



VERSI 2.0
13 FEBRUARY, 2023

BASIS DATA

MODUL 2 MATERI : PEMODELAN DATA RELASIONAL

TIM PENYUSUN :

BRIANSYAH SETIO WIYONO, S. KOM

CLARISSA SANINDITA REIKISYIFA

CHINTYA TRIA DIANA OKTAVIANI

PRESENTED BY: LAB. INFORMATIKA
UNIVERSITAS MUHAMMADIYAH MALANG

BASIS DATA

PERSIAPAN MATERI

Praktikan mempelajari konsep pemodelan data relasional sesuai dengan pedoman oracle.

TUJUAN

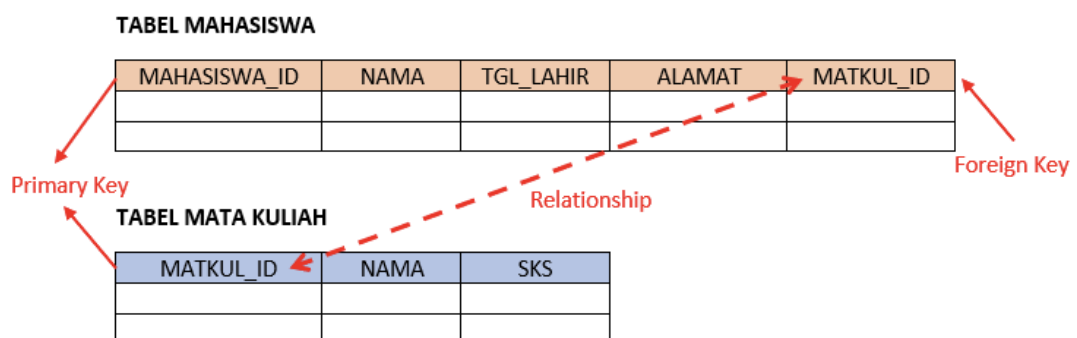
1. Praktikan memahami konsep pemodelan data relasional.
2. Praktikan mampu mengimplementasikan rancangan entitas dan keterkaitan antar entitas.
3. Praktikan mampu melakukan normalisasi

TARGET MODUL

Praktikan dapat mengimplementasikan rancangan entitas dan keterkaitan antar entitas pada skema basis data masing-masing.

MATERI POKOK**1. BASIS DATA RELASIONAL**

Sebuah database relasional menyajikan informasi dalam tabel yang terdiri atas baris dan kolom. Setiap kolom mewakili jenis informasi tertentu, dan setiap baris mencantumkan satu nilai. Tiap baris pada tabel dapat ditandai dengan pengidentifikasi unik yang disebut kunci utama (Primary Key). Tabel kemudian dihubungkan satu sama lain dengan menggunakan kunci asing (Foreign Key). Data ini dapat diakses langsung tanpa harus menyusun ulang menjadi tabel baru dalam database. Contoh :



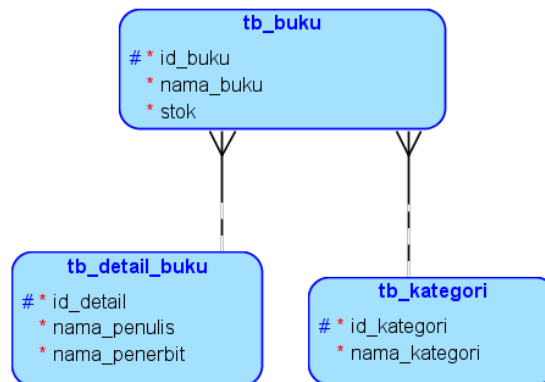
Primary key (Kunci Utama) = Pengidentifikasi unik untuk setiap baris

Foreign Key (Kunci Asing) = kolom yang merujuk ke kolom kunci utama di tabel lain

