

# NORMALISASI DATA BASE

## ***NORMALISASI***

Normalisasi adalah proses pembentukan struktur basis data sehingga sebagian besar ambiguity bisa dihilangkan. 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.

Sebuah tabel dikatakan baik (efisien) atau normal jika memenuhi 3 kriteria sebagai berikut :

a) Jika ada dekomposisi (penguraian) tabel, maka dekomposisinya harus dijamin aman (Lossless-Join Decomposition). Artinya, setelah tabel tersebut diuraikan / didekomposisi menjadi tabel-tabel baru, tabel-tabel baru tersebut bisa menghasilkan tabel semula dengan sama persis.

b) Terpeliharanya ketergantungan fungsional pada saat perubahan data (Dependency Preservation).

c) Tidak melanggar Boyce-Codd Normal Form (BCNF). Jika kriteria ketiga (BCNF) tidak dapat terpenuhi, maka paling tidak tabel tersebut tidak melanggar Bentuk Normal tahap ketiga (3rd Normal Form / 3NF). Normalisasi digunakan sebagai teknik analisis data pada database, sehingga dapat diketahui apakah pembuatan tabel – tabel yang terelasi dalam database itu sudah baik.

Kondisi sudah baik yaitu suatu kondisi pada saat proses insert, update, delete dan modifikasi pada satu atau beberapa atribut suatu tabel tidak berpengaruh terhadap integritas data yang lain dalam satu hubungan relasi database.

## **Tujuan Normalisasi Database**

adalah untuk menghilangkan dan mengurangi redudansi data dan memastikan dependensi data (Data berada pada tabel yang tepat). Jika data dalam database tersebut belum di normalisasi maka akan terjadi 3 kemungkinan yang akan merugikan sistem secara keseluruhan.

### ***BENTUK NORMALISASI***

Adapun bentuk-bentuk normalisasi sebagai berikut :

- a) Bentuk Normal Tahap Pertama (1st Normal Form / 1NF)
- b) Bentuk Normal Tahap Kedua (2nd Normal Form / 2NF)
- c) Bentuk Normal Tahap (3rd Normal Form / 3NF)
- d) Boyce-Code Normal Form (BCNF)
- e) Bentuk Normal Tahap (4th Normal Form / 4NF)
- f) Bentuk Normal Tahap (5th Normal Form / 5NF)
- g) Domain Key Normal Form (DKNF)
- h) Bentuk Normal Tahap (6th Normal Form / 6NF)

Namun dalam prakteknya dalam dunia industri bentuk normalisasi ini yang paling sering digunakan ada sekitar 5 bentuk.

Sudah disebutkan bahwa secara teori, bentuk normal suatu relasi bisa sampai ke tingkat lima 5NF, yaitu 1NF – 2NF – 3NF/BCNF – 4NF – 5NF.

Tetapi secara praktik dalam dunia nyata, relasi dalam suatu database sudah dibilang baik kalau sudah mencapai 3NF (bentuk normal ketiga).

Untuk lebih jelasnya ciri-ciri dari bentuk-bentuk tahapan normalisasi dapat dilihat pada gambar di bawah ini :



Anomali merupakan penyimpanganpenyimpangan atau error atau inkonsistensi data yang terjadi pada saat dilakukan proses delete, insert ataupun modify dalam suatu basis data.

## IMPLEMENTASI-1

### 1. INSERT Anomali

Situasi dimana tidak memungkinkan memasukkan beberapa jenis data secara langsung di database. Adanya error atau kesalahan yang terjadi sebagai akibat dari operasi menyisipkan (insert) record pada sebuah relasi.

NIM	KODE-MTK	BIAYA
92130	CS-200	75
92200	CS-300	100
92250	CS-200	75
92425	CS-400	150
92500	CS-300	100
92575	CD-500	50

**Contoh : Gambar 1. Tabel Relasi Kuliah**

Penambahan matakuliah baru (CS-600) yang akan diajarkan, maka matakuliah tersebut tidak bisa di insert / disisipkan ke dalam Relasi Kuliah di atas sampai ada mahasiswa yang mengambil matakuliah tersebut.

