

BAB III

ANALISIS DAN PERANCANGAN APLIKASI

3.1 Analisis

Dalam proses analisis, terdapat dua cara yang ditempuh, diantaranya :

a. Wawancara/Interview

Langkah ini dilakukan untuk mengetahui permasalahan-permasalahan yang terjadi di Labkom, dimana permasalahan tersebut berkaitan dengan masalah penjadwalan praktikum. Selain itu, langkah ini dilakukan untuk mengetahui kebutuhan-kebutuhan aplikasi dan keinginan pihak Labkom yang nantinya akan menggunakan aplikasi ini nantinya. Wawancara ini dikoordinasikan oleh pihak Labkom, dimana pihak Labkom ini akan mendatangkan beberapa narasumber yang akan menggunakan aplikasi ini. Narasumber tersebut diantaranya adalah Ibu Ayuningtyas selaku kepala bagian Labkom.

b. Analisis Dokumen

Analisis dokumen ini adalah langkah untuk mengamati dan menganalisis dokumen apa saja yang berkaitan tentang penjadwalan. Dokumen yang diamati diantaranya adalah adalah dokumen KRS mahasiswa.

Dokumen KRS mahasiswa akan dijadikan sebagai acuan dalam fungsi penjadwalan praktikum. Adapun dokumen lain yang digunakan untuk proses penjadwalan praktikum adalah, data pemakaian laboratorium dalam satu semester, data kriteria Labkom, data mata praktikum yang diselenggarakan dan data praktikan.

3.1.1 User Requirements

Berdasarkan hasil wawancara dengan bagian laboratorium komputer maka *user requirements* yang dibutuhkan adalah sebagai berikut.

Tabel 3.1 *User Requirement* Penjadwalan

Deskripsi	:	Fungsi ini digunakan oleh Labkom. Untuk melakukan proses penjadwalan praktikum
Aktor	:	Labkom, AAK
Input	:	Data KRS mahasiswa, data pemakaian laboratorium dalam satu semester, data kriteria Labkom, data mata praktikum yang diselenggarakan, data praktikan
Proses	:	1. Mencocokkan data praktikan dengan KRS dan kriteria yang telah dibuat oleh Labkom
Output	:	Data Jadwal Praktikum Tersimpan
Peraturan	:	1. Tidak boleh ada jadwal yang crash (dalam hal jam matapraktikum dan jam mata kuliah reguler) 2. Laboratorium yang digunakan tidak lebih dari kapasitas.

3.1.2 Software Requirements

Berdasarkan hasil analisis dari *user requirements* diatas, maka dibutuhkan *software requirements* yang dapat menunjang fungsi penjadwalan.

Terdapat 1 fungsi dalam *software requirements* yang dibutuhkan, yaitu :

Tabel 3.2 *Software Requirement* Penjadwalan

Deskripsi	:	Fungsi ini digunakan oleh Labkom. Untuk melakukan proses penjadwalan praktikum
Awal	:	Masukkan Data Praktikan dan Data KRS Mahasiswa
Alur komputerisasi (computerized-system-flow)	:	1. Aktor mengklik tombol proses jadwal 1.1. Aplikasi mengambil data KRS mahasiswa dan data praktikan 1.2. Aplikasi membuat jadwal praktikum awal berdasarkan kriteria Labkom dan data pemakaian laboratorium

	1.3. Aplikasi menampung data praktikan ke dalam array praktikan, data KRS ke dalam array KRS dan data jadwal praktikum awal ke dalam array jadwal awal 1.4. Proses Tabu array KRS, array praktikan dan array jadwal awal 1.5. Aplikasi akan meng-update data jadwal 1.6. Aplikasi akan Menyimpan data jadwal
Akhir	: Jadwal Praktikum
Non Fungsional	:

3.1.3 Data Requirements

Dari beberapa *software requirements* yang sudah dijabarkan sebelumnya, maka diperlukan beberapa data untuk mendukung *software requirements* tersebut, beberapa data yang dibutuhkan diantaranya adalah.

a. Data Kriteria Labkom

Merupakan data kriteria dari pihak Labkom yang berisi tentang ketentuan penggunaan dan jumlah peserta dalam satu laboratorium.

b. Data Mata Praktikum Yang Diselenggarakan

Merupakan data mata praktikum yang diselenggarakan dalam satu semester.

c. Data Kartu Rencana Studi(KRS) Mahasiswa Terakhir

Merupakan data kartu rencana studi sebagai acuan dalam pembuatan jadwal mata praktikum.

d. Data Praktikan

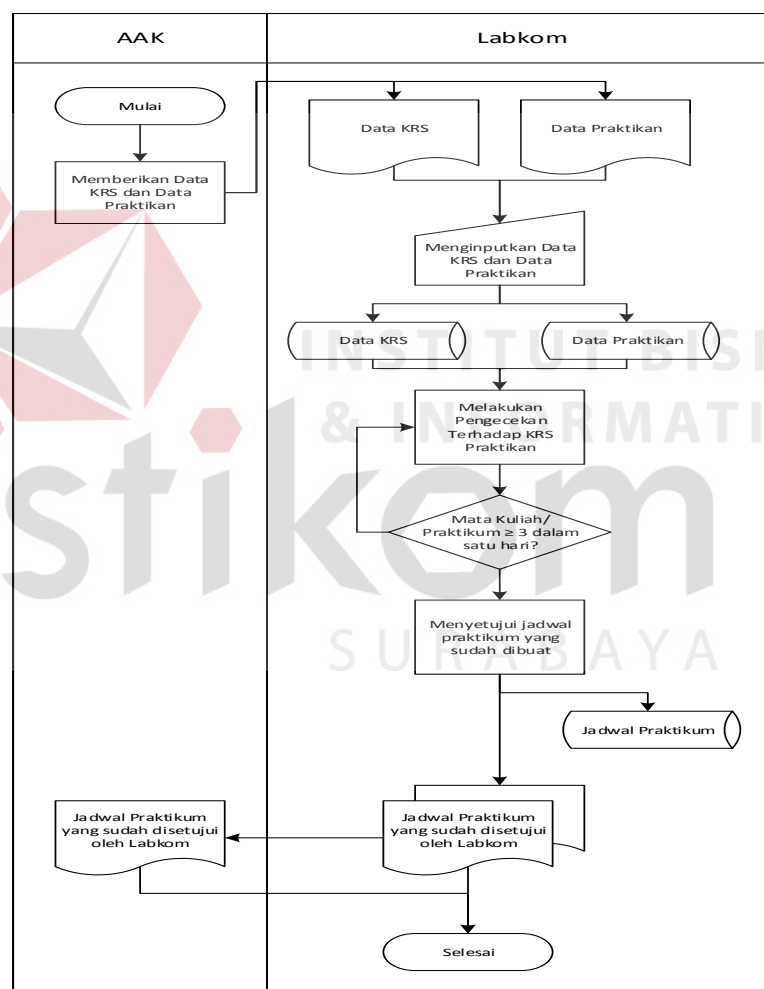
Merupakan data dari praktikan yang mengikuti mata praktikum yang diselenggarakan.