2. CDM & PDM**a) CDM (Conceptual Data Model)**

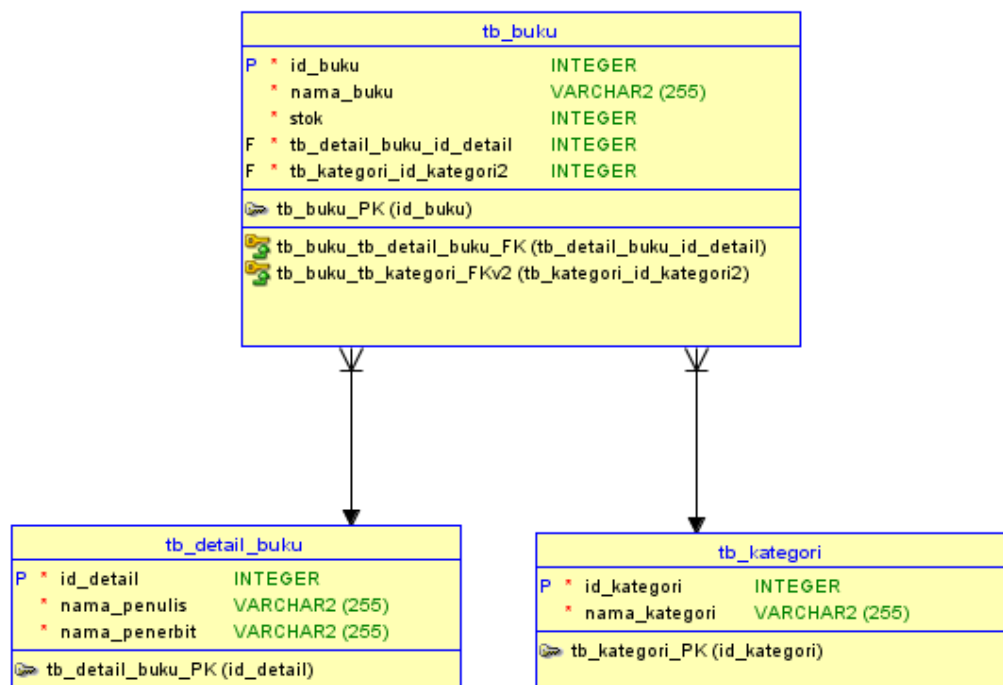
CDM dibuat sudah dalam bentuk tabel-tabel tanpa tipe data yang menggambarkan bagaimana hubungan antar tabel (relasi) untuk keperluan implementasi ke basis data.

CDM yang valid dapat dikonversi ke PDM. Contoh :



b) PDM (Physical Data Model)

PDM adalah konsep yang lebih detail dimana mendefinisikan sejumlah tabel, data, tipe data, serta hubungan antara tabel lain. PDM sudah berbentuk fisik perancangan basis data yang sudah siap diimplementasikan ke dalam DBMS sehingga menampilkan semua struktur tabel, termasuk kolom, primary key, dan foreign key. Contoh :



3. ENTITAS & ATRIBUT

a) Entitas

Entitas merupakan objek yang akan menjadi fokus utama dalam pembuatan sebuah database. Secara umum entitas sama dengan tabel yang berisi atribut-atribut penyusun yang saling berkaitan satu sama lain. Entitas dapat berupa benda, orang, pekerjaan, dan sebagainya. Penggunaan entitas umumnya berdasarkan dari tujuan pembuatan skema database yang akan dibangun.

Tipe-tipe entitas:

Nama	Deskripsi	Contoh
Prime	Berdiri sendiri	Dosen, Mahasiswa
Characteristic	Dibentuk karena entitas lain (prime)	Matakuliah
Intersection	Dibentuk karena dua atau lebih entitas	KRS, Transkrip

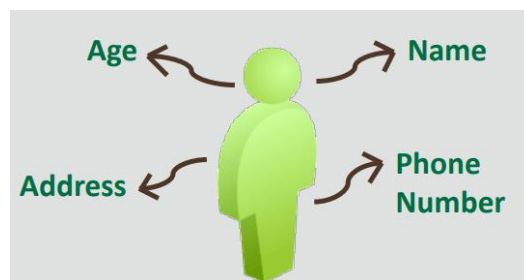
Poin-poin penting mengenai entitas:

- Setiap entitas pasti mengandung value atau nilai
- Value dari setiap entitas harus menggambarkan entitas itu sendiri
- Entitas merepresentasikan sekumpulan nilai yang merupakan kepentingan tertentu.

b) Atribut

Atribut merupakan sekumpulan informasi yang erat kaitannya dengan entitas. Atribut dan entitas merupakan satu kesatuan yang tidak dapat dipisahkan. Atribut biasa dikenal sebagai properti dari suatu entitas atau objek. Pada prinsipnya, atribut digunakan untuk mendeskripsikan sebuah entitas yang mana bersifat spesifik untuk mudah dipahami. Atribut adalah value tunggal dari detail properti dari entitas.

