

# ***Basis Data Relasional Ternormalisasi***

## **Tugas Akhir**

**Diajukan sebagai salah satu syarat  
Dalam menempuh ujian sarjana Program S1  
Jurusan Teknik Informatika, Institut Teknologi Indonesia**



**Disusun oleh :**

**H U S N I**  
**015920035 / 923206716750033**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI INDONESIA  
SERPONG  
1998**

# Lembar Pengesahan

## Tugas Akhir

### *Basis Data Relasional Ternormalisasi*

Diajukan sebagai salah satu syarat  
Dalam menempuh ujian sarjana Program S1  
Jurusan Teknik Informatika, Institut Teknologi Indonesia



Oleh :

**HUSNI**

015920035 / 923206716750033

Disetujui,

Two handwritten signatures are shown. The signature on the left is for Dr. -ing Kondar Siahaan, Ketua Jurusan Teknik informatika. The signature on the right is for Dr. -ing Kondar Siahaan, Pembimbing.

Dr. -ing Kondar Siahaan  
Ketua Jurusan Teknik informatika

Dr. -ing Kondar Siahaan  
Pembimbing



# KATA PENGANTAR

Penulisan Tugas Akhir ini merupakan salah satu syarat yang harus dipenuhi oleh mahasiswa Institut Teknologi Indonesia khususnya Jurusan Teknik Informatika dalam menempuh ujian sarjana yang bertujuan untuk melihat kemampuan mahasiswa dalam menerapkan ilmu-ilmu yang didapat di bangku kuliah dan mengekspresikannya dalam bentuk tulisan.

Ucapan terima kasih dan penghargaan yang setinggi-tingginya kepada semua pihak yang telah membantu hingga terwujudnya tugas akhir ini, khususnya kepada :

1. **ALLAH S.W.T** yang telah memberikan ridha dan karunianya.
2. Ayahanda **Let.Kol.Inf Muhammad Diah (Alm)** yang telah membiayai dan membimbing sampai akhir hayatnya juga bagi **Ibunda** dan **adik-adik** atas dukungan dan doanya yang tulus.
3. **Bapak Dr.-ing Kondar Siahaan** sebagai ketua jurusan teknik informatika, penasehat akademik, pembimbing tugas akhir, dan sebagai dosen yang telah membimbing penulis sampai selesainya tugas akhir ini.
4. **Bapak Dipl.math Syafwan Syarif, Bapak Oon Amroni Msc,** dan **Bapak Piping** sebagai dosen penguji dalam seminar tugas akhir ini.
5. Seluruh civitas akademika ITI, karyawan perpustakaan, BAAK, Jurusan Teknik Informatika, dan Fakultas Teknologi Industri.
6. Seluruh **rekan-rekan Manunggal Bhawana ITI** serta **rekan-rekan mahasiswa ITI** yang telah memberi dukungan moral serta yang sering mengganggu hingga tetap mempertahankan semangat penulis.

# **BASIS DATA RELASIONAL TERNORMALISASI**

**Oleh :**

**H u s n i**

**015920035**

## **ABSTRAK**

Dalam database relasional, penyimpanan data harus memenuhi persyaratan – persyaratan tertentu. Hal ini diperlukan antara lain untuk menjaga konsistensi data. Ketidak konsistenan data yang umum adalah karena anomali dan "Losing join information".

Anomali adalah penyimpangan data yang bisa terjadi karena tabel tidak dapat disisipkan, terjadinya kehilangan informasi pada waktu penghapusan data, atau perubahan data pada tabel tidak efisien.

"Losing join Information" adalah hilangnya informasi ketika dua tabel atau lebih yang digabung melalui operasi join menjadi satu tabel.

Untuk meminimalisasi anomali, dapat dilakukan proses normalisasi pada tabel – tabel atau relasi relasi. Metode normalisasi adalah suatu teknik penyajian data dimana suatu tabel didekomposisi menjadi tabel - tabel kecil , sederhana dan benar menurut aturan dekomposisi. Metode normalisasi ini didasarkan pada analisis hubungan fungsional yang terjadi diantara kolom-kolom tabel (field) dengan melihat ketergantungan data-data pada suatu field dengan field lain.

Dengan normalisasi ini basis data dapat terhindar dari penggandaan atau redudansi data juga dapat memelihara dan menjaga integritas data.

**Menyetujui**

**Jurusan Teknik Informatika**

**Ketua**

**Dr. Ing. Kondar Siahaan.**



# DAFTAR ISI

<b>LEMBAR PENGESAHAN</b>	<b>I</b>
<b>ABSTRAKSI</b>	<b>II</b>
<b>KATA PENGANTAR</b>	<b>III</b>
<b>DAFTAR ISI</b>	<b>IV</b>
<b>DAFTAR GAMBAR</b>	<b>V</b>
<b>BAB I PENDAHULUAN</b>	<b>1</b>
<b>UMUM</b>	<b>1</b>
1.1. SISTEM BASIS DATA	1
1.2. MODEL-MODEL BASIS DATA	3
1.2.1 Model Relasional	3
1.2.2 Model Hirarkis	4
1.2.3 Model Jaringan	4
1.2.4 Model Berorientasi Objek	6
1.3. ANOMALI BASIS DATA RELASIONAL	7
1.4. NORMALISASI BASIS DATA RELASIONAL	8
<b>RUANG LINGKUP</b>	<b>8</b>
<b>BATASAN MASALAH</b>	<b>9</b>
 <b>BAB II LANDASAN TEORI</b>	 <b>10</b>
2.1. TEORI HIMPUNAN	10
3.1.1 Himpunan Bagian	10
3.1.2 Himpunan Semesta	11
3.1.3 Kardinalitas Himpunan	11
2.2. OPERASI PADA HIMPUNAN	11
2.2.1 Gabungan	11
2.2.2 Irisan	11
2.2.3 Selisih	
12	
2.2.4 Komplemen	12

2.2.5	Selisih Simetris	12
2.3.	PASANGAN TERURUT	12
2.4.	HASIL KALI KARTESIS	13
2.5.	RELASI	13
2.5.1	Domain Relasi	13
2.5.2	Range Relasi	14
2.5.3	Relasi n-R	14
2.6.	FUNGSI	14
<b>BAB III</b>	<b>BASIS DATA MODEL RELASIONAL</b>	<b>15</b>
3.1.	KOMPONEN BASIS DATA RELASIONAL	15
3.1.1	Domain	15
3.1.2	Relasi	15
3.1.3	Atribut	16
3.1.4	Aritas	16
3.1.5	Tupel	16
3.2.	KEY	16
3.1.1	Super Key	16
3.1.2	Kandidat Key	16
3.1.3	Primary Key	17
3.1.4	Alternate Key	17
3.1.5	Foreign Key	17
3.3.	OPERASI ALJABAR	18
3.1.1	Gabungan	19
3.1.2	Irisan	19
3.1.3	Selisih	19
	19	
3.1.4	Hasil Kali Kartesis	19
3.1.5	Select	20
3.1.6	Project	20
3.1.7	Join	20



