seperti dosen, mahasiswa, mata kuliah, dan ruangan pada jadwal.
- d. Melihat Data, BAAK dapat menampilkan data seperti dosen, mahasiswa, mata kuliah, dan ruangan pada jadwal.

## 2. Membuat jadwal

BAAK sebagai *user* pada sistem penjadwalan mata kuliah yang akan dibangun dapat melakukan *generate* jadwal. *Generate* jadwal terdiri dari dua tahap, pertama yaitu menjalankan algoritma *Backtracking* dengan metode CSP. Pada bagian ini, digunakan algoritma *Backtracking* untuk menghasilkan jadwal yang tidak bentrok. Tahap kedua dilakukan *generate* Dijkstra untuk memperoleh ruangan berdasarkan jarak terdekat sesuai jadwal perharinya.

## 3. Mengunduh Jadwal

BAAK sebagai *user* pada sistem penjadwalan mata kuliah yang akan dibangun dapat melakukan unduh atau *download* jadwal yang sudah di-*generate*. Mengunduh jadwal yang telah di-*generate* yaitu dengan cara melakukan *convert* data ke *excel*.

### 3.10.2. Use Case Scenario Penjadwalan Mata Kuliah

Pada bagian ini dijelaskan prosedur atau langkah-langkah *user* BAAK menggunakan sistem penjadwalan yang digambarkan dalam bentuk *use case scenario* seperti berikut:



### 3.10.2.1. Mengelola Data

Use case scenario untuk mengelola data dapat dilihat pada tabel *use case scenario* berikut:

Tabel 3.23 Mengelola Data

<i>Use case Name</i>	Mengelola Data	
<i>Use case Description</i>	<i>Use case</i> berikut mendeskripsikan prosedur untuk BAAK menambahkan data jadwal pada system	
<i>Actor</i>	BAAK	
<i>Precondition</i>	BAAK mengakses sistem penjadwalan mata kuliah	
<i>Primary Flow of Events</i>	<i>User Action</i>	<i>System Response</i>
	1. BAAK mengakses sistem penjadwalan mata kuliah	
		2. Sistem menampilkan halaman penjadwalan mata kuliah
	3. BAAK mengisi <i>form</i> yang ditampilkan seperti Inisial Dosen, Mata Kuliah, Kelas yang diajarkan. 4. Selanjutnya BAAK klik <i>button</i> Tambah untuk menambahkan data dosen.	
		5. Sistem menampilkan daftar jadwal berisi inisial dosen, mata kuliah dan kelas yang sudah ditambahkan.
	6. Data Dosen pada sistem dihapus	
		7. Data Dosen pada sistem dihapus
	8. BAAK mengisi <i>form Set Gedung</i> berisi Kelas, Hari, Ruangan awal. Selanjutnya untuk <i>Setting gedung</i> klik <i>button Set</i> .	

		9. Sistem menampilkan daftar ruangan yang kosong dengan jarak terpendek.
<i>Alternate Flow of Events</i>	-	-
<i>Error Flow of Events</i>	-	-
<i>Post Condition</i>	Aktor berhasil mengelola data pada sistem.	

### 3.10.2.2. Membuat Jadwal

*Use case scenario* untuk membuat atau *generate* jadwal data dapat dilihat pada tabel berikut:

Tabel 3.24 Use Case Scenario Membuat Jadwal

<i>Use case Name</i>	Membuat Jadwal	
<i>Use case Description</i>	<i>Use case</i> berikut mendeskripsikan prosedur untuk BAAK <i>men-generate</i> atau membuat data jadwal pada sistem	
<i>Actor</i>	BAAK	
<i>Precondition</i>	BAAK mengakses sistem penjadwalan mata kuliah	
<i>Primary Flow of Events</i>	<i>User Action</i>	<i>System Response</i>
	4. BAAK mengakses sistem penjadwalan mata kuliah	
		5. Sistem menampilkan halaman penjadwalan mata kuliah
	6. BAAK mengisi <i>form</i> yang ditampilkan seperti Inisial Dosen, Mata Kuliah, Kelas yang diajarkan. 7. Selanjutnya BAAK klik <i>button</i> Tambah untuk menambahkan data dosen.	
		8. Sistem menampilkan daftar jadwal berisi inisial dosen, mata

		kuliah dan kelas yang sudah ditambahkan.
	9. BAAK klik ikon <i>delete</i> untuk menghapus data yang telah di <i>Set</i> .	
		10. Data Dosen pada sistem dihapus
	11. BAAK klik <i>button generate</i> jadwal yang sudah ditambahkan.	
		12. Sistem menampilkan hasil <i>generate</i> jadwal menggunakan Algoritma <i>Backtracking</i> . Tampilan pada sistem berupa <i>form</i> untuk melakukan <i>Set</i> Gedung
	13. BAAK mengisi <i>form Set</i> Gedung berisi Kelas, Hari, Ruangan awal. Selanjutnya untuk <i>Setting</i> gedung klik <i>button Set</i> .	
		14. Sistem menyimpan data dari <i>Set</i> Gedung yang dilakukan BAAK.
	15. BAAK klik <i>button Generate</i> Dijkstra	
		16. Sistem menampilkan daftar ruangan yang kosong dengan jarak terpendek.
<i>Alternate Flow of Events</i>	-	-
<i>Error Flow of Events</i>	-	-
<i>Post Condition</i>	Aktor berhasil membuat atau <i>generate</i> jadwal berdasarkan data yang sudah ditambahkan ke sistem.	

### 3.10.2.3. Mengunduh Jadwal

*Use case scenario* untuk mengunduh atau *download* data dapat dilihat pada tabel berikut:

Tabel 3.25 Mengunduh Jadwal

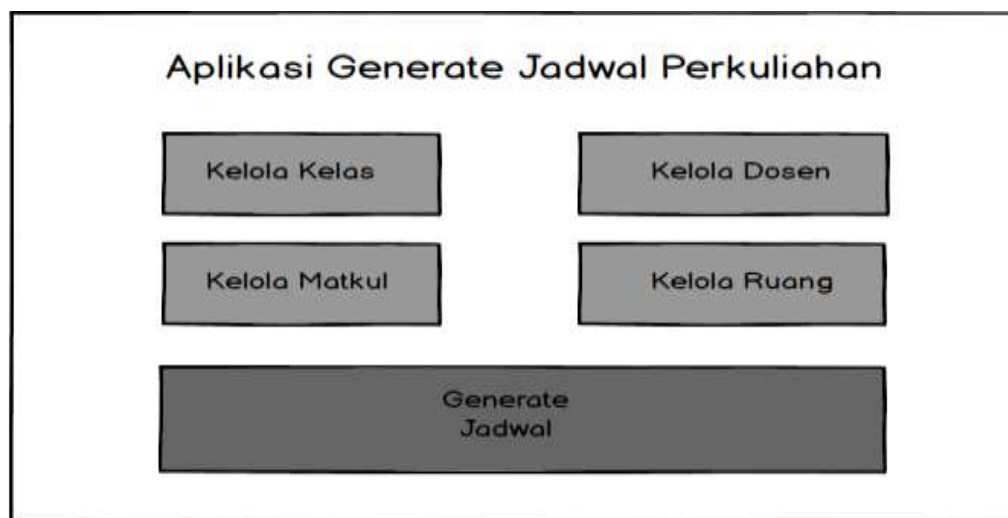
<i>Use case Name</i>	Mengunduh Jadwal	
<i>Use case Description</i>	<i>Use case</i> berikut mendeskripsikan prosedur untuk BAAK mengunduh hasil <i>generate</i> jadwal pada sistem	
<i>Actor</i>	BAAK	
<i>Precondition</i>	BAAK mengakses sistem penjadwalan mata kuliah	
<i>Primary Flow of Events</i>	<i>User Action</i>	<i>System Response</i>
	1. BAAK mengakses sistem penjadwalan mata kuliah	
		2. Sistem menampilkan halaman penjadwalan mata kuliah
	3. BAAK mengisi <i>form</i> yang ditampilkan seperti Inisial Dosen, Mata Kuliah, Kelas yang diajarkan. 4. Selanjutnya BAAK klik <i>button</i> Tambah untuk menambahkan data dosen.	
		5. Sistem menampilkan daftar jadwal berisi inisial dosen, mata kuliah dan kelas yang sudah ditambahkan.
	6. BAAK klik ikon <i>delete</i> untuk menghapus data yang telah di <i>Set</i> .	
		7. Data Dosen pada sistem dihapus

	8. BAAK klik <i>button generate</i> jadwal yang sudah ditambahkan.	
		9. Sistem menampilkan hasil <i>generate</i> jadwal menggunakan Algoritma <i>Backtracking</i> . Tampilan pada sistem berupa <i>form</i> untuk melakukan <i>Set Gedung</i>
	10. BAAK mengisi <i>form Set</i> Gedung berisi Kelas, Hari, Ruangan awal. Selanjutnya untuk <i>Setting</i> gedung klik <i>button Set</i> .	
		11. Sistem menyimpan data dari <i>Set Gedung</i> yang dilakukan BAAK.
	12. BAAK klik <i>button Generate Dijkstra</i>	
		13. Sistem menampilkan daftar ruangan yang kosong dengan jarak terpendek.
	14. BAAK melakukan <i>download</i> hasil jadwal yang sudah dibuat dengan cara klik <i>button Download</i> jadwal.	
		15. Sistem menampilkan halaman <i>Convert Jadwal</i>
	16. BAAK klik <i>button Convert</i>	

		17. Jadwal berhasil di <i>convert</i> dalam bentuk excel.
<i>Alternate Flow of Events</i>	-	-
<i>Error Flow of Events</i>	-	-
<i>Post Condition</i>	Aktor berhasil mengunduh hasil <i>generate</i> jadwal berdasarkan data yang sudah ditambahkan ke sistem.	

### 3.11. Desain

Pada bab ini penulis menjelaskan desain aplikasi yang akan dibangun pada sistem penjadwalan saat melakukan *generate* jadwal.



Gambar 3.22 Desain Menu Utama Aplikasi

Aplikasi *generate* jadwal perkuliahan memiliki menu utama yang terdiri atas beberapa menu, yaitu kelola kelas, kelola matkul, kelola dosen, kelola ruang dan *generate* jadwal.

Halaman data kelas digunakan untuk mendapatkan data mengenai kelas (mahasiswa). Pada halaman ini, *user* dapat memberikan informasi terkait nama kelas dan jumlah mahasiswa. Pada halaman ini terdapat tiga *Button*, yaitu tambah untuk menambahkan data, *update* untuk mengubah data, dan *delete* untuk menghapus data.

**Aplikasi Generate Jadwal Perkuliahan**

**Main Menu**

**Data Kelas**

Nama Kelas

Jumlah Mahasiswa

Total Data: 6

Nama	jumlah
12SI1	28
12SI2	30
33TI	27
43TI	30
13TI1	30
13TI2	30

Gambar 3.23 Halaman Kelola Data Kelas

Ketika data telah ditambahkan, maka akan tampil penambahan data pada daftar tabel daftar kelas pada halaman tersebut.

**Aplikasi Generate Jadwal Perkuliahan**

**Main Menu**

**Data Dosen**

Inisial Dosen

Total Data: 2

Nama
YHP
YBN

Gambar 3.24 Halaman Kelola Data Dosen

Pada halaman data dosen terdapat kolom inputan yaitu inisial dosen. Terdapat tiga *button* yaitu *tambah*, *update* dan *delete*. Aksi pada ketiga *button* ini yaitu menambahkan, mengubah dan menghapus data akan terlihat dalam tabel daftar nama dosen pada halaman tersebut.

**Aplikasi Generate Jadwal Perkuliahan**

**Main Menu**

**Data Mata Kuliah**

Nama Matakuliah

Jumlah SKS

Jumlah Mahasiswa

**Tambah** **Update** **Delete**

Total Data: 3

Nama	SKS	Jumlah
MATDIS	3	30
ALU	2	30
CERTAN	3	30

Gambar 3.25 Halaman Kelola Data Mata Kuliah

Pada halaman data mata kuliah terdapat *field* nama matakuliah, jumlah sks, dan jumlah mahasiswa. Terdapat tiga *button* yaitu, tambah, *update*, dan *delete*. Penambahan, perubahan dan penghapusan data akan terlihat pada tabel daftar mata kuliah yang terdapat pada halaman tersebut.

