

## SISTEM BASIS DATA

**Kompetensi Dasar**

Memahami tahapan normalisasi data dalam melakukan perancangan basis data.

**Pokok Bahasan**

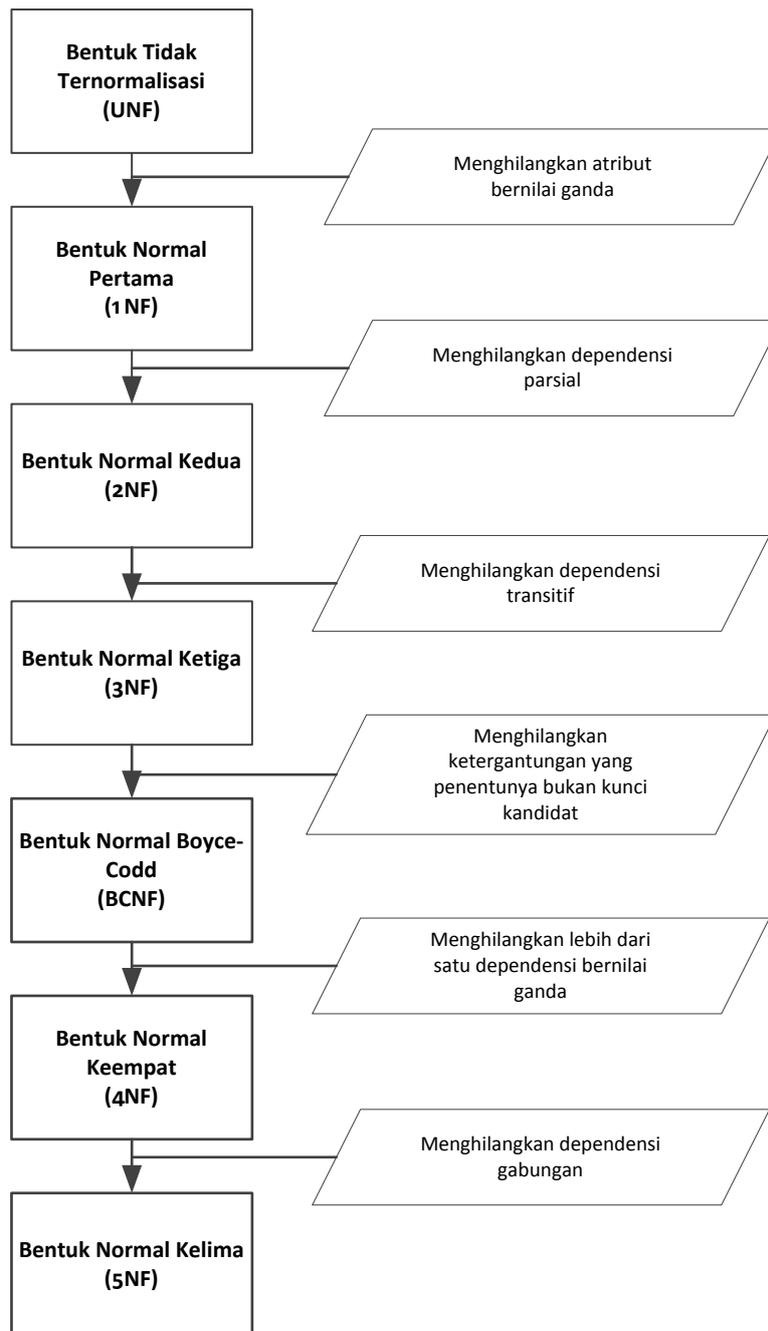
Tahapan Normalisasi Data

1. Bentuk normal pertama
2. Bentuk normal kedua
3. Bentuk normal ketiga

Normalisasi sendiri dilakukan melalui sejumlah langkah. Setiap langkah berhubungan dengan bentuk normal (*normal form*) tertentu. Dalam hal ini yang disebut dengan bentuk normal adalah suatu keadaan relasi/tabel yang dihasilkan oleh penerapan aturan-aturan yang berhubungan dengan dependensi fungsional terhadap relasi/tabel tersebut. Aturan-aturan tersebut diterapkan pada beberapa bentuk normal sebagai berikut:

1. Bentuk Normal Pertama (1NF/ *First Normal Form*)
2. Bentuk Normal Kedua (2NF/ *Second Normal Form*)
3. Bentuk Normal Ketiga (3NF/ *Third Normal Form*)
4. Bentuk Normal Boyce-Codd (BCNF/ *Boyce-Codd Normal Form*)
5. Bentuk Normal Keempat (4NF/ *Fourth Normal Form*)
6. Bentuk Normal Kelima (5NF/ *Fifth Normal Form*)

Adapun hubungan keenam bentuk normal tersebut dapat dilihat pada gambar 1 di bawah ini.