<b>BAB IV KETERGANTUNGAN</b>	<b>21</b>
4.1. KETERGANTUNGAN FUNGSIONAL (FD)	21
4.2. KETERGANTUNGAN PARSIAL DAN PENUH	21
4.3. KETERGANTUNGAN TRANSITIF	21
4.4. KETERGANTUNGAN BANYAK NILAI (MVD)	22
4.5. KETERGANTUNGAN JOIN (JD)	22
4.5.1 Dekomposisi	22
4.5.2 Lossy Join dan Lossles Join	23
4.5.3 Ketergantungan Join	24
 <b>BAB V NORMALISASI</b>	 <b>25</b>
5.1. BENTUK NORMAL PERTAMA (1 NF)	26
5.2. BENTUK NORMAL KEDUA (2 NF)	27
5.3. BENTUK NORMAL KETIGA (3 NF)	29
5.4. BOYCE CODE NORMAL FORM ( BCNF)	30
5.5. BENTUK NORMAL KEEMPAT (4 NF)	31
5.6. BENTUK NORMAL KELIMA(5 NF)	32
 <b>KESIMPULAN</b>	 <b>VI</b>
<b>DAFTAR PUSTAKA</b>	<b>VII</b>

## Daftar Gambar

Gbr. 1.1	4
Gbr 1.2	5
Gbr 1.3	6
Gbr 5.1	27
Gbr 5.2	27
Gbr 5.3	28
Gbr 5.4.1	28
Gbr 5.4.2	28
Gbr 5.5	29
Gbr 5.5.1	30
5.5.2	30
5.6	32
5.7	32
5.7.1	32
5.7.2	32



# **BAB I**

## **PENDAHULUAN**

### **UMUM**

#### **1.1 SISTEM BASIS DATA**

Perkembangan ilmu pengetahuan khususnya pada bidang teknologi informasi saat ini memungkinkan komunikasi data dari satu tempat ke tempat lain di belahan bumi ini secara langsung, sehingga hal yang paling mendasar dalam perkembangan teknologi informasi adalah data.

Data yang tersebar dan tersedia dapat saling dihubungkan, dimanipulasi dan diolah sehingga dapat memberikan informasi yang tepat untuk digunakan oleh pemakai.

Sistem basis data adalah sekumpulan data yang terstruktur dan terorganisasi yang di simpan dalam media penyimpanan dan dapat diambil atau diperbaharui untuk keperluan informasi sewaktu – waktu.

Teknologi informasi sistem basis data merupakan bagian yang menangani seluruh aktifitas pada data seperti pencarian data, penyimpanan data, pengambilan data, pengujian data, manipulasi data, distribusi dengan tingkat keakuratan dan kerahasiaan data, dan didukung oleh perangkat komputer serta media komunikasi.

Pada dasarnya sistem basis data adalah komputerisasi dari sistem penyimpanan data, artinya, komputerisasi sistem yang secara umum bertujuan untuk mengatur informasi dan membuat informasi tersebut tersedia pada waktu yang dibutuhkan.

## KEUNTUNGAN SISTEM BASIS DATA

### 1. Redudansi data dapat diminimalisasi

Dalam pengolahan data yang tidak berbasis data, seperti proses file data secara langsung, beberapa bagian data yang disajikan dalam suatu file biasanya disajikan lagi atau diduplikasi pada file lain, karena dalam proses ini biasanya file yang saling berhubungan tidak dihubungkan secara sistem, sehingga untuk kasus data besar yaitu apabila jumlah data dan manipulasi serta item data rumit, duplikasi ini akan menghabiskan media penyimpanan data yang pada akhirnya mengakibatkan pengolahan data menjadi tidak efisien.

### 2. Menghindari ketidak konsistenan data

Penyimpanan data yang berulang – ulang pada beberapa file dapat menyebabkan data tidak konsisten. Hal ini terjadi apabila ada perbedaan data pada penyimpanan data di suatu file dengan penyimpanan data pada file lain.



### 3. Standarisasi data

Standarisasi dapat dilakukan untuk mempermudah perpindahan data dan pertukaran data terutama apabila berhubungan dengan sistem basis data lain.

### 4. Keamanan data

Keamanan data pada sistem basis data menjamin hak akses dari setiap pemakai. Hanya pemakai yang memenuhi hak akses tertentu dapat mengakses data tertentu pula. Hal ini terutama sangat berguna pada sistem dengan banyak pemakai.

## 1.2 MODEL – MODEL BASIS DATA

### 1.2.1 Model Relasional

Model relasional merepresentasikan data dan relasi dalam bentuk tabel – tabel, yang setiap tabel memiliki beberapa kolom yang memiliki nama yang unik. Model ini pertama kali diperkenalkan oleh E.F Codd pada tahun 1970. Setiap tabel memiliki kolom – kolom yang disebut atribut dan setiap baris di sebut tupel atau rekord. Jika tabel (relasi) memiliki  $n$  atribut, setiap baris memiliki  $n$  tupel. Maka relasi tersebut disebut relasi  $n$ -R dan  $n$  disebut sebagai derajat atau aritas dari relasi.

Berikut adalah istilah – istilah yang sinonim yang akan digunakan dalam pembahasan lebih lanjut.

