



NORMALISASI DATA Dan KEBEBASAN DATA

Oleh : Rahmi Hidayati, S.Kom., M.Cs

Materi

Normalisasi Data : Atribut tabel, *key* dan atribut deskripsi, atribut sederhana dan atribut komposit, atribut bernilai tunggal dan atribut bernilai banyak, Atribut harus bernilai Null, Atribut turunan, Domain dan tipe data.

Kebebasan Data : Ketergantungan fungsi, Normalisasi dengan ketergantungan fungsional, *Loss Less Join Decomposition, Dependency Preservation, Boyce Codd Normal Form*, Bentuk normal tahap pertama, kedua, tahapan ketiga, ke empat dan kelima.

PENDAHULUAN

- Dalam merancang basis data dapat dilakukan dengan cara :
 1. Menerapkan normalisasi terhadap struktur tabel yang diketahui atau
 2. Langsung membuat model *Entity - Relationship*

Data Mahasiswa

NIM	NamaMahasiswa	AlamatMahasiswa	TglLahir
150001	Rudi Kurnia	Jl. Dago Pojok 91, Bandung 40135	2 Jan 1992
150002	Budi Putra	Jl. Pasantren 25D, Cimahi 40533	6 Okt 1991
160003	Indah Sari	Jl. Anggrek 15, Sumedang 45323	15 Mei 1991
160004	Andi Akbar	Jl. Titiran 12, Bandung 40133	21 Jun 1992

Atribut Tabel :

1. Key dan Atribut Deskriptif
2. Atribut sederhana dan Atribut Komposit
3. Atribut bernilai tunggal dan atribut bernilai banyak
4. Atribut harus bernilai (*mandatory*) dan nilai *Null*
5. Atribut Turunan

Domain Dan Tipe Data

- **Domain** adalah tipe data yang memiliki batasan-batasan nilai tertentu.
- Domain adalah himpunan nilai yang diijinkan pada suatu atribut.
- Setiap atribut di basis data relasional didefinisikan pada suatu domain.
- Domain-domain dapat berbeda untuk masing-masing atribut. Dua atribut atau lebih dapat mempunyai domain yang sama.

Domain Dan Tipe Data

DOMAIN

Tipe data	Domain
Character	Alpha Numeric (A-Z, 0-9, special character)
Number : <ul style="list-style-type: none">IntegerReal float	-32.768 s/d 32.767 Pecahan
Boolean	True/false
Date	Tanggal
Memo	Text
Ole Object	Object (Image, File, Resource)

TIPE DATA

Atribut	Tipe	Lebar
NIM	Character	10
NAMA_MHS	Character	20
ALAMAT	Character	30
TGL_LAHIR	Date	-

Type data pada MySQL

Tabel 3.1 Daftar tipe data numerik

Type data	Size	Keterangan
TINYINT	1 byte	<ul style="list-style-type: none"> • Secara SIGNED menampung data angka -128 sampai 127 • Secara UNSIGNED menampung data angka 0 sampai 255
SMALLINT	2 bytes	<ul style="list-style-type: none"> • Secara SIGNED menampung data angka 32.768 sampai 32.767 • Secara UNSIGNED menampung data angka 0 sampai 65535
MEDIUMINT	3 bytes	<ul style="list-style-type: none"> • Secara SIGNED menampung data angka 8.388.608 sampai 8.388.607 • Secara UNSIGNED menampung data angka 0 sampai 16777215
INT	4 bytes	<ul style="list-style-type: none"> • Secara SIGNED menampung data angka 2.147.483.648 sampai 2.147.483.647 • Secara UNSIGNED menampung data angka 0 sampai 4292967295
BIGINT	8 bytes	<ul style="list-style-type: none"> • Secara SIGNED menampung data angka $\pm 9.22 \times 10^{18}$ (9.223.372.036.854.775.808 sampai 9.223.372.036.854.775.808) • Secara UNSIGNED menampung data angka 0 sampai 18446744073709551615
SERIAL	1 bytes	Sama dengan tipe BIGINT dengan atribut AUTO INCREMENT NOT NULL PRIMARY KEY
FLOAT	4 bytes	Menampung data angka $\pm 1.175494351e-38$ sampai $\pm 3.402823466e+38$
DOUBLE	8 bytes	Menampung data angka $\pm 2.2205738585072014e-308$ sampai $\pm 1.7976931348623457e+308$
DECIMAL(m,d)	M+2	Tergantung pada nilai M dan D