Contoh: entitas manusia



Entitas manusia memiliki beberapa atribut yang merepresentasikan secara spesifik dari entitas manusia yaitu nama, umur, alamat, dan nomor telepon.

Karakteristik Atribut:

- Atribut ditampilkan dalam kotak entitas pada ERD
- Nama atribut adalah tunggal dan tertulis dengan huruf kecil. Contoh: no_hp
- Biasanya, nama atribut tidak boleh mengandung nama entitas karena atribut memenuhi syarat dari nama entitas.
- Atribut diklasifikasikan berdasarkan status:
 - Mandatory (tidak boleh bernilai kosong/ null), dilambangkan dengan *
 - Optional (boleh bernilai kosong/ nullable), dilambangkan dengan o

Atribut Volatil dan Non-volatil (*Volatile & Non-volatile*)

- Atribut volatil/ *volatile* adalah atribut yang tidak stabil. Contoh: umur
- Atribut non-volatil/ *non-volatile* adalah atribut yang stabil. Contoh: tanggal lahir

Atribut Tunggal dan Gabungan (*Single & Composite*)

- Atribut tunggal tidak memiliki dan/ atau tidak bisa memiliki bagian
- Atribut gabungan adalah atribut yang bisa memiliki bagian-bagian kecil yang saling merepresentasikan atribut dasar dan memiliki arti sendiri-sendiri.

Atribut Nilai Tunggal dan Nilai Ganda (*Single-value & Multi-value*)

- Atribut nilai tunggal adalah atribut yang bisa memiliki nilai tunggal pada waktu tertentu. Contoh: nama belakang mahasiswa
- Atribut nilai ganda adalah atribut yang dapat memiliki lebih dari satu nilai dalam satu waktu. Contoh: alamat

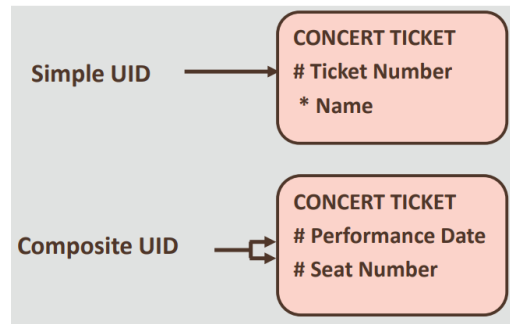
4. UNIQUE IDENTIFIERS

UID atau Unique Identifiers merupakan atribut istimewa atau kumpulan atribut yang secara unik mengidentifikasi value tertentu dari sebuah entitas. Setiap entitas harus memiliki UID, jika tidak maka tidak dapat disebut entitas. Unique identifiers adalah atribut dari entitas yang telah memenuhi beberapa aturan, yaitu:

- a) Identifier ini berada pada setiap value dari entitas
- b) Tidak terdapat nilai kosong/ null pada setiap value dari entitas untuk selamanya
- c) Memiliki value yang tidak pernah berubah untuk selamanya.

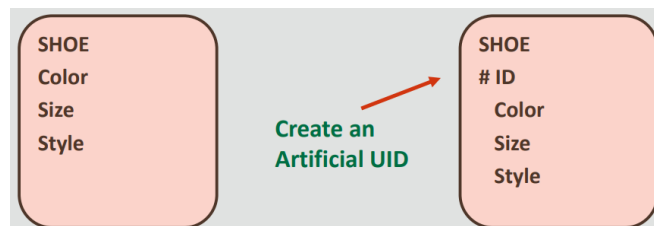
UID Sederhana & UID Gabungan (Simple UUIDs & Composite UUIDs)

- UID dengan atribut tunggal disebut dengan UID sederhana
- Terkadang atribut tunggal tidak cukup unik untuk mengidentifikasi value dari sebuah entitas
- Jika terdapat kombinasi UID maka dapat disebut juga dengan UID gabungan



Unique Identifier Buatan (Artificial Unique Identifier)

- UID buatan tercipta dari data yang ditugaskan atau dihasilkan oleh sistem
- UID buatan dibuat untuk identifikasi maksud dari sebuah sistem



5. RELASI

Relasi adalah hubungan dua arah yang signifikan antara dua entitas atau dirinya sendiri.