1. Relasi sinonim dengan tabel atau file
2. Tupel sinonim dengan baris atau rekord
3. Atribut sinonim dengan kolom , field dan data elemen

Contoh relasi :

NIM	NMHS	JURUSAN	MKULIAH
01592001	AMIR	INFORMATIKA	IF201
01593002	ABDUL	INFORMATIKA	IF304
01594001	MIMI	INFORMATIKA	IF201

MKULIAH	NKULIAH
IF201	STRUKTUR DATA
IF304	STATISTIK

Gbr. 1.2 Contoh Model Relasional

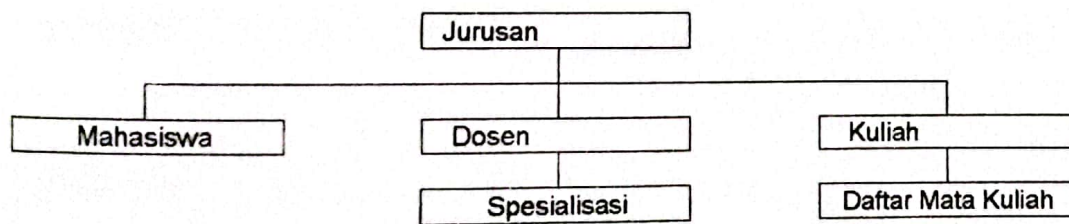
### 1.2.2 Model Hirarki

Tidak ada dokumen asli yang menjelaskan model hirarki seperti pada model relasional dan Network. Beberapa manajemen sistem informasi komputer terdahulu dikembangkan dengan menggunakan struktur penyimpanan hirarki. Contoh – contoh untuk sistem ini adalah *Control data Corporation's Multi Access Retrivial System (MARS VI)*, *IBM Management*



*System (IMS)*, dan *MRIS System 2000* (sekarang sudah dijual oleh SAS Institute)

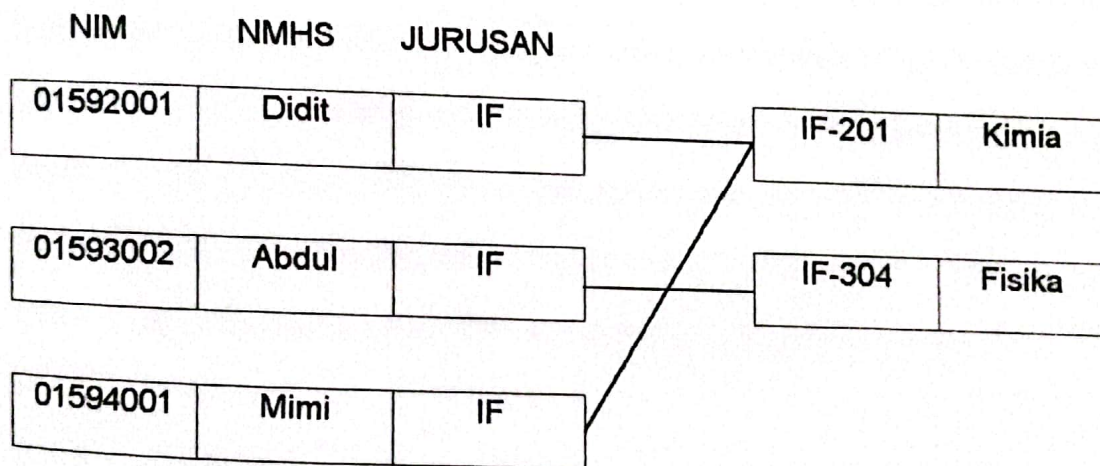
Model hirarki direpresentasikan sebagai suatu pohon yang dimulai pada suatu record tunggal yang disebut induk, dan bercabang menjadi beberapa sub pohon yang memiliki level yang lebih rendah. Setiap level berhubungan dengan level yang lebih tinggi sehingga membentuk hubungan induk dan anak. Induk dapat memiliki nol atau beberapa anak (hubungan satu ke banyak) sedangkan anak hanya berhubungan dengan satu induk (hubungan satu ke satu). Secara fisik model hirarki di sajikan dalam struktur memori "*link*"



### 1.2.2 Model Jaringan (Network)

Pada model relasional, data dan hubungan (relasi) antara data di representasikan dalam tabel – tabel. Perbedaan antara model jaringan dengan model relasional adalah data di representasikan dalam kumpulan record dan hubungan antar data direpresentasikan dalam "links" (hubungan).

Model data base jaringan berisikan kumpulan record yang dihubungkan dengan links. Suatu record dalam banyak hal mirip dengan entity pada model entity relationship. Setiap record adalah suatu kumpulan dari field (atribut), yang memiliki hanya satu nilai data. Suatu links adalah suatu hubungan antara tepat dua record. Sehingga links dapat di lihat sebagai sebuah bentuk terbatas dari hubungan pada kaitanya dengan model E-R.



Gbr. 1.3 Contoh Model Jaringan

#### 1.2.4 Model berorientasi objek

Model basis data berorientasi objek adalah model yang paling baru untuk pendesainan data. Model ini ditemukan dari konsep pemrograman berorientasi objek, yang dikembangkan pada tahun tujuh puluh delapan.



Model ini dapat digabungkan dengan model relasional dan penggabungan kedua model ini dapat menghasilkan model basis data yang efisien, handal, sehingga menjadi pilihan alternatif basis data dimasa yang akan datang.

Hal yang paling fundamental dalam basis data berorientasi objek adalah objek, class, method dan class "inheritance". Objek dalam terminologi model ini bersifat sama dengan variabel yang terdapat pada bahasa pemrograman. Perbedaananya suatu objek dapat diakses secara langsung ataupun tidak langsung sesuai definisi objek sebelumnya. Class merupakan salah satu objek yang dapat diidentikan dengan tipe record pada bahasa pemrograman, tetapi class dapat diturunkan menjadi subclass yang dapat mengakses beberapa objek data pada class asalnya secara langsung subclass- subclass ini dinyatakan sebagai class "inheritance". Sedangkan method dapat diidentikan dengan fungsi pada bahasa pemrograman.

### 1.3 ANOMALI PADA MODEL RELASIONAL

Anomali adalah penyimpangan data pada basis data akibat proses-proses penghapusan penyisipan dan perubahan data pada tabel. Penyimpangan ini dapat berupa tidak dapat dilakukannya proses penyisipan, penghapusan data tidak efisien, dan kesulitan dalam memperbaharui data. Untuk menghilangkan anomali – anomali ini maka perlu dilakukan suatu proses yang disebut normalisasi.

#### 1.4 NORMALISASI BASIS DATA RELASIONAL

Teknik normalisasi digunakan untuk meminimalisasi kasus – kasus anomali. Hal ini dilakukan dengan cara mendekoponsisikan tabel – tabel atau membagi tabel – tabel menjadi beberapa tabel yang lebih kecil. Dengan proses normalisasi ini, suatu relasi menjadi lebih sederhana sehingga memudahkan untuk dibaca, manajemen data menjadi lebih optimal karena dapat mereduksi / mengurangi masalah yang timbul pada pendisainan basis data seperti duplikasi data dan ketidak konsistenan data. Hal ini dimungkinkan karena didasari pada prinsip ideal bahwa suatu entitas hanya boleh tampil pada satu tempat, dapat diulang hanya apabila sungguh-sungguh dibutuhkan.

