

# **PERTEMUAN 6**

## **Normalisasi Database (Conoly-chap 14) (Ramakisman -chap 15)**

Normalisasi adalah teknik desain database yang dimulai dengan memeriksa hubungan antar atribut.

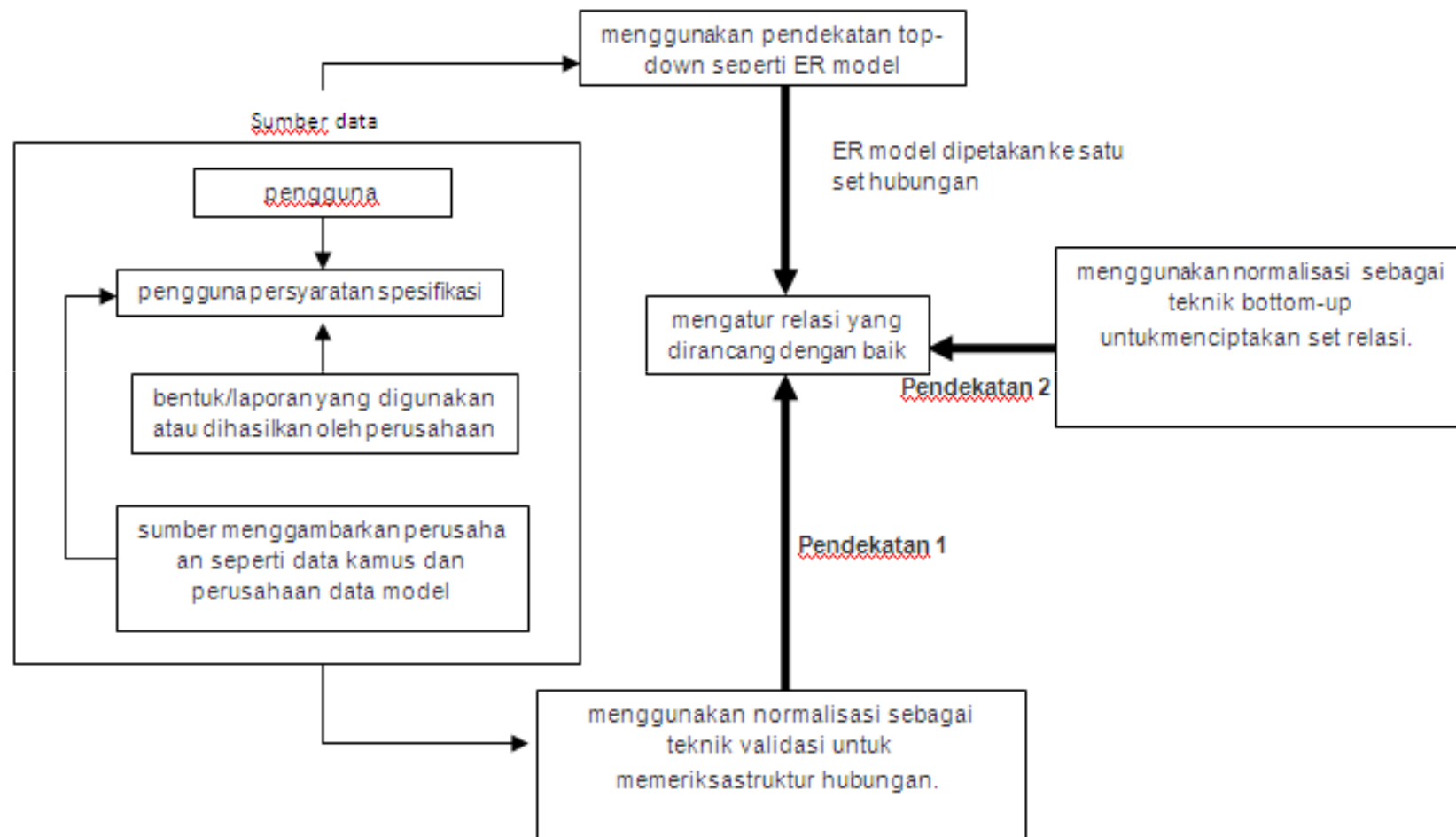
Dimana atribut menjelaskan dari data atau hubungan antara data. Normalisasi menggunakan serangkaian tes untuk membantu mengidentifikasi pengelompokan optimal untuk atribut-atribut yang akhirnya mengidentifikasi seperangkat hubungan sesuai dengan kebutuhan data perusahaan.

