

Dosen Pengampu: Adri Arisena



PRAKTIKUM 5

Normalisasi Basis Data

Asisten Laboratorium:
Najlia Intani



"Modul ini diperuntukkan bagi Mata Kuliah Pemrograman Dasar dan Basis Data Program Studi Agribisnis"

A. PENDAHULUAN

A. Pengertian

Normalisasi basis data adalah suatu proses untuk mengorganisir data dalam basis data relasional dengan tujuan menghilangkan redudansi (redundancy) dan menghindari anomali dalam pengolahan data. Normalisasi basis data bertujuan untuk memperbaiki struktur tabel dan hubungan antar tabel agar lebih efisien dan mudah dikelola. Normalisasi basis data dilakukan dengan membagi tabel besar menjadi beberapa tabel yang lebih kecil dan terkait erat sesuai dengan aturan normalisasi (Date, 2004), (Connolly & Begg, 2014).

Kriteria yang mendefinisikan level-level pada normalisasi adalah bentuk normal. Melalui normalisasi dapat membentuk struktur basis data sehingga sebagian besar ambiguity bisa dihilangkan.

B. Tujuan Normalisasi

Normalisasi perlu dilakukan agar hubungan dalam basis data menjadi mudah dimengerti, mudah dipelihara, mudah memprosesnya, dan mudah untuk dikembangkan sesuai kebutuhan baru. Selain itu, tujuan normalisasi adalah menghilangkan kerangkapan data, mengurangi kompleksitas, mempermudah pemodifikasian data.

C. Anomali

Masalah-masalah yang timbul dalam pembuatan tabel yang disebut dengan anomali. Anomali adalah proses pada basis data yang mempunyai efek samping yang tidak diharapkan. Misal : data tidak konsisten, suatu data hilang pada saat dihapus, dll. Anomali ada 3 jenis yaitu :

1) Anomali peremajaan

Anomali ini terjadi bila ada perubahan pada sejumlah data yang mubazir, tetapi tidak seluruhnya diubah.

Contoh: Tabel Mahasiswa

nama_mahasiswa	program_studi	jumlah_sks
Kartika Putri	Manajemen Informatika	5
Claudia Citra	Manajemen Informatika	2
Yudi Permadi	Manajemen Informatika	2
Claudia Citra	Manajemen Informatika	1

Seandainya Claudia Citra dengan program_studi manajemen Informatika pindah ke Komunikasi Massa maka pengubahan data hanya dilakukan pada data pertama menjadi:

Tabel Mahasiswa

nama_mahasiswa	program_studi	jumlah_sks
Kartika Putri	Manajemen Informatika	5
Claudia Citra	Komunikasi Massa	2

Yudi Permadi	Manajemen Informatika	2
Claudia Citra	Manajemen Informatika	1

Pada tabel tersebut terlihat bahwa data tentang nama_mahasiswa Claudia Citra tidak sama yang menyebabkan data tidak konsisten.

2) Anomali Penyisipan

Anomali ini terjadi pada saat penambahan data ternyata ada elemen yang kosong dan elemen tersebut justru menjadi key.

Contoh : Tabel Ekstrakurikuler

nim	ekstrakurikuler	biaya
B12110	Karate	30.000
B12112	Tenis	35.000
B12110	Tenis	35.000
B12111	Paduan Suara	20.000
B12110	Bulu Tangkis	20.000

Misalnya akan tambah ekstrakurikuler baru yaitu Pemrograman Android dengan biaya 50.000 akan tetapi belum ada seorangpun yang ikut ekstrakurikuler pemrograman android ini, sehingga data menjadi:

Tabel Ekstrakurikuler

nim	ekstrakurikuler	biaya
B12110	Karate	30.000
B12112	Tenis	35.000
B12110	Tenis	35.000
B12111	Paduan Suara	20.000
B12110	Bulu Tangkis	20.000
	Pemrograman Android	50.000

3) Anomali penghapusan

nomali ini terjadi apabila dalam satu baris/ tuple ada data yang akan dihapus sehingga akibatnya terdapat data lain yang hilang. Contoh pada tabel ekstrakurikuler data nim B12112 akan dihapus karena sudah tidak ikut ekstrakurikuler lagi sehingga akibatnya data ekstrakurikuler tenis dan biaya 35.0000 akan ikut terhapus.