## 2. DELETE Anomali

Penghapusan data yang tidak sesuai dengan yang diharapkan, artinya data yang harusnya tidak terhapus mungkin ikut terhapus. apabila terdapat salah satu baris data dari transaksi faktur dihapus, sehingga menghilangkan informasi yang tidak ingin dihapus. Contohb:

**Tabel 2. Transaksi**

NoFaktur	TglFaktur	KdSupplier	NmSupplier	KdBarang	NmBarang	Harga
F-006	14-Feb-02	S-001	Andi	PS-001	Pensil Warna	Rp.500
				SP-005	Spidol	Rp.3.000
F-007	15-Feb-02	S-002	Enya	PS-001	Pensil Warna	Rp.500
				PS-002	Pensil Hitam	Rp.1.000
				PS-003	Pensil HB	Rp.1.500
F-008	15-Feb-02	S-003	Mandy	PS-002	Pensil Hitam	Rp.1.000
				SP-005	Spidol	Rp.3.000
F-009	15-Feb-02	S-001	Andi	PS-001	Pensil Warna	Rp.500

Terdapat penghapusan data untuk transaksi nomor faktur **F-007**, sehingga data pada baris ketiga, keempat dan kelima dihapus. Hal ini termasuk ke dalam delete anomaly karena informasi mengenai kode barang PS-003 dengan nama barang “Pensil HB” akan ikut hilang akibat dari penghapusan pada transaksi F-007. Hal ini karena data mengenai barang PS-003

dengan nama barang “Pensil HB” hanya terdapat pada baris kelima. Selain itu delete anomaly juga terjadi karena informasi mengenai kode supplier S-002 dengan nama supplier “Enya” ikut hilang akibat dari penghapusan pada transaksi F-007. Hal ini terjadi karena data mengenai supplier S-002 dengan nama Supplier “Enya” hanya terdapat pada baris ketiga saja.

## 3. UPDATE Anomali

Situasi dimana nilai yang diubah menyebabkan inkonsistensi database, dalam artian data yang diubah tidak sesuai dengan yang diperintahkan atau yang diinginkan.

Contohnya Pada tabel sebelumnya, Terjadi perubahan nama barang pada baris pertama, yaitu kode barang PS-001 semula nama barang “Pensil Warna” diubah / di-update menjadi “Pensil Warna-warni” namun untuk kode barang PS-001 pada baris ketiga dan baris kedelapan tidak dilakukan perubahan nama barang menjadi “Pensil Warna-warni”.

Hal ini termasuk ke dalam update anomaly karena terdapat ketidakkonsistenan data pada kode barang PS-001, kode barang sama namun nama barang berbeda. Tahapan Normalisasi Database :

### 1) *Unnormalized Form (UNF) / Bentuk Tidak Normal*

Merupakan bentuk tidak normal berdasarkan data yang diperoleh dan mengandung kerangkapan data. Kumpulan data yang akan direkam, tidak ada keharusan mengikukti format tertentu, dapat saja data tidak lengkap atau terduplikasi. Data dikumpulkan apa adanya.

Hal 1		Rumah Impian			Tanggal : 7-Oct-95	
Perincian Pelanggan						
Nama Pelanggan : John Key				Nomor Pelanggan CR 76		
No Property	Alamat Property	Tgl Pinjam	Tgl Selesai	Biaya	No Pemilik	Nama Pemilik
PG4	Jl. Aai / 07, Jakarta	1-Jul-93	31-Aug-95	350	CO40	Ewin
PG16	Jl. Huzai / 12, Jakarta	1-Sep-95	1-Sep-96	450	CO93	Durki