#### RUANG LINGKUP

Tugas akhir ini dimulai dengan pendahuluan yang memberi gambaran umum tentang basis data serta beberapa model basis data yang ada.

Bahasan tugas akhir ini dimulai dengan landasan teori , yaitu landasan matematika bagi basis data relasional, dimulai dari konsep himpunan, operasi pada himpunan, relasi dan fungsi.

Bahasan selanjutnya tentang basis data relasional, dimulai dari komponen, key dan operasi – operasi aljabar yang digunakan dalam memanipulasi data .



Bahasan utama disusun dalam dua bab yaitu ketergantungan (dependency) dan Normalisasi. Ketergantungan menjelaskan hubungan antara atribut - atribut suatu tabel / relasi dengan atribut - atribut pada tabel / relasi lain. Konsep tentang ketergantungan ini akan digunakan sebagai dasar untuk dilakukan proses - proses normalisasi yang akan dibahas pada bab selanjutnya. Dalam bab normalisasi akan dijelaskan syarat - syarat suatu normalisasi berikut aturan - aturan dan diperlihatkan dalam contoh - contoh.

Sebagai pelengkap penulis menyertakan lampiran - lampiran yang merupakan contoh implementasi pendisainan basis data relasional yang ditampilkan dalam bentuk tabel - tabel hubungan entiti (entity relationship), dimana metode - metode normalisasi diterapkan.

## BATASAN MASALAH

Dalam tugas akhir ini disajikan konsep - konsep yang mendukung proses normalisasi, berikut alasan, syarat dan contoh. Konsep ketergantungan fungsional dibahas lebih rinci karena konsep ini lebih sering digunakan dalam proses normalisasi. Landasan teori hanya menampilkan konsep - konsep yang diperlukan dalam pembahasan bab - bab selanjutnya.



## BAB II

### LANDASAN TEORI

#### II.1 TEORI HIMPUNAN

##### Definisi

Suatu himpunan didefinisikan sebagai kumpulan objek yang telah didefinisikan dengan jelas. Objek – objek tersebut merupakan elemen dari himpunan. Notasi penulisan suatu himpunan ditulis dengan huruf besar seperti : A, B, C... dan anggota himpunan ditulis dengan huruf kecil seperti : p, q, r, s. Jika x merupakan anggota himpunan A maka dinotasikan  $x \in A$ . Jika x bukan anggota himpunan A dinotasikan  $x \notin A$ .

##### II.1.1 HIMPUNAN BAGIAN

##### Definisi

Jika anggota himpunan A merupakan anggota himpunan B dimana  $x \in A$  dan  $x \in B$  dapat dikatakan bahwa A adalah himpunan bagian dari B atau A termasuk dalam B, dinotasikan :

$$A \subseteq B$$

Himpunan A dan B dikatakan sama ditulis  $A = B$  jika himpunan A adalah himpunan bagian dari B dan himpunan B adalah bagian dari A dimana  $A = B$ ,

$$A \subseteq B, B \subseteq A.$$



### II.1.2 HIMPUNAN SEMESTA $U$

#### Definisi

Himpunan semesta adalah himpunan yang berisikan seluruh objek yang sedang dibicarakan dinotasikan  $U$ .

### II.1.3 KARDINALITAS HIMPUNAN

Suatu himpunan  $A$  dikatakan berhingga jika memiliki  $n$  buah elemen hingga, dimana  $n \in \mathbb{N}$ ,  $n$  disebut kardinal dari  $A$  ditulis  $|A|$ .

## II.2 OPERASI PADA HIMPUNAN

### II.2.1 GABUNGAN $\cup$

Gabungan  $U$  dari himpunan  $A$  dan  $B$  adalah himpunan yang beranggotakan seluruh elemen himpunan  $A$  atau  $B$  dinotasikan sebagai  $A \cup B$ .

$$A \cup B = \{x \mid x \in A \text{ atau } x \in B\}$$

### II.2.2 IRISAN $\cap$

Jika  $A$  dan  $B$  adalah himpunan – himpunan, didefinisikan irisan dari  $A$  dan  $B$  adalah suatu himpunan yang beranggotakan *seluruh anggota  $A$  dan juga anggota  $B$*  dinotasikan  $A \cap B$ .

$$A \cap B = \{x \mid x \in A, x \in B, A, B \subset U\}$$

### II.2.3 SELISIH ( - )

Jika A dan B dua himpunan, didefinisikan selisih dari A dan B adalah suatu himpunan yang berisikan seluruh anggota A tapi bukan anggota B, dinotasikan  $A - B$

$$A - B = \{x \in A \text{ dan } x \notin B, A, B \subset U\}$$

### II. 2.4 KOMPLEMEN ( ' )

Jika U adalah himpunan semesta yang berisikan A , maka  $U - A$  adalah komplemen dari himpunan A dinotasikan  $A'$  .

$$A' = \{x \mid x \notin A, x \in U, A, B \subset U\}$$

### II.2.5 SELISIH SIMETRIS ( / )

Jika A dan B adalah dua himpunan, didefinisikan selisih simetris dari A dan B adalah suatu himpunan yang berisikan seluruh anggota himpunan A atau B tapi bukan anggota himpunan A dan B dinotasikan  $A / B$ .

$$\begin{aligned} A / B &= \{x \mid (x \in A, x \notin B) \text{ atau } (x \in B, x \notin A)\} \\ &= (A - B) \cup (B - A) \end{aligned}$$

### II.2.6 PASANGAN TERURUT

Pasangan terurut (a, b) adalah pasangan yang didasarkan, a sebagai elemen pertama dan b sebagai elemen kedua.



### II.2.7 HASIL KALI KARTESIS

Hasil kali kartesis dari dua himpunan A dan B di tulis  $A \times B$  dan berisikan seluruh elemen pasangan terurut dimana elemen pertama di A dan elemen kedua di B

$$A \times B = \{ (a,b) \mid a \in A, b \in B \}$$

### II.3 RELASI

Jika A dan B adalah dua himpunan tak kosong, suatu relasi R dari A ke B adalah himpunan bagian dari  $A \times B$ , jika  $R \subseteq A \times B$  dan  $(a,b) \in R$ , dikatakan a direlasikan ke b dengan R.

Jika a tidak berelasi ke b dengan R dinotasikan  $a \not R b$ .