3.2 Perancangan Aplikasi

3.2.1 Desain Proses

Dari hasil analisis *software requirements* di atas terdapat 1 fungsi yang digunakan agar aplikasi penjadwalan praktikum dapat berjalan. Fungsi yang telah dihasilkan dari analisa kebutuhan aplikasi tersebut akan digambarkan dalam *system flow*, *context diagram* dan *data flow diagram*.

A. Document Flow

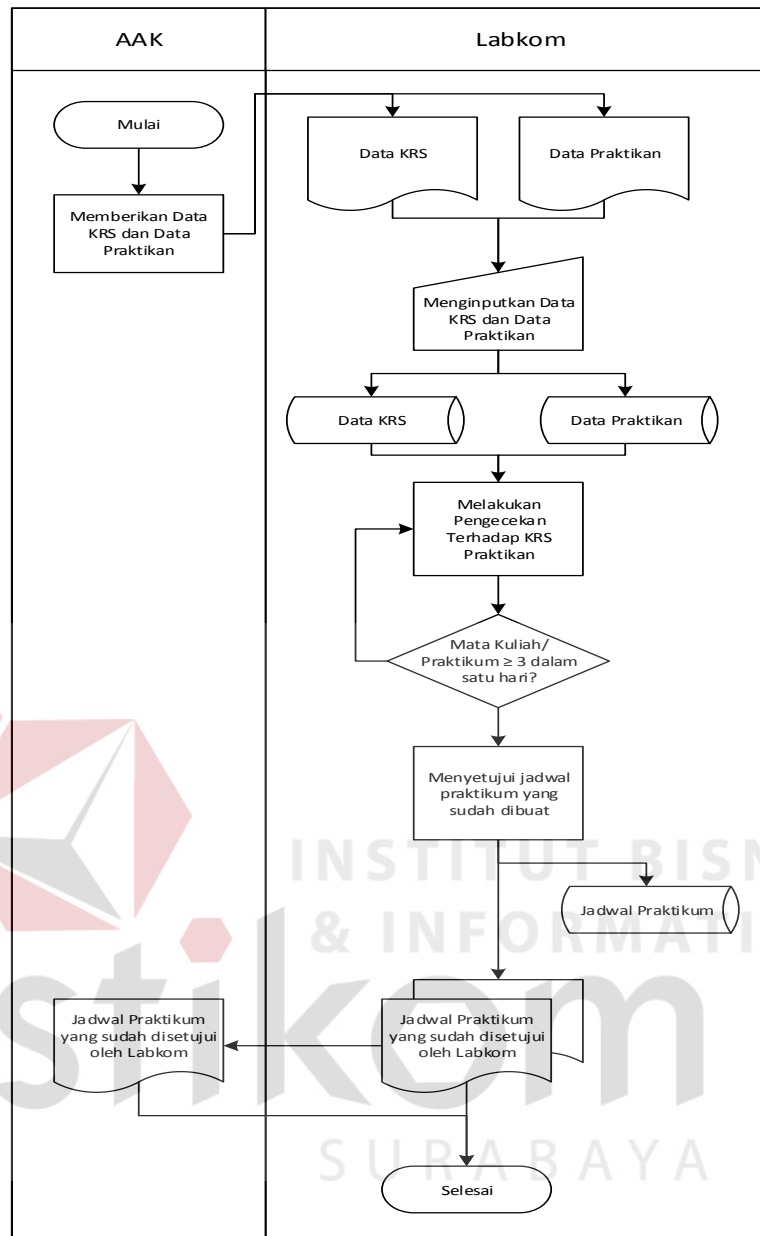


Gambar 3.1 Document Flow Penjadwalan Praktikum

Document flow ini akan menjelaskan tentang proses penjadwalan yang lama. Proses dimulai oleh bagian AAK yang menyerahkan data praktikan. Labkom akan melakukan proses penjadwalan dengan tidak memperhatikan jumlah mata kuliah atau praktikum dalam satu hari. Apabila jadwal praktikum yang dibuat sudah sesuai maka jadwal praktikum akan disetujui dan diserahkan kepada AAK. Gambar 3.1 berikut ini merupakan *document flow* dari proses penjadwalan praktikum.

B. System Flow

System flow ini akan menjelaskan tentang aplikasi yang akan dibuat. Pertama-tama, bagian AAK akan menyerahkan data KRS dan data praktikan dan bagian Labkom akan menginputkan data tersebut sehingga data tersebut dapat diolah dalam proses pengecekan jadwal KRS praktikan. Apabila dalam satu hari praktikan memiliki 3 jadwal mata kuliah atau praktikum maka, proses pengecekan akan diulangi untuk hari yang lain. Apabila jadwal praktikum yang dibuat sudah sesuai maka jadwal praktikum akan disetujui dan diserahkan kepada AAK. Gambar 3.2 ini merupakan gambar dari system flow aplikasi penjadwalan praktikum.