## Bagaimana Normalisasi Mendesign Database

Dua pendekatan utama menggunakan normalisasi:

1. Pendekatan pertama menunjukkan bagaimana normalisasi dapat digunakan sebagai teknik bottom-up desain database mandiri
2. Pendekatan 2 menunjukkan bagaimana normalisasi dapat digunakan sebagai teknik validasi untuk memeriksa struktur hubungan, yang mungkin telah dibuat dengan menggunakan pendekatan top-down

## Bagaimana normalisasi digunakan untuk desain database.



## ***Pengantar Penyempurnaan Skema: Persoalan yang Ditimbulkan oleh Redundansi***

- *Redundansi ruang penyimpanan:* beberapa data disimpan secara berulang
- *Update anomaly:* Jika satu copy data terulang tsb diubah, inkonsistensi data dpt terjadi kecuali kalau semua copy dari data tsb diubah dengan cara yang sama
- *Insertion anomaly:* Mungkin dpt terjadi kesulitan utk menyisipkan data tertentu kecuali kalau beberapa data tidak terkait lainnya juga ikut disisipkan
- *Deletion anomaly:* Mungkin dpt terjadi kesulitan utk menghapus data tertentu tanpa harus kehilangan beberapa data tidak terkait lainnya

## Persoalan yang Ditimbulkan oleh Redundansi: Contoh

| SSN         | Name      | Lot | Rating | W ages | Hours |
|-------------|-----------|-----|--------|--------|-------|
| 123-22-3666 | Attishoo  | 48  | 8      | 10     | 40    |
| 231-31-5368 | Smiley    | 22  | 8      | 10     | 30    |
| 131-24-3650 | Smethurst | 35  | 5      | 7      | 30    |
| 434-26-3751 | Guldu     | 35  | 5      | 7      | 32    |
| 612-67-4134 | Madayan   | 35  | 8      | 10     | 40    |

*Asumsi: nilai atribut wages ditentukan oleh nilai rating (utk satu nilai rating yang diberikan, hanya diperbolehkan terdapat satu nilai wages)*

- *Redundansi ruang penyimpanan:* nilai rating 8 yang berkorespondensi dg wages 10 diulang tiga kali
- *Update anomaly:* Nilai wages (yg terkait dengan nilai rating) dlm baris pertama dpt diubah tanpa membuat perubahan yg sama pada baris kedua dan kelima

## Persoalan yang Ditimbulkan oleh Redundansi: Contoh (cont'd)

| SSN         | Name      | Lot | Rating | Wages | Hours |
|-------------|-----------|-----|--------|-------|-------|
| 123-22-3666 | Attishoo  | 48  | 8      | 10    | 40    |
| 231-31-5368 | Smiley    | 22  | 8      | 10    | 30    |
| 131-24-3650 | Smethurst | 35  | 5      | 7     | 30    |
| 434-26-3751 | Guldu     | 35  | 5      | 7     | 32    |
| 612-67-4134 | Madayan   | 35  | 8      | 10    | 40    |

*Asumsi: nilai atribut wages ditentukan oleh nilai rating (utk satu nilai rating yang diberikan, hanya diperbolehkan terdapat satu nilai wages)*

- *Insertion anomaly:* Kesulitan utk menyisipkan employee baru kecuali nilai wage untuk rating dari employee tsb sudah diketahui
- *Deletion anomaly:* Jika semua baris yang terkait dg nilai rating tertentu dihapus (misalnya baris utk employee 'Smethurst' dan 'Guldu' dihapus), maka kita akan kehilangan informasi ketergantungan antara nilai *rating* dan nilai *wages* yang diasosiasikan dengan nilai rating tsb (yaitu rating = 5 dan wages = 7)



# Penyebab Anomali

Mengapa anomali - anomali ini terjadi ?