**Gambar 1.** Langkah-Langkah Dalam Normalisasi

### I. Bentuk Normal Pertama (1NF/ *First Normal Form*)

Suatu relasi/tabel dikatakan dalam bentuk normal pertama apabila setiap atribut bernilai tunggal (*Atomic Value*) untuk setiap barisnya. Untuk membentuk relasi/tabel agar berada dalam bentuk normal pertama, perlu langkah untuk menghilangkan atribut-atribut yang bernilai ganda.

Sebagai contoh, terdapat suatu bentuk yang belum ternormalisasi (UNF/*UnNormalized Form*) seperti yang ditunjukkan pada gambar 2 (tabel tersebut menyatakan informasi tentang pegawai dan klien yang ditanganinya).

no_pegawai	nama_pegawai	no_klien	nama_klien
P27	Rahayu Febrianti	K01 K02 K04	Rini Suswandi Dani Damhudi Fatwa Sari
P28	Danang	K03 K07	Randa Irwanda Suci Jelita
P29	Amira Mari	K05	Febrianti
P30	Riki Maenaki	K06 K08	Siti Aminarti Sandi Sunardi

**Gambar 2.** Contoh Bentuk yang Belum Ternormalisasi

Untuk menghilangkan atribut yang bernilai ganda, bentuk sejumlah baris hingga setiap sel berisi satu nilai. Kemudian bagian yang kosong diisi dengan data yang sesuai. Pada gambar 3 di bawah ini menunjukkan hasil setelah semua sel diatur bernilai tunggal.

**Tabel pegawai\_klien**

no_pegawai	nama_pegawai	no_klien	nama_klien
P27	Rahayu Febrianti	K01	Rini Suswandi
P27	Rahayu Febrianti	K02	Dani Damhudi
P27	Rahayu Febrianti	K04	Fatwa Sari
P28	Danang	K03	Randa Irwanda
P28	Danang	K07	Suci Jelita
P29	Amira Mari	K05	Febrianti
P30	Riki Maenaki	K06	Siti Aminarti
P30	Riki Maenaki	K08	Sandi Sunardi

**Gambar 3.** Bentuk Normal Pertama

Hal penting yang lainnya yang perlu dilakukan setelah melakukan normalisasi ke bentuk pertama adalah menentukan kunci utamanya. Kunci utama dapat dipilih melalui determinan-determinan (suatu penentu) yang muncul dalam relasi/tabel yang membuat setiap baris dapat diidentifikasi secara unik (tidak ada yang kembar). Kalau tidak ada determinan dengan satu atribut yang memenuhi, pilihlah gabungan atribut yang dapat digunakan untuk membedakan antara satu baris dengan baris lainnya. Sebagai contoh, dependensi fungsional yang ada pada relasi/tabel pegawai\_klien yaitu:

no\_pegawai -> nama\_pegawai

no\_klien -> nama\_klien

Determinan *no\_pegawai* atau *no\_klien* tidak dapat digunakan sebagai kunci utama. Satu-satunya kunci utama yang bisa digunakan adalah  $\{no\_pegawai, no\_klien\}$ . Relasi/tabel yang berada pada bentuk normal pertama ada kemungkinan masih mengandung anomali. Sebagai contoh, perhatikan gambar 3 yang masih terdapat beberapa anomali diantaranya:

1. **Anomali penyisipan.** Bila terdapat klien baru dan klien tersebut belum ditentukan akan ditangani oleh seseorang pegawai, penyisipan tidak dapat dilakukan karena kunci utama tidak boleh berupa *Null* (tidak diisi).
2. **Anomali pengubahan.** Bila nama pegawai diubah (karena ada kesalahan), ada kemungkinan lebih dari satu baris yang harus dimodifikasi. Apabila salah satu ada yang tidak diubah untuk nama yang sama, akan muncul ketidakkonsistenan.
3. **Anomali penghapusan.** Jika klien bernama Febrianti dihapus maka pegawai bernama Amira Mari akan ikut terhapus.

