

**BAB III**  
**DESAIN DAN IMPLEMENTASI**  
**SISTEM INFORMASI BIMBINGAN STUDI**

**3.1. Fase Requirement**

Fase ini merupakan bagian dari proses perancangan basisdata di dalam metode penelitian. Ada tiga teknik yang dilakukan penulis untuk mengumpulkan data atau mendapatkan daftar kebutuhan, yaitu:

1. Pencerapan secara cermat (observasi) terhadap kegiatan bimbingan studi.
2. Wawancara kepada beberapa DPA.
3. *Workshop Software Sistem Perwalian Akademik.*

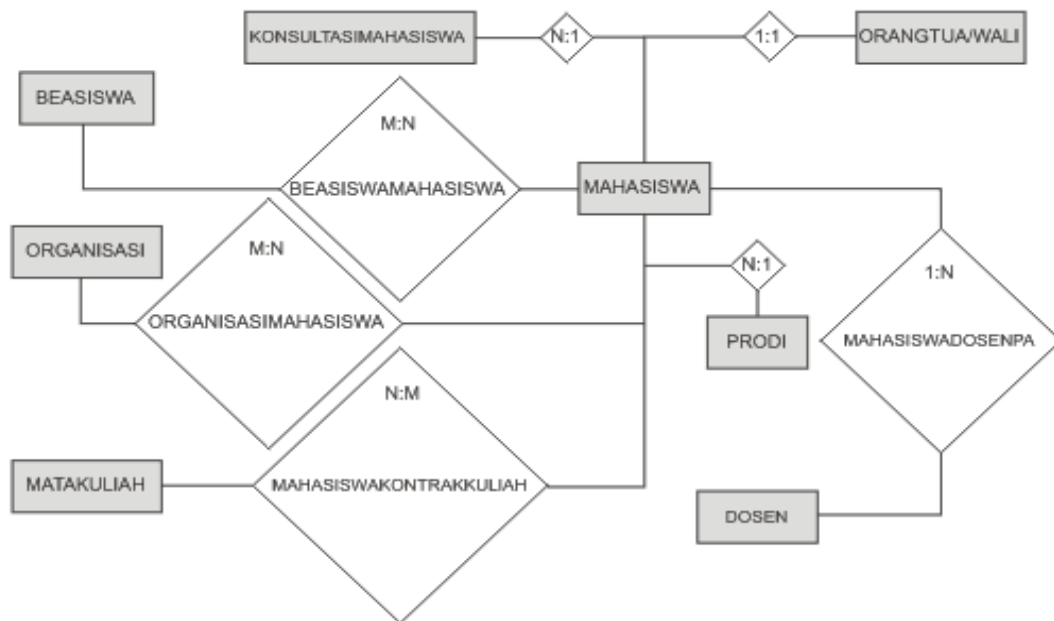
Hasil dari teknik pengumpulan data (berupa daftar kebutuhan) dapat dilihat secara rinci di dalam daftar lampiran (Lampiran 3.1: Daftar Kebutuhan) pada tugas akhir ini. Dari daftar kebutuhan, dapat disimpulkan bahwa:

- A. *Stake holder* dari Sistem Informasi Bimbingan Studi ini ialah DPA.
- B. Sistem ini dirancang sebagai media dalam kegiatan bimbingan studi, bukan sebagai media pembelajaran.
- C. Fungsi dari sistem ini ialah sebagai alat yang digunakan DPA untuk menyimpan dan mengelola data mahasiswa bimbingannya dalam kegiatan bimbingan studi.

D. Fitur-fitur yang ada di dalam sistem ialah:

1. Data mahasiswa, baik individu (lengkap) atau dikelompokkan berdasarkan DPA, minat akademik, perolehan Indeks Prestasi Kumulatif (IPK).
2. Data mahasiswa yang mendapatkan beasiswa.
3. Data mahasiswa yang aktif mengikuti kegiatan organisasi kemahasiswaan.
4. Catatan konsultasi mahasiswa.
5. Riwayat studi, transkrip nilai, dan Indeks Prestasi (IP) mahasiswa.

Selanjutnya dalam proses perancangan basisdata, menentukan kelas entitas dan hubungannya. Untuk menentukan kelas entitas dan hubungannya penulis mencermati realitas kegiatan bimbingan studi dan mengkaitkan antara hasil pemcermatan dengan teori himpunan (digunakan untuk menentukan kelas entitas) dan logika matematika (teknik berpikir logis, digunakan untuk menentukan hubungan antar entitas atau kelas entitas) sedemikian sehingga diperoleh dua belas kelas entitas yang terdiri dari delapan kelas entitas murni dan empat kelas entitas khusus; dan tujuh hubungan (*relationship*). Gambar 3.1. memperlihatkan kelas entitas dan hubungannya.



Gambar 3.1. Diagram *Extended E-R (entity-relationship)*.