**Gambar 2. Faktur Rumah\_Impian Perincian Pelanggan**

**Tabel 2. Pelanggan\_Biaya yang belum normal (unnormal)**

No_Pelanggan	Nama	Nomor Property	Alamat Property	Tgl_Pinjam	Tgl_Selesai	Biaya	No_Pemilik	Nama_Pemilik
CR76	Badi	PG4	Jl. Aai / 07, Jakarta	1-Jul-1993	31-Aug-1995	350	CO40	Ewin
		PG16	Jl. Huzai / 12, Jakarta	1-Sep-1995	1-Sep-1996	450	CO93	Durki
CR56	Faiz	PG4	Jl. Aai / 07, Jakarta	1-Sep-1995	10-Jun-1996	350	CO40	Ewin
		PG36	Jl. Azhar / 49, Jakarta	10-Oct-1996	1-Dec-1997	375	CO93	Durki
		PG16	Jl. Huzai / 12, Jakarta	1-Jan-2017	10-Aug-2017	450	CO93	Durki

### 2) *First Normal Form (1NF)*

Pada tahap ini dilakukan penghilangan beberapa group elemen yang berulang agar menjadi satu harga tunggal yang berinteraksi di antara setiap baris pada suatu tabel, dan setiap atribut harus mempunyai nilai data yang atomic (bersifat atomic value). Atom adalah zat terkecil yang masih memiliki sifat induknya, bila terpecah lagi maka ia tidak memiliki sifat induknya.

#### **Syarat :**

- Tidak ada kolom multivalue
- Tidak ada kolom dengan domain yang sama

Langkah pertama yang dilakukan pada Tabel Pelanggan Biaya (pada Tabel 2) tersebut adalah menghilangkan elemen data yang berulang dengan data-data Pelanggan yang sesuai pada setiap baris. Hasil dari tabel yang telah memenuhi bentuk normal pertama dapat dilihat pada Tabel 3.

**Tabel 3. Pelanggan\_Biaya(1NF)**

<i>No_ Pelanggan</i>	<i>Nama</i>	<i>Nomor Property</i>	<i>Alamat Property</i>	<i>Tgl_Pinjam</i>	<i>Tgl_Selesai</i>	<i>Biaya</i>	<i>No_Pem ilik</i>	<i>Nama_ Pemilik</i>
CR76	Badi	PG4	Jl. Aai / 07, Jakarta	1-Jul-2017	31-Aug- 2017	350	CO40	Ewin
CR76	Badi	PG16	Jl. Huzai /12, Jakarta	1-Sep-2017	1-Sep-2017	450	CO93	Durki
CR56	Faiz	PG4	Jl. Aai / 07, Jakarta	1-Sep-2017	10-Jun- 2017	350	CO40	Ewin
CR56	Faiz	PG36	Jl. Azhar / 49, Jakarta	10-Oct- 2017	1-Dec-2017	375	CO93	Durki
CR56	Faiz	PG16	Jl. Huzai /12, Jakarta	1-jan-2017	10-Aug- 2017	450	CO93	Durki

### **3) Second Normal Form (2NF)**

Entitas yang atribut non-primary key-nya hanya tergantung pada full primary key. Bentuk normal kedua didasari atas konsep full functional dependency (ketergantungan fungsional sepenuhnya) .

#### **Syarat :**

- Bentuk data telah memenuhi kriteria bentuk normal kesatu.
- Attribute bukan kunci (non-key) haruslah memiliki ketergantungan fungsional sepenuhnya (fully functional dependency) pada kunci utama / primary key.

**Ketiga buah relasi tersebut dapat ditulis dalam bentuk berikut ini :**

- Relasi / Tabel Pelanggan dengan atribut-atribut: No\_Pelanggan, Nama. (**No\_Pelanggan** berfungsi sebagai primary key pada tabel / relasi tersebut).
- Relasi / Tabel Biaya dengan atribut-atribut: No\_Pelanggan, No\_Property, Tgl\_Pinjam, Tgl\_Selesai. (**No\_Pelanggan** dan **No\_Property** berfungsi sebagai primary key pada tabel / relasi tersebut).

