

## **BAB 3**

### **METODOLOGI PENELITIAN**

#### **3.1 Spesifikasi Sistem**

Dalam melaksanakan penelitian ini, berikut merupakan rincian spesifikasi perangkat keras yang digunakan dalam penelitian:

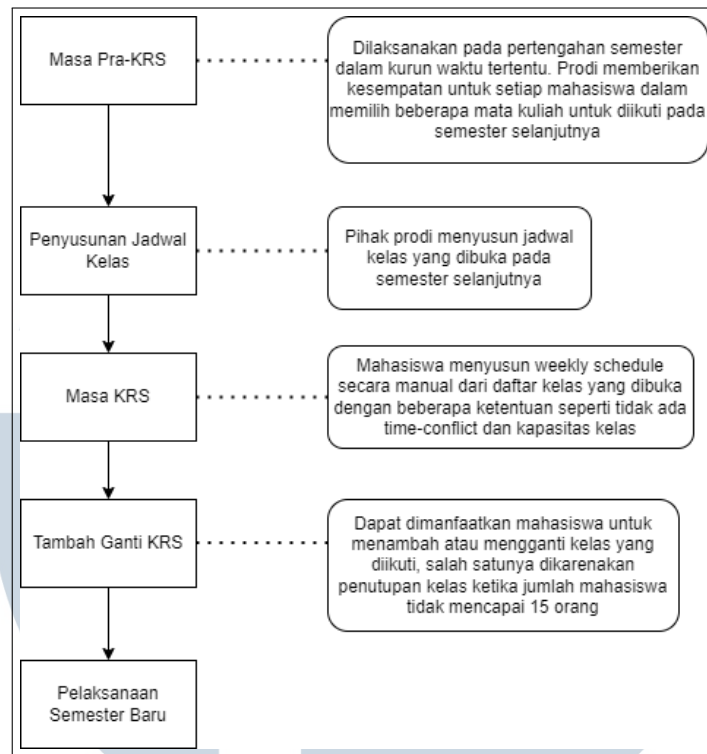
1. Prosesor Intel ® Core I7 8th Gen
2. RAM 8 GB
3. GPU Intel ® UHD Graphic 620
4. Storage 223 GB SSD
5. Google Colab Compute Engine Backend RAM 12.67 GB, Disk 107.72 GB

Berikut merupakan perincian atas perangkat lunak yang digunakan dalam penelitian:

1. Sistem operasi Windows 11
2. Google Colab
3. Python 3.10.12
4. Google Chrome

#### **3.2 Identifikasi Masalah**

Penelitian diawali dengan menganalisa permasalahan yang terjadi, tepatnya pada sistem KRS di UMN yang dijadikan sebagai objek penelitian. Sistem yang sekarang ini diterapkan pada UMN masih dilakukan secara manual dimana para mahasiswa perlu menyusun jadwal mingguan sesuai dengan kelas-kelas yang telah dibuka oleh pihak universitas. Berikut merupakan alur sistem pelaksanaan KRS (Kartu Rencana Studi) yang diterapkan.



Gambar 3.1. Sistem KRS

Seperti yang telah dijelaskan pada bagian latar belakang, proses yang menjadi fokus dalam penelitian ini berada pada masa KRS. Proses penyusunan jadwal mingguan yang masih dilakukan secara manual membuat seluruh mahasiswa dalam waktu yang bersamaan menentukan kelas untuk diambil pada semester selanjutnya dengan tidak melanggar beberapa kondisi seperti *time-conflict* dan kapasitas kelas, dengan demikian proses KRS menjadi lebih rumit dan memiliki hasil yang tidak selalu optimal. Dalam masa pengisian KRS, terdapat beberapa batasan atau aturan yang dipertimbangkan dalam pengambilan kelas bagi para mahasiswa.

1. Preferensi mata kuliah mahasiswa terpenuhi
2. Mahasiswa hanya dapat menjalankan satu kelas pada waktu yang bersamaan
3. Seorang mahasiswa hanya dapat mengambil satu kelas pada setiap mata kuliah
4. Terdapat kapasitas kelas yang tidak boleh dilebihi
5. Pemerataan pembagian mahasiswa pada setiap kelas yang dibuka dalam sebuah mata kuliah

6. Kelas hanya dibuka jika terdapat minimal 15 mahasiswa yang mengambil
7. Terdapat kelas praktikum yang perlu diambil menyesuaikan dengan kelas teori yang terambil

### 3.3 Studi Literatur

Proses ini dilakukan untuk memperluas pemahaman dengan melakukan studi melalui beberapa sumber untuk mendapatkan pengetahuan yang dapat membantu menyelesaikan permasalahan pada topik penelitian ini mencakup *constraint satisfiability problems*, *student scheduling problem*, dan *constraint integer programming*.

### 3.4 Akuisisi Data

Data yang digunakan dalam penelitian ini berasal dari pihak UMN, khususnya oleh BIA (Biro Informasi Akademik) dengan melalui persetujuan prodi. Data ini terbagi menjadi dua tabel, yaitu data kelas yang dibuka dan matakuliah preferensi mahasiswa. Berikut merupakan struktur dari kedua tabel yang digunakan.

#### 1. Tabel Jadwal Kelas

Tabel ini menyimpan informasi jadwal kelas yang dibuka oleh prodi informatika pada semester selanjutnya. Tabel ini terdiri dari 6 kolom yaitu, kode matakuliah, kode kelas, hari kelas, jam mulai, jam selesai, dan kapasitas kelas. Berikut merupakan contoh tabel jadwal kelas.

Tabel 3.1. Data Kelas yang Dibuka

Kode MK	Kode Kelas	Hari Kelas	Jam Mulai	Jam Selesai	Kapasitas Kelas
MK001	KL001	Senin	08:00	10:00	30
MK002	KL002	Selasa	10:00	12:00	25
MK003	KL003	Rabu	13:00	15:00	35
MK004	KL004	Kamis	09:00	11:00	20

#### 2. Tabel Preferensi Mahasiswa

Tabel ini menyimpan informasi terkait mata kuliah yang dipilih mahasiswa pada masa pra KRS untuk diambil mata semester selanjutnya. Tabel ini terdiri

dari 2 kolom yaitu, identifier mahasiswa dan kode matakuliah yang diminati. Berikut merupakan contoh tabel preferensi mahasiswa.