**Aplikasi Generate Jadwal Perkuliahan**

**Main Menu**

**Data Jadwal**

Dosen

Mata Kuliah

Kelas

**Tambah** **Update**

**Hapus Mata kuliah-Kelas**

**Hapus Dosen-Mata kuliah**

**Hapus Semua Relasi**

Total Data: 12

Dosen	Mata Kuliah	Kelas
ACB	PRPL	31TI
ACB	PRPL	42TI
ACB	PUPL	32TI
ACB	PUPL	42TI
RDT	PAM	32TI
RDT	PAM	42TI
RDT	PBO	32TI
RDT	PBO	42TI
TNT	PSW	31TI
TNT	PSW	41TI
TNT	KEPAL	33TI
TNT	KEPAL	43TI

**Generate Jadwal**

Gambar 3.26 Halaman Kelola Data Jadwal



Desain diatas digunakan untuk melakukan *generate* jadwal dalam menjalankan Algoritma *Backtracking* dengan metode CSP. Pada halaman tersebut terdapat *Button* tambah untuk menambahkan dan *update* untuk mengubah data dosen, mata kuliah, dan kelas. *Button* hapus mata kuliah–kelas berfungsi untuk menghapus relasi antar mata kuliah dan kelas. *Button* hapus dosen – mata kuliah berfungsi untuk menghapus data relasi antara dosen dan mata kuliah. *Button* hapus semua relasi berfungsi untk menghapus keseluruhan relasi yang telah terbentuk sebelumnya. Pada halaman tersebut terdapat daftar relasi antar dosen, matakuliah, dan kelas. Dengan menggunakan relasi tersebut akan digenerate jadwal dengan memperhatikan *constraints* yang ada. *Button generate* jadwal berfungsi untk memberikan aksi *generate* jadwal. *Generate* jadwal merupakan hasil dari data yang telah di-*input*. Pada bagian ini, digunakan Algoritma *Backtracking* untuk melakukan pencarian solusi dan menghasilkan jadwal yang tidak bentrok.

Berikut merupakan hasil yang akan ditampilkan apabila jadwal telah digenerate dengan algoritma backtracking yaitu sebagai berikut:

Jam	Senin				Selasa				Rabu			
	Matkul	Sesi	Dosen	Kelas	Matkul	Sesi	Dosen	Kelas	Matkul	Sesi	Dosen	Kelas
1	MATDIS	T	YBN	12SI1	MATDIS	P	YBN, YHP	12SI1				12SI1
				12SI2				12SI2				12SI2
				33TI				33TI	ALU	P	YBN, YHP	33TI
				43TI				43TI				43TI
	CERTAN	T	YHP	13TI1	CERTAN	P	YHP	13TI1				13TI1
				13TI2				13TI2				13TI2
2				12SI1	MATDIS	P	YBN, YHP	12SI1				12SI1
	MATDIS	T	YBN	12SI2				12SI2				12SI2
				33TI				33TI	ALU	P	YBN, YHP	33TI
				43TI				43TI				43TI
				13TI1	CERTAN	P	YHP	13TI1				13TI1
	CERTAN	T	YHP	13TI2				13TI2				13TI2
3				12SI1				12SI1				12SI1
				12SI2	MATDIS	P	YBN	12SI2				12SI2
	ALU	T	YBN, YHP	33TI				33TI				33TI
				43TI				43TI	ALU	P	YBN, YHP	43TI
				13TI1				13TI1				13TI1
				13TI2	CERTAN	P	YHP	13TI2				13TI2
4				12SI1				12SI1				12SI1
				12SI2	MATDIS	P	YBN	12SI2				12SI2
				33TI				33TI				33TI
	ALU	T	YBN, YHP	43TI				43TI	ALU	P	YBN, YHP	43TI
				13TI1				13TI1				13TI1
				13TI2	CERTAN	P	YHP	13TI2				13TI2

**Gambar 3.27 Desain Output Jadwal Backtracking**

Setelah data selesai di-generate, maka akan muncul halaman seperti gambar dibawah ini,

**Aplikasi Generate Jadwal Perkuliahan**

**Main Menu**

**Data Ruangan**

Nama Ruangan

Kapasitas

Jenis ruangan

Total Data: 15

Nama	Jenis	Kapasitas
921	Praktikum	30
932	Teori	30
912	Teori	30
714	Praktikum	30
725	Praktikum	30
514	Praktikum	30

**Gambar 3.28 Halaman Data Ruangan**

Halaman tersebut digunakan untuk memperoleh data ruangan, yang terdiri atas nama ruangan, kapasitas, dan jenis ruangan yang terdiri atas ruangan teori dan ruangan praktikum. *Button* tambah berguna untuk menambahkan data ruangan, *update* untuk mengubah, *delete* untuk menghapus data yang telah dibuat sebelumnya. Data ini digunakan untuk langkah selanjutnya setelah jadwal telah digenerate menggunakan Algoritma *Backtracking*. Hasil *generate* jadwal menggunakan *Backtracking* merupakan jadwal tanpa ruangan. Pengalokasian ruangan dilakukan dengan menggunakan Dijkstra untuk mencari jarak terdekat untuk perpindahan antar kelas, dengan jadwal perkuliahan yang beruntun pada jadwal suatu kelas perharinya.

Untuk menghasilkan jadwal perkuliahan disertai ruangan yang memiliki jarak terdekat antar ruangan yang dapat dimiliki, maka digunakan Algoritma Dijkstra. Proses kerjanya dilakukan dengan cara mengatur ruangan terlebih dahulu, untuk ruangan pertama setiap kelas perharinya. Setiap kelas harus memiliki ruangan pertama setiap harinya, hingga akhirnya jadwal akan di-*generate* menggunakan Algoritma Dijkstra untuk mencari kelas selanjutnya. Berikut merupakan rancangan halaman *generate* Dijkstra,

**Aplikasi Generate Jadwal Perkuliahan**

**Main Menu**

**Data Jadwal**

Kelas

Ruangan

Set Ruangan

Total Data : 10

Pilih Hari

Sesi	Mata Kuliah	Dosen	Kelas	Ruangan
Sesi 1	OOSD	RDT	33TI	Belum di-set
Sesi 1	PAP	RIS	33TI	Belum di-set
Sesi 1	CERTAN	YHP	33TI	Belum di-set
Sesi 2	PAP	RIS	33TI	Belum di-set
Sesi 2	CERTAN	YHP	33TI	Belum di-set
Sesi 2	OOSD	RDT	33TI	Belum di-set
Sesi 3	OOSD	RDT	33TI	Belum di-set
Sesi 3	KEPAL	TNT	33TI	Belum di-set
Sesi 3	MATDIS	YHP	33TI	Belum di-set
Sesi 4	MATDIS	YHP	33TI	Belum di-set

Gambar 3.29 Halaman Generate Dijkstra

Berikut merupakan hasil yang akan ditampilkan apabila jadwal telah di-generate dengan Algoritma Dijkstra. Gambar tersebut merupakan halaman yang digunakan untuk mengunduh jadwal yang telah di-generate dengan cara melakukan *convert* data ke excel

yaitu sebagai berikut:

Jam	Senin					Selasa					Rabu				
	Matkul	Sesi	Dosen	Kelas	Ruangan	Matkul	Sesi	Dosen	Kelas	Ruangan	Matkul	Sesi	Dosen	Kelas	Ruangan
1	MATDIS	T	YBN	12SI1	921	MATDIS	P	YBN, YHP	12SI1	712				12SI1	
				12SI2					12SI2					12SI2	
				33TI					33TI		ALU	P	YBN, YHP	33TI	712
				43TI					43TI					43TI	
	CERTAN	T	YHP	13TI1	923	CERTAN	P	YHP	13TI1	714				13TI1	
2				13TI2					13TI2					13TI2	
				12SI1		MATDIS	P	YBN, YHP	12SI1	712				12SI1	
	MATDIS	T	YBN	12SI2	924				12SI2					12SI2	
				33TI					33TI		ALU	P	YBN, YHP	33TI	712
				43TI					43TI					43TI	
3				13TI1		CERTAN	P	YHP	13TI1	714				13TI1	
	CERTAN	T	YHP	13TI2	925				13TI2					13TI2	
				12SI1					12SI1					12SI1	
				12SI2		MATDIS	P	YBN	12SI2	713				12SI2	
	ALU	T	YBN, YHP	33TI	921				33TI					33TI	
4				43TI					43TI		ALU	P	YBN, YHP	43TI	713
				13TI1					13TI1					13TI1	
				13TI2		CERTAN	P	YHP	13TI2					13TI2	
				12SI1					12SI1					12SI1	
				12SI2		MATDIS	P	YBN	12SI2	713				12SI2	
				33TI					33TI					33TI	
	ALU	T	YBN, YHP	43TI	923				43TI		ALU	P	YBN, YHP	43TI	713
				13TI1					13TI1					13TI1	
				13TI2		CERTAN	P	YHP	13TI2					13TI2	

Gambar 3.30 Desain Output Jadwal Dijkstra

## BAB 4

### Implementasi dan Testing

Pada bab ini dijelaskan mengenai kebutuhan implementasi, data dan pengujian serta hasil pengujian.

#### 4.1. Kebutuhan Implementasi

Pada subbab ini dijelaskan mengenai kebutuhan perangkat keras dan perangkat lunak dalam proses pembangunan aplikasi penjadwalan kuliah. Berikut merupakan spesifikasi perangkat keras dan perangkat lunak yang dimiliki

##### Spesifikasi Perangkat Keras

Tabel 4.1 Spesifikasi Perangkat Keras

No	Hardware	Spesifikasi	
1	Lenovo	Processor	Intel Core i5
		Memory	8 GB
		Operating System	Windows 10

##### Spesifikasi Perangkat Lunak

Tabel 4.2 Spesifikasi Perangkat Lunak

No	Software	Spesifikasi
1	Development Tools	IntelliJ IDEA
2	Programming Language	Java
3	Database	MySQL
4	Web Server	XAMPP
5	Operating System	Windows 10

#### 4.2. Hasil Aplikasi

Pada subbab ini dijelaskan mengenai aplikasi penjadwalan yang telah dibangun. Aplikasi penjadwalan memiliki beberapa halaman yang digunakan untuk mengelola sumber daya, diantaranya yaitu menu utama, kelola kelas, kelola dosen, kelola matkul, kelola ruang, data jadwal untuk *generate backtracking* dan *generate dijkstra*.

##### 4.2.1. Menu Utama

Aplikasi *generate* jadwal perkuliahan memiliki tampilan utama yang terdiri atas lima menu yang memiliki fungsinya masing-masing dalam proses pembuatan

jadwal. Diantaranya, yaitu menu kelola kelas, kelola dosen, kelola mata kuliah, kelola ruang dan *generate* jadwal.

Button perbarui data pada halaman utama berfungsi untuk melakukan *reset* pada keseluruhan relasi yang dimiliki, apabila terjadi perubahan pada relasi antara sumber daya, seperti dosen dengan mata kuliah dan mata kuliah dengan dosen. Kasus ini dapat terjadi apabila terjadi penambahan dosen atau pergantian mata kuliah yang diajarkan oleh dosen.

Berikut merupakan tampilan utama aplikasi *generate* jadwal perkuliahan,



**Gambar 4.1 Tampilan Menu Utama Aplikasi Penjadwalan Mata Kuliah**

#### **4.2.2. Kelola Kelas**

Menu kelola kelas berguna untuk mengelola data kelas yang dimiliki suatu fakultas. Atribut dari data kelas terdiri atas nama kelas dan jumlah mahasiswa yang berada pada kelas tersebut. *Button* tambah pada halaman data kelas berguna untuk menambahkan data pada daftar kelas. Data yang telah ditambahkan akan ditampilkan pada tabel daftar kelas. *Button update* berfungsi untuk mengubah data yang telah ditambahkan sebelumnya. *Button* hapus berfungsi untuk menghapus data yang tidak yang telah ditambahkan namun tidak digunakan dalam membuat

jadwal. *Button* “main menu” berfungsi untuk dapat kembali ke halaman menu utama. Berikut merupakan tampilan halaman data kelas,

**MAIN MENU**

## Aplikasi Jadwal Perkuliahan

\*\* Pilih salah satu Kelas pada tabel untuk update

### Data Kelas

Nama

Jumlah

Total Data : 4

Nama	Jumlah
11TB	30
12TB	30
13TB	30
14TB	30

**Gambar 4.2 Tampilan Menu Kelola Kelas Aplikasi Jadwal Perkuliahan**

Menu kelola dosen berfungsi untuk mengelola data dosen. Data dosen yang diperlukan yaitu nama dosen. Tersedia empat *Button* pada halaman ini, diantaranya yaitu, *Button* tambah untuk menambahkan data dosen. Setelah data dosen ditambahkan maka akan dimunculkan pada daftar nama dosen. *Button update* untuk mengubah data dosen. *Button* hapus berfungsi untuk menghapus data dosen. Main menu untuk kembali ke halaman utama, berikut merupakan tampilan halaman data dosen,

**MAIN MENU**

## Aplikasi Jadwal Perkuliahan

\*\* Pilih salah satu Dosen pada tabel untuk update

### Data Dosen

Nama

Total Data : 21

Nama
VUL
YBN
SML
SFA
RZS
RFK
REG
PN
MSL
MPR
MNP
MMM
JUN
IFY
FAKI

**Gambar 4.3 Tampilan Menu Kelola Dosen Aplikasi Jadwal Perkuliahan**

#### 4.2.3. Kelola Mata Kuliah

Menu kelola mata kuliah berfungsi untuk mengelola data mata kuliah. Terdapat tiga *field* pada data mata kuliah yaitu nama mata kuliah, jumlah SKS, dan jumlah mahasiswa yang mengambil mata kuliah tersebut. Data ini ditambahkan dengan menggunakan *Button* tambah, dan akan ditampilkan pada table daftar mata kuliah. *button update* untuk mengubah data dan *Button delete* untuk menghapus data mata kuliah yang telah ditambahkan sebelumnya.