Kelas entitas dilambangkan dengan segi empat, hubungan dilambangkan dengan ketupat atau wajik, dan kardinalitas maksimum atau minimum diperlihatkan dalam ketupat. Nama kelas entitas ditulis di dalam segi empat, dan nama hubungan ditulis di dalam wajik (bagian bawah). MAHASISWA dan KONSULTASIMAHASISWA, misalnya, masing-masing merupakan kelas entitas dan dihubungkan oleh lambang wajik dengan kardinalitas 1:N. Artinya bahwa seorang mahasiswa dapat melakukan konsultasi lebih dari satu kali. Kelas entitas dan hubungannya dapat dilihat secara rinci di dalam daftar lampiran (Lampiran 3.2: Deskripsi Entitas dan Hubungannya) pada tugas akhir ini.

Kemudian dalam proses pemodelan data, penulis menentukan *identifer* atau *primary key* dari masing-masing kelas entitas maupun kelas entitas khusus. “*Primary key* adalah kolom yang berfungsi membedakan satu baris dengan baris lainnya (membuat setiap baris jadi unik)” (Wibisono, 2006). *Primary key*

ditentukan dengan alasan bahwa kolom tersebut dapat mewakili baris di dalam kelas entitas (tabel) tersebut. Perhatikan tabel 3.1 berikut:

Tabel 3.1. *Identifer* atau *Primary Key*.

Entitas	<i>Identifer</i> atau <i>primary key</i>
MAHASISWA	NIM
BEASISWA	KodeBeasiswa
ORGANISASI	KodeOrganisasi
MATAKULIAH	KodeMatakuliah
DOSEN	NIP dan KodeDosen
PRODI	KodeProdi
KONSULTASIMAHASISWA	NIM, WaktuKonsultasi, dan PrihalKonsultasi
ORANGTUA/WALI	NIM
BEASISWAMAHASISWA	NIM, KodeBeasiswa, dan TanggalTerima
ORGANISASIMAHASISWA	NIM dan KodeOrganisasi
MAHASISWAKONTRAKKULIAH	ID, NIM, KodeMataKuliah dan Semester
MAHASISWADOSENPA	NIM

Dua langkah selanjutnya, menetapkan atribut dan domain dari masing-masing entitas. Atribut diperoleh dari daftar kebutuhan dan hasil penentuan entitas, sedangkan domain diperoleh setelah menentukan atribut (menyesuaikan dengan atribut). Misal dalam kelas entitas mahasiswa, penulis menentukan nama mahasiswa sebagai suatu atribut dengan domain berupa *text*.

Namun apabila nama beasiswa dijadikan atribut di dalam kelas entitas mahasiswa, maka penentuan entitas ini absurd. Karena tidak tepat apabila hal yang tidak berkaitan dengan identitas mahasiswa dimasukkan ke dalam kelas entitas mahasiswa. Seharusnya atribut nama beasiswa dimasukkan ke dalam Kelas entitas beasiswa. Jadi, dalam menentukan atribut harus menyesuaikan dengan kelas entitas. Hasilnya dapat dilihat di dalam daftar lampiran (Lampiran 3.3: Kelas Entitas, Atribut, dan Domain) pada tugas akhir ini.

### 3.2. Fase Desain

Hasil dari proses mentransformasi model *entity-relationship* (E-R) ke dalam desain basisdata relasional sebagai berikut:

#### 1. Kelas Entitas Mahasiswa ke dalam tabel MAHASISWA

Di bagian atas dalam tabel (sebelah kiri, yang merepresentasikan kelas entitas mahasiswa), yaitu NIM merupakan calon dari *identifer* atau *primary key* (ditandai dengan gambar sebuah kunci) dari tabel MAHASISWA. *Null* pada tabel MAHASISWA memiliki arti bahwa kolom yang diberi identitas *null* boleh tidak diisi, sedangkan kolom dengan identitas *not null* tidak boleh tidak, harus diisi.

Kelas Entitas Mahasiswa

NIM
Nama Mahasiswa
Tempat Lahir
Tanggal Lahir
Sex
Agama
Alamat Email
Alamat
Kode Pos
Kota
Telepon
Handphone Mahasiswa
Attachments
KodeProdi
Angkatan
Kelas
Minat Akademik
Asal Sekolah

Transformasi →

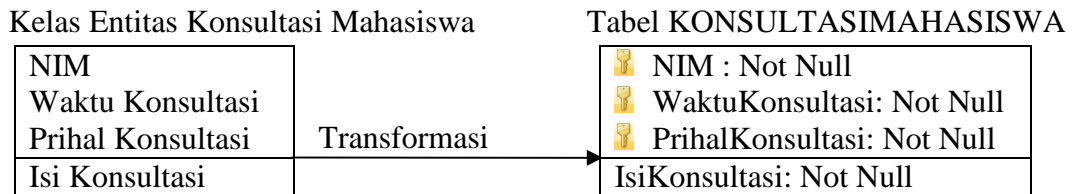
Tabel MAHASISWA

 NIM : <i>Not Null</i>
NamaMahasiswa: <i>Not Null</i>
TempatLahir: <i>Not Null</i>
TanggalLahir: <i>Not Null</i>
Sex: <i>Not Null</i>
Agama: <i>Not Null</i>
AlamatEmail: <i>Null</i>
Alamat: <i>Not Null</i>
KodePos: <i>Null</i>
Kota: <i>Null</i>
Telepon: <i>Null</i>
HandphoneMahasiswa: <i>Not Null</i>
Attachments: <i>Not Null</i>
KodeProdi: <i>Not Null</i>
Angkatan: <i>Not Null</i>
Kelas: <i>Not Null</i>
MinatAkademik: <i>Not Null</i>
AsalSekolah: <i>Not Null</i>