Tabel 3.2. Data Preferensi Mahasiswa

Identifier Mahasiswa	Kode Matakuliah yang Diminati
MHS001	MK001
MHS001	MK002
MHS002	MK002
MHS003	MK003
MHS004	MK001
MHS004	MK003
MHS004	MK004

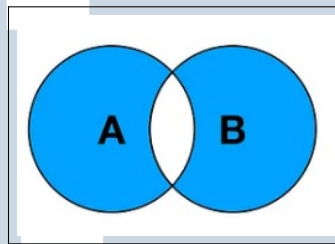
Dalam penelitian ini terdapat 4 data yang akan digunakan secara berurutan merupakan data KRS dari semester ganjil TA 2022/2023, semester genap TA 2022/2023, semester ganjil TA 2023/2024, dan semester genap TA 2023/2024, berikut merupakan spesifikasi dari setiap data.

Tabel 3.3. Keterangan Data Yang Digunakan

Keterangan	Data 1	Data 2	Data 3	Data 4
Jumlah Mahasiswa	489	453	599	587
Jumlah Mata Kuliah	62	52	61	70
Jumlah Kelas	259	206	294	272
Total Preferensi Mata Kuliah	3098	2004	4058	3101
Kapasitas Kelas Terkecil	30	30	30	15
Kapasitas Kelas Terbanyak	43	45	500	500
Rata-rata Kapasitas Kelas	38	38	41	41
Hari Pengadaan	Senin-Sabtu	Senin-Sabtu	Senin-Minggu	Senin-Sabtu

Setelah mendapatkan data tersebut, proses selanjutnya adalah *data cleaning* khususnya menghapus data yang tidak relevan atau tidak valid, dengan demikian data ini dapat digunakan dengan lebih akurat dan dapat diandalkan. Dalam konteks penelitian ini, data dinyatakan tidak relevan ketika terdapat beberapa kelas yang dibuka untuk sebuah mata kuliah, namun tidak ada mahasiswa yang mengambil mata kuliah tersebut. Berlaku juga sebaliknya jika mahasiswa memiliki minat pada

mata kuliah yang tidak dibuka oleh pihak universitas, maka data tersebut juga dianggap tidak relevan dan perlu dihapus. Pembersihan data ini dapat dilakukan dengan membuang *full outer-join excluding inner-join* dari kedua tabel berdasarkan kolom kode mata kuliah seperti gambar 3.2. Dengan demikian pada tabel jadwal kelas sudah tidak mengandung kelas yang tidak memiliki peminat, dan pada tabel preferensi mata kuliah sudah tidak mengandung preferensi kelas yang tidak dibuka.



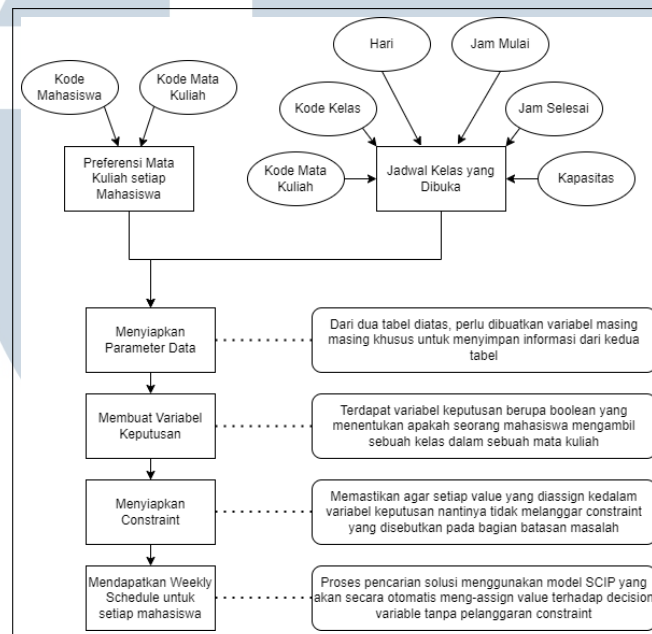
Gambar 3.2. Outer-join excluding inner-join

Proses selanjutnya adalah melakukan *data preprocessing* khususnya untuk mengubah setiap kolom yang berbentuk teks atau bentuk lainnya menjadi angka, karena pada proses merepresentasikan masalah dalam pembuatan model lebih mudah dan akurat menggunakan data berupa angka dibandingkan bentuk lainnya. Kolom seperti identifier mahasiswa, kode mata kuliah, kode kelas, dan hari kelas yang terdapat pada kedua tabel cukup diubah menjadi angka secara berurutan karena tidak terdapat urutan yang berarti pada keempat kolom tersebut. Sedangkan pada kolom jam mulai dan jam selesai, hanya perlu diubah menjadi bilangan desimal yang akan sekaligus merepresentasikan jam dan menit. Lalu pada tahap ini juga akan memastikan tidak terdapat nilai yang hilang pada setiap kolom dari kedua tabel yang digunakan.

Selain itu terdapat beberapa hal lain seperti memastikan jumlah kapasitas kelas yang dibuka memenuhi total *demand* dari preferensi mahasiswa dan penutupan kelas yang berlebihan juga perlu dilakukan karena hal tersebut berpotensi membuat model *optimization* menghasilkan *infeasible solution*. Namun untuk mempermudah beberapa penerapan hal tersebut, proses implementasi akan dilakukan setelah mendapatkan parameter data, dengan demikian data-data yang dibutuhkan menjadi lebih mudah untuk diakses.