Type data pada MySQL

Tabel 3.2 Daftar tipe data string/karakter

Tipe data	Size	Keterangan
VARCHAR	255 Bytes	Maksimal dapat menampung data sampai 255 karakter. Jika ditentukan varchar(20) maka maksimal 20 karakter dan jika lebih sedikit misal 15 karakter, akan dibaca 15 karakter.
CHAR	M bytes	Minimal atau maksimal dapat menampung data M karakter. Jika ditentukan char(3) maka baik minimal maupun maksimal harus 3 karakter. Misal tersimpan 2 tetap dibaca 3 karakter.
ENUM	1/2 bytes	Tipe data Enumerasi. Biasanya sebagai data tunggal. Seperti jenis kelamin, golongan darah, status menikah dan status kelulusan.
SET	1 – 8 bytes	Tipe data yang memiliki fungsi sama dengan ENUM.
GEOMETRY, POINT	---	Menampung tipe data obyek geometrik.
TINYTEXT	255 bytes	Mampu menampung data sampai 2^8-1 .
TINYBLOB	255 bytes	Mampu menampung data sampai 2^8-1 .
TEXT	65535 Bytes	Mampu menampung data sampai $2^{16}-1$.
BLOB	65535 Bytes	Mampu menampung data sampai $2^{16}-1$.
MEDIUMTEXT	16777215 Bytes	Mampu menampung data sampai $2^{24}-1$.
MEDIUMBLOB	16777215 Bytes	Mampu menampung data sampai $2^{24}-1$.
LOBLOB	4294967195 Bytes	Mampu menampung data sampai $2^{32}-1$.

Tipe data pada MySQL

Tabel 3.3 Daftar tipe data DATE dan TIME

Tipe data	Size	Keterangan
DATE	3 bytes	Menampung tipe data tanggal dengan format yyyy-mm-dd. Batasan nilainya dari 1000-01-01 sampai 9999-12-31.
TIME	3 bytes	Menampung tipe data waktu dengan format hh:mm:ss. Batasan nilainya dari ±838:59:59
DATETIME	8 bytes	Kombinasi tipe data DATE dan TIME dengan format yyyy-mm-dd hh:mm:ss.
YEAR	1 bytes	Menampung data tahun 4 digit dengan format yyyy.
TIMESTAMP	undefined	Digunakan untuk menyimpan data tanggal dan waktu dengan format tanpa pembatas, yyymmddhhmmdd.

Normalisasi

- **Normalisasi** merupakan sebuah teknik dalam *logical* desain sebuah basis data yang mengelompokkan atribut dari suatu relasi sehingga membentuk struktur relasi yang baik (tanpa redudansi).
- **Normalisasi** adalah proses pembentukan relasi basis data ke dalam bentuk normal, sehingga sebagian besar *ambiguity* bisa dihilangkan.

Normalisasi (Hoffer,2002)

- Sebuah proses normal untuk menentukan atribut mana yang harus dikelompokkan bersama-sama di dalam sebuah relasi.

Tujuan Normalisasi

- Normalisasi ditujukan untuk menciptakan relasi yang berstruktur baik, dengan memenuhi kondisi (Kadir, 2009) :
 - Mengandung redundansi sesedikit mungkin
 - Memungkinkan baris-baris dalam relasi disisipkan, dimodifikasi dan dihapus tanpa menimbulkan kesalahan atau ketidakkonsistenan

Redundansi

- Data yang disimpan berkali-kali
- Nama lainnya : duplikasi data

Tabel Normal

- Sebuah tabel dapat dikategorikan baik (efisien) atau normal jika telah memenuhi 3 kriteria berikut :
 1. Jika ada dekomposisi (penguraian) tabel, maka dekomposisinya harus dijamin aman (*Lossless – Join Decomposition*).
 2. Terpelihara ketergantungan fungsional pada saat perubahan data (*Dependency Preservation*).
 3. Tidak melanggar *Boyce – Code Normal Form* (BCNF).