#### 2. Kelas Entitas Konsultasi Mahasiswa ke dalam tabel KONSULTASIMAHASISWA

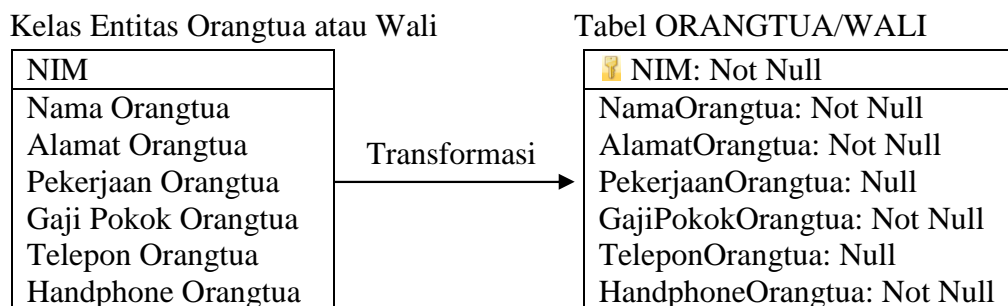
Di bagian atas dalam tabel (sebelah kiri, yang merepresentasikan kelas entitas konsultasi mahasiswa), yaitu NIM, Waktu Konsultasi, dan Prihal Konsultasi merupakan calon dari *identifer* atau *primary key* (ditandai dengan gambar sebuah kunci) dari tabel KONSULTASIMAHASISWA. *Not null* pada tabel

KONSULTASIMAHASISWA memiliki arti bahwa kolom yang diberi identitas *not null* tidak boleh tidak, harus diisi.



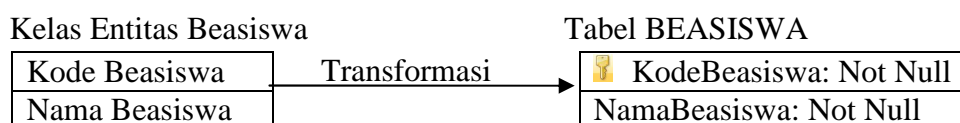
### 3. Kelas Entitas Orangtua atau Wali ke dalam tabel ORANGTUA/WALI

Di bagian atas dalam tabel (sebelah kiri, yang merepresentasikan kelas entitas orangtua atau wali), yaitu NIM merupakan calon dari *identifer* atau *primary key* (ditandai dengan gambar sebuah kunci) dari tabel ORANGTUA/WALI. *Null* pada tabel ORANGTUA/WALI memiliki arti bahwa kolom yang diberi identitas *null* boleh tidak diisi, sedangkan kolom dengan identitas *not null* tidak boleh tidak, harus diisi.



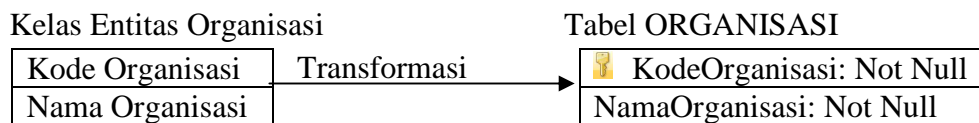
### 4. Kelas Entitas Beasiswa ke dalam tabel BEASISWA

Di bagian atas dalam tabel (sebelah kiri, yang merepresentasikan kelas entitas Beasiswa), yaitu Kode Beasiswa merupakan calon dari *identifer* atau *primary key* (ditandai dengan gambar sebuah kunci) dari tabel BEASISWA. *Not null* pada tabel BEASISWA memiliki arti bahwa kolom yang diberi identitas *not null* tidak boleh tidak, harus diisi.



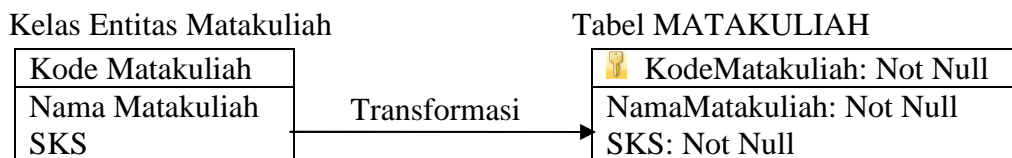
## 5. Kelas Entitas Organisasi ke dalam tabel ORGANISASI

Di bagian atas dalam tabel (sebelah kiri, yang merepresentasikan kelas entitas organisasi), yaitu Kode Organisasi merupakan calon dari *identifer* atau *primary key* (ditandai dengan gambar sebuah kunci) dari tabel ORGANISASI. *Not null* pada tabel ORGANISASI memiliki arti bahwa kolom yang diberi identitas *not null* tidak boleh tidak, harus diisi.