D. Dependensi (Ketergantungan)

Konsep dasar pada tahap normalisasi yang menjelaskan hubungan atribut atau secara lebih khusus menjelaskan nilai suatu atribut yang menentukan atribut lainnya.

Macam-macam dependensi, yaitu :

1) Dependensi Fungsional

Definisi : Suatu atribut Y mempunyai dependensi fungsional terhadap atribut X jika dan hanya jika setiap nilai X berhubungan dengan sebuah nilai Y.

Notasi : X secara fungsional menentukan Y bisa dikatakan juga Y bergantung pada X

Contoh: Tabel Nilai

nim	nama	mata_kuliah	nilai
B12110	Andri Suryanto	Basis Data	A
C12007	Tiara Putri	Basis Data	A
B12112	Andi Nur Cahyo	Basis Data	B
B12110	Andri Suryanto	Algoritma Pemrograman	A
B12111	Ilham Aris Prasetyo	Basis Data	C
B12110	Andri Suryanto	Struktur Data	B
A12007	Tiara Putri	Algoritma Pemrograman	B

Dengan demikian : **nim** = **nama** **nim** secara fungsional menentukan **nama**, sebab setiap **nim** yang sama mempunyai **nama** yang sama.

{**nim**, **mata_kuliah**} = **nilai** **nim** dan **mata_kuliah** secara fungsional menentukan **nilai**, sebab setiap **nim** dan **mata_kuliah** yang sama mempunyai **nilai** yang sama.

Keterangan:

- Bagian yang terletak di sebelah kiri tanda panah biasa disebut determinan / penentu dan bagian yang terletak di sebelah kanan panah disebut dependensi / yang tergantung.
- Tanda {} digunakan untuk menentukan lebih dari satu atribut sebagai penentu atau sebagai yang tergantung.

2) Dependensi Penuh

Definisi: Suatu atribut Y mempunyai dependensi fungsional penuh terhadap X jika

- Y mempunyai dependensi fungsional terhadap X dan/atau
- Y tidak memiliki dependensi terhadap bagian (subset) dari X

Contoh : {**nim**, **mata_kuliah**} = **nilai** (Bukan dependensi penuh. Tapi jika **mata_kuliah** dihilangkan akan menjadi dependensi penuh) **mata_kuliah** = **nilai**.

3) Dependensi Parsial

Definisi : Dependensi Parsial merupakan ketergantungan fungsional dimana beberapa atribut dapat dihilangkan dari X dengan ketergantungan tetap dipertahankan

Contoh : {nim, nama, mata_kuliah} = nilai (dimana jika nama dihapus maka ketergantungan tetap ada)

4) Dependensi Transitif

Definisi : Dependensi transitif adalah kondisi dimana X,Y, Z merupakan atribut suatu relasi, dimana $X = Y$ dan $Y = Z$. Maka dikatakan Z dependensi transitif terhadap X melalui Y

Contoh : Tabel mahasiswa

Nim	nama	mata_kuliah	dosen
B12110	Andri Suryanto	Basis Data	Marwoto
C12007	Tiara Putri	Basis Data	Marwoto
B12112	Andi Nur Cahyo	Basis Data	Marwoto
B12111	Ilham Aris Prasetyo	Basis Data	Marwoto

nim = mata_kuliah matakuliah = nim

E. Dekomposisi

Pada tahap normalisasi sering kali terjadi pemecahan tabel kedalam bentuk dua atau lebih relasi. Proses pemecahan ini disebut dengan dekomposisi. Syarat dekomposisi adalah : Tidak ada informasi yang hilang ketika suatu relasi dipecah menjadi relasi-relasi lain. Contoh : Terdapat suatu relasi awal sebagai berikut:

nim	Nama	hobi
B12110	Andri Suryanto	Memancing
C12007	Tiara Putri	Berkebun
B12112	Andi Nur Cahyo	Olahraga
B12112	Andi Nur Cahyo	Membaca