#### II.3.1 DOMAIN RELASI R

Jika  $R \subseteq A \times B$  adalah relasi dari A ke B. Domain dari R, dinotasikan  $\text{Dom}(R)$  adalah himpunan elemen di A yang berelasi dengan beberapa elemen di B.

$$\text{Dom}(R) = \{a \mid a \in A, (a,b) \in R, b \in B\}$$

### II.3.2 RANGE RELASI R

Range dari R dinotasikan  $\text{Ran}(R)$  adalah himpunan elemen di B yang merupakan elemen kedua dari pasangan di R, dimana seluruh elemen di B yang direlasikan dengan beberapa elemen di A.

$$\text{Ran}(R) = \{ b \mid b \in B, (a,b) \in R, a \in A \}$$

### II.3.3 RELASI n-R

Relasi n-r diantara n himpunan  $A_1, A_2, \dots, A_n$  didefinisikan sebagai himpunan bagian dari hasil kali kartesis  $(A_1 \times A_2 \times \dots \times A_{n-1}) \times A_n$  dengan pasangan terurut  $((a_1, a_2, \dots, a_{n-1}), a_n)$ , dengan notasi himpunan :

$$R \subseteq (A_1, A_2, \dots, A_{n-1}) \times A_n = \{(a_1, a_2, \dots, a_{n-1}), a_n \mid a_i \in A_i\}, \text{ atau}$$

$$R \subseteq A_1, A_2, \dots, \times A_n = \{(a_1, a_2, \dots, a_n) \mid a_i \in A_i\}$$

## III. 4 FUNGSI

Jika A dan B adalah himpunan tidak kosong. Fungsi  $f$  dari A ke B dinotasikan  $f : A \rightarrow B$  adalah relasi dari A ke B sehingga untuk seluruh  $a \in \text{Dom}(f)$ ,  $f(a)$  berisikan harga satu elemen dari B.



## **BAB III**

### **BASIS DATA MODEL RELASIONAL**

#### **III.1 KOMPONEN BASIS DATA RELASIONAL**

##### **III.1.1 DOMAIN**

Domain adalah suatu himpunan seluruh nilai – nilai yang mungkin di miliki oleh atribut yang memiliki type yang sama.

Domain adalah kumpulan nilai – nilai tempat atribut mengambil nilai, berisikan nilai – nilai yang memiliki tipe yang sama, yaitu tipe untuk suatu atribut . Elemen dari domain adalah atomik , yaitu tidak dapat dibagi – bagi lagi.

##### **III.1.2 RELASI**

Relasi pada database adalah tabel dua dimensi yang berdiri dari sejumlah baris dan kolom data.

Sifat – sifat relasi :

- Entiti – entiti pada tabel memiliki nilai tunggal sehingga tidak ada kelompok berulang atau array.
- Entiti – entiti pada suatu kolom memiliki tipe data yang sama.
- Tidak ada dua baris dalam suatu kolom yang identik
- Order dari baris – baris atau kolom – kolom tidak penting

### III.1.3 ATRIBUT

Setiap kolom dari suatu relasi disebut sebagai atribut suatu relasi. Nama kolom disebut nama relasi.

### III.1.4 ARITAS / DERAJAT

Derajat suatu relasi R adalah kardinal dari atribut dalam relasi tersebut.

### III.1.4 TUPEL

Tupel dalam database relasional merupakan baris – baris dalam suatu kolom.

## III.2. KEY

### III.2.1 SUPER KEY

Superkey adalah suatu himpunan dari atribut yang secara unik membedakan setiap baris pada suatu relasi.

Kunci (Key) adalah himpunan minimal dari suatu atribut yang secara unik membedakan setiap baris pada suatu relasi. Minimal dalam hal ini berarti bahwa tidak ada himpunan bagian dari himpunan atribut kunci yang dapat secara unik membedakan tupel – tupel dalam suatu relasi.

### III.2.2 CANDIDATE KEY (KUNCI KANDIDAT )

Kunci kandidat adalah key yang berisikan lebih dari satu atribut.



### III.2.3 PRIMARY KEY (KUNCI UTAMA )

Kunci Utama adalah satu atribut atau satu set minimal atribut yang tidak hanya mengidentifikasi secara unik suatu kejadian spesifik tapi juga dapat mewakili setiap kejadian dari suatu entiti.

Setiap kunci kandidat mempunyai peluang untuk menjadi kunci utama, tetapi sebaiknya dipilih satu saja yang dapat mewakili secara menyeluruh terhadap entiti yang ada.

### III.2.4 ALTERNATE KEY (KUNCI ALTERNATIF)

Kunci alternatif adalah kunci kandidat yang tidak dipakai sebagai kunci utama, kerap kali kunci alternatif digunakan sebagai kunci pengurutan dalam laporan .

### III.2.5 FOREIGN KEY (KUNCI ASING)

Kunci asing adalah suatu himpunan atribut pada suatu relasi yang menjadi kunci pada relasi yang lain (atau relasi yang sama)

### BATASAN - BATASAN

Adalah aturan yang membatasi nilai dari database. Model data relasional Codd memiliki beberapa batasan yang digunakan untuk menguji validitas data pada database untuk menambah struktur yang berarti bagi data.

Beberapa batasan :

- Integritas Entiti
- Integritas Referensial
- Ketergantungan Fungsional

### **INTEGRITAS ENTITI**

Aturan integritas entiti : tidak ada kunci atribut pada baris manapun yang memiliki nilai "null"

### **INTEGRITAS REFERENSIAL**

Aturan integritas referensial : Setiap kunci asing harus memiliki nilai "null" atau nilainya harus merupakan nilai aktual dari suatu kunci pada relasi yang lain.

Pembahasan untuk batasan ketiga yaitu ketergantungan fungsional akan dibahas dalam bab tersendiri.

### **III.3 OPERASI ALJABAR**

Operasi aljabar terdiri atas kumpulan operator yang beroperasi pada satu atau lebih relasi sebagai input dan menghasilkan relasi baru sebagai output. Diantara operasi aljabar yang banyak digunakan dalam proses normalisasi adalah Select, Project, Hasil kali kartesis dan Join



### III.3.1 GABUNGAN

Definisi :

Gabungan dari dua relasi A dan B adalah relasi dengan "heading" yang sama untuk setiap A dan B yang beranggotakan himpunan dari tupel – tupel t yang juga merupakan anggota himpunan A atau B atau keduanya.

### III.3.2 IRISAN $\cap$

Irisan dua relasi kompatibel A dan B adalah relasi dengan atribut yang sama untuk A dan B yang berisikan kumpulan tupel t yang dimiliki oleh A dan B.