**MAIN MENU**

## Aplikasi Jadwal Perkuliahan

\*\* Pilih salah satu Mata Kuliah pada tabel untuk update

### Data Mata Kuliah

Nama  
Jumlah Pertemuan  
Jumlah Mahasiswa  
Kategori ▼

Tambah Update Hapus

Total Data : 34

Nama	SKS	Jumlah	Kategori
UPL	3	30	Teori
TRB	3	30	Teori
STA	2	30	Teori
SPP	3	30	Teori
PTI	2	30	Teori
PROBSTAT	3	30	Teori
PRAK PIP	4	30	Praktikum
PRAK MIKRUM	4	30	Praktikum
PRAK KIMOR	4	30	Praktikum
PRAK KIMFIS	4	30	Praktikum
PRAK KIMDAS	4	30	Praktikum
PRAK FISDAS 1	3	30	Praktikum
PP	3	30	Teori
PIP	2	30	Teori
PRATI	2	30	Teori

**Gambar 4.4 Tampilan Menu Kelola Mata Kuliah Aplikasi Jadwal Perkuliahan**

#### 4.2.4. Kelola Data Ruangan

Menu kelola ruangan berfungsi untuk mengelola data ruangan. Halaman data ruangan memiliki *field*, yaitu nama ruangan dan kapasitas ruangan untuk membatasi muatan. Terdapat *button* jenis ruangan berupa *dropdown* yang dapat dipilih oleh *user* sebagai keterangan ruangan. *Button* tambah digunakan untuk menambahkan data ruangan, *button update* untuk mengubah data yang telah ada, dan *button* hapus untuk menghapus data yang telah dibuat sebelumnya.

**MAIN MENU**

## Aplikasi Jadwal Perkuliahan

\*\* Pilih salah satu Ruangan pada tabel untuk update

### Data Ruangan

Nama

Kapasitas

Kategori

Nama	Kapasitas	Kategori
GD514	30	Praktikum
GD515	30	Praktikum
GD516	60	Teori
GD524	30	Praktikum
GD525	30	Teori/Praktikum
GD526	30	Teori/Praktikum
GD711	30	Praktikum
GD712	30	Praktikum
GD713	30	Praktikum
GD714	30	Praktikum
GD721	60	Teori
GD722	150	Teori
GD723	60	Praktikum
GD724	30	Praktikum
GD725	30	Praktikum

**Gambar 4.5 Tampilan Menu Kelola Ruangan Aplikasi Jadwal Perkuliahan**

#### 4.2.5. Kelola Jadwal

Pada bab ini dijelaskan mengenai tampilan *generate* jadwal, metode dan algoritma yang digunakan

##### 4.2.5.1. Generate Jadwal

Halaman ini merupakan halamam akhir yang berfungsi untuk melakukan *generate* jadwal. Data dosen, mata kuliah dan kelas yang telah ditambahkan sebelumnya akan ditampilkan pada tabel yang ada pada halaman *generate* jadwal. Data masih dapat ditambah dengan menggunakan *button* tambah ataupun diubah dengan menggunakan *button update*, baik data dosen, mata kuliah ataupun kelas. Untuk menghapus data dapat digunakan *button* hapus mata kuliah-kelas, *button* ini berfungsi untuk menghapus relasi mata kuliah dan kelas. Begitu juga *button* hapus dosen-mata kuliah untuk menghapus relasi antar dosen dan mata kuliah. *user* juga dapat menghapus keseluruhan relasi yang telah ada. Untuk melakukan *generate* jadwal, menggunakan *button generate* jadwal untuk menghasilkan jadwal yang benar tanpa pengalokasian ruangan.



**MAIN MENU**

## Aplikasi Jadwal Perkuliahan

### Data Jadwal

Dosen ▼  
Mata Kuliah ▼  
Kelas ▼

Tambah Update

Hapus Mata Kuliah - Kelas

Hapus Dosen - Mata Kuliah

Hapus Semua Relasi

Kelas ▲	Mata Kuliah	Dosen
11TB	STA	MMM
11TB	PTI	ASD
11TB	PRAK KIMDAS	SFA
11TB	PRAK FISDAS 1	RZS
11TB	MADAS 1	SML
11TB	KIMDAS	SFA
11TB	FISDAS 1	RZS
11TB	ENG 1	MPR
11TB	DELCHA	ANA
12TB	PRAK MIKRUM	EAN
12TB	PRAK KIMOR	RFK
12TB	PRAK KIMFIS	YUL

Total Data : 34

Lihat Hasil Generate

**GENERATE JADWAL**

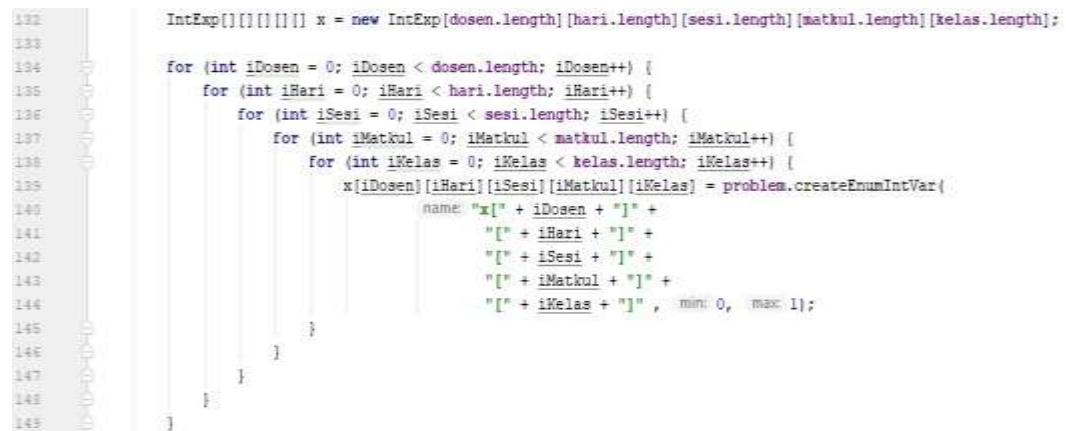
**Gambar 4.6 Tampilan Menu *Generate Jadwal***

Proses *generate* jadwal diatas dilakukan dengan menggunakan metode *Constraint Satisfaction Problem* dan *backtracking* dalam proses pencarian solusi. Setelah proses *generate* jadwal berhasil dilakukan, proses selanjutnya yang dilakukan yaitu *generate* jadwal dengan ruangan menggunakan Algoritma Dijkstra.

#### **4.2.5.2. Kode Program Pemodelan *Constraint Satisfaction Problem***

Implementasi *Constraint Satisfaction Problem* menggunakan *library choco solver*. *Choco solver* merupakan *library* yang berorientasi pada Algoritma *Backtracking* dalam pencarian solusinya. Choco solver Pada bab ini menjelaskan tentang kode-kode program yang berisi deklarasi dari setiap variabel. Variabel menyimpan nilai atau domain dalam bentuk *array*. Variabel yang dimiliki yaitu, dosen, hari, sesi, mata kuliah, kelas dan sks. Sedangkan, dosen\_matkul merupakan relasi domain antara dosen dan matkul. matkul\_kelas merupakan relasi domain antara variabel mata kuliah dan kelas. Relasi domain akan menyimpan nilai 0 atau 1 untuk setiap relasinya. Nilai 1 untuk domain yang berkaitan. Misalnya dosen A mengajar mata kuliah B, maka domain A pada variabel dosen yang berelasi dengan domain B pada variabel mata kuliah akan memiliki nilai satu. Sebaliknya, bila dosen A tidak mengajar mata kuliah B, maka nilai dari relasi kedua domain ini akan nol.

Berikut merupakan gambar dari kode program deklarasi dari setiap variabel penjadwalan mata kuliah:



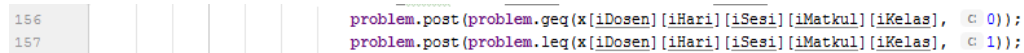
```

132 IntExp[][][][][] x = new IntExp[dosen.length][hari.length][sesi.length][matkul.length][kelas.length];
133
134 for (int iDosen = 0; iDosen < dosen.length; iDosen++) {
135     for (int iHari = 0; iHari < hari.length; iHari++) {
136         for (int iSesi = 0; iSesi < sesi.length; iSesi++) {
137             for (int iMatkul = 0; iMatkul < matkul.length; iMatkul++) {
138                 for (int iKelas = 0; iKelas < kelas.length; iKelas++) {
139                     x[iDosen][iHari][iSesi][iMatkul][iKelas] = problem.createEnumIntVar(
140                         name: "x[" + iDosen + "]" +
141                             "[" + iHari + "]" +
142                             "[" + iSesi + "]" +
143                             "[" + iMatkul + "]" +
144                             "[" + iKelas + "]", min: 0, max: 1);
145                 }
146             }
147         }
148     }
149 }

```

Gambar 4.7 Kode Program Deklarasi Variabel

Berikut merupakan gambar program untuk menentukan ketetapan nilai pada *constraint*. Dapat dilihat pada gambar bahwa relasi domain yang bernilai satu, dijadikan sebagai calon solusi pada pencarian solusi. Dan nol merupakan relasi domain yang tidak berkaitan, dan tidak dapat dijadikan sebagai solusi.



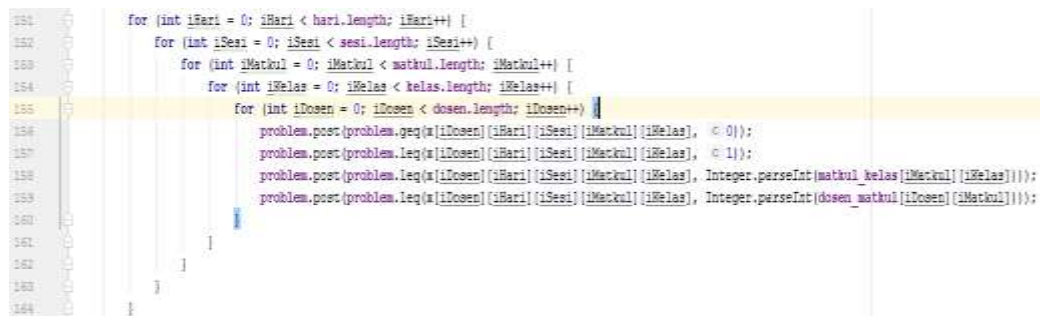
```

156 problem.post(problem.geq(x[iDosen][iHari][iSesi][iMatkul][iKelas], 0));
157 problem.post(problem.leq(x[iDosen][iHari][iSesi][iMatkul][iKelas], 1));

```

Gambar 4.8 Kode Program Ketetapan Nilai Pada *Constraint*

dijelaskan pada gambar dibawah ini bahwa relasi matkul\_kelas dan dosen\_matkul menggunakan nilai 1 untuk menandakan keterkaitan pada relasi domain tersebut.



```

155 for (int iDosen = 0; iDosen < dosen.length; iDosen++) {
156     for (int iHari = 0; iHari < hari.length; iHari++) {
157         for (int iSesi = 0; iSesi < sesi.length; iSesi++) {
158             for (int iMatkul = 0; iMatkul < matkul.length; iMatkul++) {
159                 for (int iKelas = 0; iKelas < kelas.length; iKelas++) {
160                     problem.post(problem.geq(x[iDosen][iHari][iSesi][iMatkul][iKelas], 0));
161                     problem.post(problem.leq(x[iDosen][iHari][iSesi][iMatkul][iKelas], 1));
162                     problem.post(problem.leq(x[iDosen][iHari][iSesi][iMatkul][iKelas], Integer.parseInt(matkul_kelas[iMatkul][iKelas])));
163                     problem.post(problem.leq(x[iDosen][iHari][iSesi][iMatkul][iKelas], Integer.parseInt(dosen_matkul[iDosen][iMatkul])));
164                 }
165             }
166         }
167     }
168 }