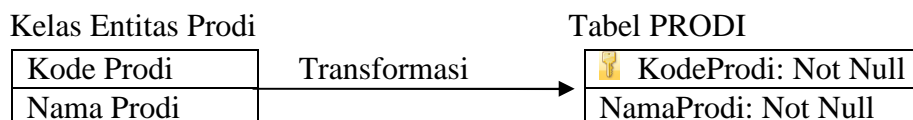
## 6. Kelas Entitas Matakuliah ke dalam tabel MATAKULIAH

Di bagian atas dalam tabel (sebelah kiri, yang merepresentasikan kelas entitas matakuliah), yaitu Kode Matakuliah merupakan calon dari *identifer* atau *primary key* (ditandai dengan gambar sebuah kunci) dari tabel MATAKULIAH. *Not null* pada tabel MATAKULIAH memiliki arti bahwa kolom yang diberi identitas *not null* tidak boleh tidak, harus diisi.



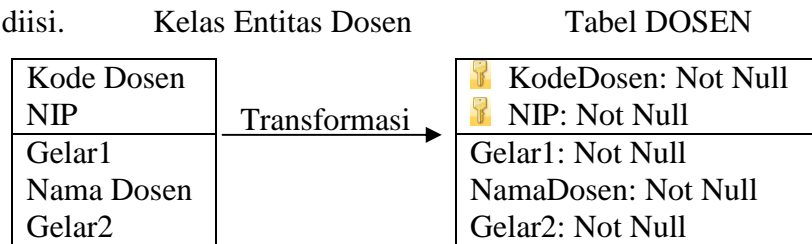
## 7. Kelas Entitas Prodi ke dalam tabel PRODI

Di bagian atas dalam tabel (sebelah kiri, yang merepresentasikan kelas entitas Prodi), yaitu Kode Prodi merupakan calon dari *identifer* atau *primary key* (ditandai dengan gambar sebuah kunci) dari tabel PRODI. *Not null* pada tabel PRODI memiliki arti bahwa kolom yang diberi identitas *not null* tidak boleh tidak, harus diisi.



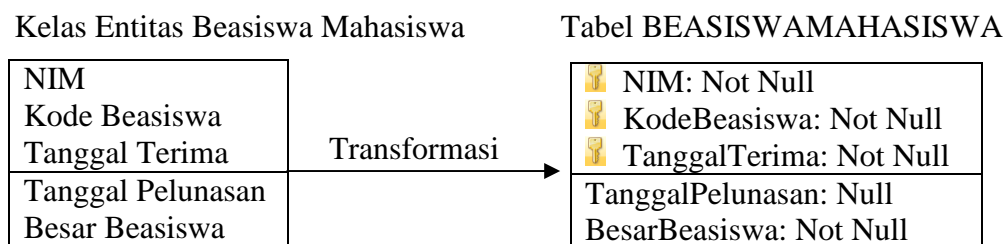
## 8. Kelas Entitas Dosen ke dalam tabel DOSEN

Di bagian atas dalam tabel (sebelah kiri, yang merepresentasikan kelas entitas dosen), yaitu Kode Dosen dan NIP merupakan calon dari *identifer* atau *primary key* (ditandai dengan gambar sebuah kunci) dari tabel DOSEN. *Not null* pada tabel DOSEN memiliki arti bahwa kolom yang diberi identitas *not null* tidak boleh tidak, harus diisi.



## 9. Kelas Entitas BeasiswaMahasiswa ke dalam tabel BEASISWAMAHASISWA

Di bagian atas dalam tabel (sebelah kiri, yang merepresentasikan kelas entitas beasiswa mahasiswa), yaitu NIM, Kode Beasiswa, dan Tanggal Terima merupakan calon dari *identifer* atau *primary key* (ditandai dengan gambar sebuah kunci) dari tabel BEASISWAMAHASISWA. *Null* pada tabel BEASISWAMAHASISWA memiliki arti bahwa kolom yang diberi identitas *null* boleh tidak diisi, sedangkan kolom dengan identitas *not null* tidak boleh tidak, harus diisi.