### III.3.3 SELISIH (DIFFERENCE)

Diberikan dua relasi kompatibel RA dan RB dengan derajat n. Selisih RA dan RB adalah relasi dengan derajat n yang berisikan tupel – tupel milik RA tapi bukan milik RB.

Ditulis :

$RA \text{ Diff } RB$

$RA - RB$

### III.3.4 HASIL KALI KARTESIAN

Hasil kali kartesian untuk sembarang dua relasi R1 dan R2 dengan derajat m dan n, adalah relasi dengan derajat m x n yang berisikan seluruh tupel– tupel t sehingga t adalah konkatenasi dari tupel t1 di R1 dan tupel t2 di R2.

### III.3.5 SELECT

Jika R adalah suatu relasi dengan derajat n dan  $\theta$  merepresentasikan setiap operator pembandingan aritmetik seperti  $=, <, >, \geq, \leq$ , dan lain – lain. Seleksi  $\theta$  dari R pada atribut – atribut  $a_1$  dan  $a_2$  didifinisikan sebagai himpunan tupel – tupel t berikut :

$$\{t \mid t \in R, \quad \text{dan} \quad t.a_1 \theta t.a_2 \text{ dipenuhi}\}$$

dimana t.a berarti nilai dari atribut a untuk tupel t. Sementara  $t.a_2$  dapat digantikan dengan konstanta. Sehingga seleksi  $\theta$  dari R dapat didifinisikan :

$$\{t \mid t \in R, \quad \text{dan} \quad t.a_1 \theta c \text{ dipenuhi}\}$$

### III.3.6 PROJECT

Jika R suatu relasi dengan derajat n dan atribut –atribut  $a_1, \dots, a_n$  Jika  $a_{i1}, \dots, a_{ik}$ ,  $k < n$ , merupakan subset dari n atribut tersebut. Proyeksi dari R pada atribut  $a_{i1}, \dots, a_{ik}$  adalah himpunan dari semua tupel – tupel  $t = (t_1, \dots, t_k)$  sehingga t pada R bernilai  $t_1, \dots, t_k$  untuk atribut – atribut  $a_{i1}, \dots, a_{ik}$

### III.3.6 JOIN

Jika R1 dan R2 dua relasi dengan derajat m dan n, dan  $\theta$  adalah operator matematika sembarang. Maka  $\theta$  join dari relasi R1 pada atribut a dan R2 pada atribut b adalah himpunan semua tupel t sehingga t adalah hasil konkatenasi dari tupel  $t_1 \in R_1$  dan tupel  $t_2 \in R_2$  dengan kondisi  $t_1.a \theta t_2.b$  terpenuhi.



## BAB IV

### KETERGANTUNGAN

#### 4.1 KETERGANTUNGAN FUNGSIONAL

**Definisi:**

Diberikan sebuah relasi  $R$ , atribut  $y$  dari  $R$  adalah bergantung fungsi pada atribut  $x$  dari  $R$ , jika dan hanya jika setiap nilai  $x$  dalam  $R$  punya hubungan dengan tepat satu nilai  $y$  dalam  $R$ .

**Notasi:**

Notasi  $\rightarrow$  menyatakan suatu hubungan logika implikasi.

$x \rightarrow y$  berarti  $x$  menentukan secara fungsional  $y$  di  $r(U)$

#### 4.2 KETERGANTUNGAN PARSIAL DAN PENUH

Jika  $x \rightarrow A$  adalah FD tak trivial ( $A \subsetneq X$ ), atribut  $A$  adalah bergantung secara parsial pada  $X$  jika  $y \rightarrow A$  ada untuk  $y \subset x$ . Dalam hal  $X \rightarrow A$  adalah FD tak trivial dan  $\exists y \subset x$ , sehingga  $y \not\rightarrow A$ , maka  $x \rightarrow A$  adalah bergantung penuh.

#### 4.3 KETERGANTUNGAN TRANSITIF

**Definisi :**

Jika  $x$  dan  $y$  atribut tidak kosong, subset dari  $U$ ,  $\exists A \subset U$  atribut  $A$  adalah bergantung secara transitif pada  $x$  melalui  $y$  jika memenuhi semua kondisi

:

1.  $X \rightarrow Y$
2.  $Y \not\rightarrow X$
3.  $Y \rightarrow A$
4.  $A \not\subset XY$

Jika suatu FD  $X \rightarrow A$  memenuhi kondisi diatas, maka  $X \rightarrow A$  adalah ketergantungan transitif.

#### 4.4 KETERGANTUNGAN BANYAK NILAI (MVD)

Definisi :

Jika R suatu relasi, dan A,B,C adalah huruf besar sembarang mempunyai atribut – atribut di R, maka dikatakan B bergantung banyak nilai (*multidependent*) pada A.

Notasi :

$$A \twoheadrightarrow B$$

Teorema Fagin:

Jika R {A,B,C} adalah relasi, dimana A, B, dan C adalah himpunan atribut. Maka R sama dengan hasil join dari proyeksinya terhadap {A,B} dan {A,C} jika dan hanya jika R memenuhi MVD  $A \twoheadrightarrow B \ C$ .

#### 4.5 KETERGANTUNGAN JOIN

##### 4.5.1 DEKOMPOSISI

Jika R adalah relasi dari  $R_1, \dots, R_n$  adalah himpunan proyeksi R pada atribut-atributnya sehingga  $R_1, \dots, R_n$  dapat digabungkan melalui atribut-



atribut untuk memproduksi  $R$ . Jika  $t \in$  adalah tupel sembarang,  $t$  mengembangkan  $t_i$  pada setiap  $R_i$ ,  $i=1, \dots, n$ . Sejumlah  $n$  tupel  $t_i$  ini dapat digabung kembali untuk membangun kembali  $t$  dengan menggunakan join  $R_1 \text{ join } R_2 \dots \text{Join } R_n$ . Dengan kata lain kita selalu mendapatkan

$$R \subseteq R_1 \text{ join } R_2 \dots \text{Join } R_n.$$

#### 4.5.2 LOSSY JOIN DAN LOSSLESS JOIN

Jika suatu relasi dedekomposisi menjadi dua atau lebih relasi-relasi yang lebih kecil, maka dengan memperlakukan operasi *natural join* ( $\bowtie$ ) kepada relasi-relasi tersebut hasilnya akan berupa relasi baru yang mungkin mempunyai ukuran tupel yang lebih banyak dari ukuran aslinya.

Penggunaan relasi baru tersebut dalam sistem basis data akan menghasilkan informasi yang salah (*Lossy information*) khususnya dalam pengambilan keputusan.

