# MODUL 3

# NORMALISASI

## Tujuan Instruksional Umum

Mahasiswa mampu membangun desain logika basis data relasional untuk menghasilkan struktur tabel yang normal

## Tujuan Instruksional Khusus

1. Mahasiswa mampu mengidentifikasi tabel bentuk normal 1, bentuk normal 2 dan bentuk normal 3.
2. Mahasiswa mampu melakukan proses normalisasi

# MATERI PRAKTIKUM

## NORMALISASI

**Normalisasi** merupakan Teknik/ pendekatan yang digunakan dalam membangun design logic database relasional melalui organisasi himpunan data dengan tingkat ketergantungan fungsional dan keterkaitan yang tinggi sedemikian sehingga menghasilkan struktur tabel yang normal.

**Tujuan Normalisasi adalah:**

* 1. Minimalisasi redundansi (pengulangan data)
  2. Memudahkan identifikasi entitas
  3. Mencegah terjadinya anomali

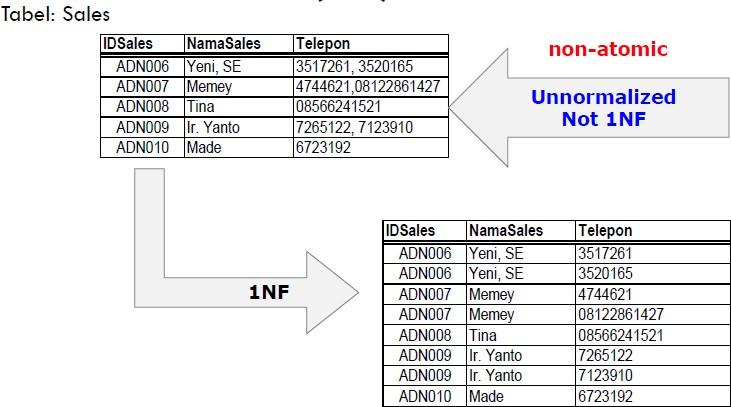
## Beberapa bentuk normal (normal forms, NF) :

* 1NF, 2NF, 3NF, BCNF: based on keys and functional dependencies
* 4NF, 5NF: based on keys and multivalued dependencies

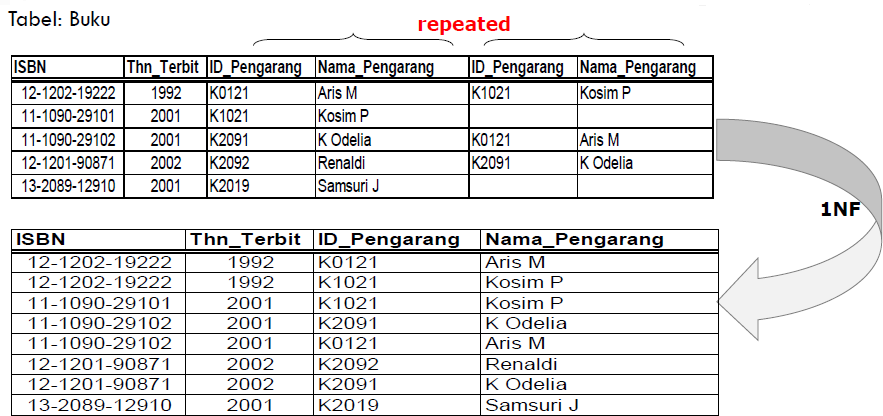
## FIRST NORMAL FORM (1NF)

Suatu relasi disebut memenuhi bentuk normal pertama (1NF) jika dan hanya jika setiap atribut dari relasi tersebut hanya memiliki nilai tunggal dan tidak ada pengulangan grup atribut dalam baris.

Contoh 1:



Contoh 2:



## SECOND NORMAL FORM (2NF)

Suatu relasi disebut memenuhi bentuk normal kedua (2NF) jika dan hanya jika:

1. Memenuhi 1NF
2. setiap atribut yang bukan kunci utama (primary key) tergantung secara fungsional terhadap semua atribut kunci dan bukan hanya sebagian atribut kunci (fully functionally dependent).