- Karena menggabungkan dua tema (konsep entitas) dalam satu relasi. Ini mengakibatkan duplikasi – duplikasi sebagai akibat dari ketergantungan antar atribut yang tidak pada tempatnya.

Solusi : **Normalisasi**



# 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.

# Normalisasi

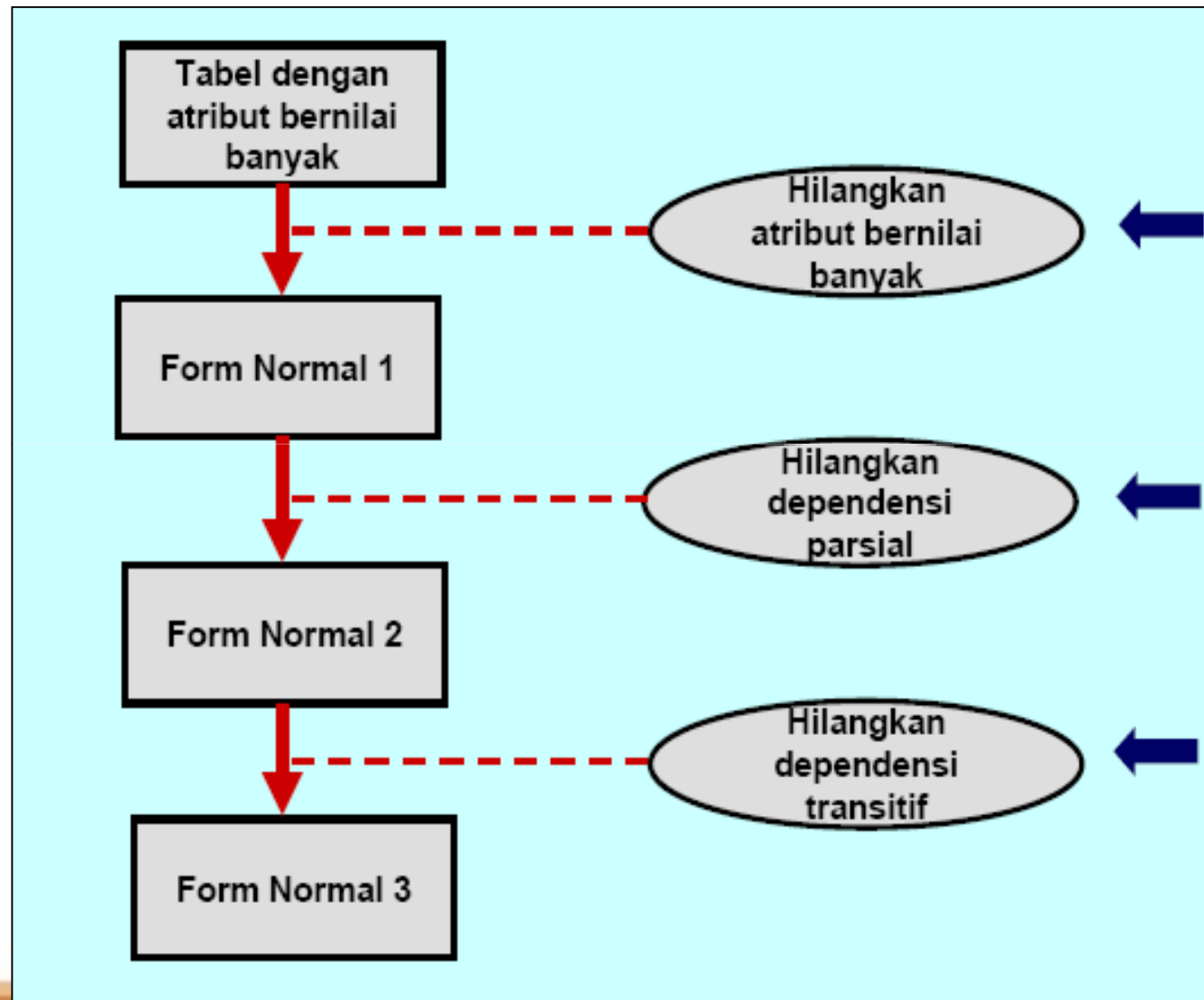
Sebuah tabel dikatakan baik (efisien) atau normal jika memenuhi 3 kriteria sbb:

1. 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.
2. Terpeliharanya ketergantungan fungsional pada saat perubahan data (Dependency Preservation).
3. Tidak melanggar Boyce-Codd Normal Form (BCNF) (-akan dijelaskan kemudian-)

# Normalisasi

Jika kriteria ketiga (BCNF) tidak dapat terpenuhi, maka paling tidak tabel tersebut tidak melanggar Bentuk Normal tahap ketiga (3rd Normal Form / 3NF).

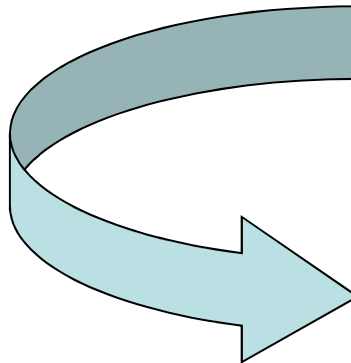
# Langkah – Langkah Normalisasi



# Tabel Universal

Tabel Universal (*Universal / Star Table*) → sebuah tabel yang merangkum semua kelompok data yang saling berhubungan, bukan merupakan tabel yang baik.

Misalnya:



# Tabel Universal

|   | nrp      | mhs.nama         | mhs.alamat                     | kodekul | namakul              | sks | kodesem | nihuruf | dsn.nama         | dsn.alamat                 |
|---|----------|------------------|--------------------------------|---------|----------------------|-----|---------|---------|------------------|----------------------------|
| ▶ | 11020001 | Abdullah Machrus | Jl. Sinoman 1/ 11 Mojokerto    | SP      | Software Perkantoran | 2 1 |         | A       | Imam Kuswardayan | Jl. Teknik Komputer 18 Sby |
|   | 11020002 | Achmad Fajril    | Jl. Panglima Sudirman XII / 30 | SP      | Software Perkantoran | 2 1 |         | A       | Imam Kuswardayan | Jl. Teknik Komputer 18 Sby |
|   | 11020003 | Achmad Ridho     | Geluran RT 13 / 03 Sepanjang S | SP      | Software Perkantoran | 2 1 |         | E       | Imam Kuswardayan | Jl. Teknik Komputer 18 Sby |
|   | 11020004 | Adi Christanto   | Jl. Wonorejo IV / 45 Surabaya  | SP      | Software Perkantoran | 2 1 |         | AB      | Imam Kuswardayan | Jl. Teknik Komputer 18 Sby |
|   | 11020005 | Aloysius Rendy   | Pucangan VII / 9 Surabaya      | SP      | Software Perkantoran | 2 1 |         | D       | Imam Kuswardayan | Jl. Teknik Komputer 18 Sby |
|   | 11020006 | Anita Rachmawati | Perum Canda Bhirawa Asri N - 1 | SP      | Software Perkantoran | 2 1 |         | E       | Imam Kuswardayan | Jl. Teknik Komputer 18 Sby |
|   | 11020007 | Arif Fachrudin   | Jl. Gubernur Suryo No.15       | SP      | Software Perkantoran | 2 1 |         | E       | Imam Kuswardayan | Jl. Teknik Komputer 18 Sby |
|   | 11020008 | Arohman Agung    | Kupang Gunung Timur IV / 24 A  | SP      | Software Perkantoran | 2 1 |         | C       | Imam Kuswardayan | Jl. Teknik Komputer 18 Sby |

# Functional Dependency

- **Notasi:  $A \rightarrow B$**

A dan B adalah atribut dari sebuah tabel. Berarti secara fungsional A menentukan B atau B tergantung pada A, jika dan hanya jika ada 2 baris data dengan nilai A yang sama, maka nilai B juga sama

- **Notasi:  $A \nrightarrow B$  atau  $A \not\rightarrow B$**

Adalah kebalikan dari notasi sebelumnya.