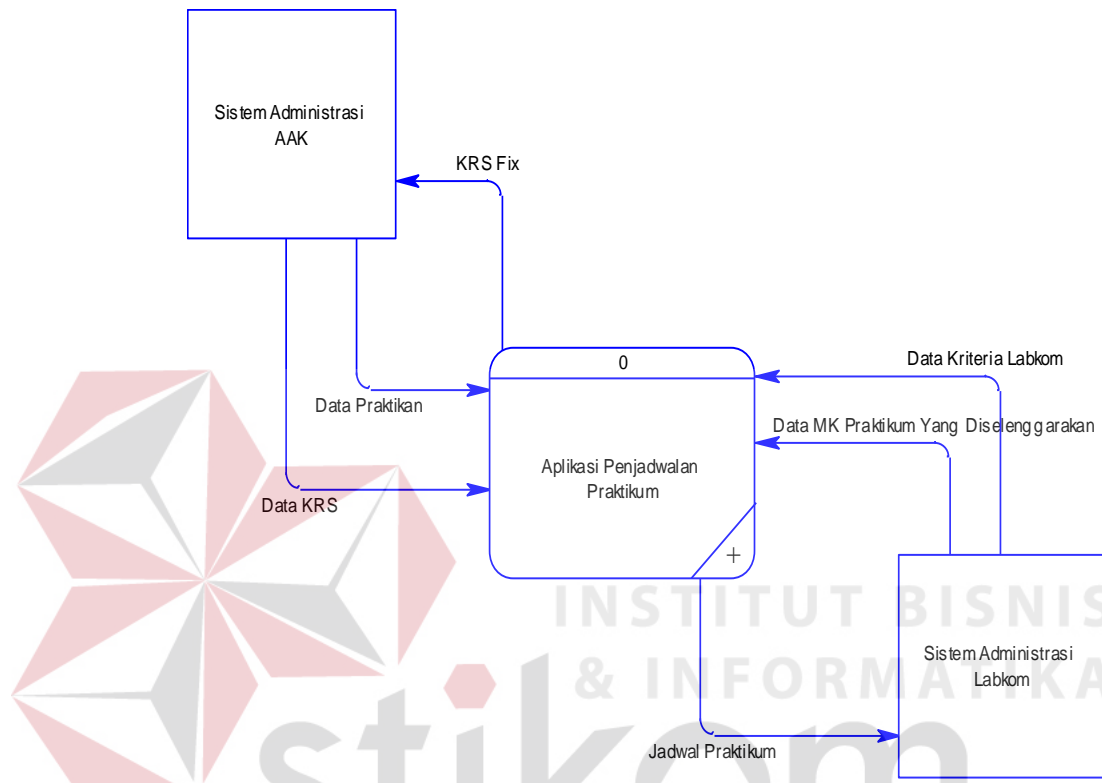


Gambar 3.2 System Flow Aplikasi Penjadwalan Praktikum

C. Context Diagram

Context diagram dibuat untuk menampilkan entitas apa saja yang akan berinteraksi dengan aplikasi penjadwalan ini. *Context diagram* ini dibuat berdasarkan hasil analisis *software requirements*. Terdapat satu *software requirements* yang dihasilkan yaitu penjadwalan. Dimana dari *software requirements* tersebut digunakan oleh dua aktor yakni bagian Labkom dan AAK,

sehingga dalam hal ini bagian Labkom dan AAK otomatis menjadi entitas yang ikut membangun aplikasi tersebut. Gambar 3.3 ini merupakan gambar dari *Context Diagram* aplikasi penjadwalan praktikum.

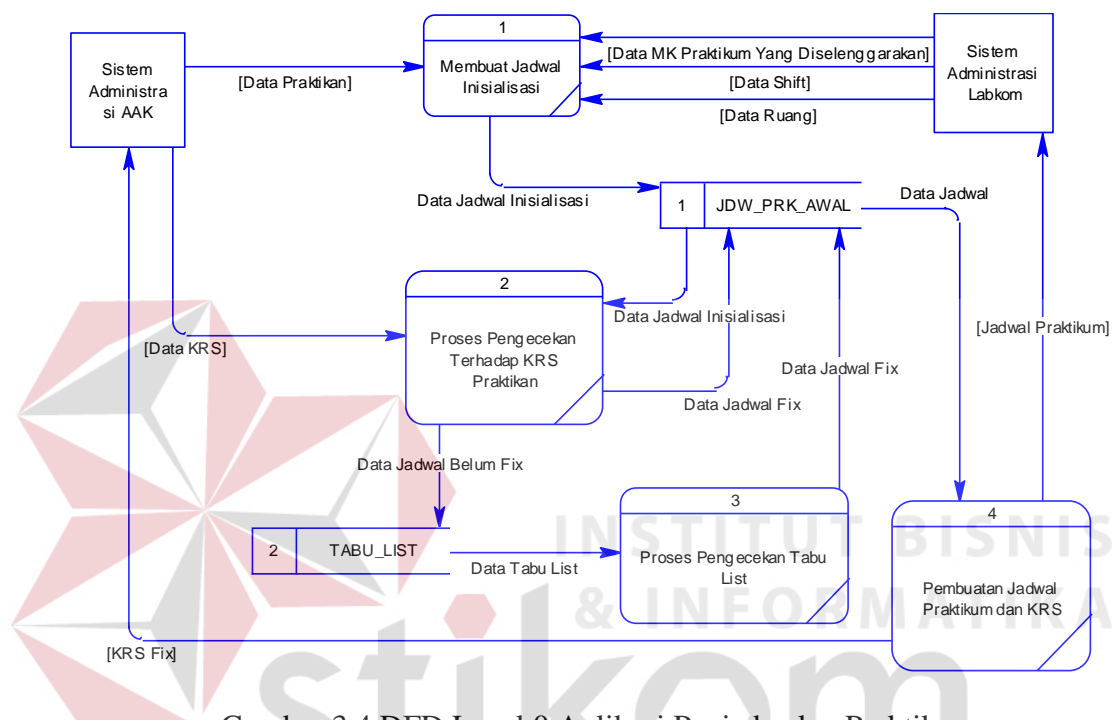


Gambar 3.3 *Context Diagram*

Bagian Labkom yang akan menggunakan aplikasi penjadwalan praktikum ini akan memberikan inputan data pemakaian laboratorium dan data kriteria Labkom dan akan menerima output berupa jadwal praktikum. Bagian AAK akan membantu aplikasi penjadwalan praktikum dengan memberikan inputan data KRS mahasiswa, data peserta praktikum(praktikan) dan data mata praktikum yang diselenggarakan.

D. DFD Level 0 Aplikasi Penjadwalan Praktikum

Sama halnya dengan *context diagram*, DFD level 0 aplikasi penjadwalan praktikum ini dibuat berdasarkan *software requirements*. Gambar 3.4 ini merupakan DFD level 0 aplikasi penjadwalan praktikum.



Gambar 3.4 DFD Level 0 Aplikasi Penjadwalan Praktikum

Proses pertama dari DFD level 0 adalah proses membuat jadwal inisialisasi. Pada proses ini, sistem administrasi AAK akan memasukkan data praktikan. Sistem administrasi Labkom akan memasukkan data MK praktikum yang diselenggarakan, data hari, dan data shift praktikum. Hasil dari proses ini akan dimasukkan ke dalam tabel JDW_PRK_AWAL.

Proses yang kedua adalah proses pengecekan terhadap KRS praktikan, proses ini dimulai dari sistem administrasi AAK memberikan data KRS. Proses pengecekan terhadap KRS praktikan ini akan menghasilkan dua data jadwal, yaitu data jadwal fix dan data jadwal belum fix. Data jadwal fix akan memperbaharui

data yang ada pada tabel JDW_PRK_AWAL. Sedangkan, data jadwal belum fix akan dimasukkan ke dalam tabel TABU_LIST.

Proses yang ketiga adalah proses pengecekan *tabu list*. Proses ini akan mengecek data dalam tabel TABU_LIST terhadap KRS praktikan sehingga dapat dicari data jadwal praktikum yang cocok dengan KRS praktikan. Proses ini akan menghasilkan data jadwal fix yang akan kembali dimasukkan ke dalam tabel JDW_PRK_AWAL. Proses terakhir adalah proses menampilkan data jadwal praktikum dan KRS praktikan. Hasil dari proses ini adalah hasil dari proses penjadwalan yaitu, jadwal praktikum beserta praktikan dan KRS praktikan.