```

Gambar 4.9 Kode Program Relasi Domain

Berikut merupakan implementasi *constraint* dimana satu dosen tidak dapat mengajar dua mata kuliah, tidak dapat mengajar dua kelas dan dua mata kuliah dalam waktu yang sama. *Constraint* tersebut dimodelkan dalam bentuk  $\sum MKX[Dd][M][K][Ss][Hh] \leq 1$ . Begitu juga dengan *constraint* dimana dosen

dipisahkan berdasarkan tugas mata kuliah yang diajarkan dan kelas yang diajar. *constraint* tersebut dimodelkan dalam bentuk  $\sum shX[Dd][Mm][Kk][S][H] \leq 1$ . Pemodelan tersebut dikaitkan oleh variabel solusi, yaitu X. Pada program dibawah ini dapat dilihat bahwa, untuk pencarian solusi dalam permasalahan ini akan menggunakan relasi domain yang bernilai 1 untuk pencarian solusinya.

```

186      IntExp plus1;
187      for (int iDosen=0; iDosen < dosen.length; iDosen++) {
188          for (int iHari=0; iHari < hari.length; iHari++) {
189              for (int iSesi=0; iSesi < sesi.length; iSesi++) {
190                  plus1 = null;
191
192                  for (int iMatkul=0; iMatkul < matkul.length; iMatkul++) {
193                      for (int iKelas=0; iKelas < kelas.length; iKelas++) {
194                          if (plus1 == null)
195                              plus1 = x[iDosen][iHari][iSesi][iMatkul][iKelas];
196                          else
197                              plus1 = problem.plus(plus1, x[iDosen][iHari][iSesi][iMatkul][iKelas]);
198                      }
199                  }
200
201                  problem.post(problem.leq(plus1, C 1));
202              }
203          }
204      }

```

**Gambar 4.10 Kode Program Constraint Pertama**

Berikut merupakan implementasi *constraint* dimana setiap mata kuliah yang berbeda tidak boleh berada pada kelas dan waktu yang sama. *Constraint* ini dimodelkan dalam bentuk  $\sum d X[D][Mm][Kk][Ss][Hh] \leq 1$ . *Constraint* yang tidak dapat dilanggar untuk menghindari kesalahan pada penjadwalan. Pemodelan tersebut dikaitkan oleh variabel solusi, yaitu X. Pemodelan *constraint* tersebut dapat dilihat pada program dibawah ini. Pada program ini dapat dilihat bahwa, untuk pencarian solusi dalam permasalahan ini akan menggunakan relasi yang bernilai 1 untuk pencarian solusinya. *Constraint* berikut ini telah berkaitan dengan program plus1, sehingga gabungan dari implementasi *constraint* pertama dan kedua telah dapat menyelesaikan permasalahan mata kuliah berbeda tidak boleh berada pada kelas dan waktu yang sama.

```

206      IntExp plus2;
207      for (int iHari=0; iHari < hari.length; iHari++) {
208          for (int iSesi=0; iSesi < sesi.length; iSesi++) {
209              for (int iKelas=0; iKelas < kelas.length; iKelas++) {
210                  plus2 = null;
211
212                  for (int iDosen=0; iDosen < dosen.length; iDosen++) {
213                      for (int iMatkul=0; iMatkul < matkul.length; iMatkul++) {
214                          if (plus2 == null)
215                              plus2 = x[iDosen][iHari][iSesi][iMatkul][iKelas];
216                          else
217                              plus2 = problem.plus(plus2, x[iDosen][iHari][iSesi][iMatkul][iKelas]);
218                      }
219                  }
220
221                  problem.post(problem.leq(plus2, C 1));
222              }
223          }
224      }

```

Gambar 4.11 Kode Program Constraint Kedua

Setiap mata kuliah memiliki SKS yang harus dipenuhi dan diatur pada jadwal setiap minggunya. *Constraint* ini dimodelkan seperti berikut ini  $\sum dksh X[D][Mm][K][S][H] = C[m]$ . SKS dimiliki oleh mata kuliah sehingga variabel SKS (C) harus sama dengan variabel solusi X untuk memenuhi *constraint* yang ada. Jumlah pertemuan untuk proses pembelajaran yang diatur pada jadwal dan terkait pada dosen, hari, sesi, dan kelas harus sesuai dengan SKS yang telah ditetapkan. Pada program dibawah ini diperlihatkan mata kuliah akan di jadwalkan sebanyak jumlah SKS yang dimiliki. Jumlah SKS merupakan ketentuan jumlah sesi pertemuan dalam perkuliahan. Satu SKS menandakan satu sesi pertemuan.

```

166      IntExp plus;
167      for (int iMatkul=0; iMatkul < matkul.length; iMatkul++) {
168          plus = null;
169
170          for (int iDosen=0; iDosen < dosen.length; iDosen++) {
171              for (int iHari=0; iHari < hari.length; iHari++) {
172                  for (int iSesi=0; iSesi < sesi.length; iSesi++) {
173                      for (int iKelas = 0; iKelas < kelas.length; iKelas++) {
174                          if (plus == null)
175                              plus = x[iDosen][iHari][iSesi][iMatkul][iKelas];
176                          else
177                              plus = problem.plus(plus, x[iDosen][iHari][iSesi][iMatkul][iKelas]);
178                      }
179                  }
180              }
181          }
182
183          problem.post(problem.eq(plus, Integer.parseInt(sks[iMatkul])));
184      }

```

Gambar 4.12 Kode Program Constraint Ketiga

*Library choco solver* telah memudahkan pengguna dalam penggunaannya, *choco solver* menggunakan Algoritma *Backtracking* dalam pencarian solusi. Setelah *constraint* telah dibuat dalam program, dengan menggunakan nilai yang telah ditetapkan berdasarkan relasi antar domain. *Choco solver* akan mencari solusi dengan memperhatikan solusi yang memungkinkan saja, yaitu relasi yang bernilai satu. Berikut merupakan kode program yang memperlihatkan penggunaan *method "problem.solve()",* dalam pencarian solusi.

```

226     try {
227         problem.solve();
228     }
229     catch (SolverException ce) {
230         System.out.println("Error [SolverException]: " + ce);
231         System.exit(1);
232     }

```

**Gambar 4.13 Kode Program Problem Solver**

Pencarian solusi yang dihasilkan akan ditampilkan dalam bentuk susunan jadwal dengan urutan dosen, hari, sesi, mata kuliah, dan kelas. Berikut merupakan program yang memperlihatkan pencetakan jadwal apabila solusi dinyatakan benar dan merupakan solusi yang diperoleh dari relasi yang bernilai satu. Diperlihatkan juga, bahwa hasil jadwal akan disimpan ke tabel jadwal pada *database*. Penyimpanan jadwal ini ditujukan untuk pemakaian Algoritma Dijkstra untuk pencarian jalur terpendek.

```

233     for (int iDosen = 0; iDosen < dosen.length; iDosen++) {
234         for (int iHari = 0; iHari < hari.length; iHari++) {
235             for (int iSesi = 0; iSesi < sesi.length; iSesi++) {
236                 for (int iMatakul = 0; iMatakul < matakul.length; iMatakul++) {
237                     for (int iKelas = 0; iKelas < kelas.length; iKelas++) {
238                         if (x[iDosen][iHari][iSesi][iMatakul][iKelas].toString().substring(x[iDosen][iHari][iSesi][iMatakul][iKelas].toString().length() - 1).equals("1")) {
239                             System.out.println(
240                                 "x[" + dosen[iDosen] + "]" +
241                                 "[" + hari[iHari] + "]" +
242                                 "[" + sesi[iSesi] + "]" +
243                                 "[" + matakul[iMatakul] + "]" +
244                                 "[" + kelas[iKelas] + "]" + " = " +
245                                 x[iDosen][iHari][iSesi][iMatakul][iKelas]);
246
247                             String sql = "INSERT INTO jadwal (no_dosen, no_hari, no_sesi, no_matakul, no_kelas) "
248                                 + "VALUES (" + iDosen + ", " + iHari + ", " + iSesi + ", " + iMatakul + ", " + iKelas + ")";
249                             DBHelper.insertDB(sql);
250                         }
251                     }
252                 }
253             }
254         }
255     }

```

**Gambar 4.14 Kode Program Penyimpanan Jadwal**

#### 4.2.5.3. Algoritma Dijkstra

Algoritma Dijkstra digunakan untuk menentukan jarak terdekat antar kelas yang harus ditempuh pada jadwal yang memiliki jam beruntun pada satu hari tertentu. Hal ini bertujuan untuk meminimalisasi penggunaan waktu pada perpindahan. Pada halaman data jadwal tersedia tabel data jadwal yang telah memenuhi hari, sesi, mata kuliah, dosen dan kelas hasil dari pencarian Algoritma *Backtracking*. Untuk proses selanjutnya yaitu dengan melakukan *generate* dijkstra. Berikut merupakan potongan kode dijkstra,

```
private void generateDijkstra(int[][] matrix, int source) {
    int node;
    for (int i = 1; i <= nodes; i++)
        for (int j = 1; j <= nodes; j++)
            arrayMatrix[i][j] = matrix[i][j];

    for (int i = 1; i <= nodes; i++)
    {
        distances[i] = Integer.MAX_VALUE;
    }

    unsettled.add(source);
    distances[source] = 0;
    while (!unsettled.isEmpty())
    {
        node = getMinDisFromUnsettled();
        unsettled.remove(node);
        settled.add(node);
        getMinNode(node);
    }
}
```

**Gambar 4.15 Kode Program Dijkstra**

Potongan *code* diatas berfungsi untuk menyimpan data jarak antar ruangan, penelusuran dari ruangan asal ke ruangan lainnya, untuk menemukan jarak terdekat dengan ruangan lainnya.

```

private int getMinDisFromUnsettled() {
    int min ;
    int node;

    Iterator<Integer> iterator = unsettled.iterator();
    node = iterator.next();
    min = distances[node];
    for (int i = 1; i <= distances.length; i++)
    {
        if (unsettled.contains(i))
        {
            if (distances[i] <= min)
            {
                min = distances[i];
                node = i;
            }
        }
    }
    return node;
}

```

Gambar 4.16 Kode Program Dijkstra

Potongan *code* diatas berfungsi untuk mengambil jarak terkecil dari semua ruangan yang telah dibandingkan jaraknya.

### 4.3. Testing

Pada bab ini akan dibahas mengenai pengujian yang dilakukan terhadap *constraints*, yang berguna untuk memastikan bahwa tidak ada *constraint* yang dilanggar setelah jadwal selesai di-*generate*.

#### 4.3.1. Metode Test

Pengujian menggunakan metode *blackbox*. Metode *blackbox* merupakan, metode yang melakukan pengujian aplikasi tanpa melihat isi program secara langsung. Pengujian ini dilakukan setelah aplikasi selesai dibangun. Pengujian *blackbox* bertujuan untuk memastikan bahwa aplikasi yang dapat dilihat oleh *user* dapat berjalan dengan baik.

#### 4.3.2. Tujuan Pengujian

1. Pengujian dilakukan untuk memastikan bahwa tidak ada error pada aplikasi.

2. Memastikan bahwa aplikasi berhasil menghasilkan jadwal yang tidak melanggar batasan-batasan yang ada.

#### 4.3.3. Test Plan

Subbab *Test plan* merupakan subbab yang menjelaskan bagaimana *Test* dilakukan dan perencanaan *Test* berdasarkan scenario pengujian dengan tujuan menjelaskan alur pengujian, hingga hasil akhir yang seharusnya diperoleh

##### 4.3.3.1. Test Plan Pengujian Mengelola Data Kelas

Data kelas merupakan data dari kelompok mahasiswa yang dikelompokkan berdasarkan program studi dan angkutannya berdasarkan jumlah yang memenuhi untuk satu kelas. *Test plan* mengelola data kelas, bertujuan untuk merancang perencanaan *Test* yang bertujuan untuk memastikan proses pengelolaan data kelas, telah berjalan sesuai dengan yang diharapkan.