Anomali Data (Kehilangan Data)

- Anomali Penyisipan (*INSERT*)
 - Kunci ada yang bernilai *null*
 - Ada data yang perlu diisikan ulang (redundansi tinggi)
- Anomali Pengubahan (*UPDATE*)
 - Apabila ada satu sel yang berubah, tupel seluruh tabel dengan data yang mirip juga harus diubah (kerancuan data).
Contoh : ada perusahaan yang pindah alamat
- Anomali Penghapusan (*DELETE*)
 - Saat data dihapus ada data lain yang ikut hilang

Contoh Tabel Aktivitas Siswa

SID	Aktivitas	Biaya
100	Ski	200
120	Berenang	50
150	Basket	50
175	Berenang	50



SID	Aktivitas
100	Ski
120	Berenang
150	Basket
175	Berenang

Aktivitas	Biaya
Ski	200
Berenang	50
Basket	50

DEPENDENSI

- Dependensi sering digunakan dalam normalisasi.
- Ada beberapa jenis :
 - Dependensi Fungsional
 - Dependensi Parsial
 - Dependensi Transitif

Dependensi Fungsional

- Sering kali disebut dependensi saja.
- Merupakan kekangan antara dua buah atribut atau dua buah himpunan atribut.
- Contoh : $R(A, B, \dots)$
- B memiliki **dependensi fungsional** terhadap A apabila nilai A memiliki hubungan **satu** nilai dengan B

Dependensi Fungsional

- Notasi : $A \rightarrow B$
- Dapat dibaca sebagai:
 - A panah B,
 - A menentukan B, atau
 - B tergantung secara fungsional pada A

Item di sebelah kiri disebut **penentu** atau **determinan**, sebelah kanan disebut **dependen** (“yang tergantung”)

Dependensi Fungsional

- Contoh :

Tabel PEMASOK_BARANG

Kode	Nama_Brg	Harga	Kd_Pemasok	Nm_Pemasok	Kota
T-001	TV SN 14"	600.000	P22	PT Sumber	Jakarta
T-002	TV SN 21"	950.000	P22	PT Sumber	Jakarta
T-003	TV SS 14"	450.000	P11	PT Tunas Jaya	Surabaya
T-004	TV M 34"	4.500.000	P33	PT Mekar	Semarang
T-005	TV S 24"	1.200.000	P44	PT Holic	Semarang

Dependensi Fungsional

Berdasarkan tabel tersebut, diperoleh:

- Kode \rightarrow Nama_Brg
- Kode \rightarrow Harga
- Kode \rightarrow Kd_Pemasok
- Kode \rightarrow Nm_Pemasok
- Kd_Pemasok \rightarrow Nm_Pemasok

Setiap **Kode** pasti berhubungan dengan satu **Nama_Brg** begitu juga antara **Kode** dan **Harga**. Begitu seterusnya.

Misalnya : T-001 hanya cocok dengan 1 barang, yaitu TV SN 14"

Dependensi Fungsional

- Perhatikan bagian ini:
- **Kode \rightarrow Nama_Brg**
- **Kode \rightarrow Harga**
- **Kode \rightarrow Kd_Pemasok**
- Kode \rightarrow Nm_pemasok
- Kd_Pemasok \rightarrow Nm_Pemasok

Ternyata, Kode menentukan lebih dari satu atribut. Notasinya dapat diganti sebagai berikut:

- Kode \rightarrow {Nama_Brg, Harga, Kd_Pemasok}

- Contoh lain:

Tabel DOSEN_PENDIDIKAN

No_ Dosen	Nama_Dosen	Jns_ Kelamin	Strata	Lulus_Thn
D41	Lintang	Pria	S1	1987
D41	Lintang	Pria	S2	1990
D42	Murni	Wanita	S1	1988
D42	Murni	Wanita	S2	1990
D42	Murni	Wanita	S3	1999
D43	Rio	Pria	S1	1994

- Sebuah atribut bisa tergantung lebih dari satu atribut. Contoh dalam kasus ini:

$\{\text{No_Dosen}, \text{Strata}\} \rightarrow \text{Lulus_Thn}$

Dependensi Parsial

- Syaratnya adalah sebagai berikut:
 1. B adalah atribut bukan kunci primer dan A adalah kunci primer
 2. B memiliki dependensi terhadap bagian dari A

Contoh:

Pada tabel DOSEN_PENDIDIKAN memiliki **kunci primer** {*No_Dosen*, *Strata*}.