Maka dari itu relasi/tabel yang memenuhi bentuk normal pertama masih harus diproses untuk menjadi bentuk normal kedua.

## II. Bentuk Normal Kedua (2NF/ *Second Normal Form*)

Suatu relasi/tabel dikatakan dalam bentuk normal kedua apabila:

1. Berada pada bentuk normal pertama.
2. Semua atribut bukan kunci memiliki dependensi sepenuhnya terhadap kunci utama atau tidak mengandung dependensi parsial.

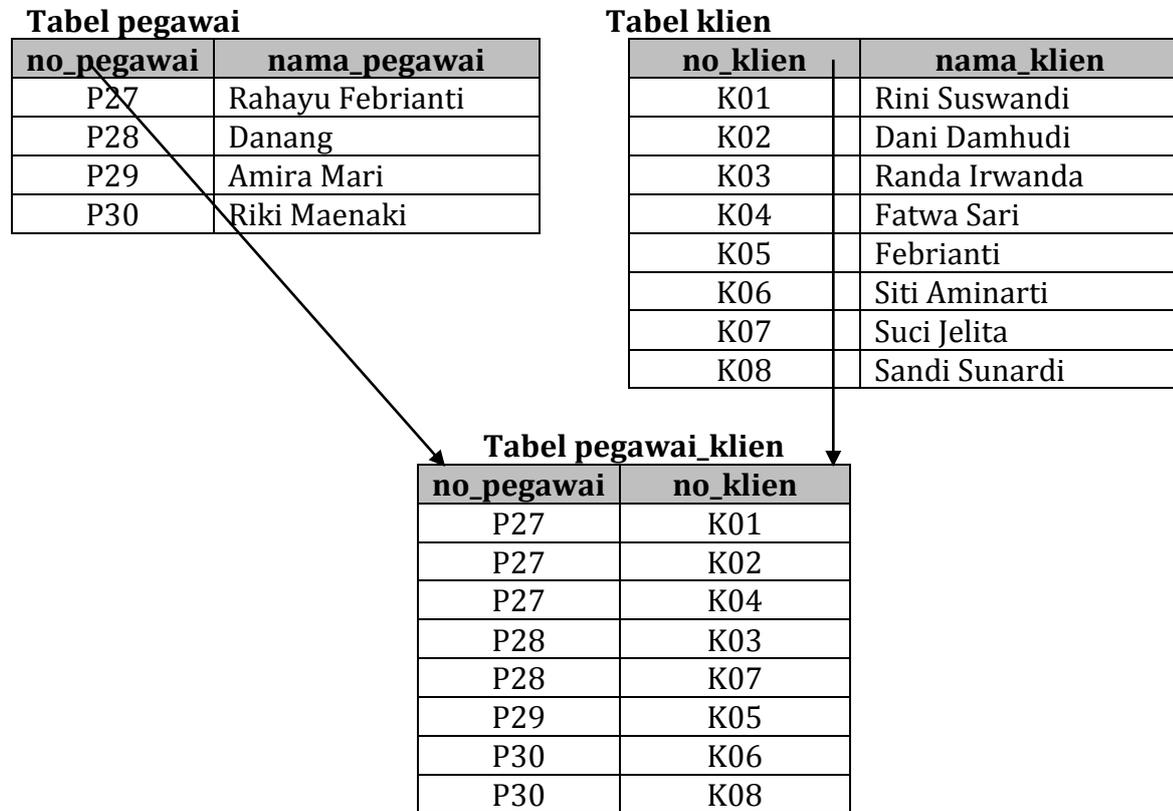
Sebagai contoh pada gambar 3 (tabel pegawai\_klien) masih terdapat dependensi parsial diantaranya:

1.  $\{no\_pegawai, no\_klien\} \rightarrow nama\_pegawai$ , dimana *nama\_pegawai* hanya bergantung pada *no\_pegawai* (bagian dari kunci utama).
2.  $\{no\_pegawai, no\_klien\} \rightarrow nama\_klien$ , dimana *nama\_klien* hanya bergantung pada *no\_klien* (bagian dari kunci utama).