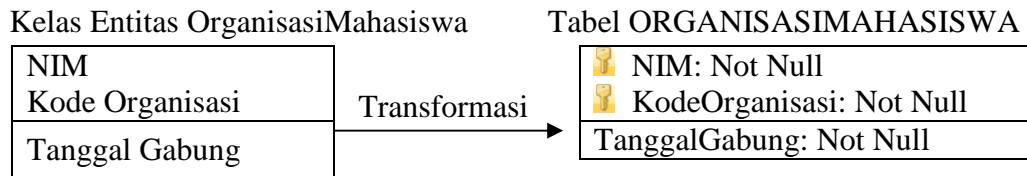


## 10. Entitas Organisasi Mahasiswa ke dalam tabel ORGANISASIMAHASISWA

Di bagian atas dalam tabel (sebelah kiri, yang merepresentasikan kelas entitas Organisasi Mahasiswa), yaitu NIM dan Kode Organisasi merupakan calon dari *identifer* atau *primary key* (ditandai dengan gambar sebuah kunci) dari tabel ORGANISASIMAHASISWA. *Not null* pada tabel ORGANISASIMAHASISWA

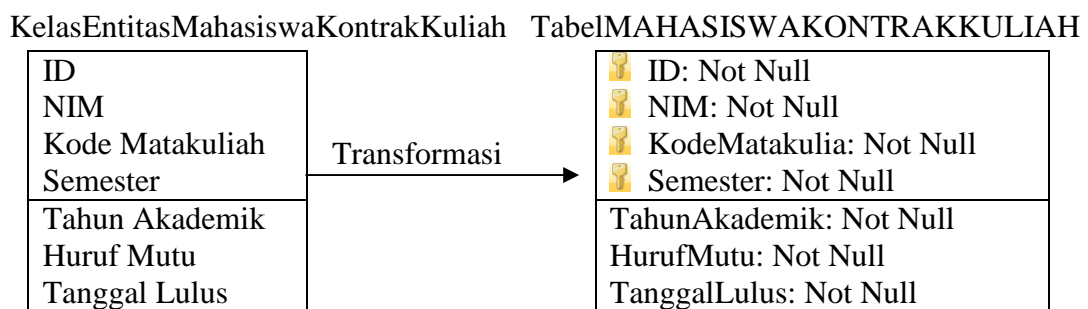


memiliki arti bahwa kolom yang diberi identitas *not null* tidak boleh tidak, harus diisi.



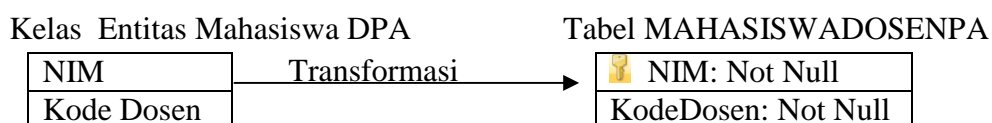
### 11. Kelas Entitas Mahasiswa Kontrak Kuliah ke dalam tabel MAHASISWAKONTRAKKULIAH

Di bagian atas dalam tabel (sebelah kiri, yang merepresentasikan kelas entitas mahasiswa kontrak kuliah), yaitu ID, NIM, Kode Matakuliah dan Semester merupakan calon dari *identifer* atau *primary key* (ditandai dengan gambar sebuah kunci) dari tabel MAHASISWAKONTRAKKULIAH. *Not null* pada tabel MAHASISWAKONTRAKKULIAH memiliki arti bahwa kolom yang diberi identitas *not null* tidak boleh tidak, harus diisi.