### 3.5 Perancangan Model

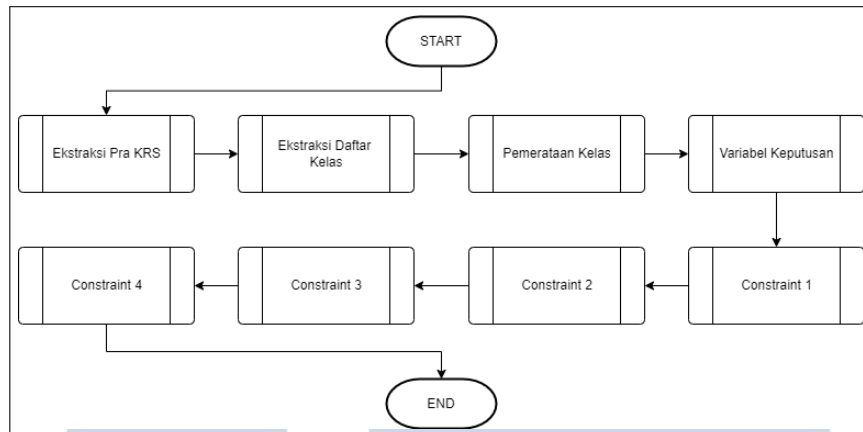
Dalam model penjadwalan yang akan dibuat, jadwal kelas mingguan untuk setiap mahasiswa secara otomatis akan terbentuk dalam waktu yang bersamaan. Dalam membuat model penjadwalan tersebut, berikut merupakan tahapan yang perlu dilakukan.



Gambar 3.3. Tahapan Pembuatan Model Penjadwalan

Proses dimulai dengan menyiapkan data preferensi mata kuliah setiap mahasiswa dan data jadwal kelas yang dibuka dari prodi. Kedua data tersebut akan diproses terlebih dahulu agar terdapat variabel yang merepresentasikan setiap informasi kolom didalam tabel, melalui variabel itu memungkinkan model yang akan dibuat mendapatkan informasi seperti hari pelaksanaan kelas, peminatan mahasiswa, dan beberapa informasi lainnya. Lalu proses dilanjutkan dengan pembuatan variabel keputusan yang akan merepresentasikan apakah seorang mahasiswa mengambil sebuah kelas dalam sebuah mata kuliah. Variabel yang terbentuk akan berjumlah total mahasiswa dikalikan dengan total kelas pada setiap mata kuliah. Setelah itu proses pembuatan *constraint* akan dilakukan dengan memberi batasan nilai pada setiap *decision variable*. Setelah semua tahapan diatas maka model sudah dapat digunakan untuk meng-*generate* jadwal kelas mingguan untuk setiap mahasiswa dalam waktu yang bersamaan. Berikut merupakan flowchart yang akan diterapkan dalam membangun sistem penjadwalan, khususnya

pada tahap parameter data hingga merepresentasikan *constraint*.



Gambar 3.4. Flowchart Sistem Penjadwalan

Pembuatan model penjadwalan dilakukan dengan beberapa tahap, dimulai dari merepresentasikan parameter data yang didapatkan dari setiap tabel yang telah dijelaskan pada bagian 3.4. Setelah itu dilanjutkan dengan penerapan *decision variable* dan *constraints* yang secara detail akan dijelaskan pada sesi dibawah ini.

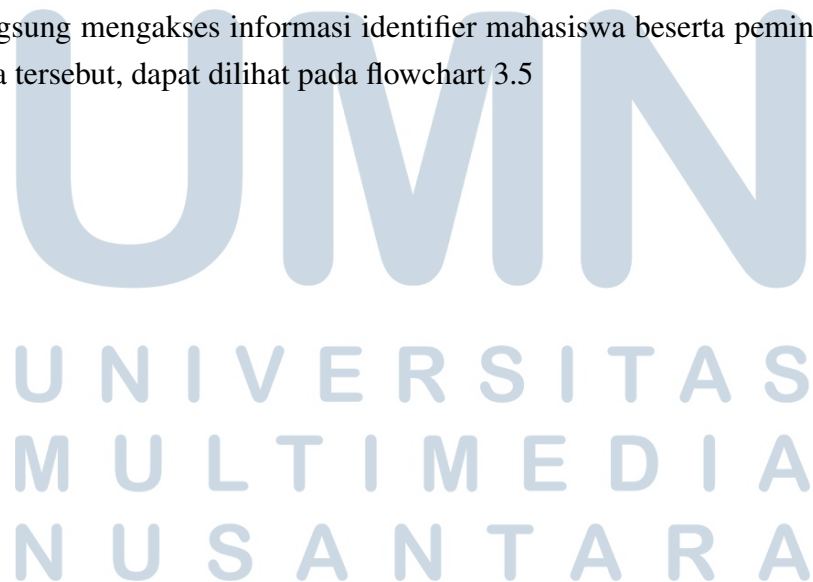
### 3.5.1 Parameter Data

Berdasarkan *requirement* yang teridentifikasi serta data yang digunakan, berikut merupakan beberapa parameter data yang perlu dibuat dalam proses perancangan model.

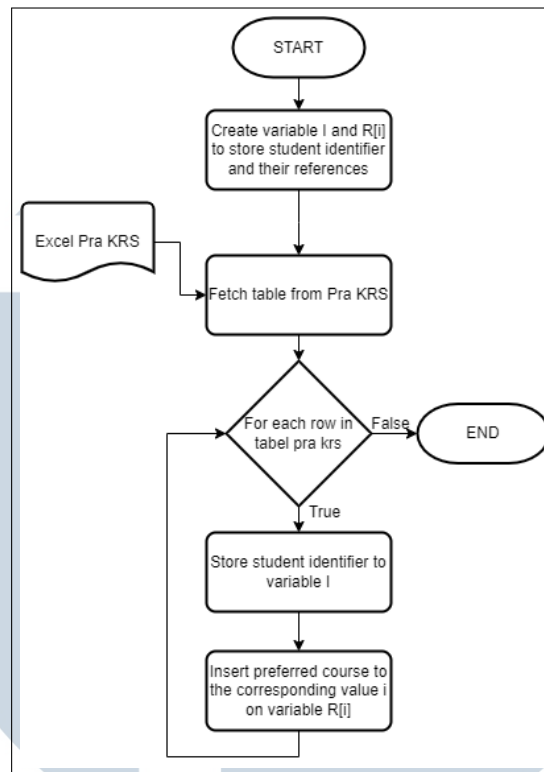
- $I$  : Kumpulan mahasiswa
- $R_i$  : Kumpulan mata kuliah preferensi mahasiswa
- $J$  : Kumpulan kelas yang dibuka
- $K_j$  : Kapasitas sebuah kelas
- $H_j$  : Hari pelaksanaan sebuah kelas
- $S_j$  : Jam mulai pelaksanaan kelas
- $E_j$  : Jam berakhir pelaksanaan kelas
- $L_j$  : Keterikatan kelas praktikum terhadap kelas teori