Untuk normalisasi ke bentuk 2NF, maka tabel 1NF didekomposisi menjadi beberapa tabel yang masing-masing memenuhi 2NF.

Contoh 1:

Diketahui tabel R = (A,B,C,D,E) ; A,B kunci utama (primary key) Dengan FD : A,B 🡪 C,D,E

Maka tabel R memenuhi 2NF sebab:

A,B 🡪 C,D,E berarti:

A,B 🡪 C,

A,B 🡪 D dan A,B 🡪 E

Jadi semua atribut bukan kunci utama tergantung penuh pada (A,B).

Contoh 2:

Bagaimana bila R = (A,B,C,D,E) tetapi dengan FD : (A,B) 🡪 (C,D) dan B 🡪 E. Apakah memenuhi 2NF ?

Jelas bahwa R bukan 2NF karena ada atribut E yang bergantung hanya pada atribut B saja dan bukan terhadap (A,B).

Dari FD : (A,B) 🡪 (C,D) juga mencerminkan bahwa hanya C dan D saja yang bergantung secara fungsional terhadap (A,B), tidak untuk E.

Jadi bukan 2NF.

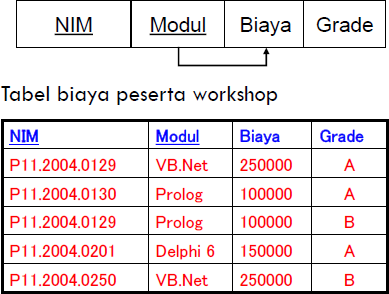
Untuk mengubah menjadi 2NF, lakukan dekomposisi menjadi:

R1= (A,B,C,D) dan R2= (B,E).

R1dan R2 memenuhi 2NF.

Contoh 3:

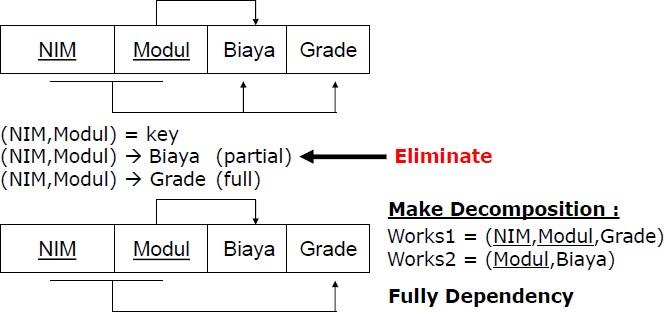
Diketahui suatu entitas Workshop = (NIM, Modul, Biaya, Grade)

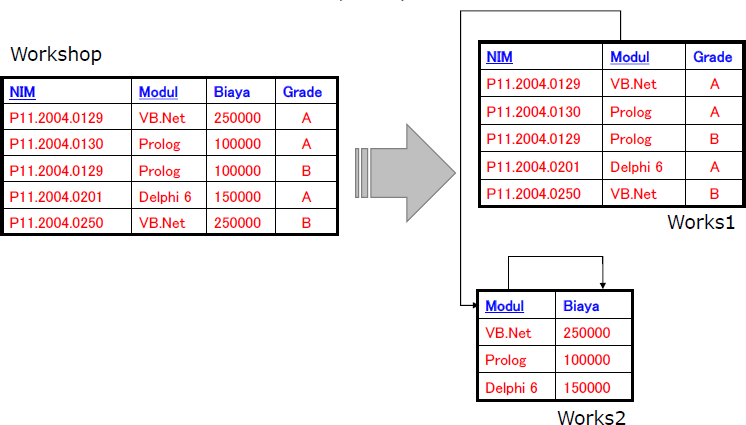
Key : (NIM, Modul)

FD : Modul🡪Biaya

(Biaya ditentukan oleh Modul yang diambil mahasiswa) : sudah memenuih bentuk 1NF tetapi tidak 2NF

Sebab dalam tabel ini, biaya tidak bergantung penuh pada atribut kunci (NIM, Modul)





## THIRD NORMAL FORM (3NF)

Suatu relasi disebut memenuhi bentuk normal ketiga (3NF) jika dan hanya jika:

1. memenuhi 2NF

Key (tidak terdapat ketergantungan transitif pada atribut bukan kunci).