# Functional Dependency

Contoh tabel nilai

| Namakul          | Nrp    | namaMhs       | NiHuruf |
|------------------|--------|---------------|---------|
| Struktur Data    | 980001 | Ali Akbar     | A       |
| Struktur Data    | 980004 | Indah Susanti | B       |
| Basis Data       | 980001 | Ali Akbar     |         |
| Basis Data       | 980002 | Budi Haryanto |         |
| Basis Data       | 980004 | Indah Susanti |         |
| Bahasa Indonesia | 980001 | Ali Akbar     | B       |
| Matematika I     | 980002 | Budi Haryanto | C       |

# Functional Dependency

Functional Dependency dari tabel nilai

- **Nrp  $\rightarrow$  namaMhs**

Karena untuk setiap nilai nrp yang sama, maka nilai namaMhs juga sama.

- **{NamaKul, nrp}  $\rightarrow$  NiHuruf**

Karena atribut Nihuruf tergantung pada NamaKul dan nrp secara bersama-sama. Dalam arti lain untuk NamaKul dan nrp yang sama, maka NiHuruf juga sama, karena NamaKul dan nrp merupakan key (bersifat unik).

- **NamaKul  $\nrightarrow$  nrp**
- **Nrp  $\nrightarrow$  NiHuruf**

# Bentuk-bentuk Normal

1. Bentuk Normal Tahap Pertama (1st Normal Form / 1NF)
2. Bentuk Normal Tahap Kedua (2nd Normal Form / 2NF)
3. Bentuk Normal Tahap (3rd Normal Form / 3NF)
4. Boyce-Code Normal Form (BCNF)
5. Bentuk Normal Tahap (4th Normal Form / 4NF)
6. Bentuk Normal Tahap (5th Normal Form / 5NF)

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

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

# Contoh 1

Misal data mahasiswa sbb:

| Nrp      | nama         | Hobi                               |
|----------|--------------|------------------------------------|
| 12020001 | Heri Susanto | Sepakbola, membaca komik, berenang |
| 12020013 | Siti Zulaiha | Memasak, mrogram komputer          |
| 12020015 | Dini Susanti | Menjahit, membuat roti             |

Atau:

| Nrp      | nama         | hobi1      | hobi2            | Hobi3    |
|----------|--------------|------------|------------------|----------|
| 12020001 | Heri Susanto | Sepak Bola | Membaca komik    | berenang |
| 12020013 | Siti Zulaiha | Memasak    | mrogram komputer |          |
| 12020015 | Dini Susanti | Menjahit   | membuat kue      |          |

Tabel-tabel di atas tidak memenuhi syarat 1NF

# Contoh 1

Didekomposisi menjadi:

- Tabel Mahasiswa

| <u>Nrp</u> | <u>Nama</u>  |
|------------|--------------|
| 12020001   | Heri Susanto |
| 12020013   | Siti Zulaiha |
| 12020015   | Dini Susanti |

- Tabel Hobi

| <u>Nrp</u> | <u>Hobi</u>      |
|------------|------------------|
| 12020001   | Sepakbola        |
| 12020001   | membaca komik    |
| 12020001   | Berenang         |
| 12020013   | Memasak          |
| 12020013   | mrogram komputer |
| 12020015   | Menjahit         |
| 12020015   | membuat roti     |

# Contoh 2 (composite)

Jadwalkuliah

| Kodekul | NamaKul | Dosen | Kelas | Jadwal |
|---------|---------|-------|-------|--------|
|---------|---------|-------|-------|--------|

- Dimana nilai pada atribut jadwal berisi gabungan antara Hari dan Jam.
- Jika asumsi hari dan jam memegang peranan penting dalam sistem basis data, maka atribut Jadwal perlu dipisah sehingga menjadi JadwalHari dan JadwalJam sbb:

Jadwalkuliah

| Kodekul | NamaKul | Dosen | Kelas | JadwalHari | JadwalJam |
|---------|---------|-------|-------|------------|-----------|
|---------|---------|-------|-------|------------|-----------|