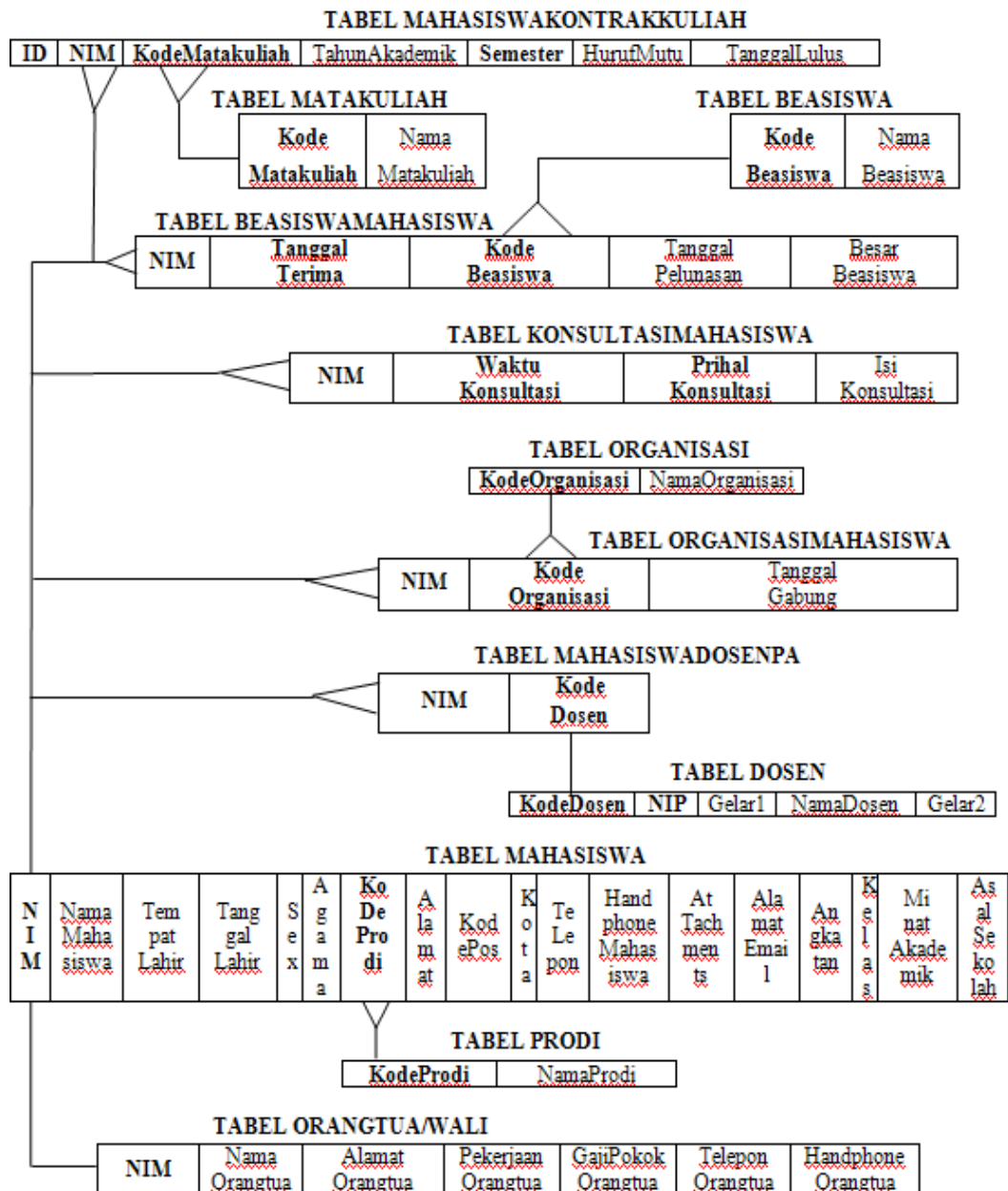


### 12. Kelas Entitas Mahasiswa DPA ke dalam tabel MAHASISWADOSENPA

Di bagian atas dalam tabel (sebelah kiri, yang merepresentasikan kelas entitas DPA), yaitu NIM merupakan calon dari *identifer* atau *primary key* (ditandai dengan gambar sebuah kunci) dari tabel DOSENPA. *Not null* pada tabel DOSENPA memiliki arti bahwa kolom yang diberi identitas *not null* tidak boleh tidak, harus diisi.



Selanjutnya, melakukan elaborasi dari sebuah contoh struktur data Sistem Informasi Bimbingan Studi, yaitu diagram yang digunakan untuk memperlihatkan tabel dan hubungannya. Berikut adalah penjelasan lebih rinci dari gambar 3.2 mengenai representasi dari garis dan cabang (*fork*).



Gambar 3.2. Struktur Data Sistem Informasi Bimbingan Studi.

## **1. Hubungan di antara tabel MAHASISWA dengan tabel MATAKULIAH**

Tabel MAHASISWAKONTRAKKULIAH menjadi relasi (tabel khusus) yang menghubungkan antara tabel MAHASISWA dengan tabel MATAKULIAH. Hubungan di antara tabel MAHASISWA dengan tabel MAHASISWAKONTRAKKULIAH dan tabel MAHASISWAKONTRAKKULIAH dengan tabel MATAKULIAH masing-masing pada atribut NIM dan KodeMatakuliah berarti bahwa sebuah baris pada tabel MAHASISWA dapat berkaitan dengan banyak baris di tabel MAHASISWAKONTRAKKULIAH (karena cabang berada di tabel MAHASISWAKONTRAKKULIAH) dan sebuah baris pada tabel MATAKULIAH dapat berkaitan dengan banyak baris di tabel MAHASISWAKONTRAKKULIAH. Dengan kata lain, seorang mahasiswa dapat mengkontrak lebih dari satu matakuliah dan sebuah matakuliah dapat di kontrak banyak mahasiswa.

## **2. Hubungan di antara tabel MAHASISWA dengan tabel ORGANISASI**

Tabel ORGANISASIMAHASISWA menjadi relasi (tabel khusus) yang menghubungkan antara tabel MAHASISWA dengan tabel ORGANISASI. Hubungan di antara tabel MAHASISWA dengan tabel ORGANISASIMAHASISWA dan tabel ORGANISASIMAHASISWA dengan tabel ORGANISASI masing-masing pada atribut NIM dan KodeOrganisasi berarti bahwa sebuah baris pada tabel MAHASISWA dapat berkaitan dengan banyak baris di tabel ORGANISASIMAHASISWA (karena cabang berada di tabel ORGANISASIMAHASISWA) dan sebuah baris pada tabel ORGANISASI dapat berkaitan dengan banyak baris di tabel ORGANISASIMAHASISWA. Dengan kata

lain, seorang mahasiswa dapat bergabung atau mengikuti lebih dari satu organisasi dan sebuah organisasi dapat diikuti banyak mahasiswa.