Atau

Suatu relasi disebut memenuhi bentuk normal ketiga (3NF) jika dan hanya jika setiap FD nontrivial : X 🡪A, di mana X dan A atribut (atau kompositnya), memenuhi salah satu kondisi:

1. X adalah superkey
2. A merupakan anggota candidate key (A disebut prime attribute)

Jika suatu relasi sudah memenuhi 2NF tapi tidak memenuhi 3NF, maka untuk normalisasi ke bentuk 3NF, tabel 2NF didekomposisi menjadi beberapa tabel hingga masing-masing memenuhi 3NF.

Catatan:

*Jika suatu relasi memenuhi 2NF dan hanya memiliki tepat satu atribut yang bukan kunci utama maka relasi tersebut sudah memenuhi 3NF*

Contoh:

Diketahui tabel R = (A,B,C,D,E) ; A, B kunci utama(primary key) dengan FD : A,B 🡪C,D,E dan C 🡪 D,E maka R bukan 3NF sebab: Atribut D dan E (bukan kunci utama) bergantung secara fungsional pada C (yang juga bukan kunci utama).

Melalui FD :

Diketahui A,B 🡪 C,D,E.

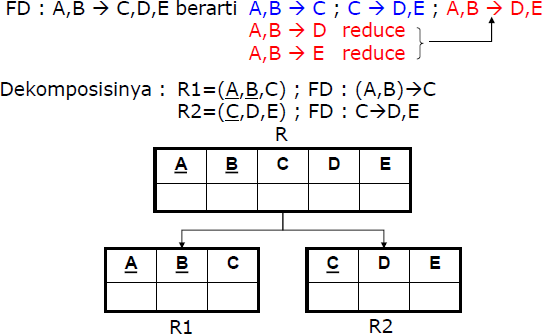
Karena sifat refleksif maka A,B 🡪 A,B. Sehingga A,B 🡪 A,B,C,D,E (A,B) : Superkey.

Diketahui C 🡪 D,E.

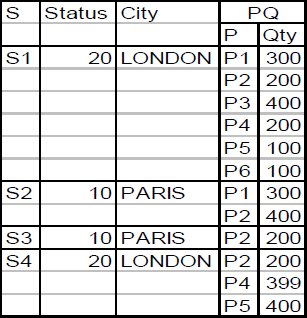
Karena sifat refleksif maka C 🡪 C. Sehingga C 🡪C,D,E. Karena C 🡪 A,B,C,D,E maka C bukan superkey.

Jadi: Tidak memenuhi definisi 3NF. Jadi R bukan 3NF.

Agar R memenuhi 3NF maka didekomposisi menjadi: R1=(A,B,C) dan R2=(C,D,E) sehingga R1 dan R2 memenuhi 3NF.



Contoh:

Misal diketahui struktur informasi dari suatu dokumen supplier :

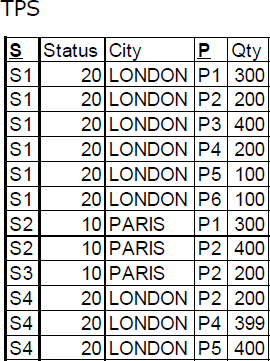
Akan dibentuk suatu tabel dengan skema TPS =( S, Status ,City, P, Qty)

Dengan (S,P) = primary key Dan berlaku FD :

S 🡪 Status S 🡪 City

City 🡪 Status

Lakukan normalisasi dari 1NF hingga3NF.

JAWAB:

* 1NF
* Not 2NF Problem :

Redundansi 🡪 inconsistency (low speed process)

* Anomaly :

S 🡪 (Status, City) tapi tidak bisa dilakukan insert data (S5, 30,JAKARTA) tanpa diikuti data P (khususnya) dan Qty.

Menghapus 1 baris data akan merusak keutuhan informasi.

Solusi:

Dekomposisi menjadi: TPS1 dan TPS2