Tabel 4.3 Test Plan Pengujian Mengelola Data Kelas

<b>Nama Kasus Uji</b>	Mengelola data Kelas		
<b>Tujuan</b>	Menguji aplikasi apakah data dapat ditambahkan, diubah atau dihapus		
<b>Deskripsi</b>	Fungsi ini akan digunakan untuk mengelola data sumber daya kelas yang digunakan untuk melakukan <i>generate</i> jadwal.		
<b>Kondisi Awal</b>	<i>User</i> telah masuk pada halaman utama aplikasi		
<b>Skenario Uji</b>			
1. <i>User</i> memilih menu kelola kelas 2. <i>User</i> menambahkan data 3. <i>User</i> mengubah data 4. <i>User</i> menghapus data			
<b>Kriteria Evaluasi Hasil</b>			
<i>User</i> berhasil melakukan penambahan, perubahan dan penghapusan data.			
<b>Kasus dan Hasil Uji (Data Normal)</b>			
<b>Data masukan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
<i>User</i> menambahkan nama dan jumlah	Daftar data kelas akan bertambah dengan data yang telah diinput	Sesuai dengan yang diharapkan	Diterima
<i>User</i> mengubah nama dan jumlah	Daftar data kelas yang dipilih untuk diubah dapat berubah.	Sesuai dengan yang diharapkan	Diterima



<i>User</i> menghapus nama dan jumlah	Daftar data kelas yang dipilih untuk dihapus, berhasil dihapus	Sesuai dengan yang diharapkan	Diterima
<b>Kasus dan Hasil Uji Coba (Data Tidak Normal)</b>			
<i>User</i> menambahkan data tanpa mengisi nama dan jumlah	Tetap berada pada halaman kelola data kelas dan memberikan notifikasi bahwa data harus diisi	Sesuai yang diharapkan	Diterima
<i>User</i> mengubah data dengan cara mengosongkan salah satu inputan	Tetap berada pada halaman kelola data kelas dan memberikan notifikasi bahwa data harus diisi	Sesuai dengan yang diharapkan	Diterima
<b>Catatan</b>			

#### 4.3.3.2. Test Plan Pengujian Mengelola Data Dosen

Data dosen merupakan kumpulan data dosen yang terdiri atas inisial dosen dan jumlah. *Test plan* pengujian mengelola data dosen merupakan pengujian yang bertujuan untuk memastikan bahwa bahwa proses pengelolaan data dosen, baik untuk menambah, mengubah dan menghapus telah berjalan dengan baik.

Tabel 4.4 Test Plan Pengujian Mengelola Data Dosen

<b>Nama Kasus Uji</b>	Mengelola Data Dosen
<b>Tujuan</b>	Menguji aplikasi apakah data dosen dapat ditambahkan, diubah atau dihapus
<b>Deskripsi</b>	Fungsi ini akan digunakan untuk mengelola data sumber daya dosen yang digunakan untuk melakukan <i>generate</i> jadwal.
<b>Kondisi Awal</b>	<i>User</i> telah masuk pada halaman utama aplikasi
<b>Skenario Uji</b>	
1. <i>User</i> memilih menu kelola dosen 2. <i>User</i> menambahkan data dosen 3. <i>User</i> mengubah data dosen 4. <i>User</i> menghapus data dosen	
<b>Kriteria Evaluasi Hasil</b>	
<i>User</i> berhasil melakukan penambahan, perubahan dan penghapusan data.	
<b>Kasus dan Hasil Uji (Data Normal)</b>	
<b>Data masukan</b>	<b>Yang diharapkan Pengamatan Kesimpulan</b>

<i>User</i> menambahkan nama dan jumlah	Daftar data kelas akan bertambah dengan data yang telah diinput	Sesuai yang diharapkan	Diterima
<i>User</i> mengubah nama dan jumlah	Daftar data kelas yang dipilih untuk diubah dapat berubah.	Sesuai dengan yang diharapkan	Diterima
<i>User</i> menghapus nama dan jumlah	Daftar data kelas yang dipilih untuk dihapus berhasil dihapus	Sesuai yang diharapkan	Diterima
<b>Kasus dan Hasil Uji Coba (Data Tidak Normal)</b>			
<i>User</i> menambahkan data dosen tanpa menginput nama dosen	Tetap berada di halaman kelola data dosen dan memberikan notifikasi bahwa data harus diisi	Sesuai dengan yang diharapkan	Diterima
<i>User</i> menambahkan data dosen dengan inputan yang sama dengan data dosen yang sudah ada	Tetap berada di halaman kelola data dosen dan memberikan notifikasi bahwa data dosen tidak boleh sama	Sesuai dengan yang diharapkan	Diterima
<i>User</i> mengubah data dosen dengan mengosongkan inputan	Tetap berada di halaman kelola data dosen dan memberikan notifikasi bahwa data harus diisi	Sesuai dengan yang diharapkan	Diterima
<b>Catatan</b>			

#### 4.3.3.3. Test plan Pengujian Mengelola Data Mata Kuliah

*Test plan* pengujian mata kuliah bertujuan untuk perancangan pengujian terkait pengelolaan data matakuliah yang terdiri atas nama matakuliah, jumlah SKS dan jumlah mahasiswa yang mengambil mata kuliah tersebut.

Tabel 4.5 Test Plan Pengujian Mengelola Data Mata Kuliah

<b>Nama Kasus Uji</b>	Mengelola Data Mata kuliah		
<b>Tujuan</b>	Menguji aplikasi apakah data matakuliah dapat ditambahkan, diubah atau dihapus		
<b>Deskripsi</b>	Fungsi ini akan digunakan untuk mengelola data sumber daya mata kuliah yang digunakan untuk melakukan <i>generate</i> jadwal.		
<b>Kondisi Awal</b>	<i>User</i> telah masuk pada halaman utama aplikasi		
<b>Skenario Uji</b>			
1. <i>User</i> memilih menu kelola mata kuliah 2. <i>User</i> menambahkan data mata kuliah 3. <i>User</i> mengubah data mata kuliah 4. <i>User</i> menghapus data mata kuliah			
<b>Kriteria Evaluasi Hasil</b>			
<i>User</i> berhasil melakukan penambahan, perubahan dan penghapusan data.			
<b>Kasus dan Hasil Uji (Data Normal)</b>			
<b>Data masukan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
<i>User</i> menambahkan nama, sks dan jumlah, serta kategori dari mata kuliah	Daftar data kelas akan bertambah dengan data yang telah diinput	Sesuai dengan yang diharapkan	Diterima
<i>User</i> mengubah nama, sks dan jumlah, serta kategori dari mata kuliah	Daftar data kelas yang dipilih untuk diubah dapat berubah.	Sesuai yang diharapkan	Diterima
<i>User</i> menghapus nama, sks dan jumlah, serta kategori dari mata kuliah	Daftar data kelas yang dipilih untuk dihapus berhasil dihapus	Sesuai yang diharapkan	Diterima
<b>Kasus dan Hasil Uji Coba (Data Tidak Normal)</b>			
<i>User</i> menambahkan data mata kuliah tanpa menginput data mata kuliah	Tetap di halaman utama dan memberikan notifikasi bahwa data harus diisi	Sesuai dengan yang diharapkan	Diterima
<i>User</i> menambahkan data mata kuliah dengan menginput data mata kuliah yang sudah ada	Tetap di halaman utama dan memberikan notifikasi bahwa data harus diisi	Sesuai dengan yang diharapkan	Diterima

User mengubah data mata kuliah dengan menginput data mata kuliah yang sama	Tetap berada pada halaman utama dan memberikan notifikasi bahwa data dosen tidak boleh sama	Sesuai dengan yang diharapkan	Diterima
<b>Catatan</b>			

#### 4.3.3.4. Test Plan Pengujian Mengelola Ruangan

*Test plan* pengujian mengelola data ruangan yang terdiri atas data nama ruangan, kapasitas dan jenis ruangan. *Test plan* ini bertujuan untuk merancang proses pengujian guna memastikan bahwa pengelolaan ruangan telah sesuai dan berjalan dengan baik.

Tabel 4.6 Test Plan Pengujian Mengelola Ruangan

<b>Nama Kasus Uji</b>	Mengelola Data Ruangan		
<b>Tujuan</b>	Menguji aplikasi apakah data ruangan dapat ditambahkan, diubah atau dihapus		
<b>Deskripsi</b>	Fungsi ini akan digunakan untuk mengelola data sumber daya ruangan yang digunakan untuk melakukan <i>generate</i> jadwal.		
<b>Kondisi Awal</b>	<i>User</i> telah masuk pada halaman utama aplikasi		
<b>Skenario Uji</b>			
1. <i>User</i> memilih menu kelola ruangan 2. <i>User</i> menambahkan data ruangan 3. <i>User</i> mengubah data ruangan 4. <i>User</i> menghapus data ruangan			
<b>Kriteria Evaluasi Hasil</b>			
<i>User</i> berhasil melakukan penambahan, perubahan dan penghapusan data ruangan.			
<b>Kasus dan Hasil Uji (Data Normal)</b>			
<b>Data masukan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
<i>User</i> menambahkan nama, kapasitas dan kategori ruangan	Daftar data ruangan akan bertambah dengan data yang telah diinput	Sesuai dengan yang diharapkan	Diterima
<i>User</i> mengubah data dosen dengan nama, kapasitas dan jenis ruangan	Daftar data ruangan yang dipilih untuk diubah dapat berubah.	Sesuai dengan yang diharapkan	Diterima

<i>User</i> menambahkan nama, kapasitas dan jenis ruangan	Daftar data ruangan yang dipilih untuk dihapus berhasil dihapus	Sesuai dengan yang diharapkan	Diterima
<b>Kasus dan Hasil Uji Coba (Data Tidak Normal)</b>			
<i>User</i> menambahkan ruangan tanpa nama, kapasitas dan jenis ruangan	Tetap di halaman kelola ruangan dan memberikan notifikasi bahwa data harus diisi	Sesuai dengan yang diharapkan	Diterima
<i>User</i> mengubah data ruangan namun dengan mengosongkan salah satu inputan	Tetap di halaman kelola ruangan dan memberikan notifikasi bahwa data harus diisi	Sesuai dengan yang diharapkan	Diterima
<b>Catatan</b>			

#### 4.3.3.5. Test Plan Pengujian *Generate Jadwal*

*Test plan* pengujian *generate* jadwal bertujuan untuk membuat perancangan pengujian terkait penambahan, pengubahan dan penghapusan data dosen, matakuliah dan kelas, serta penghapusan data relasi variabel yang telah terbentuk, diantaranya matakuliah dan kelas, dosen dan matakuliah dan menghapus keseluruhan relasi. Hal ini dilakukan apabila terdapat relasi yang salah saat pengelolaan masing-masing data jadwal yang diperlukan.