Atribut *Jns_Klmn* bergantung pada *No_Dosen* namun tidak pada *Strata*, dikatakan memiliki **dependensi parsial**.

Dependensi Transitif

- Dikatakan memiliki dependensi transitif jika:
 1. C memiliki dependensi fungsional terhadap B
 2. B memiliki dependensi fungsional terhadap A

- Notasinya : $A \rightarrow B \rightarrow C$

Pada contoh di depan:

- $Kode \rightarrow Kd_Pemasok$
- $Kd_Pemasok \rightarrow Nm_Pemasok$
- $Kode \rightarrow Nm_Pemasok$

Jadi ketiga dependensi ini sebenarnya dapat ditulis:

- $Kode \rightarrow Kd_Pemasok \rightarrow Nm_Pemasok$

Menghilangkan Dependensi Transitif

Kode	Nama_Brg	Harga	Kd_Pemasok	Nm_Pemasok	Kota
T-001	TV SN 14"	600.000	P22	PT Sumber	Jakarta
T-002	TV SN 21"	950.000	P22	PT Sumber	Jakarta
T-003	TV SS 14"	450.000	P11	PT Tunas Jaya	Surabaya
T-004	TV M 34"	4.500.000	P33	PT Mekar	Semarang
T-005	TV S 24"	1.200.000	P44	PT Holic	Semarang

- Jika diaplikasikan untuk tabel PEMASOK_BARANG, dapat dihasilkan:
- PEMASOK(Kd_Pemasok, Nm_Pemasok, Kota)
- BARANG(Kode, Nama_Brg, Harga, Kd_Pemasok)

Proses Normalisasi

- Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat.
- Apabila tabel yang diuji belum memenuhi persyaratan tertentu, maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal.

Tahapan Normalisasi

- Tahap Normalisasi dimulai dari tahap paling ringan (1NF) hingga paling ketat (5NF).
- Biasanya hanya sampai pada tingkat 3NF atau BCNF karena sudah cukup memadai untuk menghasilkan tabel-tabel yang berkualitas baik.
- Urutan : 1NF, 2NF, 3NF, BCNF, 4NF, 5NF

Tahapan Normalisasi

1. 1st Normal Form
 - Atribut dengan nilai ganda (atau kelompok yang berulang-ulang) dihilangkan, menyisakan nilai tunggal (atau juga *null*) pada setiap *cell*.
2. 2nd Normal Form
 - **Dependensi parsial** dihilangkan
3. 3rd Normal Form
 - **Dependensi transitif** dihilangkan
4. Boyce/Codd Normal Form
 - Semua **anomali** yang tersisa dari **dependensi fungsional** dihilangkan
5. 4th Normal Form
 - Dependensi nilai ganda dihilangkan
6. 5th Normal Form
 - Anomali yang tersisa dihilangkan

1NF (*First Normal Form*)

- Bentuk normal 1NF terpenuhi jika sebuah tabel tidak memiliki atribut bernilai ganda atau banyak (*multivalued attribute*), atribut *composite*.
- Setiap atribut dalam tabel tersebut harus bernilai *atomic* (tidak dapat dibagi-bagi lagi).

2NF (*Second Normal Form*)

- Syarat dari 2NF adalah :
 - Sudah berada dalam bentuk 1NF
 - Setiap atribut bukan kunci primer telah bergantung sepenuhnya pada kunci primer.
 - Jika terdapat atribut yang tidak memiliki ketergantungan terhadap *primary key*, maka atribut tersebut harus dipindah atau dihilangkan.

Hilangkan Dependensi Parsial, langkahnya:

- Ubahlah tiap dependensi parsial menjadi relasi, dengan kunci primer sebagai penentunya.
- Ubah dependensi yang terkait langsung dengan kunci primer sebagai relasi tersendiri dan kunci primernya adalah kunci primer dalam relasi semula.

Menghilangkan Dependensi Parsial

Jika terdapat sebuah relasi R dengan:

- **$R(A,B,C,D)$**
- Kunci Primer **(A,B)**
- **$A \rightarrow D$**

Maka relasi R dapat digantikan dengan dua buah relasi R1 dan R2 sebagai berikut:

- **$R1(A,D)$** dengan *Primary Key*-nya (A)
- **$R2(A,B,C)$** dengan *Primary Key*-nya (A,B) dan *Foreign Key*-nya (A) Referensi R1

3NF (*Third Normal Form*)

- Syarat dari bentuk normal ketiga adalah :
 - Sudah berada dalam bentuk normal kedua
 - Tidak ada dependensi transitif
- Bentuk normal 3NF terpenuhi jika telah memenuhi bentuk 2NF, dan jika **tidak ada** atribut *non primary key* yang memiliki ketergantungan terhadap atribut *non primary key* yang lainnya.