- Relasi / Tabel Property\_Pemilik dengan atribut-atribut: No\_Property, Alamat\_Property, Biaya, No\_Pemilik, Nama\_Pemilik. (**No\_Property** berfungsi sebagai primary key pada tabel / relasi tersebut).

**Tabel / Relasi yang telah memenuhi kriteria normal ke dua (2-NF) adalah sebagai berikut :**

***Tabel 4. Tabel Pelanggan Biaya dalam bentuk normal kedua (2-NF)***

No_Pelanggan	Nama
CR76	Badi
CR56	Faiz

No_Pelanggan	No_Property	Tgl_Pinjam	Tgl_Selesai
CR76	PG4	1-Jul-93	31-Aug-95
CR76	PG16	1-Sep-95	1-Sep-96
CR56	PG4	1-Sep-95	10-Jun-93
CR56	PG36	10-Oct-93	1-Dec-94
CR56	PG16	1-Jan-95	10-Aug-95

No_Property	Alamat_Property	Biaya	No_Pemilik	Nama_Pemilik
PG4	Jl. Aai / 07, Jakarta	3530	CO40	Ewin
PG16	Jl.Huzai /12,Jakarta	450	CO93	Durki
PG36	Jl.Suciana / 68, Bogor	375	CO93	Durki

#### **4) Third Normal Form (3NF)**

Walaupun relasi 2-NF memiliki redundansi yang lebih sedikit dari pada relasi 1-NF, namun relasi tersebut masih mungkin mengalami kendala bila terjadi anomaly peremajaan (update) terhadap relasi tersebut.

Jika kita hanya mengupdate satu baris saja, sementara baris yang lainnya tidak, maka data di dalam database tersebut akan inkonsisten / tidak teratur. Anomaly update ini disebabkan oleh suatu ketergantungan transitif (transitive dependency). Kita harus menghilangkan ketergantungan tersebut dengan melakukan normalisasi ketiga (3-NF).

**Syarat :**

- Bentuk data telah memenuhi kriteria bentuk normal kedua.
- Attribute bukan kunci (non-key) harus tidak memiliki ketergantungan transitif, dengan kata lain suatu atribut bukan kunci (non\_key) tidak boleh memiliki ketergantungan fungsional (functional dependency) terhadap atribut bukan kunci lainnya, seluruh atribut bukan kunci pada suatu relasi hanya memiliki ketergantungan fungsional terhadap primary key di relasi itu saja.

Hasil akhir normalisasi tabel Pelanggan\_Biaya sampai ke bentuk normal ketiga adalah sebagai berikut.

**Tabel 5. Tabel Pelanggan Biaya dalam bentuk normal ketiga (3-NF)**

No_Pelanggan	Nama
CR76	Badi
CR56	Faiz

No_Property	Alamat_Property	Biaya	No_Pemilik
PG4	Jl. Aai / 07, Jakarta	3530	CO40
PG16	Jl.Huzai /12,Jakarta	450	CO93
PG36	Jl.Suciana / 68, Bogor	375	CO93

No_Pemilik	Nama_Pemilik
CO40	Ewin
CO93	Durki
CO93	Durki

No_Pelanggan	No_Property	Tgl_Pinjam	Tgl_Selesai
CR76	PG4	1-Jul-93	31-Aug-95
CR76	PG16	1-Sep-95	1-Sep-96
CR56	PG4	1-Sep-95	10-Jun-93
CR56	PG36	10-Oct-93	1-Dec-94
CR56	PG16	1-Jan-95	10-Aug-95

### 5) Boyce Code Normal Form (BCNF)

Secara praktis, tujuan rancangan database adalah cukup sampai pada level 3NF. Akan tetapi untuk kasus-kasus tertentu kita bisa mendapatkan rancangan yang lebih baik lagi apabila bisa mencapai ke BCNF. BCNF ditemukan oleh: R.F. Boyce dan E.F. Codd.