### **3. Hubungan di antara tabel MAHASISWA dengan tabel BEASISWA**

Tabel BEASISWAMAHASISWA menjadi relasi (tabel khusus) yang menghubungkan antara tabel MAHASISWA dengan tabel BEASISWA. Hubungan di antara tabel MAHASISWA dengan tabel BEASISWAMAHASISWA dan tabel BEASISWAMAHASISWA dengan tabel BEASISWA masing-masing pada atribut NIM dan KodeBeasiswa berarti bahwa sebuah baris pada tabel MAHASISWA dapat berkaitan dengan banyak baris di tabel BEASISWAMAHASISWA (karena cabang berada di tabel BEASISWAMAHASISWA) dan sebuah baris pada tabel BEASISWA dapat berkaitan dengan banyak baris di tabel BEASISWAMAHASISWA. Dengan kata lain, seorang mahasiswa dapat memperoleh lebih dari satu beasiswa dan satu beasiswa dapat diperoleh banyak mahasiswa.

### **4. Hubungan di antara tabel MAHASISWA dengan tabel KONSULTASIMAHASISWA**

Hubungan di antara tabel MAHASISWA dengan tabel KONSULTASIMAHASISWA pada atribut NIM berarti bahwa sebuah baris pada tabel MAHASISWA dapat berkaitan dengan banyak baris di tabel KONSULTASIMAHASISWA (karena cabang berada di tabel KONSULTASIMAHASISWA). Dengan kata lain, seorang mahasiswa dapat melakukan lebih dari satu kali konsultasi kepada DPA.

## **5. Hubungan di antara tabel MAHASISWA dengan tabel ORANGTUA/WALI**

Hubungan di antara tabel MAHASISWA dengan tabel ORANGTUA/WALI pada atribut NIM berarti bahwa sebuah baris di tabel MAHASISWA berikatan dengan tepat satu baris pada tabel ORANGTUA/WALI. Dengan kata lain, seorang mahasiswa memiliki tepat satu orangtua atau wali.

## **6. Hubungan di antara tabel MAHASISWA dengan tabel DOSEN**

Tabel MAHASISWADOSENPA menjadi relasi (tabel khusus) yang menghubungkan antara tabel MAHASISWA dengan tabel DOSEN. Hubungan di antara tabel MAHASISWA dengan tabel MAHASISWADOSENPA dan tabel MAHASISWADOSENPA dengan tabel DOSEN masing-masing pada atribut NIM dan KodeDosen berarti bahwa sebuah baris pada tabel MAHASISWA dapat berkaitan dengan banyak baris di tabel MAHASISWADOSENPA (karena cabang berada di tabel MAHASISWADOSENPA) dan sebuah baris pada tabel DOSEN dapat berkaitan dengan banyak baris di tabel MAHASISWADOSENPA. Dengan kata lain, lebih dari satu orang (banyak) mahasiswa memiliki satu DPA dan satu DPA dapat memiliki lebih dari satu mahasiswa bimbingan.

## **7. Hubungan di antara tabel MAHASISWA dengan tabel PRODI**

Hubungan di antara tabel MAHASISWA dengan tabel PRODI pada atribut KodeProdi berarti bahwa lebih dari satu baris pada tabel MAHASISWA dapat berkaitan dengan tepat satu baris di tabel PRODI (karena cabang berada di tabel MAHASISWA). Dengan kata lain, lebih dari satu orang (banyak) mahasiswa dapat memiliki tepat satu prodi.

### 3.3. Fase Implementasi

Tabel dan hubungannya dibuat selama fase ini menggunakan *access 2007*. Langkah-langkah dalam membuat tabel dan hubungan yang ada di dalam Sistem Informasi Bimbingan Studi dapat dilihat di dalam daftar lampiran (Lampiran 3.4: Membuat Tabel dan Hubungannya) pada tugas akhir ini.

Selanjutnya setelah tabel dan hubungannya terbentuk, penulis menggunakan SQL yang dilandasi oleh kalkulus relasional, aljabar relasional, dan operator-operator relasional dalam memanipulasi data untuk memperoleh informasi dan membuat fitur-fitur. Berikut ini adalah beberapa contoh penggunaan aljabar relasional dan kalkulus relasional dalam fase implementasi, membuat *query*.