Suatu relasi mengalami *lossy join* (hilangnya informasi akibat operasi join) jika untuk suatu relasi  $R$ , dan  $R_1, \dots, R_n$  adalah hasil dekomposisi  $R$  memenuhi hubungan :

$$R \subseteq R_1 \bowtie R_2 \bowtie \dots \bowtie R_n$$

Sedangkan suatu relasi mengalami Lossless Join (Informasi utuh ketika join) jika memenuhi :

$$R = R_1 \bowtie R_2 \bowtie \dots \bowtie R_n$$

#### 4.5.3 KETERGANTUNGAN JOIN

Jika R suatu relasi A,B, C.....,Z sembarang himpunan bagian dari himpunan atribut-atribut R. Dikatakan R memenuhi ketergantungan join (JD) dinotasikan :

$$*(A,B,.....,Z)$$

Jika dan hanya jika R sama dengan hasil join proyeksi-proyeksinya di A,B,C....,Z.



## BAB V

# NORMALISASI

Basis data model relasional selalu melibatkan duplikasi data. Hal ini di butuhkan untuk menghubungkan banyak relasi selama operasi join. Bagaimanapun redudansi harus dijaga pada level minimum.

Proses normalisasi mempersiapkan cara yang sistematis untuk memastikan kemungkinan minimum redudansi data pada relasi-relasi di pendisainan database secara logika. Hal ini berdasarkan pada prinsip ideal bahwa entiti hanya tampil pada suatu tempat dan hanya dapat berulang apabila sungguh-sungguh dibutuhkan.

Teori normalisasi dibangun berdasarkan konsep dari bentuk normal (*normal form*). Suatu relasi dapat dikatakan sebagai suatu bentuk normal jika memenuhi himpunan kondisi yang telah dispesifikasikan secara pasti. Banyak bentuk normal didefinisikan, pada kesempatan ini akan di jelaskan enam model, yaitu:

1. Bentuk normal pertama 1NF
2. Bentuk normal kedua 2NF
3. Bentuk normal ketiga 3NF
4. Boyce Codd Normal Form (BCNF)

5. Bentuk normal kelima 4 NF
6. Project Join Normal Form (PJNF) 5 NF

Codd pada awalnya mendefinisikan bentuk normal 1 NF, 2 NF, dan 3 NF pada tahun 1972. Motivasi dibelakang definisi Codd ini adalah bahwa 1 NF dan 2 NF tidak memuaskan untuk pendisainan database yang baik, maka relasi-relasi tersebut harus memenuhi 3 NF. Definisi awal Codd untuk 3 NF terdapat banyak kekurangan. Sebuah revisi definisi yang lebih ketat didefinisikan selama Boyce dan Codd hadir pada tahun 1974. Definisi ini lebih ketat dalam hal bahwa relasi yang memenuhi definisi 3NF baru akan secara pasti memenuhi definisi 3 NF lama, tetapi relasi yang memenuhi definisi 3 NF lama tidak memenuhi definisi 3 NF baru. Bentuk 3 NF yang baru sering disebut sebagai "*Boyce Codd Normal Form*" (BCNF) untuk membedakan dengan bentuk yang lama. Kemudian pada tahun 1977 Fagin mendefinisikan bentuk normal baru (keempat) 4 NF dan lebih baru pada tahun 1979, ia memperkenalkan bentuk normal lain, disebut "*Project Join Normal Form*" (PJNF) 5 NF.

### 5.1 BENTUK NORMAL PERTAMA (1 NF)

Relasi pada bentuk normal pertama (1 NF) jika memenuhi kondisi- kondisi :

1. Tidak ada atribut yang berulang atau bernilai array
2. Tidak ada penggandaan tupel



Contoh:

Pada relasi Gbr 5.1 berikut terdapat pengulangan atribut yaitu atribut Kuliah\_1, Kuliah\_2, dan Kuliah\_3. Agar relasi mahasiswa tersebut dalam 1 NF maka relasi mahasiswa harus dinormalisasi sampai tidak ada pengulangan atribut.

Nim#	Nama	P. Akademik	Kuliah_1	Kuliah_2	Kuliah_3
1592001	Amir	Ade	Basic	Fortran	Pascal
1592003	Abdul	Bambang	Pancasila	Kimia	Pde
1593001	Mimi	Kadir	Basic	Kimia	Pascal

Gbr. 5.1 Bentuk Tidak Normal

Nim#	Nama	P. Akademik	Kuliah
1592001	Amir	Ade	Basic
1592001	Amir	Ade	Fortran
1592001	Amir	Ade	Pascal
1592003	Abdul	Bambang	Pancasila
1592003	Abdul	Bambang	Kimia
1592003	Abdul	Bambang	Pde
1593001	Mimi	Kadir	Basic
1593001	Mimi	Kadir	Kimia
1593001	Mimi	Kadir	Pascal

Gbr. 5.2 Bentuk 1 NF

## 5.2 BENTUK NORMAL KEDUA (2 NF)

Suatu relasi dikatakan dalam bentuk normal ke dua apabila memenuhi syarat

1. Dalam bentuk normal pertama
2. Kunci (key) berisikan atribut tunggal

3. Tidak ada atribut yang tidak mempunyai kunci (key)
4. Setiap atribut bukan kunci tergantung fungsional secara penuh kepada seluruh kunci yang komposit.

Nim#	N_Mahasiswa	P_Akademik	K_Kuliah
1592001	Amir	Ade	101
1592003	Abdul	Bambang	102
1593001	Mimi	Kadir	101

Gbr. 5.3 Tidak dalam 2 NF

Pada relasi Gbr. 5.3 diatas :

$Nim \rightarrow N\_Mahasiswa, P\_Akademik$

Tetapi

$Nim \nrightarrow K\_Kuliah$

Sehingga relasi mahasiswa diatas harus didekomposisi menjadi dua relasi baru yaitu relasi Kuliah(Nim,K\_Kuliah) dan Perwalian (Nim, N\_Mahasiswa, P\_Akademik)

Nim#	N_Mahasiswa	P_Akademik
1592001	Amir	Ade
1592003	Abdul	Bambang
1593001	Mimi	Kadir

Gbr. 5.4.1 Dalam 2 NF

Nim#	K_Kuliah
1592001	101
1592003	102
1593001	101

Gbr. 5.4.2 Dalam 2 NF



#### 5.4 BENTUK NORMAL KETIGA (3 NF)

Suatu relasi dikatakan dalam normal ketiga jika dan hanya jika setiap tupel mempunyai nilai primary key yang menjelaskan suatu entiti, bersamaan dengan suatu himpunan kosong atau ketergantungan dari nilai atribut yang telah dijelaskan sebelumnya. atau bentuk 3 NF jika dan hanya jika atribut yang bukan key ( jika ada) memenuhi ;

1. Saling bebas ( *mutualy independent*)
2. Bergantung pada kunci utama.
3. Non key atribut adalah setiap atribut yang tidak berhubungan pada primary key pada relasi yang ditentukan.

