

BAB IV

Normalisasi Data



BAB IV

1. Pengertian Normalisasi Data



Perancangan basis data diperlukan, agar dapat terbentuk basis data yang efisien dalam penggunaan ruang penyimpanan, cepat dalam pengaksesan dan mudah dalam manipulasi (tambah, ubah, hapus) data

Perancangan basis data dilakukan dengan cara :

1. Menerapkan **Normalisasi** terhadap struktur tabel yang telah diketahui
Normalisasi -> merupakan cara pendekatan lain dalam membangun desain logik sebuah **basis data relasional** dengan menerapkan sejumlah aturan dan kriteria standar untuk menghasilkan struktur tabel yang **normal/baik**.
2. Langsung membuat **Model Entity-Relationship (Model E-R)**
Model E-R -> merupakan sebuah perangkat konseptual yang menterjemahkan/mentransformasikan kelompok-kelompok data dan relasi antar kelompok data tersebut kedalam bentuk diagram

BAB IV

1. Pengertian Normalisasi Data



Dalam pelaksanaannya, desain logik basis data relasional yang didasari baik oleh prinsip normalisasi maupun yang didasari oleh transformasi secara hati-hati dari Model E-R ke bentuk fisik akan menghasilkan hasil yang mirip.

Dalam pendekatan **Normalisasi** -> perancang basis data bertitik tolak dari situasi yang nyata dimana ia telah memiliki item-item data yang siap ditempatkan dalam baris dan kolom pada tabel-tabel relasional dan ia telah mengetahui sejumlah aturan tentang keterhubungan antara item-item data tersebut.

Dalam pendekatan **Model E-R** -> perancang basis data dengan langsung membuat model data jika yang telah diketahui baru prinsip-prinsip sistem secara keseluruhan karena adanya kelangkaan data/fakta yang dimiliki.

Faktanya di lapangan, kedua pendekatan ini dilakukan bersama-sama, berganti-ganti, dan dapat saling memperkuat satu sama lain, dari contoh data (fakta) yang telah dimiliki dilakukan Normalisasi kemudian hasilnya diwujudkan dalam Model E-R, setelah itu diimplementasikan dalam bentuk sejumlah struktur tabel dalam sebuah basis data, struktur tabel ini dapat diuji kembali dengan menerapkan aturan-aturan Normalisasi, hingga akhirnya diperoleh sebuah struktur basis data yang benar-benar **efektif dan efisien**.

BAB IV

2. Atribut Tabel (*Table attribute*)



Atribut adalah karakteristik/sifat-sifat dari entity atau relationship, yang menyediakan penjelasan detail tentang entity atau relationship tersebut.

Atribut identik dengan kolom data (*field*) pada sebuah tabel.

Nilai Atribut merupakan suatu data aktual atau informasi yang disimpan pada suatu atribut di dalam suatu entity atau relationship.

Dari bab sebelumnya, dapat diketahui :

- tabel Mahasiswa memiliki 4 buah atribut : *nim, nama_mhs, alamat_mhs, tgl_lahir*
- tabel Dosen memiliki 2 buah atribut : *nama_dos, alamat_dos*
- tabel Kuliah memiliki 4 buah atribut : *kode_kul, nama_kul, sks, semester*
- tabel Nilai memiliki 4 buah atribut : *nama_kul, nim, nama_mhs, indeks_nilai*
- tabel Jadwal memiliki 4 buah atribut : *nama_kul, tempat, waktu, nama_dos*

Atribut dapat dibedakan menjadi 5 kelompok, yaitu :

BAB IV 2. Atribut Tabel (*Table attribute*)



2.1 Key dan Atribut Deskriptif

Key adalah satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris data (*row*) dalam tabel secara unik

Jika suatu atribut dijadikan sebagai key, maka tidak boleh ada dua atau lebih baris data dengan nilai yang sama untuk atribut tersebut.

Ada 4 macam key yang dapat diterapkan pada suatu tabel, yaitu :

1. *Super key*
2. *Candidate key*
3. *Primary key*
4. *Foreign key*

BAB IV 2. Atribut Tabel (*Table attribute*)



2.1 Key dan Atribut Deskriptif

Super key -> merupakan satu atau lebih atribut (kumpulan atribut) yang dapat membedakan setiap baris data dalam sebuah tabel secara unik.

nim	nama_mhs	alamat_mhs	tgl_lahir
980001	I Made Suta	Jl. Dewi Sartika No.12, Bangli 40121	05 Desember 1980
980002	I Wayan Sura	Jl. Kartini No.10, Badung 45123	06 Maret 1980
980003	Dewa Made Gita	Jl. Flamboyan No.23, Singaraja 40151	17 Juni 1980
980004	Dewi Asih	Jl. A Yani 5, Gianyar 40124	08 Nopember 1980