Tabel 4.7 Test Plan Pengujian *Generate Jadwal*

<b>Nama Kasus Uji</b>	<i>Generate Jadwal</i>
<b>Tujuan</b>	Menguji aplikasi apakah data untuk <i>generate</i> jadwal dapat ditambahkan, diubah atau dihapus
<b>Deskripsi</b>	Fungsi ini akan digunakan untuk mengelola data sumber daya yang telah dimuat pada satu daftar jadwal yang terdiri atas data dosen, matakuliah dan kelas, yang digunakan untuk melakukan <i>generate</i> jadwal.
<b>Kondisi Awal</b>	<i>User</i> telah masuk pada halaman utama aplikasi
<b>Skenario Uji</b>	
<ol style="list-style-type: none"> <li>1. <i>User</i> memilih menu kelola <i>generate</i> jadwal</li> <li>2. <i>User</i> menambahkan data dosen, matakuliah, kelas</li> <li>3. <i>User</i> mengubah data dosen, matakuliah, kelas</li> <li>4. <i>User</i> menghapus data dosen, matakuliah, kelas</li> </ol>	
<b>Kriteria Evaluasi Hasil</b>	

<i>User</i> berhasil melakukan penambahan, perubahan dan penghapusan data relasi antar domain.			
<b>Kasus dan Hasil Uji (Data Normal)</b>			
<b>Data masukan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
<i>User</i> menambahkan data dosen, matakuliah dan kelas.	Daftar data kelas akan bertambah dengan data yang telah diinput	Sesuai dengan yang diharapkan	Diterima
<i>User</i> mengubah dosen, matakuliah dan kelas.	Daftar data kelas yang dipilih untuk diubah dapat berubah.	Sesuai dengan yang diharapkan	Diterima
<i>User</i> menghapus relasi matakuliah dan kelas.	relasi mata kuliah dan kelas yang dipilih untuk dihapus berhasil dihapus	Sesuai dengan yang diharapkan	Diterima
<i>User</i> menghapus relasi dosen dan mata kuliah	relasi dosen dan mata kuliah yang dipilih untuk dihapus berhasil dihapus	Sesuai dengan yang diharapkan	Diterima
<i>User</i> menghapus semua relasi	Relasi dosen, matakuliah dan kelas yang telah dipilih akan dihapus	Sesuai dengan yang diharapkan	Diterima
<i>User</i> melakukan <i>generate</i> jadwal	Menghasilkan data jadwal berdasarkan data jadwal yang sudah diinput	Sesuai dengan yang diharapkan	Diterima
<b>Kasus dan Hasil Uji Coba (Data Tidak Normal)</b>			
<i>User</i> menambahkan data jadwal tanpa mengisi inputan	Tetap di halaman <i>generate</i> jadwal dan memberikan notifikasi bahwa data harus diisi	Sesuai dengan yang diharapkan	Diterima
<i>User</i> mengubah data dosen , mata kuliah dan kelas dengan data yang sama dengan data yang sudah ada	Tetap dihalaman <i>generate</i> jadwal dan memberikan notifikasi bahwa data telah tersedia	Sesuai dengan yang diharapkan	Diterima
<i>User</i> menambah data dosen , mata	Tetap dihalaman <i>generate</i> jadwal	Sesuai dengan yang diharapkan	Diterima

kuliah dan kelas dengan data yang sama dengan data yang sudah ada	dan memberikan notifikasi bahwa data telah tersedia		
<b>Catatan</b>			

Berikut merupakan pengujian yang dilakukan untuk setiap *constraint* yang ada:

Tabel 4.8 Pengujian Constraint

No	<i>Constraint</i>	Keterangan
1	satu dosen yang mengajar dua mata kuliah, tidak dapat mengajar dua kelas dan dua mata kuliah dalam waktu yang sama.	Diterima
2	Memisahkan dosen berdasarkan tugas mata kuliah yang diajarkan dan kelas yang diajar	Diterima
3	Setiap mata kuliah yang berbeda tidak boleh berada pada kelas dan waktu yang sama.	Diterima
4	Kapasitas ruangan tidak diperbolehkan lebih kecil dibandingkan jumlah mahasiswa yang mengambil mata kuliah tersebut.	Diterima
5	Setiap mata kuliah harus dijadwalkan sesuai dengan banyaknya jumlah SKS	Diterima
6	Sesi praktikum dapat dilakukan apabila sudah terlebih dahulu melakukan sesi teori.	Diterima
7	Jumlah kelas yang memiliki jam perkuliahan tidak boleh melebihi jumlah ruangan yang tersedia	Diterima

## BAB 5

### Hasil dan Pembahasan

Pada bab ini dijelaskan mengenai hasil yang didapatkan dan pembahasan aplikasi yang telah dibangun.

#### 5.1. Hasil Jadwal

Pada subbab ini dijelaskan mengenai hasil jadwal yang dihasilkan menggunakan metode *Constraint Catisfaction Problem* dan Algoritma *Backtracking* dalam proses pencarian solusi untuk membentuk sebuah jadwal. Proses *generate* jadwal dilakukan perfakultas secara bertahap tanpa memasukkan ruangan. Dikarenakan jumlah data yang diproses mencapai 21.638.400 data, mengakibatkan tidak mampunya memori perangkat keras yang digunakan untuk menyimpan data dalam proses pengerjaan

##### 5.1.1 Hasil Penjadwalan FTB (Fakultas Teknik Bioproses)

Dengan menggunakan *Constraint Satisfaction Problem*, telah dibentuk *constraints* yang berguna untuk membatasi tugas dan fungsi dari masing-masing sumber daya yang ada. Hasil jadwal dibagi atas hari, sesi, kelas, mata kuliah, dosen, kategori dan ruangan. Berikut merupakan keluaran jadwal dari aplikasi penjadwalan pada FTB (Fakultas Teknik Bioproses):



Hari	Sesi	Kelas	Mata Kuliah	Dosen	Kategori	Ruangan
SENIN	SESI 1	14TB	AITB	EAN	Teori	GD912
SENIN	SESI 1	12TB	BIOKIM	RFK	Teori	GD913
SENIN	SESI 1	11TB	FISDAS 1	RZS	Teori	GD914
SENIN	SESI 2	14TB	AITB	EAN	Teori	GD912
SENIN	SESI 2	12TB	BIOKIM	RFK	Teori	GD913
SENIN	SESI 2	11TB	FISDAS 1	RZS	Teori	GD914
SENIN	SESI 3	14TB	PEN I	RFK	Teori	GD912
SENIN	SESI 3	12TB	MIKRUM	EAN	Teori	GD913
SENIN	SESI 3	11TB	PRAK FISDAS 1	RZS	Praktikum	LAB FISIKA
SENIN	SESI 4	14TB	PEN I	RFK	Teori	GD913
SENIN	SESI 4	12TB	MIKRUM	EAN	Teori	GD914

Total Data : 23

Gambar 5.1 Hasil Generate Jadwal



Hasil *generate* jadwal diatas, merupakan hasil akhir yang dimiliki oleh FTB. Jadwal yang diperoleh merupakan hasil *generate* dari relasi satu mata kuliah hanya diajar oleh satu dosen saja. Proses yang dilalui untuk menciptakan jadwal FTB yaitu dengan melakukan set domain, membangun relasi antar domain, dan menjalankan *constraint* yang ada. Algoritma *Backtracking* yang berfungsi untuk menemukan solusi terbaik dari semua calon solusi dengan mempertimbangkan *constraint-constraint* yang ada. Berikut ini merupakan hasil ekspor *generate* hasil jadwal oleh aplikasi ke dalam file excel,

	A	B	C	D	E	F	G	H
1								
2		Hari	Sesi	Mata Kuliah	Dosen	Kelas	Ruangan	
3		SENIN	SESI 1	AITB	EAN	14TB	GD913	
4		SENIN	SESI 1	BIOKIM	RFK	12TB	GD914	
5		SENIN	SESI 1	FISDAS 1	RZS	11TB	GD915	
6		SENIN	SESI 2	AITB	EAN	14TB	GD913	
7		SENIN	SESI 2	BIOKIM	RFK	12TB	GD914	
8		SENIN	SESI 2	FISDAS 1	RZS	11TB	GD915	
9		SENIN	SESI 3	PEN I	RFK	14TB	GD923	
10		SENIN	SESI 3	MIKRUM	EAN	12TB	GD924	
11		SENIN	SESI 3	PRAK FISDAS 1	RZS	11TB	LAB FISIKA	
12		SENIN	SESI 4	PEN I	RFK	14TB	GD913	
13		SENIN	SESI 4	MIKRUM	EAN	12TB	GD914	
14		SENIN	SESI 4	PRAK FISDAS 1	RZS	11TB	LAB FISIKA	
15		SENIN	SESI 5	AKRB	YUL	13TB	GD913	
16		SENIN	SESI 5	PRAK KIMOR	RFK	12TB	GD826	
17		SENIN	SESI 5	KIMDAS	SFA	11TB	GD914	
18		SENIN	SESI 6	AKRB	YUL	13TB	GD913	
19		SENIN	SESI 6	PRAK KIMOR	RFK	12TB	GD826	
20		SENIN	SESI 6	KIMDAS	SFA	11TB	GD914	
21		SENIN	SESI 7	FENOPOR	MNP	13TB	GD913	
22		SENIN	SESI 7	PRAK MIKRUM	EAN	12TB	GD825	
23		SENIN	SESI 7	PRAK KIMDAS	SFA	11TB	GD812	
24		SENIN	SESI 8	SPP	DDA	14TB	GD923	
25		SENIN	SESI 8	FENOPOR	MNP	13TB	GD924	
26		SELASA	SESI 1	AKRB	YUL	13TB	GD913	
27		SELASA	SESI 1	BIOKIM	RFK	12TB	GD914	

**Gambar 5.2 Hasil Export Jadwal TB pada File Excel**

Pada gambar diatas dapat dilihat bahwa, jadwal yang dihasilkan tidak mengalami pelanggaran pada ketentuan *constraint* yang ada. Sumber daya yang dimiliki, seperti dosen, kelas, mata kuliah dan ruangan tidak mengalami bentrok. Dari hasil pengujian *constraint* yang telah dilakukan, diperoleh bahwa ketujuh *constraint* yang ada, telah dapat terpenuhi. Untuk FTB, ruangan praktikum memiliki pengalokasian tersendiri. Oleh karena itu, mata kuliah yang memiliki ruangan tersendiri, secara otomatis akan terisi tanpa harus di-*set*. Ruangan

praktikum tidak dapat digunakan untuk jam teori. Jumlah data yang dimiliki oleh FTB untuk diproses sejumlah 114.240 dengan waktu *generate* selama 1 menit.

### 5.1.2 Hasil Penjadwalan FTI (Fakultas Teknik Industri)

Fakultas Teknik Industri memiliki data sebanyak 76.000. Jadwal FTI (Fakultas Teknik Industri) berhasil di-*generate* dengan hasil jadwal yang dimiliki, memperlihatkan bahwa jadwal mampu memenuhi keseluruhan *constraint* yang ada, dengan kasus setiap mata kuliah memiliki satu dosen pengampu. Waktu *generate* jadwal untuk Fakultas Teknik Industri yaitu selama satu menit. Berikut merupakan halaman aplikasi yang memperlihatkan hasil *generate* jadwal,



Hari	Sesi	Kelas	Mata Kuliah	Dosen	Kategori	Ruangan
SENIN	SESI 1	13MR	PPBO	ANA	Teori	GD913
SENIN	SESI 1	12MR	ENG 3	SAM	Teori	GD914
SENIN	SESI 1	11MR	FISDAS 1	AFS	Teori	GD915
SENIN	SESI 2	13MR	PPBO	ANA	Teori	GD913
SENIN	SESI 2	12MR	ENG 3	SAM	Teori	GD914
SENIN	SESI 2	11MR	FISDAS 1	AFS	Teori	GD915
SENIN	SESI 3	14MR	MANSTRA	SAM	Teori	GD913
SENIN	SESI 3	12MR	PENGPRO	ASD	Teori	GD914
SENIN	SESI 3	11MR	PRAK FISDAS 1	AFS	Praktikum	LAB FISIKA
SENIN	SESI 4	14MR	MANSTRA	SAM	Teori	GD913
SENIN	SESI 4	12MR	PENGPRO	ASD	Teori	GD914

Total Data : 23

EXPORT KE EXCEL

**Gambar 5.3 Tampilan Hasil Generate Jadwal FTI**

Setelah jadwal telah berhasil di-*generate* yang disertai dengan ruangan. Jadwal dapat diexport ke dalam bentuk file excel. Berikut merupakan gambar dari hasil *export* jadwal ke dalam bentuk file excel,

	A	B	C	D	E	F	G
1							
2		Hari	Sesi	Mata Kuliah	Dosen	Kelas	Ruangan
3		SENIN	SESI 1	PPBO	ANA	13MR	GD913
4		SENIN	SESI 1	ENG 3	SAM	12MR	GD914
5		SENIN	SESI 1	FISDAS 1	AFS	11MR	GD915
6		SENIN	SESI 2	PPBO	ANA	13MR	GD913
7		SENIN	SESI 2	ENG 3	SAM	12MR	GD914
8		SENIN	SESI 2	FISDAS 1	AFS	11MR	GD915
9		SENIN	SESI 3	MANSTRA	SAM	14MR	GD913
10		SENIN	SESI 3	PENGPRO	ASD	12MR	GD914
11		SENIN	SESI 3	PRAK FISDAS 1	AFS	11MR	LAB FISIKA
12		SENIN	SESI 4	MANSTRA	SAM	14MR	GD913
13		SENIN	SESI 4	PENGPRO	ASD	12MR	GD914
14		SENIN	SESI 4	PRAK FISDAS 1	AFS	11MR	LAB FISIKA
15		SENIN	SESI 5	KP	ANA	14MR	GD913
16		SENIN	SESI 5	PRAK PENGPRO	ASD	12MR	GD912
17		SENIN	SESI 5	ENG 1	SAM	11MR	GD914
18		SENIN	SESI 6	KP	ANA	14MR	GD923
19		SENIN	SESI 6	PRAK PENGPRO	ASD	12MR	GD916
20		SENIN	SESI 6	ENG 1	SAM	11MR	GD913
21		SENIN	SESI 7	PMB	NSS	13MR	GD913