Dua atau lebih atribut dikatakan *mutualy independen* jika tidak satupun dari mereka tergantung fungsional pada kombinasi yang ada. Independen berarti bahwa setiap atribut tertentu dapat di ubah secara bebas terhadap data atribut yang lain.

Contoh:

N_Kajur	Jurusan	Telp_Jurusan
Abas	Informatika	7561236
Badu	Mesin	7561245
Budi	Elektro	7564545

Gbr. 5.5 Tidak Dalam 3 NF

Terlihat bahwa :

N\_Kajur → Jurusan

Jurusan → Telp\_Jurusan

Agar relasi dalam 3 NF harus didekomposisi menjadi :

N_Kajur	Jurusan
Abas	Informatika
Badu	Mesin
Budi	Elektro

Gbr 5.5.1 Dalam 3 NF

Jurusan	Telp_Jurusan
Informatika	7561230
Mesin	7561245
Elektro	7564545

Gbr 5.5.2 Dalam 3 NF

#### 5.4 BENTUK BOYCE CODD NORMAL FORM (BCNF)

Boyce-Codd normal form memiliki aturan yang lebih ketat dari bentuk normal ketiga. Untuk menjadi BCNF setiap atribut harus bergantung fungsi pada atribut superkey.

Kondisi BCNF apabila memenuhi.

1. Terdapat dua atau lebih kunci kandidat
2. Kunci kandidat merupakan kunci komposit
3. Terdapat atribut-atribut yang "overlapping" pada kandidat key-nya

Contoh :

- Diberikan relasi berikut dengan ketentuan:
- Mahasiswa dapat mengambil lebih dari satu praktikum



- Setiap praktikum diawasi oleh tiga asisten
- Mahasiswa dibimbing oleh satu asisten/praktikum
- Asisten boleh memimpin lebih dari satu praktikum

Nim#	Praktikum	Asisten
1592001	Basic	Nina Minah Udin
1592003	Fortran	
1593001	Pascal	

Gbr. 5.6 Tidak dalam BCNF

Relasi diatas dalam 3 NF tetapi tidak dalam BCNF karena praktikum bergantung kepada asisten. Maka relasi diatas didekomposisi menjadi

Nim#	Asisten
1592001	Nina
1592003	Minah
1593001	Udin

Gbr 5.6.1 Dalam BCNF

Asisten	Praktikum
Nina	Basic
Minah	Fortran
Udin	Pascal

Gbr 5.6.1 Dalam BCNF

## 5.5 BENTUK NORMAL KEEMPAT (4 NF)

Suatu relasi R dikatakan dalam bentuk normal keempat 4 NF jika dipenuhi MVD pada R dari  $A \twoheadrightarrow B$ , seluruh atribut dari R juga tergantung fungsional pada A.

Dengan kata lain, R dalam 4NF jika R dalam BCNF, dan setiap MVD juga FD pada R.

K_Jurusan	Jurusan	N_Kajur
15	Informatika	Abas
12	Mesin	Badu
11	Elektro	Budi

## 5.6 BENTUK NORMAL KELIMA (5NF)

Bentuk normal kelima digunakan untuk menghilangkan anomali akibat ketergantungan join (FD).

Suatu relasi R dalam 5 NF jika dan hanya jika setiap ketergantungan join (FD) di R termasuk dalam kunci kandidat R.

K_Jurusan	Jurusan	N_Kajur
15	Informatika	Abas
12	Mesin	Badu
11	Elektro	Budi

Gbr. 5.7 Tidak dalam 5 NF

Relasi jurusan(K\_jurusan#, Kajur, Jurusan memenuhi 5 NF karena dapat ditulis sebagai hasil join dua relasi berikut.

K_Jurusan	N_Kajur
15	Abas
12	Badu
11	Budi

Gbr 5.7.1 Dalam 5 NF

K_Jurusan	Jurusan
15	Informatika
12	Mesin
11	Elektro

Gbr 5.7.1 Dalam 5 NF



## KESIMPULAN

Penggunaan model relasional menuntut adanya normalisasi untuk menjamin integritas dan menghilangkan redudansi data. Metode normalisasi ini tidak hanya digunakan pada pendisainan basis data yang rumit / berukuran besar, tetapi harus dilakukan untuk seluruh pendisainan basis data. Karena tabel yang kecil belum berarti sudah memenuhi aturan – aturan normalisasi, dan tabel yang besar belum tentu melanggar aturan-aturan normalisasi. Cara yang paling aman dan baik adalah menguji setiap tabel yang dibuat dengan aturan-aturan normalisasi.

### **Keuntungan:**

Data yang disimpan dalam bentuk tabel-tabel kecil mudah dibaca

Menghilangkan redudansi data

Walaupun relasi sudah didekomposisi namun hasil dekomposisi tersebut tidak akan menghilangkan data apabila relasi digabungkan kembali

### **Kerugian:**

Karena hasil dekomposisi akan menghasilkan banyak tabel sehingga dibutuhkan lebih banyak ruang memori.

## Daftar Pustaka

1. Henry F. Korth, Abraham Silberschatz; Database System Concepts; McGrawhill; 1991
2. C.J. Date; An Introduction to Database Design; 6<sup>th</sup> ed; Addison Welley; 1995
3. Sitansu S. Mitra; Principles of Relational Database System; Prentice Hall; 1991
4. Harianto Kristianto, Ir, Konsep dan Perancangan Database; Andi Offset Yogyakarta; 1994
5. Ludwig Slusky, Cases in Database Design, Times Mirror/ Mosby Colledge Publishing; 1988
6. Kolman, Busby; Discrete Mathematical Structure for Computer Science; 2<sup>nd</sup> ed; Prentice Hall; 1996
7. Kendal, Kenneth E, Kendall, Julie E, System Analysis and Design, 3<sup>rd</sup> ed; 1995
8. Hansen Gari W, Hansen James S.V; Database Management and Design; 2<sup>nd</sup> Ed; Prentice Hall; 1996