Dari tabel Mahasiswa diatas yang dapat menjadi *Super key* :

- (*nim, nama_mhs, alamat, tgl_lahir*)
- (*nim, nama_mhs, alamat*)
- (*nim, nama_mhs*)
- (*nama_mhs*), jika bisa dijamin tidak ada nilai yang sama untuk atribut ini.
- (*nim*)

BAB IV

2. Atribut Tabel (Table attribute)

2.1 Key dan Atribut Deskriptif

Candidate key -> merupakan kumpulan atribut **minimal** yang dapat membedakan setiap baris data dalam sebuah tabel secara unik.

nim	nama_mhs	alamat_mhs	tgl_lahir
980001	I Made Suta	Jl. Dewi Sartika No.12, Bangli 40121	05 Desember 1980
980002	I Wayan Sura	Jl. Kartini No.10, Badung 45123	06 Maret 1980
980003	Dewa Made Gita	Jl. Flamboyan No.23, Singaraja 40151	17 Juni 1980
980004	Dewi Asih	Jl. A Yani 5, Gianyar 40124	08 Nopember 1980

Dari tabel Mahasiswa diatas yang dapat menjadi *Candidate key* :

- (nim)
- (nama_mhs), jika bisa dijamin tidak ada nilai yang sama untuk atribut ini.

Salah satu dari *Candidate key* diatas dapat dijadikan sebagai *Primary key* (kunci utama)

Pemilihan *Primary key* dari sejumlah *Candidate key* tersebut umumnya didasari oleh :

- Key tersebut lebih sering (lebih natural) untuk dijadikan sebagai acuan
- Key tersebut lebih ringkas
- Jaminan keunikan key tersebut lebih baik

Dari pertimbangan diatas, dari kedua *Candidate key* yang ada di tabel Mahasiswa, maka yang lebih cocok dipilih sebagai *Primary key* adalah (nim)

BAB IV

2. Atribut Tabel (Table attribute)

2.1 Key dan Atribut Deskriptif

Foreign key / Kunci tamu -> Atribut dengan domain yang sama yang menjadi kunci utama pada sebuah tabel tetapi pada tabel lain atribut tersebut hanya sebagai atribut biasa

Kunci tamu ditempatkan pada entitas anak dan sama dengan *Primary key* induk direlasikan

nama_kul	nim	nama_mhs	indeks_nilai
Struktur Data	980001	I Made Suta	A
Struktur Data	980002	I Wayan Sura	B
Basis Data	980001	I Made Suta	
Basis Data	980004	Dewi Asih	
...

Yang menjadi *Foreign key* pada tabel Nilai diatas adalah (nim)

nim	nama_mhs	alamat_mhs	tgl_lahir
980001	I Made Suta	Jl. Dewi Sartika No.12, Bangli 40121	05 Desember 1980
980002	I Wayan Sura	Jl. Kartini No.10, Badung 45123	06 Maret 1980
980003	Dewa Made Gita	Jl. Flamboyan No.23, Singaraja 40151	17 Juni 1980
...

Atribut Deskriptif adalah atribut-atribut yang tidak menjadi atau merupakan anggota dari *Primary key*. Dari tabel mahasiswa diatas yang termasuk dalam atribut deskriptif adalah : (nama_mhs, alamat_mhs, tgl_lahir)

BAB IV

2. Atribut Tabel (Table attribute)

2.2 Atribut Sederhana (Simple Attribute) dan Atribut Komposit (Composite Attribute)

Atribut sederhana -> atribut atomik yg tidak dapat diuraikan lagi menjadi sub-sub atribut.

Atribut komposit -> atribut yang masih dapat diuraikan lagi menjadi sub-sub atribut yang masing-masing memiliki makna.

nim	nama_mhs	alamat_mhs	tgl_lahir
980001	I Made Suta	Jl. Dewi Sartika No.12, Bangli 40121	05 Desember 1980
980002	I Wayan Sura	Jl. Kartini No.10, Badung 45123	06 Maret 1980
980003	Dewa Made Gita	Jl. Flamboyan No.23, Singaraja 40151	17 Juni 1980
980004	Dewi Asih	Jl. A Yani 5, Gianyar 40124	08 Nopember 1980

Atribut sederhana
(nama_mhs)

Atribut komposit
(alamat_mhs)

dapat diuraikan menjadi 3 atribut

Atribut Komposit dapat didekomposisi menjadi Atribut Sederhana disesuaikan dengan kebutuhan sistem

alamat	nama_kota	kode_pos
Jl. Dewi Sartika No.12	Bangli	40121
Jl. Kartini No.10	Badung	45123
...