Komponen pada relasi meliputi :

a) Nama

Label yang muncul di dekat entitas yang ditugaskan padanya. Pastikan semua nama relasi menggunakan huruf kecil.

b) Kardinalitas

Kardinalitas menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain. Kardinalitas relasi merujuk kepada hubungan maksimum yang terjadi dari himpunan entitas yang satu ke himpunan entitas yang lain dan begitu juga sebaliknya. Derajat kardinalitas dibagi menjadi 3 bagian :

- **One-to-One**

Relasi ini menggunakan notasi 1:1. Kardinalitas one to one terjadi jika satu entitas X hanya berelasi dengan satu entitas Y ataupun sebaliknya.

Contohnya: Relasi antara tabel siswa dan tabel orang tua yang mana satu baris siswa hanya berhubungan dengan satu baris orang tua begitu juga sebaliknya.

- **One-to-Many atau Many-to-One**

Relasi ini menggunakan notasi 1:N. Kardinalitas ini terjadi jika satu entitas X berelasi dengan banyak entitas Y, ataupun sebaliknya. Dalam data modeller disimbolkan dengan one / single toe di satu ujung dan many / crow's feet di ujung lainnya.

Contohnya: Relasi antara tabel guru dan tabel siswa. Yang dimana satu baris guru atau satu guru bisa berhubungan dengan satu baris atau lebih siswa.

- **Many-to-Many**

Relasi ini menggunakan notasi M:N. Kardinalitas Many to Many terjadi jika banyak entitas X berelasi dengan banyak entitas Y, ataupun sebaliknya. Dalam data modeller disimbolkan dengan many / crow's feet di kedua ujung.

Contohnya: Relasi antar tabel siswa dan tabel mata kuliah. Dimana satu baris siswa bisa berhubungan dengan banyak baris mata pelajaran begitu juga sebaliknya.

c) Opsionalitas

Opsional

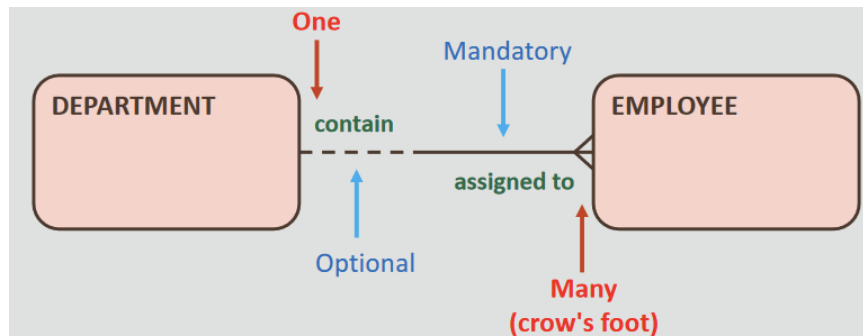
Opsional dapat berupa nol (tidak ada) catatan yang sesuai. Dalam SQL Data Modeller disimbolkan dengan garis putus-putus.

Contoh : Setiap jurusan tidak harus memiliki Mahasiswa.

Mandatory

Mandatory merupakan sejumlah atribut yang ada pada suatu table yang harus berisi data dan tidak boleh kosong (wajib), setidaknya satu data yang cocok di setiap entitas. Dalam SQL Data Modeller disimbolkan dengan garis solid.

Contoh : Setiap Mahasiswa harus memilih satu jurusan.



6. ENTITY RELATIONSHIP MODELING (ERDs)

ERD adalah salah satu jenis diagram yang sifatnya lebih struktural dan bisa digunakan untuk dimanfaatkan dalam suatu desain pada suatu database ataupun pada sebuah *business plan*. ERD pada database mengidentifikasi konsep atau entitas yang ada dalam suatu sistem dan hubungan antar entitas yang berbeda.