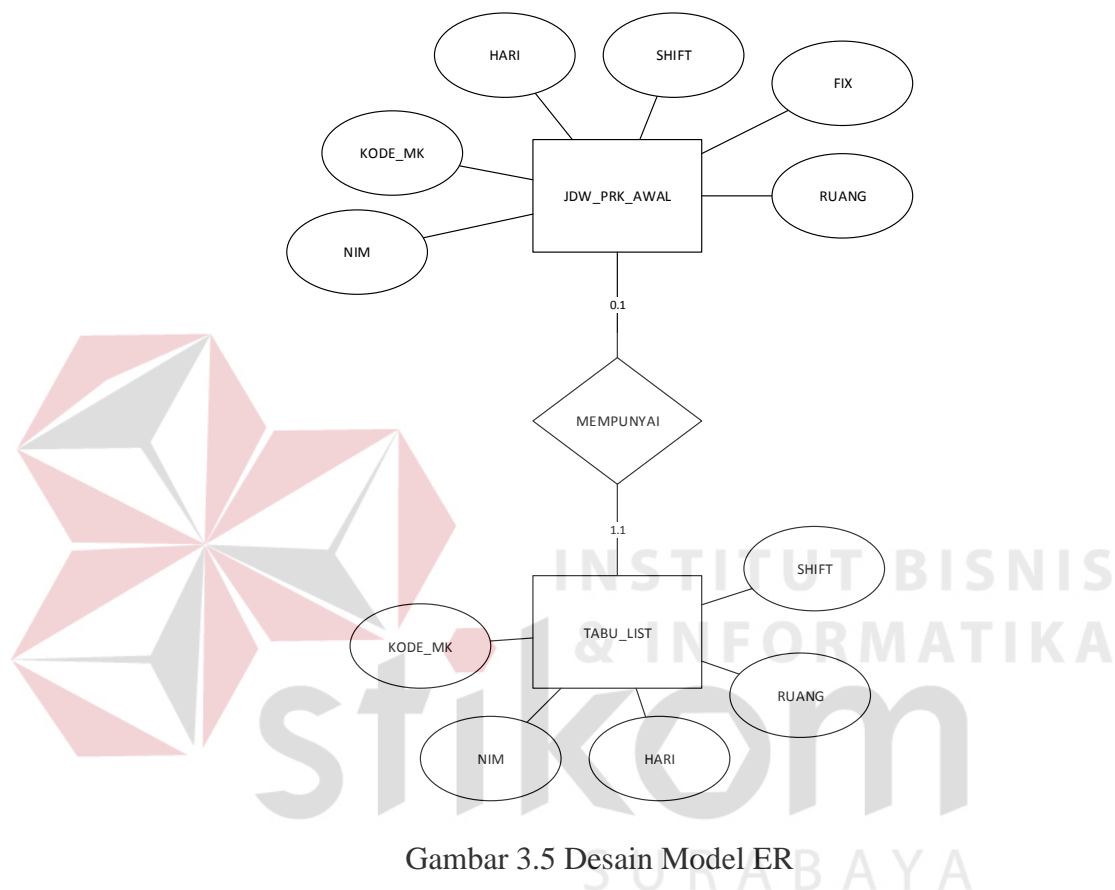
3.2.2 Desain Data

Setelah selesai menggambarkan desain proses diatas, maka dapat diketahui desain data yang dibutuhkan dalam menunjang berjalannya aplikasi penjadwalan praktikum. Dari Gambar 3.4 DFD level 0 aplikasi penjadwalan praktikum di atas, dibutuhkan dua buah desain data yang diperlukan dalam pembuatan aplikasi penjadwalan praktikum ini. Kedua data tersebut diantaranya adalah JDW_PRK_AWAL dan TABU_LIST. Kedua desain data tersebut akan digambarkan dalam bentuk ER-Model dan normalisasi

A. Desain Konseptual

Desain konseptual kali ini diawali dengan pembuatan desain *entity relationship model* (model ER). Model ER ini nantinya digunakan untuk memetakan hubungan antara entitas dalam proses yang akan ditangani oleh aplikasi, yang kemudian digunakan untuk mendesain model data konseptual. Desain model data konseptual digunakan untuk menentukan data apa saja yang

harus disimpan atau dibutuhkan pada sebuah entitas atau pada sebuah hubungan antar entitas. Dari desain data konseptual tersebut maka dapat dihasilkan model data fisikal, yaitu daftar tabel yang akan digunakan pada aplikasi. Desain model ER tersebut dapat dilihat pada Gambar 3.5.



Gambar 3.5 Desain Model ER

B. Normalisasi

Dari Model ER yang telah dibuat perlu dinormalisasikan. Di sini dilakukan tiga tahap normalisasi yaitu normalisasi pertama untuk mengecek apakah setiap nilai atribut dalam tabel ada yang memiliki dua nilai yang berbeda. Normalisasi kedua adalah pengecekan apakah tabel yang ada bergantung penuh pada *primary key* setiap tabel. Normalisasi ketiga adalah tidak adanya ketergantungan transitif, yaitu atribut bukan key tidak tergantung secara transitif terhadap atribut *key*.

1. Tabel JDW_PRK_AWAL

Tabel 3.3 Tabel JDW_PRK_AWAL

NIM	KODE_MK	HARI_MF_ID	SHIFT_MF_ID	RUANG_MF_ID	FIX
-----	---------	------------	-------------	-------------	-----

a. 1NF/ First Normal Form (Bentuk Normal Prima)

Kriteria	Pass
Semua nilai atribut harus simple/atomic yang tidak bisa dibagi-bagi lagi (tidak boleh ada atribut yang komposit/multivalued)	✓

b. 2NF/ First Normal Form (Bentuk Normal Kedua)

Kriteria	Pass
Memenuhi Kriteria 1NF	✓
Setiap attribute bergantung penuh pada primary key	✓

c. 3NF/ First Normal Form (Bentuk Normal Ketiga)

Kriteria	Pass
Memenuhi Kriteria 2NF	✓
Tidak ada ketergantungan transitif, yaitu atribut bukan key tidak tergantung secara transitif terhadap atribut key	✓

2. Tabel Jadwal

Tabel 3.4 Tabel TABU_LIST

NIM	KODE_MK	HARI_MF_ID	SHIFT_MF_ID	RUANG_MF_ID
-----	---------	------------	-------------	-------------

a. 1NF/ First Normal Form (Bentuk Normal Prima)

Kriteria	Pass
Semua nilai atribut harus simple/atomic	✓

Kriteria	Pass
yang tidak bisa dibagi-bagi lagi (tidak boleh ada atribut yang komposit/multivalue)	

b. *2NF/ First Normal Form* (Bentuk Normal Kedua)

Kriteria	Pass
Memenuhi Kriteria 1NF	✓
Setiap attribute bergantung penuh pada primary key	✓

c. *3NF/ First Normal Form* (Bentuk Normal Ketiga)

Kriteria	Pass
Memenuhi Kriteria 2NF	✓
Tidak ada ketergantungan transitif, yaitu atribut bukan key tidak tergantung secara transitif terhadap atribut key	✓

Dari hasil pengecekan diatas, dapat diketahui bahwa tabel *jdw_prk_awal* dan *tabu_list* telah memenuhi bentuk 3NF, dimana atribut bukan key yang ada di dalam kedua tabel tersebut tidak tergantung secara transitif terhadap atribut key.