Gambar 5.4 Hasil Export Jadwal FTI pada File Excel

### 5.1.3 Hasil Penjadwalan (FTIE)Fakultas Teknik Informatika dan Elektro Prodi Diploma

Program studi diploma merupakan program studi yang menerapkan sistem paket untuk pengambilan mata kuliah untuk setiap semester. Program studi diploma untuk FTIE terdiri dari 3 prodi yaitu Diploma 3 Teknik Informatika, Diploma 4 Teknik Informatika, dan Diploma 3 Teknik Komputer. Apabila ketiga prodi tersebut digabungkan, data yang dimiliki sejumlah 1.288.000. Penjadwalan untuk diploma yang terdiri atas tiga program studi, jadwal tidak dapat digabungkan. Hasil analisis yang diperoleh apabila setiap prodi diploma dipisahkan, maka data jadwal untuk Teknik Komputer sejumlah 76.800, Diploma 3 Teknik Informatika sejumlah 83.640, dan Diploma 4 Teknik Informatika sejumlah 123.120. Untuk uji implementasi pada diploma digunakan penjadwalan pada Teknik Komputer, dengan data yang dimiliki jadwal teknik komputer dapat di-generate. Berikut merupakan hasil *generate* oleh aplikasi penjadwalan kuliah untuk program studi teknik komputer,

MAIN MENU

## Aplikasi Jadwal Perkuliahan

SENIN

Hari	Sesi	Kelas	Mata Kuliah	Dosen	Kategori	Ruangan
SENIN	SESI 1	33TK	ADJAR	TLS	Teori	GD913
SENIN	SESI 1	32TK	DACOM	IPM	Teori	GD914
SENIN	SESI 1	31TK	ENG 1	RMM	Teori	GD915
SENIN	SESI 2	33TK	PRAK ADJAR	TLS	Praktikum	GD912
SENIN	SESI 2	32TK	PRAK DACOM	IPM	Praktikum	GD917
SENIN	SESI 2	31TK	PRAK ENG	RMM	Praktikum	GD916
SENIN	SESI 3	33TK	PRAK ADJAR	TLS	Praktikum	GD916
SENIN	SESI 3	32TK	PRAK DACOM	IPM	Praktikum	GD912
SENIN	SESI 3	31TK	PRAK ENG	RMM	Praktikum	GD917
SENIN	SESI 4	33TK	TA 1	ESS	Teori	GD914
SENIN	SESI 4	32TK	ENG 3	MPR	Teori	GD913

Total Data : 21

EXPORT KE EXCEL

**Gambar 5.5 Tampilan Hasil Generate Jadwal Teknik Komputer**

Berdasarkan hasil *generate* jadwal yang diperoleh, jadwal perkuliahan Teknik Komputer dapat memenuhi semua *constraint* dengan setiap matakuliah diampuh oleh satu dosen. Setelah jadwal telah di-*generate* jadwal dapat diexport ke bentuk file excel. Berikut merupakan hasil export dari jadwal perkuliahan Teknik Komputer,

	A	B	C	D	E	F	G
1							
2		Hari	Sesi	Mata Kuliah	Dosen	Kelas	Ruangan
3		SENIN	SESI 1	ADJAR	TLS	33TK	GD913
4		SENIN	SESI 1	DACOM	IPM	32TK	GD914
5		SENIN	SESI 1	ENG 1	RMM	31TK	GD915
6		SENIN	SESI 2	PRAK ADJAR	TLS	33TK	GD912
7		SENIN	SESI 2	PRAK DACOM	IPM	32TK	GD917
8		SENIN	SESI 2	PRAK ENG	RMM	31TK	GD916
9		SENIN	SESI 3	PRAK ADJAR	TLS	33TK	GD916
10		SENIN	SESI 3	PRAK DACOM	IPM	32TK	GD912
11		SENIN	SESI 3	PRAK ENG	RMM	31TK	GD917
12		SENIN	SESI 4	TA 1	ESS	33TK	GD914
13		SENIN	SESI 4	ENG 3	MPR	32TK	GD913
14		SENIN	SESI 4	MATDIS	SML	31TK	GD923
15		SENIN	SESI 5	ANJAR 2	ESS	33TK	GD915
16		SENIN	SESI 5	PRAK ENG 3	MPR	32TK	GD917
17		SENIN	SESI 5	MATDIS	SML	31TK	GD913
18		SENIN	SESI 6	ANJAR 2	ESS	33TK	GD516
19		SENIN	SESI 6	PRAK ENG 3	MPR	32TK	GD525
20		SENIN	SESI 6	PRAK MATDIS	SML	31TK	GD526
21		SENIN	SESI 7	PRAK TA 1	ESS	33TK	GD526
22		SENIN	SESI 7	BASDAT	VES	32TK	GD937
23		SENIN	SESI 7	PRAK MATDIS	SML	31TK	GD711

Gambar 5.6 Hasil Export Jadwal Teknik Komputer pada File Excel

#### 5.1.4 Hasil Penjadwalan Sarjana FTIE

Program studi sarjana pada FTIE terdiri atas 3 program studi yang diantaranya, Sistem Informasi, Teknik Informatika, Teknik Elektro dengan jumlah data yang dimiliki sejumlah 1.893.120 dengan matakuliah teori dan praktikum sebanyak 116 mata kuliah. *Generate* jadwal tidak dapat dilakukan akibat data yang terlalu banyak. Untuk mengetahui batas maksimal data yang dapat di-*generate* dilakukan iterasi untuk pengurangan data. Setelah dilakukannya iterasi pertama dengan melakukan pengurangan matakuliah sebanyak 5, didapatkan hasil data sejumlah 1.811.520. Iterasi kedua dilakukan dengan mengurangi 15 mata kuliah dari jumlah asli mata kuliah, maka diperoleh data sejumlah 1.648.320. Untuk iterasi ketiga dengan mengurangi 35 mata kuliah, diperoleh data sejumlah 1.321.920. Hingga iterasi ketiga *generate* jadwal tidak dapat dilakukan dikarenakan data yang melebihi batas maksimum *generate*. Untuk iterasi terakhir yang dilakukan dapat diperoleh jumlah data yang dapat diproses yaitu 347.760 data dengan jumlah relasi sebanyak 46 dengan waktu *generate* selama 4 menit. Dalam kasus ini penjadwalan sarjana FTIE tidak dapat digabungkan dikarenakan data yang berlebih dalam proses *generate* jadwal.

## 5.2. Hasil Dijkstra

Setelah jadwal selesai di-generate tanpa ruangan menggunakan *Backtracking*. Pencarian ruangan dilakukan menggunakan Algoritma Dijkstra. Berikut merupakan halaman aplikasi untuk melakukan *generate* dijkstra,

**MAIN MENU**

**Aplikasi Jadwal Perkuliahan**

**Data Jadwal**

Pilih Hari:

\*\* Pilih salah satu Kelas pada tabel

Sesi	Kelas	Mata Kuliah	Dosen	Kategori	Ruangan
SESI 1	14TB	AITB	EAN	Teori	GD912
SESI 5	13TB	AKRB	VUL	Teori	Belum di-set
SESI 1	12TB	BIOKIM	RFK	Teori	Belum di-set
SESI 1	11TB	FISDAS 1	RZS	Teori	Belum di-set

Total Data : 4

Gambar 5.7 Tampilan Menu Generate Dijkstra

Proses yang dilalui yaitu dengan terlebih dahulu melakukan set ruangan pada sesi yang memiliki mata kuliah pertama dihari tersebut. *Button* hari digunakan untuk memilih hari yang ingin di set ruangnya untuk setiap matakuliah pertama. Setelah keseluruhan hari selesai di-set, jadwal telah dapat di-generate. Dengan Algoritma Dijkstra, setiap ruangan yang telah di-set akan mencari ruangan terdekat untuk mata kuliah selanjutnya yang akan dimasuki. Sehingga perpindahan ruangan kelas tidak akan terlalu membutuhkan waktu yang terlalu banyak.

Untuk kebutuhan khusus, seperti permintaan ruangan oleh dosen atau kelas kasus ruangan yang mengalami kendala dalam penggunaannya, diberikan alternative penempatan ruangan secara manual, namun tetap memperhatikan *constraint* ruangan agar tidak mengalami bentrok pada ruangan. Berikut merupakan halaman aplikasi untuk pengalokasian ruangan dengan cara set ruangan manual,

**MAIN MENU**

Data Jadwal

Kelas

Ruangan

**SET RUANGAN (Sesi)**

**SET RUANGAN (Hari)**

## Aplikasi Jadwal Perkuliahan

SENIN
SESI 1

\*\* Pilih salah satu Kelas pada tabel

Sesi	Kelas	Mata Kuliah	Dosen	Kategori	Ruangan
SESI 1	14TB	AITB	EAN	Teori	GD912
SESI 1	12TB	BIOKIM	RFK	Teori	Belum di-set
SESI 1	11TB	FISDAS 1	RZS	Teori	Belum di-set

Total Data : 3

**Lihat Hasil Jadwal**

**Gambar 5.8 Tampilan Set Ruang Manual**

Untuk proses set ruangan manual dilakukan dengan terlebih dahulu melakukan set sesi perhari. Ruang yang telah ditempati oleh satu kelas pada satu sesi tidak dapat ditempati oleh kelas lain kembali, dan pada list ruangan kelas yang telah dipakai tidak akan ditampilkan kembali. Setelah satu sesi selesai di-set, *button* “set ruangan” berfungsi untuk menyimpan perubahan *set*. apabila keseluruhan sesi telah di *set* perhari, maka *button* “set ruangan hari” berfungsi untuk menyimpan perubahan keseluruhan sesi pada hari tersebut. Hingga keseluruhan hari telah di set, maka hasil jadwal dapat dilihat dengan mengklik *buton* “Lihat hasil Jadwal”. Berikut merupakan hasil *generate* oleh aplikasi penjadwalan fakultas teknik bioproses yang diperoleh menggunakan algoritma dijkstra,



**MAIN MENU**

## Aplikasi Jadwal Perkuliahan

SENIN ▼

Hari	Sesi	Kelas	Mata Kuliah	Dosen	Kategori	Ruangan
SENIN	SESI 1	14TB	AITB	EAN	Teori	GD925
SENIN	SESI 1	12TB	BIOKIM	RFK	Teori	GD935
SENIN	SESI 1	11TB	FISDAS 1	RZS	Teori	GD943
SENIN	SESI 2	14TB	AITB	EAN	Teori	GD927
SENIN	SESI 2	12TB	BIOKIM	RFK	Teori	GD937
SENIN	SESI 2	11TB	FISDAS 1	RZS	Teori	GD814
SENIN	SESI 3	14TB	PEN I	RFK	Teori	GD928
SENIN	SESI 3	12TB	MIKRUM	EAN	Teori	GD938
SENIN	SESI 3	11TB	PRAK FISDAS 1	RZS	Praktikum	LAB FISIKA
SENIN	SESI 4	14TB	PEN I	RFK	Teori	GD933
SENIN	SESI 4	12TB	MIKRUM	EAN	Teori	GD812

Total Data : 23

**EXPORT KE EXCEL**

**Gambar 5.9 Generate Dijkstra**

Setelah jadwal telah berhasil untuk di-*generate*, maka dicari ruangan menggunakan algoritma dijkstra untuk memperoleh jarak terdekat antar kelas. Berikut merupakan hasil *generate* dijkstra yang telah diexport kedalam file excel



	A	B	C	D	E	F	G	H
2		Hari	Sesi	Mata Kuliah	Dosen	Kelas	Ruangan	
3		SENIN	SESI 1	AITB	EAN	14TB	GD925	
4		SENIN	SESI 1	BIOKIM	RFK	12TB	GD935	
5		SENIN	SESI 1	FISDAS 1	RZS	11TB	GD943	
6		SENIN	SESI 2	AITB	EAN	14TB	GD927	
7		SENIN	SESI 2	BIOKIM	RFK	12TB	GD937	
8		SENIN	SESI 2	FISDAS 1	RZS	11TB	GD814	
9		SENIN	SESI 3	PEN I	RFK	14TB	GD928	
10		SENIN	SESI 3	MIKRUM	EAN	12TB	GD938	
11		SENIN	SESI 3	PRAK FISDAS 1	RZS	11TB	LAB FISIKA	
12		SENIN	SESI 4	PEN I	RFK	14TB	GD933	
13		SENIN	SESI 4	MIKRUM	EAN	12TB	GD812	
14		SENIN	SESI 4	PRAK FISDAS 1	RZS	11TB	LAB FISIKA	
15		SENIN	SESI 5	AKRB	YUL	13TB	GD711	
16		SENIN	SESI 5	PRAK KIMOR	RFK	12TB	GD826	
17		SENIN	SESI 5	KIMDAS	SFA	11TB	GD815	
18		SENIN	SESI 6	AKRB	YUL	13TB	GD923	
19		SENIN	SESI 6	PRAK KIMOR	RFK	12TB	GD826	
20		SENIN	SESI 6	KIMDAS	SFA	11TB	LAB FISIKA	
21		SENIN	SESI 7	FENOPOR	MNP	13TB	GD933	
22		SENIN	SESI 7	PRAK MIKRUM	EAN	12TB	GD825	
23		SENIN	SESI 7	PRAK KIMDAS	SFA	11TB	GD812	
24		SENIN	SESI 8	SPP	DDA	14TB	GD934	

**Gambar 5.10 Hasil Export Hasil Generate Jadwal**

## **BAB 6**

### **Kesimpulan dan Saran**

Pada bab ini dijelaskan mengenai kesimpulan dan saran dari Tugas Akhir yang telah dilakukan.