Berikut merupakan fungsi umum dari *Entity Relationship Diagram* :

- Membantu menganalisis suatu database dengan cara yang lebih cepat dan juga lebih mudah.
- Mampu menjalankan relasi antar setiap data yang mempunyai keterkaitan dengan berdasarkan objek yang dihubungkan dengan suatu relasi khusus.
- Membantu menjalankan dokumentasi data yang terdapat dalam suatu database dengan cara melakukan analisis dan identifikasi pada setiap objek ataupun entitas serta relasinya.
- Melakukan suatu pengujian model yang sebelumnya sudah dibuat.

Cara Membuat *Entity Relationship Diagram* (ERD) :

1. Melakukan identifikasi seluruh entitas yang akan digunakan
2. Melakukan identifikasi ulang dan menjelaskan hubungan antar entitas (relasi)
3. Menambahkan atribut kunci pada setiap entitas
4. Mengisi diagram ERD secara keseluruhan

7. NORMALISASI

Normalisasi merupakan sebuah pola dalam logical desain sebuah basis data. Normalisasi adalah konsep dari relasional basis data. Normalisasi juga bentuk dari proses pembuatan suatu tabel atau entitas dalam basis data dengan tujuan untuk mengurangi kemubaziran.

Proses normalisasi adalah proses penggolongan data elemen menjadi tabel-tabel yang menunjukkan entitas beserta relasinya.

Mengapa Harus Dilakukan Normalisasi?

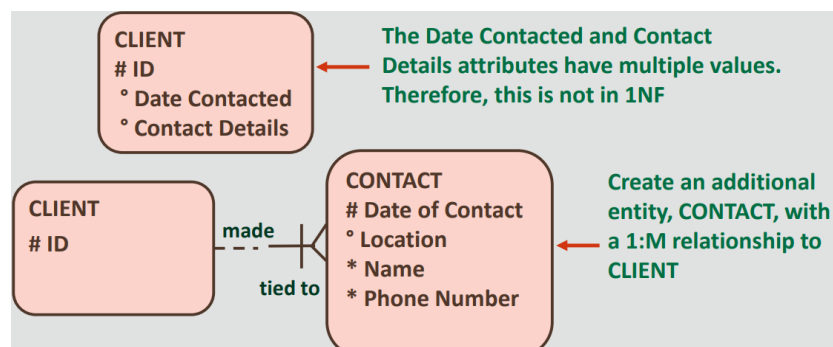
- Mengurangi data redundansi dari desain yang sudah ada
- Meningkatkan integritas data dan stabilitas desain
- Mengurangi tipe data yang tidak konsisten dan anomali
- Mengidentifikasi tabel dan kolom yang hilang serta kendala

Bentuk-bentuk aturan normalisasi:

Aturan	Deskripsi
First Normal Form (1NF)	Semua atribut harus ber-value tunggal (single-value)
Second Normal Form (2NF)	Atribut harus bergantung pada seluruh UID dari entitas
Third Normal Form (3NF)	Tidak ada atribut non-UID yang dapat bergantung pada atribut non-UID lainnya

a) 1NF

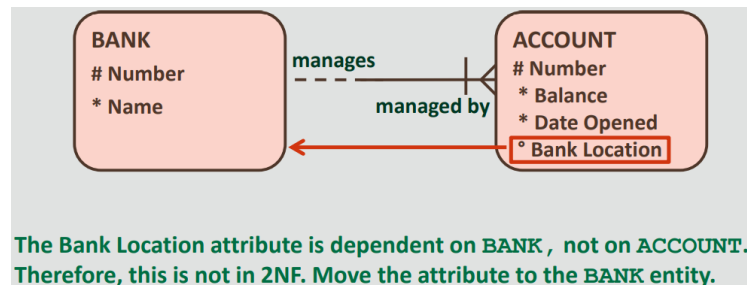
- Pada aturan ini tidak ada atribut dengan nilai ganda (multi-value)
- Untuk mengetahui 1NF, validasi setiap atribut yang bernilai tunggal (single-value) kepada setiap value pada entitas
- Jika terdapat atribut nilai ganda (multi-value), buat entitas tambahan yang menghubungkannya dengan entitas induk dengan relasi 1: M