BAB IV

2. Atribut Tabel (Table attribute)

2.3 Atribut Bernilai Tunggal (Single-valued attribute) dan Atribut Bernilai Banyak (Multivalued attribute)

Atribut Bernilai Tunggal -> ditujukan pada atribut-atribut yang memiliki paling banyak satu nilai untuk setiap baris data

Atribut Bernilai Banyak -> ditujukan pada atribut-atribut yang dapat diisi dengan lebih dari 1 (satu) nilai, tetapi jenisnya sama

nim	nama_mhs	alamat_mhs	tgl_lahir	hobbi
980001	I Made Suta	Jl. Dewi Sartika No.12 ...	05 Desember 1980	memancing, bulu tangkis, nonton tv
980002	I Wayan Sura	Jl. Kartini No.10 ...	06 Maret 1980	membaca
980003	Dewa Made Gita	Jl. Flamboyan No.23 ...	17 Juni 1980	
...

Atribut Bernilai Tunggal
(nim, nama_mhs, alamat_mhs, tgl_lahir)

Atribut Bernilai Banyak
(hobbi)

Atribut-atribut bernilai tunggal diatas hanya dapat berisi 1 (satu) nilai. Jika misalnya ada mahasiswa memiliki 2 alamat maka hanya salah satu saja yang boleh diisikan kedalam atribut alamat_mhs

Seorang mahasiswa ada yang mempunyai 1 hobbi, banyak hobbi dan ada yg tidak mempunyai hobbi

BAB IV

2. Atribut Tabel (Table attribute)

2.4 Atribut Harus Bernilai (Mandatory Attribute) dan Nilai Null (Non Mandatory Attribute)

Atribut harus bernilai -> adalah atribut pada sebuah tabel yang harus berisi data (nilainya tidak boleh kosong)

Nilai Null -> adalah atribut yang nilainya boleh kosong. Nilai (konstanta) Null digunakan untuk mengisi atribut yang nilainya memang belum siap atau tidak ada. Nilai Null tidak sama dengan spasi.

nim	nama_mhs	alamat_mhs	tgl_lahir	hobbi
980001	I Made Suta	Jl. Dewi Sartika No.12 ...	05 Desember 1980	memancing, bulu tangkis, nonton tv
980002	I Wayan Sura			membaca
980003	Dewa Made Gita	Jl. Flamboyan No.23 ...	17 Juni 1980	

Mandatory Attribute (nim, nama_mhs)

Non Mandatory Attribute (alamat_mhs, tgl_lahir, hobbi)

Atribut *nim* dan *nama_mhs* digolongkan sebagai *Mandatory Attribute*, karena setiap mahasiswa yg datanya ingin disimpan ke tabel Mahasiswa, paling tidak harus diketahui NIM dan Nama-nya.

Ada atribut-atribut pada suatu tabel yang nilainya boleh dikosongkan (*Non Mandatory Attribute*)

BAB IV

2. Atribut Tabel (Table attribute)

2.5 Atribut Turunan (Derived Attribute)

Atribut turunan -> atribut yang nilai-nilainya diperoleh dari pengolahan atau dapat diturunkan dari atribut atau tabel lain yang berhubungan

Atribut turunan sebenarnya dapat ditiadakan dari sebuah tabel, karena nilai-nilainya bergantung pada nilai yang ada di atribut lainnya.

Penambahan atribut *angkatan* dan *ip* (indeks prestasi) pada tabel Mahasiswa berikut merupakan contoh penggunaan Atribut Turunan

nim	nama_mhs	alamat_mhs	tgl_lahir	angkatan	ip
980001	I Made Suta	Jl. Dewi Sartika No.12 ...	05 Desember 1980	1998	2.85
980002	I Wayan Sura			1998	3.40
980003	Dewa Made Gita	Jl. Flamboyan No.23 ...	17 Juni 1980	1998	2.98

Nilai - nilai pada atribut *angkatan* dapat diketahui dari atribut *nim*, dimana 2 karakter pertama dalam *nim* menyatakan 2 digit bilangan tahun masuknya mahasiswa bersangkutan

Nilai - nilai pada atribut *ip* diperoleh dari pengolahan dengan menerapkan formula tertentu yang melibatkan *indeks_nilai* di tabel Nilai dan atribut *sks* yang ada di tabel Kuliah

BAB IV

3. Domain dan Tipe Data



Tipe Data lebih merujuk pada kemampuan penyimpanan data **yang mungkin** bagi suatu atribut secara fisik, tanpa melihat layak/tidaknya data tersebut bila dilihat dari kenyataan pemakaiannya.