C. CDM (*Conceptual Data Model*)

Pada conceptual data model (CDM) ini terdapat 2 entitas (tabel) antara entitas *jdw_prk_awal* dan *tabu_list*, dimana entitas *jdw_prk_awal* dan *tabu_list* bergantung pada entitas mahasiswa dengan hubungan relasi (one to one). Kedua tabel ini merupakan tabel asli yang telah dibuat oleh peneliti guna melengkapi tabel yang sudah disediakan oleh perusahaan untuk kepentingan penelitian ini. Desain CDM bisa dilihat pada Gambar 3.11.

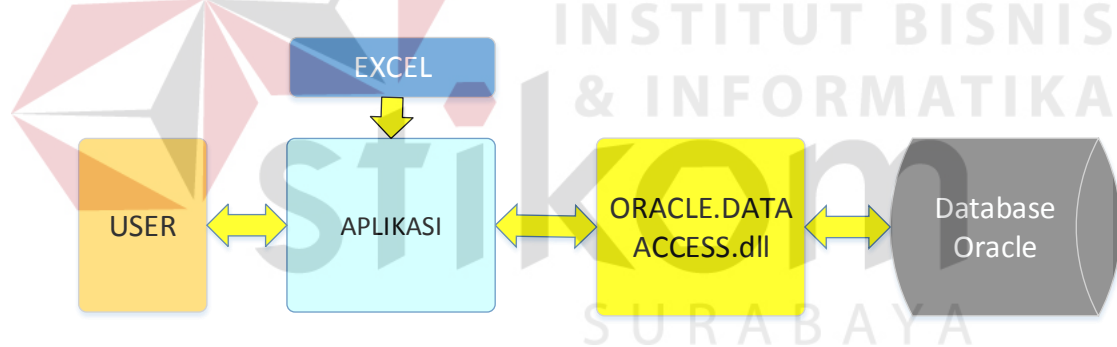


Gambar 3.6 CDM(*Conceptual Data Model*)

3.2.3 Desain Antarmuka

A Antarmuka Perangkat Lunak

Aplikasi penjadwalan praktikum yang akan dibuat ini membutuhkan data excel untuk dapat menghasilkan jadwal praktikum. Selain itu, dibutuhkan sebuah driver oracle yang bernama Oracle.DataAccess.dll, driver ini berfungsi untuk menghubungkan antara aplikasi dengan database oracle. Gambar 3.6 ini merupakan gambar desain antarmuka perangkat lunak.



Gambar 3.7 Gambar Antarmuka Perangkat Lunak

B Antarmuka Pengguna

Antar muka pengguna adalah sebuah titik dimana aplikasi dan user saling berinteraksi. Interaksi ini dapat melalui layar dan keyboard (interaksi langsung) atau melalui laporan yang dicetak dan form-form yang didesain untuk menangkap data (interaksi tidak langsung). Fokus desain antar muka pengguna adalah pada interaksi tidak langsung. Pada bagian ini, digambarkan terlebih dahulu alur kerja

GUI secara keseluruhan. Missal, dari form login lalu ke form utama, dan seterusnya

1. Desain *Form* Penjadwalan

Form penjadwalan digunakan oleh pengguna untuk melakukan proses penjadwalan praktikum. Pada *form* penjadwalan ini terdapat dua buah *textbox* yang berfungsi untuk menampung alamat dari data KRS dan data praktikan, dalam hal ini berupa data *excel*. Di dalam form ini juga terdapat empat buah *button*. Untuk *button* pilih digunakan untuk memilih data KRS dan data praktikan yang akan digunakan untuk proses penjadwalan praktikum. Sedangkan untuk *button* proses digunakan untuk memproses data KRS dan data Praktikan tersebut menjadi jadwal praktikum. *Button* batal digunakan untuk keluar dari aplikasi. Desain form penjadwalan dapat dilihat pada Gambar 3.8.

Gambar 3.8 Desain Form Penjadwalan

Alur fungsional utama untuk halaman penjadwalan adalah sebagai berikut

```
function proses()
```

```
{
```

```
    membuat jadwal inisialisasi
```

```
    mengambil data KRS dan data praktikan;
```

```
    menampung data KRS dan data praktikan ke dalam array;
```

melakukan perulangan sebanyak data praktikan yang mengambil mata praktikum

```

foreach($datapraktikan)
{
    mencari jadwal yang tidak berbenturan dengan jadwal
    kuliah reguler;
    mencari hari di mana praktikan memiliki jadwal kuliah
    reguler di bawah 3 mata kuliah;
    if (KRS praktikan < 3)
    {
        if (kriteria Labkom terpenuhi)
        {
            simpan data jadwal ke dalam tabel jadwal;
        }
        else
        {
            mencari jadwal lain yang cocok dengan
            kriteria Labkom
        }
    }
}
else
{
    mencari jadwal lain yang tidak melebihi jadwal mata
    kuliah praktikan
}
}

```

2. Desain Form Tampil Jadwal Praktikum

Form tampil jadwal ini digunakan untuk menampilkan jadwal yang sudah dibuat oleh aplikasi. Jadwal yang ditampilkan jadwal untuk satu mata praktikum. Pada form ini terdapat satu buah data grid view yang berguna untuk menampilkan jadwal praktikum harian. Gambar 3.8 menunjukkan desain dari form tampil jadwal praktikum.

Tampil Jadwal Praktikum

Nama Lab Shift

Nama Praktikum

NIM	Nama
Text	Text
Text	Text
Text	Text

Gambar 3.9 Desain Form Tampil Jadwal Praktikum

3. Desain Form Tampil KRS Praktikan

Form tampil KRS praktikan ini digunakan untuk menampilkan KRS praktikan. Form ini berfungsi untuk membuktikan penggunaan metode tabu search sehingga, satu praktikan tidak memiliki jadwal mata kuliah atau praktikum yang melebihi tiga jadwal. Gambar 3.10 menunjukkan desain form tampil KRS praktikan.