Mudahnya, dependensi transitif di dalam sebuah relasi adalah sebuah dependensi fungsional di antara dua atau lebih atribut yang bukan kunci primer.

Menghilangkan Dependensi Transitif

Bila terdapat sebuah relasi R :

- R(A, B, C) dengan kunci primer (A)
- $B \rightarrow C$

Maka relasi R dapat digantikan dengan dua buah relasi R1 dan R2 sebagai berikut:

- R1 (B, C) dengan kunci primer (B)
- R2 (A, B) dengan kunci primer (A) dan kunci tamu (B) referensi R1

BCNF (*Boyce-Codd Normal Form*)

- Suatu relasi sudah dalam BCNF bila semua dependensi atau ketergantungan dalam relasi tersebut hanya terhadap kunci.
- BCNF disebut juga sebagai 3,5NF dan jarang terjadi karena pada umumnya 3NF sudah cukup.

4th Normal Form

- Bentuk normal 4NF terpenuhi dalam sebuah tabel jika telah memenuhi bentuk BCNF dan tabel tersebut tidak boleh memiliki lebih dari sebuah *multivalued attribute*.
- Untuk setiap *multivalued dependencies* (MVD) juga harus merupakan *functional dependencies*.

Contoh

Misal, tabel berikut tidak memenuhi 4NF :

Employee	Project	Skill
Jim	11	Program
Mary	5	Design
Mary	NULL	Analysis

Setiap *employee* dapat bekerja di lebih dari *project* dan dapat memiliki lebih dari satu *skill*. Untuk kasus seperti ini tabel tersebut harus di-dekomposisi menjadi :

(Employee, Project)

(Employee, Skill)

5th Normal Form

- Bentuk normal 5NF terpenuhi jika tidak dapat memiliki sebuah *lossless decomposition* menjadi tabel-tabel yang lebih kecil.
- Jika 4 bentuk normal sebelumnya dibentuk berdasarkan *functional dependency*, 5NF dibentuk berdasarkan konsep *join dependence*.
- Yakni apabila sebuah tabel telah di-dekomposisi menjadi tabel-tabel lebih kecil, harus bisa digabungkan lagi (*join*) untuk membentuk tabel semula.

Contoh Normalisasi Data

NoProyek	NamaProyek	NoPegawai	NamaPegawai	Golongan	BesarGaji
NP001	BRR	Peg01	Anton	A	1.000.000
		Peg02	Paula	B	900.000
		Peg06	Koko	C	750.000
NP002	PEMDA	Peg01	Anton	A	1.000.000
		Peg12	Sita	B	900.000
		Peg14	Yusni	B	900.000

Untuk mendapatkan hasil yang paling normal, maka proses normalisasi dimulai dari normal pertama.

Field-field tabel di atas yang merupakan group berulang : NoPegawai, NamaPegawai, Golongan, BesarGaji.

Normalisasi Pertama

Solusinya hilangkan duplikasi dengan mencari ketergantungan parsial.

Menjadikan *field-field* menjadi tergantung pada satu atau beberapa *field*.

Karena yang dapat dijadikan kunci adalah *NoProyek* dan *NoPegawai*, maka langkah kemudian dicari *field-field* mana yang tergantung pada *NoProyek* dan mana yang tergantung pada *NoPegawai*.

NoProyek	NamaProyek	NoPegawai	NamaPegawai	Golongan	BesarGaji
NP001	BRR	Peg01	Anton	A	1.000.000
NP001	BRR	Peg02	Paula	B	900.000
NP001	BRR	Peg06	Koko	C	750.000
NP002	PEMDA	Peg01	Anton	A	1.000.000
NP002	PEMDA	Peg12	Sita	B	900.000
NP002	PEMDA	Peg14	Yusni	B	900.000

Normalisasi Kedua

- *Field-field* yang tergantung pada satu *field* harus dipisah dengan tepat, misalnya *NoProyek* menjelaskan *NamaProyek* dan *NoPegawai* menjelaskan *NamaPegawai*, *Golongan* dan *BesarGaji*.