- $C$  : Kumpulan mata kuliah
- $M_c$  : Daftar kelas yang dibuka pada setiap mata kuliah
- $T$  : Kumpulan mata kuliah teori
- $V_t$  : Rata-rata pembagian mahasiswa pada sebuah mata kuliah

Pertama terdapat variabel  $I$  yang berisi informasi seluruh mahasiswa yang mengambil mata kuliah preferensi pada masa pra KRS, dan akan digunakan sebagai salah satu index pada variabel keputusan dan membantu proses penerapan constraint. Variabel  $I$  bisa didapatkan dengan mengambil seluruh *unique value* dari kolom identifier mahasiswa pada tabel 3.2. Lalu pada tabel yang sama terdapat informasi kumpulan preferensi mata kuliah yang akan direpresentasikan pada variabel  $R_i$ , variabel ini akan menggunakan informasi kumpulan mahasiswa pada variabel  $I$  sebagai *key* yang memiliki *value* berupa daftar mata kuliah preferensi dari mahasiswa  $i$ . Perlu diketahui juga bahwa kumpulan nilai dari data  $R_i$  merupakan salah satu dari nilai variabel  $T$  yang juga dijelaskan pada bagian ini. Dengan demikian model penjadwalan yang dibuat dapat memastikan mahasiswa perlu dan hanya mengambil kelas yang sesuai dengan preferensi-nya saja pada saat memberikan nilai variabel keputusan. Proses pengambilan kedua variabel tersebut bisa dilakukan iterasi setiap baris pada tabel pra krs, dan bisa secara langsung mengakses informasi identifier mahasiswa beserta peminatan dari mahasiswa tersebut, dapat dilihat pada flowchart 3.5