Tabel Kuliah

kode_kul	nama_kul	sks	semester
INF1014	Struktur Data	3	1
INF1012	Basis Data	3	2
INF2011	Algoritma	3	1
INF3044	Matematika I	3	2

Contoh : pada tabel Kuliah diatas, Tipe data untuk atribut *sks* adalah integer, dengan begitu secara fisik dapat menyimpan nilai -1, 0 atau 100 untuk atribut *sks* tersebut, tetapi kita mengetahui dengan pasti, bahwa nilai-nilai itu (-1, 0 atau 100) sebetulnya tidak pantas (*invalid*) untuk menjadi data pada atribut *sks*.

Domain lebih ditekankan pada batas-batas nilai yang diperbolehkan bagi suatu atribut, dilihat dari kenyataan yang ada.

Contoh : Domain nilai untuk atribut *sks* adalah 1, 2, 3 atau 4 karena nilai-nilai tersebut lebih layak (*valid*) dan sesuai dengan kenyataan yang ada.

BAB IV

3. Domain dan Tipe Data



Domain nilai bagi setiap atribut perlu dilihat dan dipertimbangkan pada saat perancangan basis data.

Tipe data bagi setiap atribut relevan untuk diperhitungkan pada saat implementasi basis data.

BAB IV

4. Ketergantungan Fungsional (*Functional Dependency*)



Diberikan sebuah tabel T berisi paling sedikit 2 buah atribut, yaitu A dan B. Kita dapat menyatakan notasi berikut ini :

$$A \rightarrow B$$

yang berarti A secara fungsional menentukan B atau B secara fungsional tergantung pada A, jika dan hanya jika untuk setiap kumpulan baris data (*row*) yang ada di tabel T, pasti ada 2 baris data (*row*) di tabel T dengan nilai untuk A yang sama, maka nilai untuk B pasti juga sama.

Definisi yang lebih formal untuk hal diatas yaitu : diberikan 2 row r1 dan r2 dalam tabel T dimana $A \rightarrow B$, Jika $r1(A) = r2(A)$, maka $r1(B) = r2(B)$

BAB IV

4. Ketergantungan Fungsional (*Functional Dependency*)



Untuk menjelaskan Ketergantungan Fungsional tersebut, berikut diberikan contoh menggunakan tabel Nilai berikut :

Tabel Nilai

nama_kul	nim	nama_mhs	indeks_nilai	
Struktur Data	980001	I Made Suta	A	... baris 1
Struktur Data	980002	I Wayan Sura	B	... baris 2
Basis Data	980001	I Made Suta		... baris 3
Basis Data	980004	Dewi Asih		... baris 4
Basis Data	980002	I Wayan Sura		... baris 5
Algoritma	980002	I Wayan Sura	C	... baris 6

Dari data tabel Nilai diatas (sesuai fakta yg ada), maka Ketergantungan Fungsional (KF) yang dapat diajukan adalah :

1. $nim \rightarrow nama_mhs$

Yang berarti bahwa atribut *nama_mhs* hanya tergantung pada atribut *nim*.

Faktanya : untuk setiap nilai *nim* yang sama, maka pasti nilai *nama_mhs*-nya juga sama (Contoh : baris 1 dan baris 3 nilai *nim* yang sama, maka pasti nilai *nama_mhs*-nya juga sama)

BAB IV

4. Ketergantungan Fungsional (*Functional Dependency*)



2. $\text{nama_kul nim} \rightarrow \text{indeks_nilai}$

Yang berarti bahwa atribut *indeks_nilai* tergantung pada atribut *nama_kul* dan *nim* secara bersama-sama

Faktanya : bahwa untuk setiap nilai *nama_kul* dan *nim* yang sama, maka nilai *indeks_nilai*-nya juga sama, karena (*nama_kul, nim*) merupakan *key* (yang unik) untuk tabel Nilai.

KF tersebut sesuai dengan pengertian bahwa setiap indeks nilai diperuntukkan pada mahasiswa tertentu untuk mata kuliah tertentu yang diambilnya.

Dari tabel Nilai juga dapat diajukan sejumlah ketidaktergantungan (non KF) dengan hanya melihat fakta yang ada, yaitu :

1. $\text{nama_kul} \not\rightarrow \text{nim}$

Yang berarti atribut *nim* tidak tergantung pada atribut *nama_kul*.

Buktinya terlihat pada baris 1 dan baris 2 : dengan nilai *nama_kul* yang sama, tapi nilai *nim*-nya berbeda.

BAB IV

4. Ketergantungan Fungsional (*Functional Dependency*)



2. $\text{nim} \not\rightarrow \text{indeks_nilai}$

Yang berarti atribut *indeks_nilai* tidak hanya tergantung pada atribut *nim*

Buktinya terlihat pada baris 2 dan baris 6 : dengan nilai *nim* yang sama, tapi nilai *indeks_nilai*-nya berbeda.

Dalam perancangan basis data, maka tabel harus ditentukan seluruh KF yang ada (bisa terdapat lebih dari satu KF di sebuah tabel).