Normalisasi Kedua

TABEL PROYEK

NoProyek	NamaProyek
NP001	BRR
NP002	PEMDA

TABEL PEGAWAI

NoPegawai	NamaPegawai	Golongan	BesarGaji
Peg01	Anton	A	1.000.000
Peg02	Paula	B	900.000
Peg06	Koko	C	750.000
Peg12	Sita	B	900.000
Peg14	Yusni	B	900.000

Untuk membuat hubungan antara dua tabel, dibuat suatu tabel yang berisi *key-key* dari tabel yang lain.

TABEL PROYEKPEGAWAI

NoProyek	NoPegawai
NP001	Peg01
NP001	Peg02
NP001	Peg06
NP002	Peg01
NP002	Peg12
NP002	Peg14

Normalisasi Ketiga

Pada tabel diatas masih terdapat masalah, bahwa *BesarGaji* tergantung kepada *Golongan*-nya. Padahal disini *Golongan* bukan merupakan *field* kunci.

Artinya kita harus memisahkan *field* non-kunci *Golongan* dan *BesarGaji* yang tadinya tergantung secara parsial kepada *field* kunci *NoPegawai*, untuk menghilangkan ketergantungan transitif.

TABEL PROYEK

NoProyek	NamaProyek
NP001	BRR
NP002	PEMDA

TABEL PEGAWAI

NoPegawai	NamaPegawai	Golongan
Peg01	Anton	A
Peg02	Paula	B
Peg06	Koko	C
Peg12	Sita	B
Peg14	Yusni	B

TABEL PROYEKPEGAWAI

NoProyek	NoPegawai
NP001	Peg01
NP001	Peg02
NP001	Peg06
NP002	Peg01
NP002	Peg12
NP002	Peg14

TABEL GOLONGAN

Golongan	BesarGaji
A	1.000.000
B	900.000
C	750.000

Contoh lain Kasus Normalisasi

No-Mhs	Nm-Mhs	Jurusan	Kd-MK	Nama-MK	Kd-Dosen	Nm_Dosen	Nilai
2683	Welli	MI	MI350 MI465	Manajemen Basis Data Analisis Prc. Sistem	B104 B317	Ati Dita	A B
5432	Bakri	AK	MI350 AKN201 MKT300	Manajemen Basis Data Akuntansi Keuangan Dasar Pemasaran	B104 D310 B212	Ati Lia Lola	C B A

1 NF

No-Mhs	Nm-Mhs	Jurusan	Kd-MK	Nama-MK	Kd-Dosen	Nm_Dosen	Nilai
2683	Welli	MI	MI350	Manajemen Basis Data	B104	Ati	A
2683	Welli	MI	MI465	Analisis Prc. Sistem	B317	Dita	B
5432	Bakri	AK	MI350	Manajemen Basis Data	B104	Ati	C
5432	Bakri	AK	AKN201	Akuntansi Keuangan	D310	Lia	B
5432	Bakri	AK	MKT300	Dasar Pemasaran	B212	Lola	A

2NF

Tabel Mahasiswa

<u>No-Mhs</u>	Nama-Mhs	Jurusan
2683	Welli	MI
5432	Bakri	AK

Tabel Kuliah

<u>Kode-MK</u>	Nama-MK	Kode-Dosen	Nama-Dosen
MI350	Manajemen Basis Data	B104	Ati
MI465	Analisis Prc. Sistem	B317	Dita
AKN201	Akuntansi Keuangan	D310	Lia
MKT300	Dasar Pemasaran	B212	Lola

Tabel Nilai

<u>No-Mhs</u>	<u>Kode MK</u>	Nilai
2683	MI350	A
2683	MI465	B
5432	MI350	C
5432	AKN201	B
5432	MKT300	A

3NF

Tabel Mata Kuliah

<u>Kode-MK</u>	Nama-MK	Kode-Dosen
MI350	Manajemen Basis Data	B104
MI465	Analisis Prc. Sistem	B317
AKN201	Akuntansi Keuangan	D310
MKT300	Dasar Pemasaran	B212

Tabel Dosen

<u>Kode-Dosen</u>	Nama-Dosen
B104	Ati
B317	Dita
D310	Lia
B212	Lola