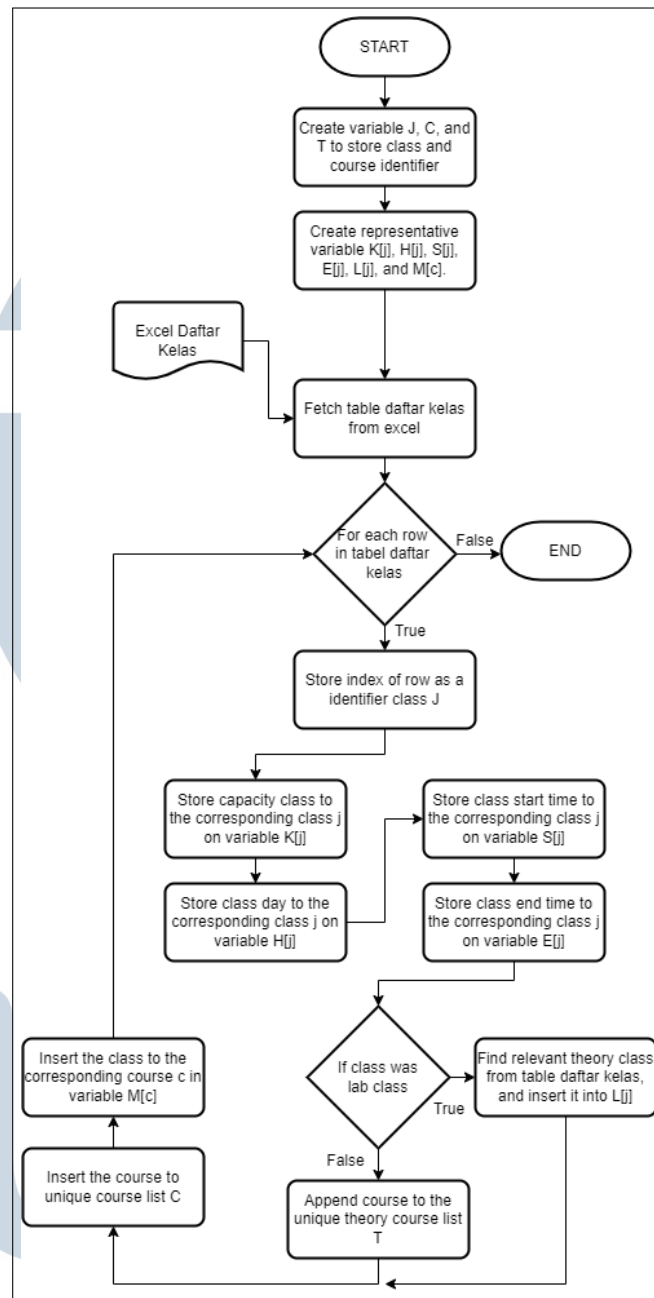
Gambar 3.5. Flowchart Ekstraksi Pra KRS

Setelah itu terdapat variabel  $J$  yang merepresentasikan seluruh kelas yang dibuka pada masa KRS terlepas dari kode mata kuliah masing masing kelas, variabel ini akan digunakan sebagai index dalam variabel keputusan juga membantu dalam proses penerapan constraint. Variabel  $J$  cukup didapatkan melalui identifer *row* pada tabel 3.1 karena setiap *row* pada tabel tersebut menunjukkan kelas yang berbeda-beda, perlu diketahui juga bahwa kolom kode kelas pada tabel 3.1 hanya merepresentasikan kode kelas yang unique untuk masing masing kode mata kuliah, seperti kelas A, B, C, D, dan E. Bersamaan dengan variabel  $J$ , terdapat beberapa data lainnya yang diperlukan untuk merepresentasikan detail informasi lainnya mencakup kapasitas kelas, hari pelaksanaan kelas, jam mulai pelaksanaan kelas, dan jam berakhirnya pelaksanaan kelas. Memiliki pemodelan yang serupa, keempat detail informasi tersebut secara berurutan akan direpresentasikan melalui variabel  $K_j, H_j, S_j, E_j$ . Variabel  $K_j$  akan digunakan untuk memastikan jumlah mahasiswa yang dimasukkan kedalam sebuah kelas tidak melebihi dari kapasitas, sedangkan ketiga variabel lainnya akan digunakan untuk memastikan mahasiswa hanya dapat menjalankan satu mata kuliah pada waktu yang bersamaan.

Selain itu diperlukan juga variabel yang merepresentasikan keterikatan kelas

praktikum terhadap kelas teori, Sehingga model yang dibuat memastikan ketika mahasiswa mengambil kelas teori A pada mata kuliah MK01, maka mahasiswa tersebut harus mengambil kelas praktikum AL untuk mata kuliah MK01. Pada data yang digunakan, perbedaan antara kelas teori dan praktikum terdapat pada kode L setelah kode kelas teorinya. Maka dari itu perlu disiapkan data parameter  $L_j$ , dimana data tersebut menggunakan identifier kelas teori dari variabel  $J$  sebagai *key* yang memiliki *value* berupa kelas praktikum yang bersangkutan. Pada proses yang sama juga akan dibentuk parameter  $T$  yang berisi kumpulan mata kuliah teori dan merupakan subset dari variabel  $C$ , yang nantinya dapat digunakan dalam rangka pemerataan mahasiswa dan mengurangi jumlah *constraint* yang dibutuhkan. Perlu diketahui juga pada variabel  $C$ , kelas praktikum memiliki kode mata kuliah yang berbeda dengan kelas teori pada mata kuliah yang sama. Hal ini dilakukan agar mempermudah proses merepresentasikan constraint 3.2, dimana terdapat pembatasan jumlah kelas untuk diambil sebanyak satu untuk mata kuliah yang sama. Lalu terdapat variabel  $C$  merepresentasikan kumpulan dari mata kuliah yang didapatkan dari seluruh *unique value* pada kolom kode mata kuliah ditabel 3.1. Variabel ini juga digunakan untuk pembuatan variabel keputusan dan pembuatan *constraint* yang berkaitan dengan mata kuliah. Selanjutnya diperlukan juga variabel  $M_c$  untuk mendapatkan daftar kelas yang dibuka pada setiap mata kuliah  $c$ . Gambar 3.6 dapat diperhatikan terkait proses penarikan parameter data yang telah disebutkan. Dengan demikian akan memperkecil jumlah iterasi dan ukuran dari variabel keputusan, dimana eksplorasi hanya dibatasi pada kelas yang sesuai dengan kode mata kuliah-nya saja

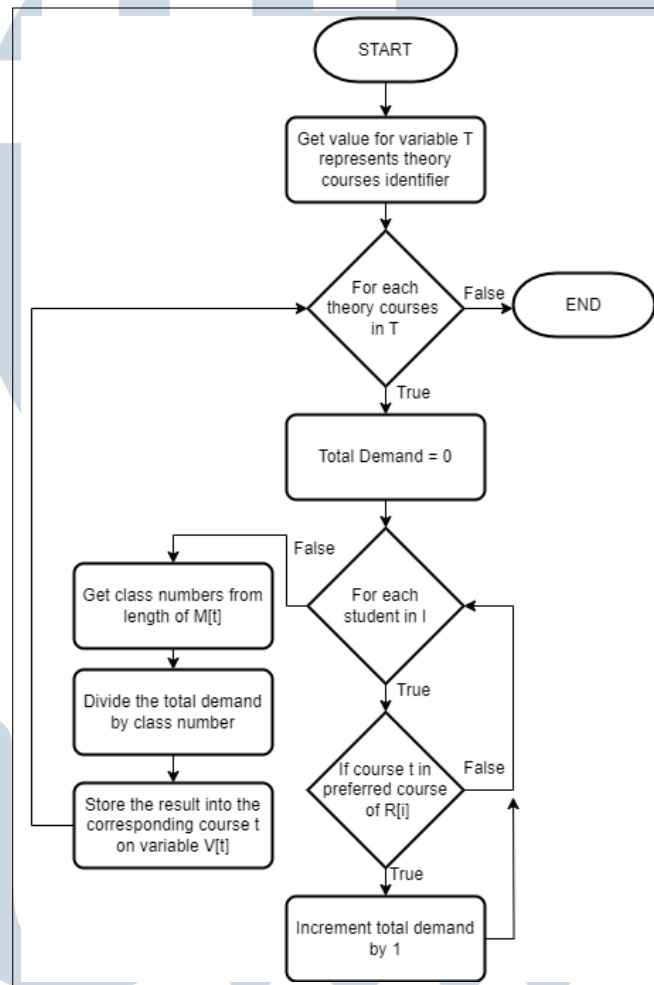
U M M N  
U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 3.6. Flowchart Ekstraksi Daftar Kelas

Proses dimulai dengan melakukan iterasi pada setiap baris dalam tabel daftar kelas, dimana index pada setiap baris merupakan identifier sebuah kelas. Lalu akan diambil seluruh informasi kapasitas, hari pelaksanaan, jam mulai, dan jam selesai kedalam variabel yang telah dibuat. Selanjutnya jika kelas tersebut merupakan kelas praktikum, maka akan dicari terlebih dahulu kelas teori yang terkait untuk dimasukkan kedalam variabel  $L[j]$ , sebaliknya akan dimasukkan kedalam variabel

T jika kelas tersebut merupakan teori. Setelah itu identifier matakuliah dari kelas tersebut dapat ditambahkan beserta memasukkan kelas kedalam variabel  $M[c]$ . Berdasarkan parameter data  $R_i$ ,  $T$ , dan  $M_c$ , parameter data  $V_t$  dapat dibuat untuk menyimpan rata rata pembagian mahasiswa pada seluruh kelas dalam setiap mata kuliah. Berikut merupakan *flowchart* dalam menarik informasi pemerataan kelas.