Tampil KRS Praktikan

NIM : xxxxxx
Nama : xxxxxx

Kode MK	HARI	Jam Mulai
Text	Text	Text
Text	Text	Text
Text	Text	Text

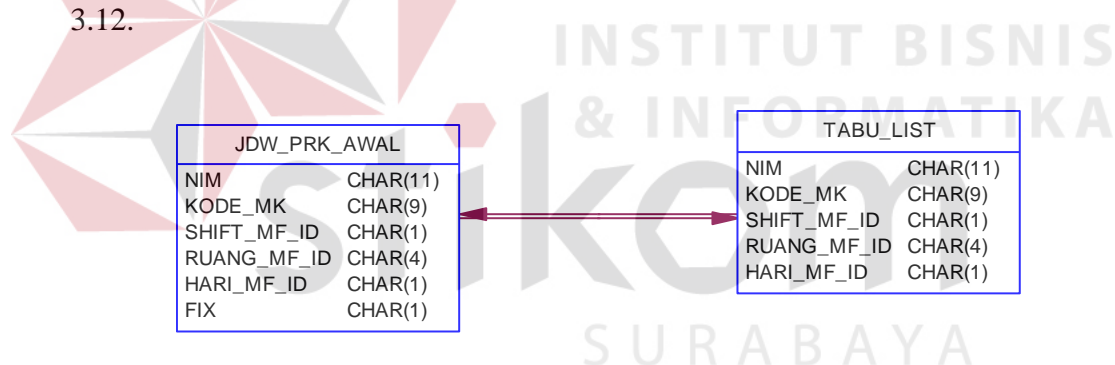
Gambar 3.10 Desain Form Tampil KRS Praktikan

3.2.4 Desain Fisik

Setelah mengetahui desain data yang dibutuhkan, maka langkah selanjutnya adalah menggambarkan desain fisik. Dalam aplikasi penjadwalan tambat kapal ini, *Database management systems* (DBMS) yang digunakan adalah *oracle database*. Terdapat 3 tabel baru yang dibutuhkan dalam aplikasi ini, yaitu tabel KRS, mahasiswa dan jadwal. Ketiga tabel tersebut akan digambarkan dalam *Physical Data Model* (PDM).

a. PDM (*Physical Data Model*)

Pada *Physical Data Model* (PDM) ini, terdapat 3 tabel sama halnya dengan *Conceptual Data Model* yang telah dibuat sebelumnya. PDM ini dihasilkan dari proses generate CDM diatas. Desain PDM tersebut dapat dilihat pada Gambar 3.12.



Gambar 3.11 PDM(*Physical Data Model*)

b. Struktur Tabel

1. Nama Tabel : JDW_PRK_AWAL

Keterangan : Untuk menyimpan data jadwal praktikum.

Tabel 3.5 Struktur Tabel JDW_PRK_AWAL

Nama Kolom	Tipe Data	Constraint	Keterangan
NIM	Char(11)	FK	NIM Mahasiswa
KODE_MK	Char(9)	FK	Kode Mata Kuliah
HARI_MF_ID	Char(1)	FK	Hari Praktikum

Nama Kolom	Tipe Data	Constraint	Keterangan
SHIFT_MF_ID	Char(1)	FK	Shift Praktikum
RUANG_MF_ID	Char(4)	FK	Ruang Praktikum
FIX	Char(1)	NN	

2. Nama Tabel : TABU_LIST

Keterangan : Untuk menyimpan data tabu.

Tabel 3.6 Struktur Tabel TABU_LIST

Nama Kolom	Tipe Data	Constraint	Keterangan
NIM	Char(11)	FK	NIM Mahasiswa
KODE_MK	Char(9)	FK	Kode Mata Kuliah
HARI_MF_ID	Char(1)	FK	Hari Praktikum
SHIFT_MF_ID	Char(1)	FK	Shift Praktikum
RUANG_MF_ID	Char(4)	FK	Ruang Praktikum

3.2.5 Perancangan Algoritma *Tabu Search*

Algoritma tabu search pada penjadwalan praktikum ini dibagi menjadi empat bagian di mana terdiri dari satu bagian utama dan tiga bagian pendukung yaitu pembuatan jadwal inisialisasi, pengecekan jadwal inisialisasi dan pengecekan tabu list. Keempat algoritma tersebut dihubungkan dengan connector off page antara satu dengan lainnya. Tujuan dari pembagian tersebut tidak lain adalah memudahkan dalam pembacaan algoritma tabu search tersebut. Di bawah ini merupakan penjelasan dari keempat algoritma tabu search penjadwalan praktikum.

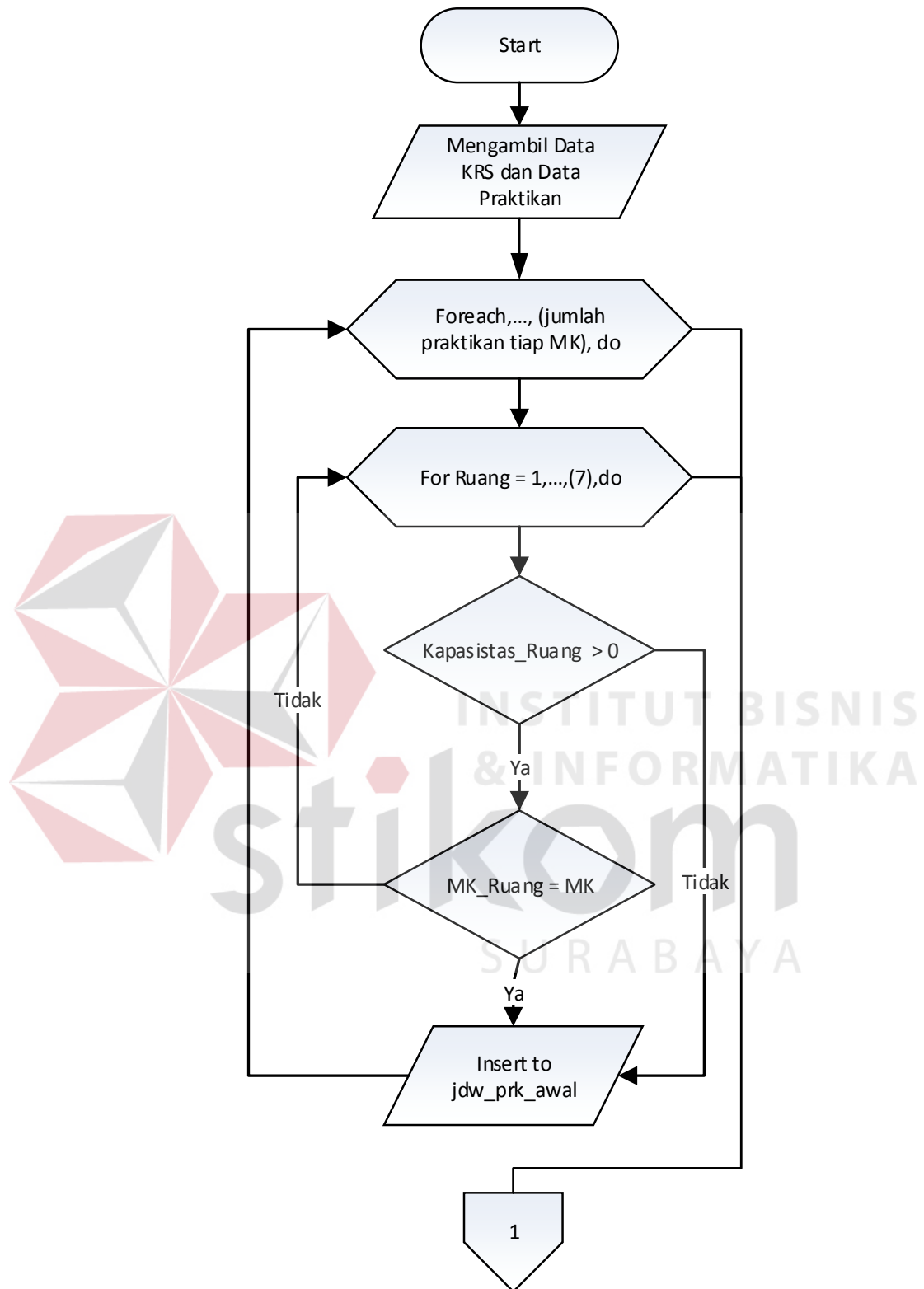
Algoritma tabu search ini dimulai dari bagian pertama, di mana algoritma akan mengambil data KRS dan data praktikan. Data tersebut akan langsung diolah pada algoritma bagian kedua untuk proses pembuatan jadwal inisialisasi. Proses pembuatan jadwal inisialisasi ini dimulai dengan perulangan data praktikan untuk