#### **6.1. Kesimpulan**

Dari hasil pengerjaan Tugas Akhir ini diperoleh bahwa,

1. *Constraint Satisfaction Problem* dapat dijadikan sebagai metode yang tepat, untuk menyelesaikan masalah penjadwalan karena menghasilkan solusi penjadwalan yang unique (tetap).
2. Algoritma *Backtracking* mampu memperoleh solusi terbaik untuk jadwal melalui penelusuran keseluruhan calon solusi yang ada dengan kondisi setiap matakuliah diampu oleh satu dosen. Perpaduan antara CSP dan *backtracking* mampu menghasilkan solusi terbaik berdasarkan batasan dan aturan yang harus diikuti
3. Algoritma dijkstra dapat digunakan untuk pencarian jarak terdekat antar ruangan, untuk pergantian mata kuliah.
4. Aplikasi yang telah dibangun mampu menghasilkan jadwal perkuliahan yang memperhatikan *constaint-constraint* yang ada namun terbatas pada total data yang harus dijalankan.
5. Aplikasi yang dibangun belum dapat menyelesaikan penjadwalan di seluruh IT Del karena keseluruhan fakultas tidak dapat disatukan untuk di-*generate* dalam menghasilkan jadwal perkuliahan.

#### **6.2. Saran**

Saran yang dapat diperhatikan dalam pengembangan Tugas Akhir selanjutnya adalah,

1. Berdasarkan hasil pengujian, data yang terlalu besar mengakibatkan kebutuhan perangkat keras dengan spesifikasi yang lebih tinggi. Untuk permasalahan data yang besar, dibutuhkan metode untuk meningkatkan kinerja komputer

2. Kombinasi data dari keseluruhan sumber daya disimpan dalam *database* untuk diproses. Sehingga saat aplikasi dijalankan dengan kombinasi data yang banyak, jadwal tetap dapat di-*generate*.
3. Aplikasi yang dibangun masih membutuhkan penyempurnaan dalam menangani mata kuliah yang diampu oleh lebih dari satu dosen.

## Rujukan dan Daftar Pustaka

- Abdelhalim & Khayat, E., 2016. Penerapan algoritma genetika untuk sistem penjadwalan kuliah. *Jurnal Ilmiah Ilmu Komputer*, p. 2.
- Anamisa & Djunaidy, 2014. Penyelesaian penjadwalan maat kuliah menggunakan hibridisasi.
- Andriani, 2017. Implementasi algoritma dijkstra dalam menemukan jarak terdekat dari lokasi pengguna ke tanaman yang di tuju berbasis android (studi kasus di kebun raya purwodadi). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume 1, pp. 1779-1787.
- Anon., n.d. s.l.:s.n.
- Ardana, D. & Saputra, R., 2016. Penerapan algoritma djikstra pada aplikasi pencarian rute bus trans semarang. *Departemen Ilmu Komputer/Informartika*, pp. 299-306.
- Arsyd, 2009. Pengembangan media pembelajaran berupa komik fisika berbantuan sosial media instagram sebagai alternatif pembelajaran. *Jurnal Ilmiah Pendidikan Fisika* , pp. 33-42.
- Arviani, Y., 2013. *Algoritma ant colony system dalam penjadwalan kegiatan belajar mengajar di sekolah dasar*. Medan, Universitas Sumatera: s.n.
- Bartak & Roman, 2003. Penerapan *constraint* satisfaction problem pada metode priority scheduling untuk penjadwalan khutbah jum'at para mubaligh di IKMI pekanbaru. *Jurnal Sains, Teknologi dan Industri*, Volume 13, p. 194.
- Beligiannis, 2009. Particle swarm optimization untuk sistem informasi penjadwalan resource Di perguruan tinggi. p. 15.
- Burke, E. & Varley, D., 1997. Penerapan algoritma genetika untuk sistem penjadwalan kuliah. *Jurnal Ilmiah Ilmu Komputer*, Volume 2, p. 2.
- Cormen, 2009. s.l.:s.n.
- Cormen, 2009. *Pembangunan aplikasi pencarian rute angkutan kota di pematangsiantar*. Laguboti: s.n.
- E.Burke & D.Varley, 1997. Penerapan algoritma genetika untuk sistem penjadwalan kuliah. *Jurnal Ilmiah Ilmu Kompute*, Volume 2, p. 2.
- Fahrurozi, 2011. *Sistem informasi penjadwalan mata kuliah pada international programs fakultas SAINS dan teknologi UIN syarif hidayatullah jakarta berbasis website*. Jakarta: s.n.
- F. & Rizal, A., 2013. Penerapan algoritma dijkstra untuk menentukan rute terpendek pembacaan water meter induk PDAM tirta kerta raharja kabupaten tangerang. *TICOM*, Volume II, pp. 51-57.
- Gunawan, C. A. & Toba, H., 2016. Pembangunan solusi penjadwalan berprioritas melalui penerapan *constraint* satisfaction problem (studi kasus: laboratorium fakultas teknologi informasi universitas XXX).. *Jurnal Teknik Informatika dan Sistem Informasi*, Volume II, pp. 43-45.
- Gurari, 1999. Sistem rekomendasi pengambilan mata kuliah dengan menggunakan algoritma genetika dan metode *constraint* satisfaction. *Konferensi Nasional Teknologi Informasi dan Komunikasi*.
- Hayati & Yohanes, 2014. *Pembangunan aplikasi pencarian rute angkutan kota di Pematangsiantar*. Laguboti: s.n.

- Levitin, 2012. *Pembangunan aplikasi pencarian rute angkutan kota di pematangsiantar*. Laguboti: s.n.
- Levitin, A., 2012. *Introduction to the design & analysis of algorithms*. 3 ed. s.l.:s.n.
- Levitin, A., n.d. *Introduction to the design & analysis of algorithms*. 3 ed. s.l.:s.n.
- Lubis, 2009. Implementasi algoritma dijkstra dalam menemukan jarak terdekat dari lokasi pengguna ke tanaman yang di tuju berbasis android (studi kasus di kebun raya purwodadi). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume 1, pp. 1779-1787.
- Lumbantoruan, R. et al., 2012. Penjadwalan kuliah dengan algoritma backtracking. p. 258.
- Mansur, 2014. *Jurnal Ilmiah Ilmu Komputer*, p. 3.
- Mansur, 2014. *Jurnal Ilmiah Ilmu Komputer*. *Jurnal Ilmiah Ilmu Komputer*, p. 2.
- Maulana, H. F., 2017. Prototype pengontrolan sistem kerja hidrolik.
- Mittal & dkk, 2015. Penerapan algoritma genetika untuk sistem penjadwalan kuliah. *Jurnal Ilmiah Ilmu Kompute*, p. 2.
- Muhyi, Y., 2017. Penjadwalan kuliah otomatis dengan *constraint programming*. 15 August, p. 6.
- Munadhi, 2014. Pengembangan media pembelajaran berupa komik fisika berbantuan sosial media instagram sebagai alternatif pembelajaran. *Jurnal Ilmiah Pendidikan Fisika*, pp. 33-42.
- Napitupulu, S. S., Lubis, Y. Y. E. & Pardosi, A. J. M., 2016. *Pembangunan aplikasi pencarian rute angkutan kota di pematangsiantar*. Toba Samosir: s.n.
- Napitupulu, S. S., Lubis, Y. Y. E. & Pardosi, A. J., 2016. *Pembangunan aplikasi pencarian rute angkuatan kota di pematang siantar*. Laguboti: s.n.
- Nilsson, N. J., 1998. *Artificial Intelligence a new synthesis*. United States of America: Morgan Kaufmann Publishers, Inc.
- N. & M, Z., 2013. Penerapan algoritma backtracking berbasis blind search untuk menentukan penjadwalan mengajar. *Seminar Nasional Aplikasi Teknologi Informasi*, 15 Juni, pp. 15-16.
- Norberciak, 2009. Particle Swarm Optimization Untuk Sistem Informasi Penjadwalan Resource Di Perguruan Tinggi. p. 19.
- Oates, 2014. Particle Swarm Optimization Untuk Sistem Informasi Penjadwalan Resource Di Perguruan Tinggi. *Jurnal Sistem Informasi Bisnis*, p. 15.
- Panjaitan, G. H. & Tambunan, I. R., 2015. *Optimasi Penjadwalan Kuliah Menggunakan*. s.l.:s.n.
- Puspaningrum, W. A., Djunaidy, A. & Vinarti, R. A., 2013. Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika di Jurusan Sistem Informasi ITS. *JURNAL TEKNIK POMITS*, p. 5.
- Qashlim, A. & Assidiq, M., 2016. Penerapan Algoritma Genetika untuk Sistem Penjadwalan Kuliah. *Jurnal Ilmiah Ilmu Komputer*, Volume 2, p. 7.
- Ratnawati, L. R. & M. 2., n.d.

- Regasari, L. R. & Muflikhah, 2012. Sistem rekomendasi pengambilan mata kuliah dengan menggunakan algoritma genetika dan metode *constraint satisfaction*. *Konferensi Nasional Teknologi Informasi dan Komunikasi*, p. 29.
- Rizal, A., Mola, S. A. & Widiastuti, T., 2014. Penerapan algoritma dijkstra pada permasalahan lintasan terpendek objek wisata alam kota kupang berbasis web. *Jurusan Ilmu Komputer*, pp. 1-9.
- Rohini, V., 2013. A Phased Approach to Solve the University Course Scheduling System. *International Journal of Computational Engineering Research*, 03(4), p. 261.
- Rowe, N. C., 1998. *Artificial Intelligence Through Prolog*. United State of America: Prentice-Hall International Editions.
- Russell, S. J. & Norvig, P., 2010. *Artificial Intelligence A Modern Approach*. 3rd ed. s.l.:s.n.
- S.Jat & S.Yang, 2009. Alokasi ruang dan waktu pada universitas memiliki hubungan yang kuat. *Jurnal Ilmiah Ilmu Komputer*, Volume 2, p. 2.
- Sadiman, 2012. Pengembangan media pembelajaran berupa komik fisika berbantuan sosial media instagram sebagai alternatif pembelajaran. *Jurnal Ilmiah Pendidikan Fisika*, pp. 33-42.
- Sam'ani, 2012. s.l.:s.n.
- Shiau, 2011. penerapan algoritma genetika untuk sistem penjadwalan kuliah. *Jurnal Ilmiah Ilmu Komputer*, p. 3.
- Sirait, R., 2014. Perancangan aplikasi game labirin dengan menggunakan algoritma backtracking. *Pelita Informatika Budi Darma*, Volume V, pp. 100-103.
- Staereling, 2012. Penerapan Algoritma Genetika untuk Sistem Penjadwalan Kuliah. *Jurnal Ilmiah Ilmu Komputer*, p. 2.
- Tassopoulos & Beligiannis, 2012. rapan Algoritma Genetika untuk Sistem Penjadwalan Kuliah. *Jurnal Ilmiah Ilmu Komputer*, p. 2.
- Winston, P. H., 1992. *Artificial Intellegence*. 3 rd ed. United States of America: Addison Wesley Publishing Company.
- Yumarsono, M., n.d. *Penjadwalan Kuliah Otomatis*, s.l.: s.n.
- Yusaputra, 2013. *Pembangunan aplikasi pencarian rute angkutan kota di Pematangsiantar*. Laguboti: s.n.
- Yusuf, M. S., Az-Zahra, H. M. & Apriyanti, D. H., 2017. Implementasi algoritma dijkstra dalam menemukan jarak terdekat dari lokasi pengguna ke tanaman yang di tuju berbasis android. *Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume I, pp. 1779-1787.