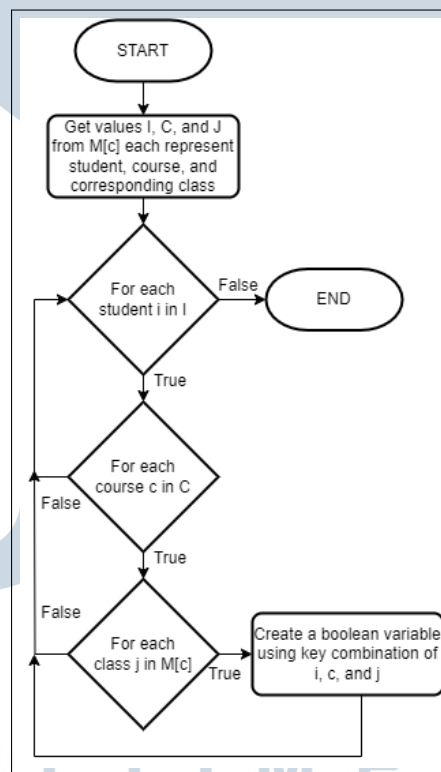
Gambar 3.7. Flowchart Pemerataan Kelas

Proses dapat dimulai dengan mengiterasikan setiap matakuliah teori pada variabel  $T$ . Pada setiap iterasi, akan dihitung jumlah mahasiswa  $i$  yang meminati kelas tersebut pada variabel  $R[i]$ . Dan pada akhirnya, total peminat tersebut akan dibagi dengan jumlah kelas yang tersimpan pada variabel  $M[c]$ . Hasil pembagian tersebut akan dimasukkan kedalam variabel  $V[c]$  sebagai rata rata pembagian mahasiswa pada setiap kelas dalam sebuah matakuliah.

### 3.5.2 Variabel Keputusan

Setelah semua parameter data disiapkan, selanjutnya bisa dilanjutkan membentuk variabel keputusan. Seperti yang telah dijelaskan sebelumnya, variabel ini akan digunakan untuk menyimpan hasil keputusan dari model penjadwalan yang dibentuk. Dalam penelitian ini, variabel keputusan yang akan digunakan berbentuk *boolean* atau biner yang merepresentasikan apakah seorang mahasiswa mengambil sebuah kelas yang telah dibuka. Berikut merupakan variabel keputusan yang digunakan.

$$X_{icj} = \begin{cases} 1, & \text{Jika mahasiswa } i \text{ ditempatkan pada kelas } j \text{ untuk mata kuliah } c \\ 0, & \text{Sebaliknya} \end{cases} \quad (3.1)$$



Gambar 3.8. Flowchart Variabel Keputusan

Variabel keputusan yang dibuat bersifat 3 dimensi, dimana setiap kombinasi dari tiga variabel *I*, *C*, dan *J* memiliki variabel keputusan masing masing. Namun kumpulan kombinasi tersebut dapat dipersempit dengan memanfaatkan variabel  $M_c$ ,

sehingga tidak terdapat kombinasi dari kelas yang tidak sesuai dengan mata kuliahnya.

### 3.5.3 Constraint

Setelah mendapatkan variabel keputusan dan parameter data, maka proses pembuatan constraint sudah dapat dilakukan. Dalam *constraint* yang telah disebutkan pada bagian identifikasi masalah, terdapat beberapa yang dianggap sebagai *soft constraint* dan beberapa lainnya sebagai *hard constraint*. Pada dasarnya semua constraint bisa saja dijadikan sebagai *hard constraint*, namun jika terdapat kondisi dimana adanya konflik antar *hard constraint* yang digunakan. Misal ketika total kapasitas dari kelas yang dibuka pada sebuah mata kuliah tidak cukup untuk menampung semua mahasiswa yang meminatinya, dengan demikian model penjadwalan yang dibuat akan menghasilkan *infeasible solution* tanpa mencari *least optimal solution* dari permasalahan yang diberikan. Berikut merupakan beberapa *constraint* yang perlu diterapkan.

1. Setiap mahasiswa  $i$  hanya boleh mengikuti paling banyak 1 kelas  $j$  pada setiap mata kuliah  $c$ .

$$\sum_{j \in M_c} X_{icj} \leq 1, \quad \forall i \in I, \quad \forall c \in T \quad (3.2)$$

2. Mahasiswa  $i$  hanya dapat mengikuti 1 kelas  $j$  pada timeslot  $H_j$ ,  $S_j$ , dan  $E_j$ .

$$\text{if is\_conflict}(j_1, j_2) : \quad x_{i,c_1,j_1} + x_{i,c_2,j_2} \leq 1, \quad \forall i \in I, \quad \forall c_1 \in C, \quad \forall c_2 \in C, \\ c_1 \neq c_2, \quad \forall j_1 \in M_{c_1}, \quad \forall j_2 \in M_{c_2} \quad (3.3)$$

3. Jumlah mahasiswa  $i$  yang terdaftar pada setiap kelas  $j$  dalam sebuah mata kuliah  $c$  harus diperhatikan keseimbangannya dan tidak boleh lebih sedikit dari 15 mahasiswa juga tidak boleh melebihi kapasitas kelas  $K_j$  untuk setiap kelasnya.

$$\max(15, V_c - 10) \leq \sum_{i \in I} X_{icj} \leq \min(V_c + 10, K_j), \quad \forall c \in T, \quad \forall j \in M_c \quad (3.4)$$