tiap mata praktikum. Dilanjutkan dengan proses perulangan untuk mencari ruang laboratorium yang cocok untuk praktikan tersebut. Apabila sudah menemukan kecocokan antara ruangan dan praktikan, maka proses akan dilanjutkan ke bagian ketiga.

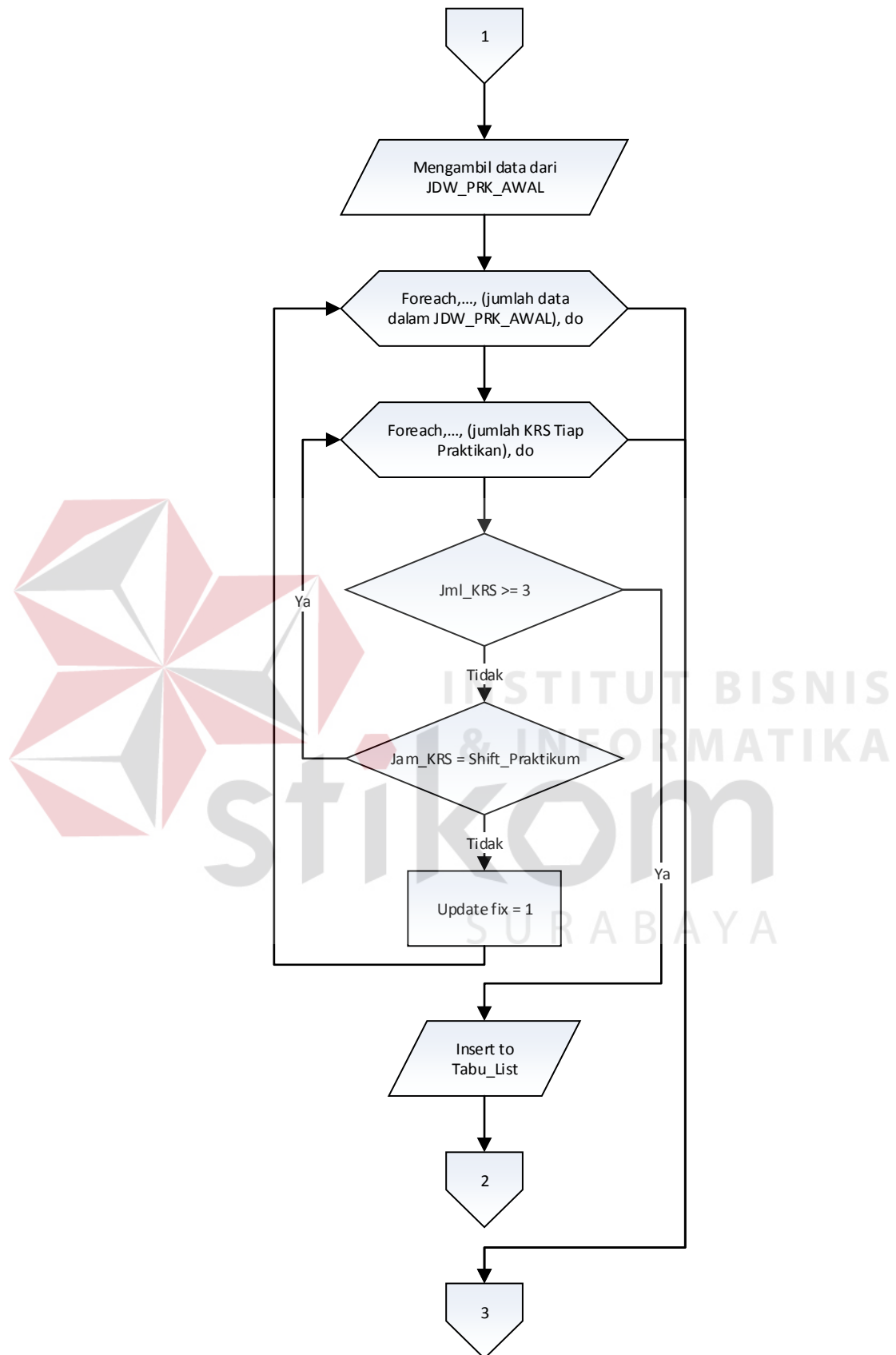
Pada algoritma bagian ketiga ini, proses di mulai dengan mengambil data dari hasil penjadwalan inisialisasi, data tersebut akan dibandingkan dengan data KRS masing-masing praktikan. Apabila KRS praktikan pada hari dan shift tersebut cocok maka, proses dilanjutkan dengan memperbaharui data pada tabel JDW_PRK_AWAL. Jika tidak menemui kecocokan maka, data praktikan yang tidak cocok tersebut akan dimasukkan ke dalam tabel TABU_LIST.

Pada algoritma bagian keempat ini, proses di mulai dengan mengambil data dari tabel TABU_LIST. Pada proses ini data pada tabel TABU_LIST akan dibandingkan dengan KRS praktikan. Apabila ada kecocokan antara KRS dan data pada tabel TABU_LIST, maka data pada TABU_LIST akan dihapus dan dipindah ke dalam tabel JDW_PRK_AWAL. Apabila tidak ada kecocokan antara KRS dan data pada tabel TABU_LIST, maka akan dilakukan perulangan untuk hari dan shift berikutnya.