Suatu relasi R dikatakan dalam bentuk BCNF jika dan hanya jika setiap atribut kunci (*Key*) pada suatu relasi adalah kunci kandidat (*candidate key*). Kunci kandidat (*candidate key*) adalah atribut-atribut dari entitas yang mungkin dapat digunakan sebagai kunci (*key*) atribut.

BCNF hampir sama dengan 3NF, dengan kata lain setiap BCNF adalah 3NF. BCNF terjadi jika masih terdapat anomaly pada bentuk 3NF dikarenakan relasi memiliki lebih dari satu *candidate key*.

#### **Pembuktian BCNF :**

*Tabel 6. Supplier*

<b>Kdsupplier</b>	<b>Nmsupplier</b>
S-001	Andi
S-002	Enya
S-003	Mandy

#### **Syarat :**

- memenuhi 3<sup>rd</sup> NF, dan
- Untuk semua Functional Dependency, ruas kiri dari Functional Dependency tersebut adalah superkey

#### **Analisis :**

- Tabel sudah memenuhi 3rd NF
- Tentukan *candidate key* dan tentukan functional dependency *Candidate key* : (KdSupplier)
- FD1: (KdSupplier) -> (NmSupplier)

Berdasarkan FD, ruas kiri adalah (KdSupplier). Dalam tabel 1-1: Supplier (KdSupplier) merupakan primary key sehingga dapat disimpulkan ruas kiri adalah superkey.

#### **Kesimpulan :**

Tabel Supplier sudah memenuhi 3rd NF dan ruas kiri dari FD yang ada dalam tabel sudah merupakan superkey sehingga tabel 1-1: Supplier sudah memenuhi BCNF.

## 6) Fifth Normal Form (5NF)

Tahapan ini dilakukan untuk mengatasi terjadinya join dependent pemecahan relasi menjadi dua sehingga relasi tersebut tidak dapat digabungkan kembali menjadi satu.

### IMPLEMENTASI-2

Untuk mengimplementasikan normalisasi ke dalam sebuah rancangan database dapat diikuti Langkah-langkah sebagai berikut :

#### 1) Un-normalization

Form Bentuk yang tidak normal dimaksudkan suatu kumpulan data yang akan diolah yang diperoleh dari format– format yang beraneka ragam, masih terdapat duplikasi atau pengulangan data, bisa saja tidak sempurna atau tidak lengkap, dan sesuai fakta lapangan.

Bentuk ini didapat dari dokumen yang ada dilapangan atau manual dengan atribut bukan nilai sederhana. Untuk lebih jelasnya dapat dilihat pada tabel 1 di bawah ini :

Tabel 1. Un-normalisasi Tabel

Tabel Mahasiswa							
nim	nama	prodi	kode mtk	nama mtk	id dosen	nama dosen	nilai
1234	Roma	TI	TI4801	Sistem Basis Data	SSD	Surya	A
			TI4815	Rekayasa Perangkat Lunak	RNW	Ronal	C
2345	Beni	SI	TI4801	Sistem Basis Data	SSD	Surya	B
			UN121	Kalkulus	WHY	Wahyu	B
			UN125	Bahasa Indonesia	SAB	Sabrina	A

#### 2) Bentuk Normal Tahap Pertama (1st Normal Form / 1NF)

Adapun ciri-ciri bentuk normal 1NF adalah :

- Jika sebuah tabel tidak memiliki atribut bernilai banyak (multivalued attribute) dengan arti harus bernilai tunggal.
- Jika sebuah tabel tidak memiliki atribut composite atau kombinasinya dalam domain data yang sama. Setiap atribut dalam tabel tersebut harus bernilai atomic (tidak dapat dibagi-bagi lagi).
- Jika sebuah tabel tidak memiliki atribut turunan/derivatied value.
- Jika sebuah tabel tidak memiliki record yang bernilai ganda/ redundancy.

e) atribut composite atau kombinasinya dalam domain data yang sama.

f) Setiap atribut dalam tabel tersebut harus bernilai atomic (tidak dapat dibagi-bagi lagi).