b) 2NF

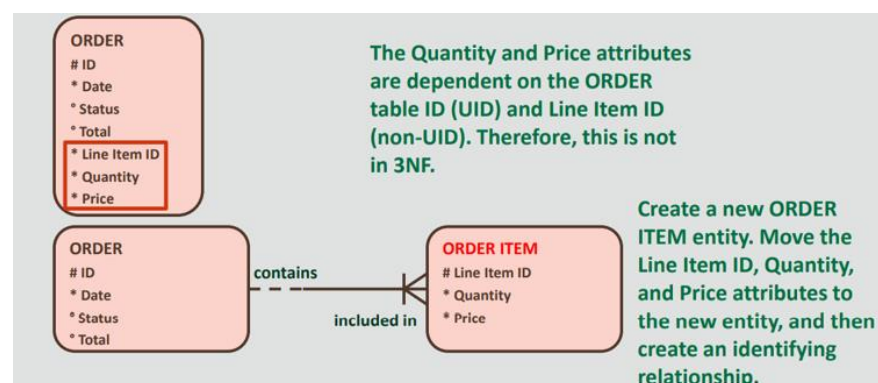
- Aturan ini memerlukan atribut non-UID yang bergantung pada seluruh UID
- Jika UID-nya merupakan composite UID, maka setiap atribut harus bergantung pada semua bagian dari composite UID

- Jika atribut tidak bergantung pada seluruh UID, buat entitas tambahan dengan UID parsial



c) 3NF

- Aturan ini menyatakan bahwa tidak ada atribut non-UID yang dapat bergantung kepada atribut non-UID lainnya
- 3NF melarang ketergantungan transitif
- Ketergantungan transitif dapat dilakukan ketika suatu atribut pada sebuah entitas bergantung pada atribut non-UID lainnya di entitas tersebut.
- Anda perlu memindahkan suatu atribut non-UID yang bergantung pada atribut non-UID lainnya dalam entitas baru



CONTOH NORMALISASI

Di bawah ini terdapat gambar nota pembelian suatu toko. Toko tersebut menjual banyak jenis barang dan juga memiliki banyak pembeli. Hal yang perlu dilakukan adalah menyimpan data nota dalam bentuk database.

FAKTUR PEMBELIAN BARANG				
PT REVANDA JAYA Jl. Bekasi Timur No. 2 Bekasi Timur				
Kode Supplier : G01			Tanggal : 07/02/2001	
Nama Supplier : Gobel Nustra			Nomor : 998	
Kode	Nama Barang	Qty	Harga	Jumlah
A01	AC Split ½ PK	10	1.350.000	13.500.000
A02	AC Split 1 PK	10	2.000.000	20.000.000
			Total Faktur	33.500.000
Jatuh Tempo Faktur : 09/03/2001				

FAKTUR PEMBELIAN BARANG				
PT REVANDA JAYA Jl. Bekasi Timur No. 2 Bekasi Timur				
Kode Supplier : S02			Tanggal : 02/02/2001	
Nama Supplier : Hitachi			Nomor : 779	
Kode	Nama Barang	Qty	Harga	Jumlah
R01	Rice Chocker C3	10	150.000	1.500.000
			Total Faktur	1.500.000
Jatuh Tempo Faktur : 09/03/2001				

1. Bentuk Tabel Belum Dinormalisasi (Unnormalized Form)

Gambar di bawah adalah data yang direkam dan dimasukkan secara mentah dalam suatu tabel. Pada bentuk ini sangat mungkin terjadi inkonsistensi dan anomali data. maka perlu dilakukan normalisasi database.

tb_faktur	
o	no_faktur
o	kode_supplier
o	nama_supplier
o	kode_barang
o	nama_barang
o	tanggal
o	jatuh_tempo
o	qty
o	harga
o	jumlah_harga
o	grand_total

2. 1NF

Bisa disebut 1NF jika dan hanya jika setiap atribut dari data tersebut hanya memiliki nilai tunggal dalam satu baris atau tidak boleh ada grouping ataupun duplikasi data.