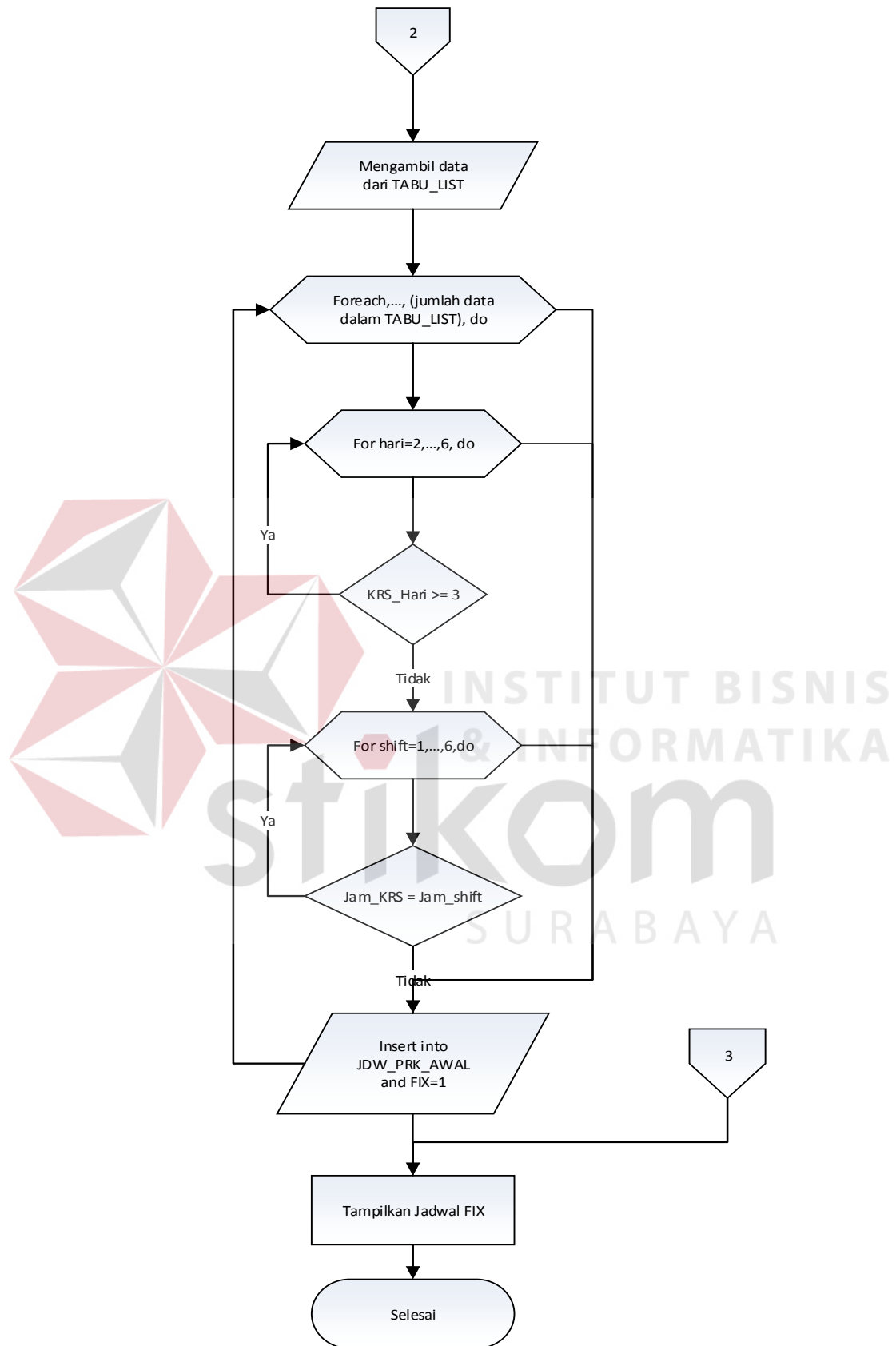
Gambar flowchart dari proses penjadwalan inisialisasi dapat dilihat pada Gambar 3.12. Gambar flowchart dari proses pengecekan jadwal inisialisasi dapat dilihat pada Gambar 3.13. Gambar flowchart dari proses pengecekan tabu list dapat dilihat pada Gambar 3.14.



Gambar 3.12 Algoritma Penjadwalan Inisialisasi



Gambar 3.13 Algoritma Pengecekan Jadwal Inisialisasi



Gambar 3.14 Algoritma Pengecekan *Tabu List*

3.3 Desain Uji Coba

3.3.1 Desain Uji Fungsional

Desain uji fungsional ini digunakan sebagai dasar melakukan pengujian terhadap aplikasi dan untuk membuktikan bahwa fungsi dari metode Tabu Search pada aplikasi sudah berjalan sesuai dengan semestinya. Pengujian fungsional ini akan dilakukan dengan membandingkan antara hasil dari penjadwalan praktikum menggunakan metode *tabu search* secara manual dan hasil dari penjadwalan praktikum menggunakan metode *tabu search* secara aplikasi. Diharapkan setelah pengujian akan ada kesamaan hasil dari penjadwalan secara manual dan aplikasi.

Pada pengujian fungsional ini akan menggunakan data KRS semester 142.

Data yang digunakan dalam penjadwalan manual dapat dilihat pada Tabel 3.7.

Tabel 3.7 Data KRS Semester 142

MHS_NIM	kode MK	Hari	Mulai	selesai
08410100071	410104015	1	07:32	07:33
08410100071	410103092	2	13:30	16:00
08410100071	410102053	3	10:30	13:00
08410100071	410102055	3	13:30	15:10
08410100071	410102054	4	15:45	17:25
08410100071	410103083	5	10:30	13:00
08410100071	410103086	5	15:15	17:45
08410100071	410103078	6	07:30	10:00
08410100071	410109041	7	07:42	07:43
08410100071	410109042	7	07:58	07:59
08410100071	410109044	7	07:50	07:51

Terdapat beberapa langkah yang harus dilalui dalam perhitungan *tabu search* secara manual ini. Langkah-langkah tersebut diantaranya sebagai berikut.

1. Mengambil data KRS dan data praktikan.

2. Memasukkan semua praktikan yang memiliki mata praktikum yang sama dalam satu laboratorium dengan kapasitas 18 peserta.
3. Melakukan pengecekan jumlah KRS dalam satu hari apabila dalam satu hari sudah ada tiga jadwal mata kuliah, maka praktikan tersebut akan dipindah di hari yang tidak memiliki tiga jadwal mata kuliah .
4. Melakukan pengecekan KRS dengan jadwal praktikum apabila ada jadwal mata kuliah dan jadwal mata praktikum yang berbenturan maka, praktikan tersebut akan dipindah di mana tidak ada jadwal yang tidak berbenturan.
5. Hasil akhir dari pemrosesan manual ini dapat dilihat pada Tabel 3.8.

Tabel 3.8 Hasil *Tabu Search* Secara Manual

NIM	KODE_MK	HARI	SHIFT	RUANG
08410100071	410109041	2	1	B606
08410100071	410109042	2	2	B609
08410100071	410109044	3	1	B610

3.3.2 Desain Uji Data

Desain uji data ini digunakan sebagai dasar melakukan pengujian terhadap aplikasi dan membuktikan bahwa hasil penjadwalan menggunakan metode tabu search dari aplikasi sudah menjawab rumusan masalah di atas. Pengujian data ini akan dilakukan dengan membandingkan antara hasil dari penjadwalan secara manual yang lakukan oleh Labkom dan hasil dari penjadwalan secara aplikasi. Diharapkan setelah pengujian hasil dari aplikasi dapat menjawab rumusan masalah di atas.