Tabel dari un-normalisasi pada langkah pertama dapat dekomposisi menjadi tabel di bawah ini :

**Tabel 2. Tabel Mahasiswa Bentuk 1NF**

Tabel Mahasiswa							
nim	nama	prodi	kode_mtk	nama_mtk	id_dosen	nama_dosen	nilai
1234	Roma	TI	TI4801	Sistem Basis Data	SSD	Surya	A
1234	Roma	TI	TI4815	Rekayasa Perangkat Lunak	RNW	Ronal	C
2345	Beni	SI	TI4801	Sistem Basis Data	SSD	Surya	B
2345	Beni	SI	UN121	Kalkulus	WHY	Wahyu	B
2345	Beni	SI	UN125	Bahasa Indonesia	SAB	Sabrina	A

Dari tabel di atas masih terdapat atribut yang muncul secara berulang, untuk itu harus melanjutkan ke tahap normalisasi kedua.

### 3) Bentuk Normal Tahap Kedua (2nd Normal Form)

Bentuk normal 2NF terpenuhi dalam sebuah tabel jika telah memenuhi bentuk 1NF, dan semua atribut selain primary key, secara utuh memiliki Functional Dependency pada primary key

Sebuah tabel tidak memenuhi 2NF, jika ada atribut yang ketergantungannya (Functional Dependency) hanya bersifat parsial saja (hanya tergantung pada sebagian dari primary key)

Jika terdapat atribut yang tidak memiliki ketergantungan terhadap primary key, maka atribut tersebut harus dipindah atau dihilangkan. Hal ini dapat dilihat pada tabel 3 di bawah ini :

**Tabel 3. Tabel Kuliah Bentuk 2NF**

Tabel Kuliah			
<u>kode_mtk</u>	nama_mtk	<u>id_dosen</u>	nama_dosen
TI4801	Sistem Basis Data	SSD	Surya
TI4815	Rekayasa Perangkat Lunak	RNW	Ronal
UN121	Kalkulus	WHY	Wahyu
UN125	Bahasa Indonesia	SAB	Sabrina

#### 4) Bentuk Normal Tahap Ketiga (3rd Normal Form /3NF)

a) Bentuk normal 3NF terpenuhi jika telah memenuhi bentuk 2NF, dan jika **tidak ada** atribut *non primary key (biasa)* yang memiliki ketergantungan terhadap atribut *non primary key (biasa)* yang lainnya.

b) Untuk setiap *Functional Dependency* dengan notasi  $X \rightarrow A$ , maka:

- X harus menjadi *superkey* pada tabel tersebut.
- Atau A merupakan bagian dari *primary key* pada tabel tersebut.

Hal ini dapat dilihat pada tabel-tabel di bawah ini, yakni tabel mahasiswa, tabel dosen, tabel matakuliah dan tabel nilai.

**Tabel 4. Tabel-tabel Bentuk 3NF**

Tabel Mahasiswa			Tabel Dosen	
nim	nama	prodi	id_dosen	nama_dosen
1234	Roma	TI	SSD	Surya
2345	Beni	SI	RNW	Ronal
			WHY	Wahyu
			SAB	Sabrina

Tabel Matakuliah			Tabel Nilai		
kode_mtk	nama_mtk	id_dosen	nim	kode_mtk	nilai
TI4801	Sistem Basis Data	SSD	1234	TI4801	A
TI4815	Rekayasa Perangkat Lunak	RNW	1234	TI4815	C
UN121	Kalkulus	WHY	2345	TI4801	B
UN125	Bahasa Indonesia	SAB	2345	UN121	B
			2345	UN125	A

Selanjutnya **langkah kelima** dilakukan pengecekan *composite* dan *multivalued attribute* dengan cara melihat data yang mengandung tanda koma. Jika tidak ada data yang mengandung nilai koma, maka tabel yang dihasilkan tetap dan proses normalisasi selesai, dan tabel dapat diimplementasikan ke database relational.

oooOooo