Bentuk normal yang pertama atau 1NF, yang sudah dilakukan adalah :

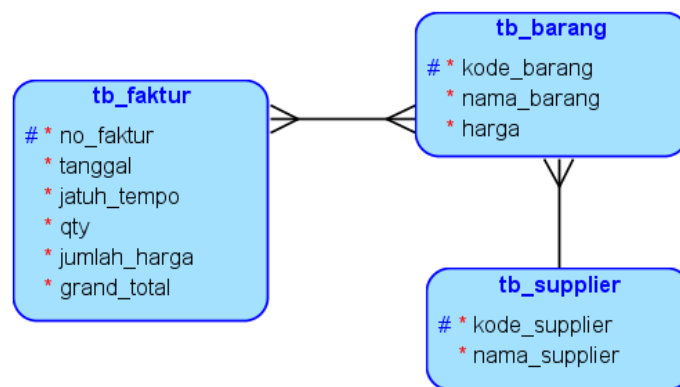
- Mengisi data pada kolom yang kosong dan menjadikannya nilai tunggal
- Namun, masih terdapat redundancy data pada kode Supp dan Nama Supp

pada nota ini tidak terdapat 1NF karena tidak ada multi value

3. 2NF

Intinya adalah pada tahap normalisasi 2NF ini tabel tersebut harus dipecah berdasarkan primary key. Jadi hal yang perlu dilakukan :

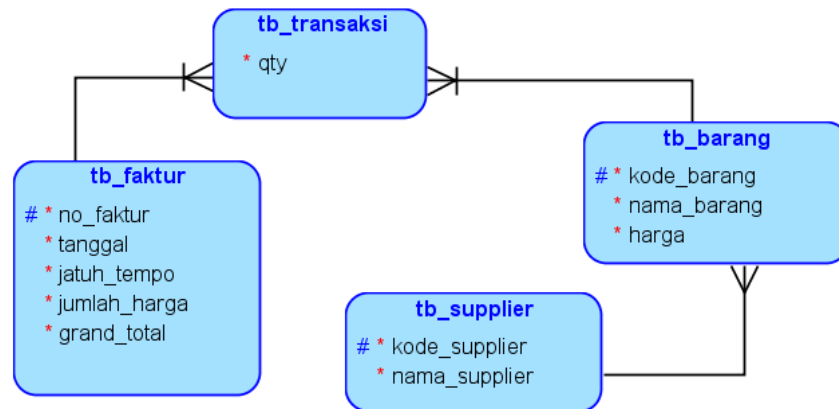
- Menghilangkan duplikasi kolom dari tabel yang sama.
- Buat tabel terpisah untuk masing-masing kelompok data terkait dan mengidentifikasi setiap baris dengan kolom yang unik (primary key).



4. 3NF

Pada 3NF jika terdapat suatu atribut yang tidak bergantung pada primary key tapi bergantung pada field yang lain maka atribut-atribut tersebut perlu dipisah ke tabel baru.

Contohnya pada atribut **qty**, kolom tersebut tidak bergantung langsung pada primary key **no_faktur** sehingga perlu ditentukan di entitas mana dia akan masuk. Jadi setelah di normalisasi 3NF akan menghasilkan tabel berikut:



TUGAS

1. Buatlah normalisasi konseptual database mulai dari 1NF hingga 3NF dari gambar nota dibawah ini

Kartu pengobatan masyarakat							
No Pasien : 1234/PO/IV/99				Tanggal Pendaftaran : 1 Mei 1999			
Data Pasien dari, NOPEN : 1000019999				Nama Pasien : Bachtiar Jose			
Alamat Pasien, Jalan : Kebon Jeruk No. 27				Kelurahan : Palmerah			
Kecamatan : Kemanggisan				Wilayah : Jakarta Barat			
Kode Pos : 11530				Telepon : 5350999			

Tgl periksa	Kode Dokter	Nama Dokter	KodeSakit	Diskripsi sakit	Kode obat	Nama obat	Dosis
1/5/99	D01	Dr Zurmaini	S11	Tropicana	B01 B02	Sulfa Anymiem	3dd1 4dd1
4/7/99	D01	Dr Zurmaini	S12	Ulcer Triombis	B01 B03	Sulfa Supralin	3dd2 3dd1
4/4/99	D02	Dr Harjono	S12	Ulcer Triombis	B04	Adrenalin	4dd2
7/8/99	D04	Dr Mahendra	S12	Ulcer Triombis	B01 B02 B03	Sulfa Anymiem Supralin	3dd2 4dd2 3dd1

Nb. Tugas modul 2 wajib dikumpulkan di ilab (berbobot 20%)

RUBRIK PENILAIAN PRAKTIKUM

Detail	Bobot
Pemahaman Materi	40%
Tugas	20%
Kegiatan 1	15%
Kegiatan 2	25%