## 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

# Contoh

Tabel berikut memenuhi 1NF tapi tidak termasuk 2NF:

|         |          |            |         |         |        |         |
|---------|----------|------------|---------|---------|--------|---------|
| Mhs_nrp | mhs_nama | mhs_alamat | mk_kode | mk_nama | mk_sks | nihuruf |
|---------|----------|------------|---------|---------|--------|---------|

- Tidak memenuhi 2NF, karena {Mhs\_nrp, mk\_kode} yang dianggap sebagai primary key sedangkan:

{Mhs\_nrp, mk\_kode}  $\rightarrow$  mhs\_nama

{Mhs\_nrp, mk\_kode}  $\rightarrow$  mhs\_alamat

{Mhs\_nrp, mk\_kode}  $\rightarrow$  mk\_nama

{Mhs\_nrp, mk\_kode}  $\rightarrow$  mk\_sks

{Mhs\_nrp, mk\_kode}  $\rightarrow$  nihuruf

- Tabel di atas perlu didekomposisi menjadi beberapa tabel yang memenuhi syarat 2NF

# Contoh

Functional dependencynya sbb:

|                    |   |                        |       |
|--------------------|---|------------------------|-------|
| {Mhs_nrp, mk_kode} | → | nihuruf                | (fd1) |
| Mhs_nrp            | → | {mhs_nama, mhs_alamat} | (fd2) |
| Mk_kode            | → | {mk_nama, mk_sks}      | (fd3) |

|     |  |                    |
|-----|--|--------------------|
| fd1 | ( <u>mhs_nrp</u> , <u>mk_kode</u> , nihuruf) | → Tabel Nilai      |
| fd2 | ( <u>Mhs_nrp</u> , mhs_nama, mhs_alamat)     | ➔ Tabel Mahasiswa  |
| fd3 | ( <u>mk_kode</u> , mk_nama, mk_sks)          | ➔ Tabel MataKuliah |

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

- 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.
- Untuk setiap Functional Dependency dengan notasi  $X \rightarrow A$ , maka:
  - X harus menjadi superkey pada tabel tsb.
  - Atau A merupakan bagian dari primary key pada tabel tsb.

# Contoh

Tabel berikut memenuhi 2NF, tapi tidak memenuhi 3NF:

## Mahasiswa

| Nrp | Nama | Alm_Jalan | Alm_Kota | Alm_Provinsi | Alm_Kodepos |
|-----|------|-----------|----------|--------------|-------------|
|-----|------|-----------|----------|--------------|-------------|

- karena masih terdapat atribut *non primary key* (yakni **alm\_kota** dan **alm\_Provinsi**) yang memiliki ketergantungan terhadap atribut *non primary key* yang lain (yakni **alm\_kodepos**):

$\text{alm\_kodepos} \rightarrow \{\text{alm\_Provinsi}, \text{alm\_kota}\}$

- Sehingga tabel tersebut perlu didekomposisi menjadi:

Mahasiswa (Nrp, nama, alm\_jalan, alm\_kodepos)

Kodepos (alm\_kodepos, alm\_provinsi, alm\_kota)

# Boyce-Code Normal Form (BCNF)

- Bentuk BCNF terpenuhi dalam sebuah tabel, jika untuk setiap *functional dependency* terhadap setiap atribut atau gabungan atribut dalam bentuk:  $X \rightarrow Y$  maka **X** adalah **super key**
- tabel tersebut harus di-dekomposisi berdasarkan *functional dependency* yang ada, sehingga X menjadi *super key* dari tabel-tabel hasil dekomposisi
- Setiap tabel dalam BCNF merupakan 3NF. Akan tetapi setiap 3NF belum tentu termasuk BCNF .  
Perbedaannya, untuk functional dependency  $X \rightarrow A$ , BCNF tidak membolehkan A sebagai bagian dari primary key.

# Bentuk Normal Tahap Keempat (4th Normal Form /4NF)

- 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)

## Bentuk Normal Tahap Keempat (5th Normal Form /5NF)

- Bentuk normal 5NF terpenuhi jika tidak dapat memiliki sebuah *lossless decomposition* menjadi tabel-tabel yg 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