4. Mahasiswa  $i$  harus mengambil mata kuliah praktikum  $L_j$  jika mengambil

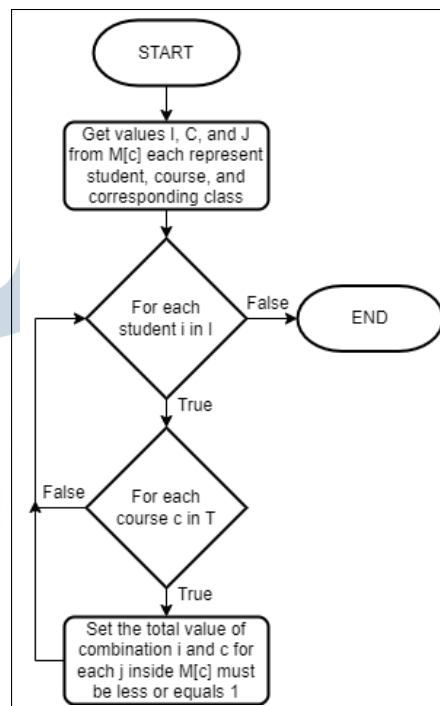
mata kuliah teori  $j$

$$X_{icj} = x_{i,cl,jl}, \quad \forall i \in I, \quad \forall c \in T, \quad \forall j \in M_c \cap L, \quad jl = L_j, \quad cl = \text{get\_course}(L_j) \quad (3.5)$$

5. Memaksimalkan preferensi mahasiswa  $i$  untuk mengikuti mata kuliah  $c$  yang terdapat didalam  $R_i$

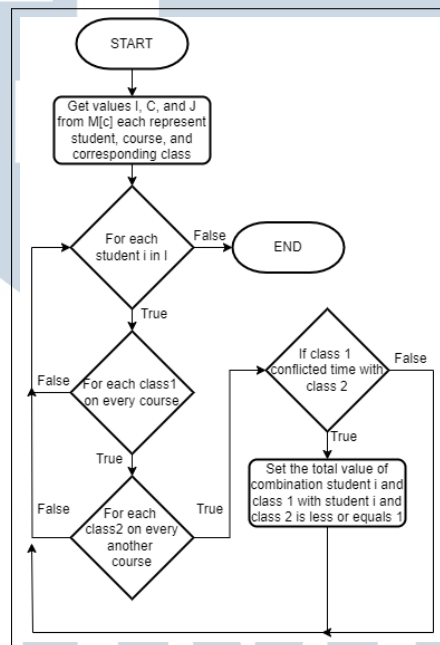
$$\text{maximize} \left( \sum_{(i,c,j) \in x : c \in R_i} x_{icj} \right) \quad (3.6)$$

Persamaan 3.2 merupakan penerapan constraint untuk memastikan bahwa untuk setiap mahasiswa hanya dapat mengikuti maksimal 1 kelas untuk setiap mata kuliah, dengan demikian hasil penjadwalan dari model yang dibuat tidak memiliki mata kuliah yang duplikat untuk diambil oleh mahasiswa. *Constraint* ini diterapkan dengan membatasi jumlah seluruh variabel keputusan untuk setiap kelas yang dibuka agar tidak lebih dari satu. Pada *constraint* ini dan beberapa *constraint* lainnya hanya perlu diterapkan pada setiap mata kuliah teori saja, karena mata kuliah praktikum hanya akan perlu mengikuti nilai variabel keputusan dari kelas teori yang terkait. Berikut merupakan flowchart dari penerapan constraint ini.



Gambar 3.9. Flowchart Constraint 1

Pada persamaan 3.3, *constraint* untuk memastikan bahwa seorang mahasiswa hanya dapat menjalankan satu kelas pada waktu yang bersamaan dapat diterapkan. *Constraint* ini diterapkan dengan memastikan untuk setiap mahasiswa, jumlah dari variabel keputusan tidak melebihi 1 untuk setiap kombinasi kelas yang merupakan *time conflict*. Sementara itu, metode pengecekan *timeconflict* dilakukan dengan membandingkan ketiga variabel  $H_j$ ,  $S_j$ , dan  $E_j$ , dimana jika hari pelaksanaan antar kelas sama, lalu jam kelas pertama dimulai sebelum jam kelas kedua selesai, dan jam kelas kedua dimulai sebelum jam kelas pertama selesai, maka kedua kelas tersebut bentrok dan tidak dapat diikuti secara bersamaan. Berikut merupakan flowchart dari penerapan *constraint* ini.

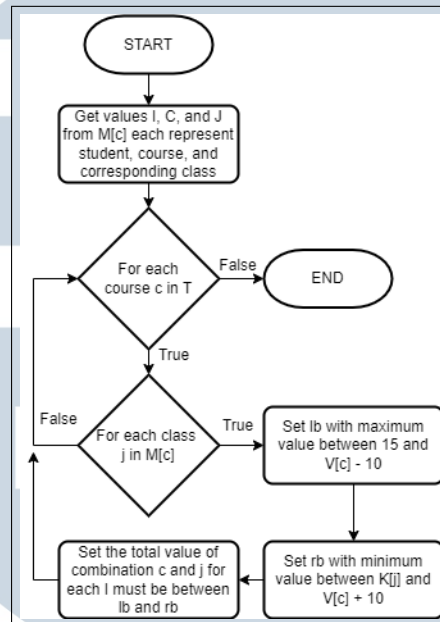


Gambar 3.10. Flowchart Constraint 2

Setelah itu pada persamaan 3.4 digunakan untuk memberikan batas bawah dan batas atas pada setiap kelas berkaitan dengan pemerataan jumlah mahasiswa dalam sebuah mata kuliah. Batasan tersebut didapatkan dengan menambahi atau mengurangi parameter data  $V_i$  dengan sebuah konstanta atau bisa disebut sebagai *offset*, misalnya 10 seperti pada persamaan diatas. Namun jika *offset* yang digunakan terlalu kecil, dapat berpotensi menghasilkan infeasible solution atau banyak mahasiswa yang tidak mendapatkan mata kuliah preferensi-nya. Disamping itu juga terdapat pembatasan pada batas bawah, dimana terdapat batas minimal jumlah mahasiswa sebanyak 15 orang agar sebuah kelas dapat diadakan. Sedangkan pada batas atas juga dibatasi dengan nilai kapasitas kelas. Dengan demikian