#### Contoh 3.1

```
SELECT DISTINCT MAHASISWAKONTRAKKULIAH.NIM,
MAHASISWA>NamaMahasiswa, MAHASISWA.Angkatan,
MAHASISWA.KodeProdi, PRODI>NamaProdi,
MAHASISWAKONTRAKKULIAH.KodeMatakuliah,
MATAKULIAH>NamaMatakuliah, MATAKULIAH.SKS,
MAHASISWAKONTRAKKULIAH.TahunAkademik,
MAHASISWAKONTRAKKULIAH.Semester,
MAHASISWAKONTRAKKULIAH.HurufMutu,
Iif([HurufMutu]="A",4,Iif([HurufMutu]="B",3,Iif([HurufMutu]="C",2,Iif([Huruf
Mutu]="D",1,0)))) AS NilaiMutu,
MAHASISWAKONTRAKKULIAH.TanggalLulus

FROM MATAKULIAH RIGHT JOIN ((PRODI RIGHT JOIN MAHASISWA ON
PRODI.KodeProdi=MAHASISWA.KodeProdi) LEFT JOIN
MAHASISWAKONTRAKKULIAH ON
MAHASISWA.NIM=MAHASISWAKONTRAKKULIAH.NIM) ON
MATAKULIAH.KodeMatakuliah=MAHASISWAKONTRAKKULIAH.KodeMat
akuliah

WHERE (MAHASISWAKONTRAKKULIAH.ID IN(SELECT
Max(MAHASISWAKONTRAKKULIAH.ID) AS ID FROM PRODI RIGHT JOIN
(MATAKULIAH RIGHT JOIN (MAHASISWA LEFT JOIN
```

MAHASISWAKONTRAKKULIAH ON MAHASISWA.NIM =  
 MAHASISWAKONTRAKKULIAH.NIM) ON MATAKULIAH.KodeMatakuliah  
 = MAHASISWAKONTRAKKULIAH.KodeMatakuliah) ON PRODI.KodeProdi =  
 MAHASISWA.KodeProdi GROUP BY MATAKULIAH.KodeMatakuliah,  
 MAHASISWA.NIM));

Penggunaan kalkulus relasional, aljabar relasional, dan operator relasional di atas berfungsi untuk memanipulasi data dengan tujuan untuk memperoleh informasi tentang hasil studi mahasiswa. Kolom-kolom yang dimunculkan ialah NIM, nama mahasiswa, angkatan, kode prodi, nama prodi, kode matakuliah, nama matakuliah, SKS, tahun akademik, semester, huruf mutu, nilai mutu dan tanggal lulus. Informasi di atas diperoleh dengan menggabung (*join*) empat tabel, yaitu MAHASISWA, MATAKULIAH, MAHASISWAKONTRAKKULIAH dan PRODI.

kalkulus relasional, aljabar relasional, dan operator relasional tidak digunakan dalam membuat *form*, akan tetapi digunakan dalam pembuatan *report*. Berikut ini adalah beberapa contoh penggunaan aljabar relasional dan kalkulus relasional dalam fase implementasi, membuat *report*.

### Contoh 3.2

**SELECT** MAHASISWADOSENPA.NIM, MAHASISWA>NamaMahasiswa,  
 MAHASISWA.Angkatan, MAHASISWA.KodeProdi, PRODI>NamaProdi,  
 MAHASISWADOSENPA.KodeDosen, DOSEN.Gelar1, DOSEN>NamaDosen,  
 DOSEN.Gelar2, MAHASISWA.AsalSekolah, MAHASISWA.Kelas,  
 MAHASISWA.TempatLahir, MAHASISWA.TanggalLahir, MAHASISWA.Sex,  
 MAHASISWA.Agama, MAHASISWA.AlatEmail, MAHASISWA.Alat,  
 MAHASISWA.KodePos, MAHASISWA.Kota, MAHASISWA.Telepon,  
 MAHASISWA.HandphoneMahasiswa, MAHASISWA.Attachments,  
 MAHASISWA.MinatAkademik

**FROM** PRODI RIGHT JOIN (MAHASISWA LEFT JOIN (DOSEN RIGHT JOIN  
 MAHASISWADOSENPA ON DOSEN.KodeDosen =

MAHASISWADOSENPA.KodeDosen) ON MAHASISWA.NIM =  
 MAHASISWADOSENPA.NIM) ON PRODI.KodeProdi =  
 MAHASISWA.KodeProdi

**WHERE** (((MAHASISWADOSENPA.KodeDosen)=[Masukan kode dosen]))  
**ORDER BY** MAHASISWADOSENPA.NIM,  
 MAHASISWADOSENPA.KodeDosen;

Penggunaan kalkulus relasional, aljabar relasional, dan operator relasional pada *report* di atas bertujuan untuk menampilkan informasi dan mencetak data dengan lengkap) semua mahasiswa bimbingan, seperti kode prodi, nama prodi, kode DPA, nama DPA, NIM, angkatan, kelas, minat akademik, asal sekolah, foto mahasiswa, nama mahasiswa, tempat dan tanggal lahir, jenis kelamin, agama, *email*, alamat, kota, kode pos, telepon dan *handphone*.

Penggunaan kalkulus relasional, aljabar relasional, dan operator relasional pada *query* dan *report* di dalam Sistem Informasi Bimbingan Studi secara rinci dapat dilihat di dalam Lampiran 3.5: Penggunaan Kalkulus Relasional, Aljabar Relasional, dan Operator-operator Relasional dan teknik membuat *form* pada Lampiran 3.6: Membuat *Form*.