Agar relasi/tabel yang mengandung dependensi parsial memenuhi bentuk normal kedua, dependensi parsialnya harus dihilangkan. Adapun cara untuk mengkonversi bentuk normal pertama ke bentuk normal kedua adalah sebagai berikut:

1. Ubahlah setiap dependensi parsial menjadi sebuah relasi/tabel, dengan kunci utama adalah determinannya.
2. Ubahlah dependensi yang terkait langsung dengan kunci utama sebagai relasi/tabel tersendiri dan kunci utamanya adalah kunci utama dalam relasi/tabel semula.

Berdasarkan cara diatas maka akan tercipta dua buah relasi/tabel baru karena terdapat dua buah dependensi parsial. Selain itu, langkah nomor dua juga menghasilkan sebuah relasi/tabel. Dengan demikian relasi/tabel pegawai\_klien didekomposisi (dipecah) menjadi tiga buah relasi/tabel seperti pada gambar 4 di bawah ini.



**Gambar 4.** Bentuk Normal Kedua

### III. Bentuk Normal Ketiga (3NF/ *Third Normal Form*)

Suatu relasi/tabel dikatakan dalam bentuk normal ketiga apabila:

1. Berada pada bentuk normal kedua.
2. Setiap atribut bukan kunci tidak memiliki dependensi transitif terhadap kunci utama.

Agar suatu relasi/tabel masuk ke dalam bentuk normal ketiga, dependensi transitif harus dibuang. Adapun cara mendekomposisi relasi/tabel yang mengandung dependensi transitif adalah sebagai berikut:

1. Bentuk relasi/tabel yang mewakili dependensi fungsional yang tidak melibatkan kunci utama dalam relasi semula. Determinannya menjadi kunci utama relasi yang dibentuk.
2. Bentuk relasi/tabel yang berisi kunci utama relasi semula. Kemudian pindahkan semua atribut bukan kunci utama yang bergantung pada kunci utama tetapi tidak

bergantung pada determinan lain ke relasi tersebut. Jadikan atribut yang menjadi kunci utama relasi semula sebagai kunci utama relasi baru. Adapun atribut yang berasal dari determinan yang menjadi perantara dalam dependensi transitif akan bertindak sebagai kunci tamu/asing.

Sebagai contoh, pada tabel `barang_pemasok` (gambar 5) terdapat dependensi transitif sebagai berikut:

**Tabel `barang_pemasok`**

<code>kode_barang</code>	<code>nama_barang</code>	<code>harga_jual</code>	<code>kode_pemasok</code>	<code>nama_pemasok</code>	<code>kota</code>
T-001	TV ABC 14"	600000	P22	PT. Citra Jaya	Bogor
T-002	TV ABC 21"	950000	P22	PT. Citra Jaya	Bogor
T-003	TV XYZ 14"	450000	P11	PT. Amerta	Bandung
T-004	TV Rhino 29"	1750000	P33	PT. Kartika	Yogya
T-005	TV Kirana 14"	475000	P44	PT. Nindya	Tangerang

**Gambar 5.** Relasi/Tabel `barang_pemasok`

`kode_barang` -> `kode_pemasok` -> {`nama_pemasok`, `kota`}

Berdasarkan cara di atas, relasi/tabel `barang_pemasok` (gambar 5) dipecah menjadi tabel `barang` dan tabel `pemasok` seperti pada gambar 6.

**Tabel `barang`**

<code>kode_barang</code>	<code>nama_barang</code>	<code>harga_jual</code>	<code>kode_pemasok</code>
T-001	TV ABC 14"	600000	P22
T-002	TV ABC 21"	950000	P22
T-003	TV XYZ 14"	450000	P11
T-004	TV Rhino 29"	1750000	P33
T-005	TV Kirana 14"	475000	P44

**Tabel `pemasok`**

<code>kode_pemasok</code>	<code>nama_pemasok</code>	<code>kota</code>
P11	PT. Amerta	Bandung
P22	PT. Citra Jaya	Bogor
P33	PT. Kartika	Yogya
P44	PT. Nindya	Tangerang

**Gambar 6.** Relasi/Tabel `barang` dan Relasi/Tabel `pemasok`