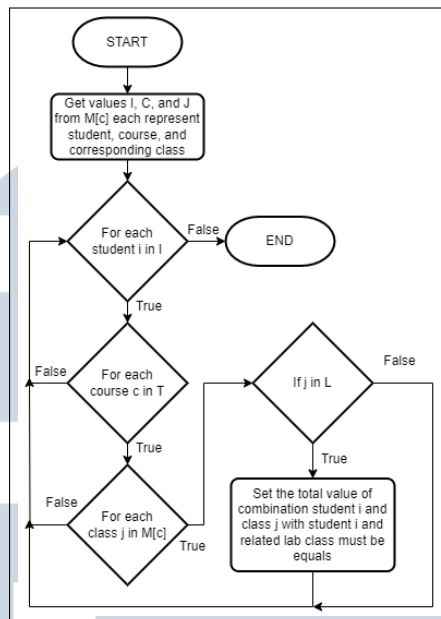


distribusi mahasiswa kedalam sebuah kelas pada setiap mata kuliah menjadi merata dan tetap mematuhi aturan jumlah minimal dan maksimal mahasiswa dalam pengadaan kelas. Berikut merupakan flowchart dari penerapan constraint ini.



Gambar 3.11. Flowchart Constraint 3

Lalu pada persamaan 3.5 akan digunakan untuk memastikan pada setiap mata kuliah yang memiliki kelas praktikum, jika mahasiswa di-assign sebuah kelas teori, maka mahasiswa tersebut juga harus di-assign dengan kelas praktikum yang bersangkutan. *Constraint* ini dapat diterapkan dengan memberikan persamaan antara dua variabel keputusan yang masing masing merupakan kombinasi dari mahasiswa yang sama untuk masing masing kelas teori dan praktikum yang bersangkutan. Perlu diperhatikan juga bahwa iterasi yang dilakukan untuk penetapan *constraint* dilakukan terhadap setiap mata kuliah teori pada variabel  $T$ , lalu informasi terkait kelas praktikum yang bersangkutan beserta kode mata kuliah-nya, masing masing didapatkan melalui variabel  $L_j$  dan melalui metode  $getcourse(L_j)$ . Metode  $getcourse$  ini dapat diterapkan dengan mengambil *key* dari parameter data  $M_c$  berdasarkan value  $L_j$  sebagai kode kelas praktikum yang bersangkutan. Berikut merupakan flowchart dari penerapan constraint ini.



Gambar 3.12. Flowchart Constraint 4

Berbeda dengan kelima *constraint* sebelumnya yang diterapkan sebagai *hard constraint*. Pada persamaan 3.6 diterapkan untuk memaksimalkan jumlah mahasiswa yang berhasil mendapatkan mata kuliah preferensi-nya. *Constraint* ini diterapkan dengan mendefinisikan model untuk memaksimalkan nilai dari jumlah variabel keputusan untuk kombinasi setiap mahasiswa, kelas, dan mata kuliah dimana mata kuliah tersebut merupakan salah satu dari peminatan mahasiswa yang tersimpan pada variabel  $R_j$ .

### 3.6 Pengujian

Sesuai dengan tujuan dari *optimization* dalam mendapatkan solusi terbaik pada ruang solusi dari sebuah permasalahan dengan waktu yang wajar, proses pengujian model akan dilakukan dengan memperhatikan waktu komputasi yang diperlukan dan tingkat optimalitas solusi berdasarkan jumlah pelanggaran *constraint* yang dihasilkan [29]. Teknik *optimization* umumnya melibatkan *trade-off* antara tingkat optimalitas dan waktu komputasi yang dibutuhkan [29]. Oleh karena itu, semakin cepat sebuah teknik optimisasi dalam memaksimalkan tingkat optimalitas pada solusi yang dihasilkan, semakin bagus teknik optimisasi tersebut dalam menyelesaikan masalah [15]. Tahap pengujian ini juga akan dilakukan terhadap beberapa dataset yang berbeda-beda, masing masing memiliki jumlah mata kuliah dan mahasiswa yang berbeda. Selain itu juga akan dilakukan validasi

terhadap hasil penjadwalan yang dibuat, untuk memastikan hasil penjadwalan tersebut sudah sesuai dengan *constraint* yang ditetapkan.

### **3.7 Evaluasi User**

Tahap ini dilakukan untuk mendapatkan *feedback* dari pihak yang terkait mengenai hasil penjadwalan yang telah dilakukan secara otomatis. Pihak yang dilibatkan untuk proses evaluasi adalah prodi informatika. *Feedback* yang didapatkan bertujuan untuk mengetahui apakah hasil penjadwalan yang terbentuk sudah layak dan dapat diintegrasikan dengan sistem yang diterapkan pada universitas. Selain itu diperlukan juga untuk mengetahui pertimbangan-pertimbangan lainnya untuk mengembangkan model penjadwalan yang telah terbuat. Teknik evaluasi dapat dilakukan melalui 2 cara, yaitu secara langsung melakukan *Question and Answer* atau melalui surat elektronik. Validasi terhadap *feedback* juga akan dilakukan untuk memastikan kebenarannya dengan membuat perjanjian yang ditandatangani kedua pihak antara penulis dan responden.

### **3.8 Penulisan Laporan**

Tahap ini dilakukan selama proses penelitian hingga selesai dengan melakukan dokumentasi terhadap penelitian yang telah dilakukan mulai dari rincian tahapan penelitian, hasil penelitian, hingga pengambilan kesimpulan.

U M I N  
U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A