Akan dilakukan dekomposisi, dengan tidak menghilangkan informasi. Hasil dekomposisi sebagai berikut:

Nim	nama
B12110	Andri Suryanto
C12007	Tiara Putri
B12112	Andi Nur Cahyo

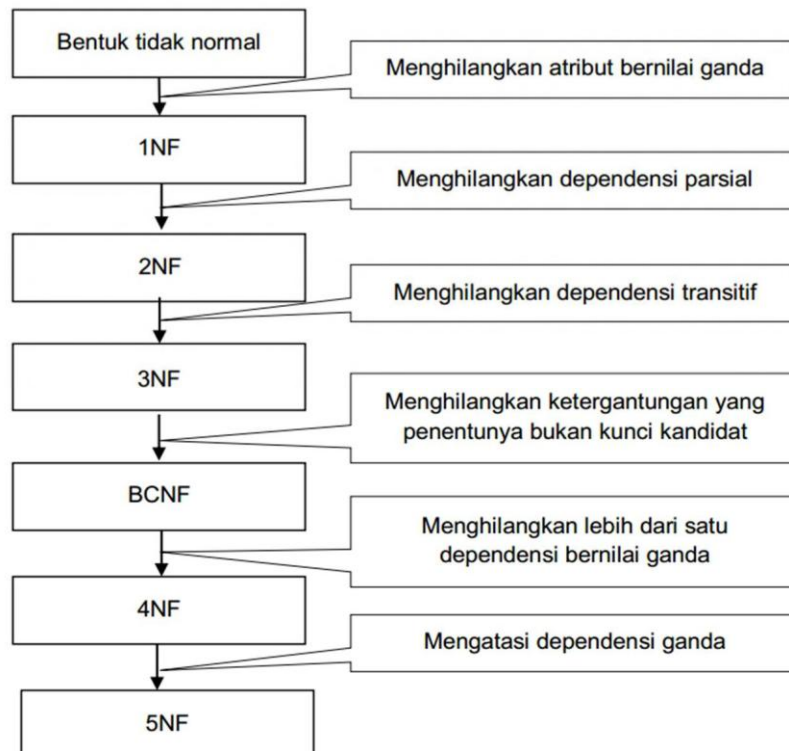
nim	hobi
B12110	Memancing
C12007	Berkebun
B12112	Olahraga
B12112	Membaca

B. BENTUK NORMALISASI

A. Kriteria normalisasi basis data:

1. Setiap atribut dalam tabel hanya memiliki satu nilai
Setiap atribut pada tabel harus memiliki nilai tunggal (singlevalue) atau atomik, artinya tidak boleh memiliki kumpulan nilai atau nilai terpisah yang harus dikelompokkan bersama. Hal ini membantu memastikan bahwa data dapat dikelola dan dimanipulasi dengan mudah dan akurat.
2. Setiap tabel harus memiliki sebuah primary key
Setiap tabel harus memiliki sebuah primary key yang unik dan tidak berubah-ubah. Primary key digunakan untuk mengidentifikasi setiap baris dalam tabel secara unik.
3. Tidak ada atribut yang tergantung pada atribut non-key
Tabel harus disusun sedemikian rupa sehingga tidak ada atribut yang bergantung pada atribut non-key. Hal ini dapat dicapai melalui normalisasi tabel hingga mencapai minimal bentuk normal.
4. Tidak ada ketergantungan siklik
Tidak ada ketergantungan siklik antar tabel. Artinya, tidak ada tabel yang saling tergantung satu sama lain sehingga tidak mungkin menentukan urutan tabel yang tepat saat mengakses data.

Dengan memenuhi kriteria di atas, normalisasi dapat membantu memperbaiki integritas data, meningkatkan efisiensi database, dan mengurangi redundansi data.



Normalisasi dilakukan melalui serangkaian level atau tingkatan, yaitu:

1. First Normal Form (1NF)

Pada level ini, setiap kolom dalam tabel hanya memiliki nilai atomik atau tidak dapat dibagi lagi. Setiap baris dalam tabel harus unik, dan tabel tersebut tidak boleh memiliki kolom-kolom yang dapat berubah-ubah.

2. Second Normal Form (2NF)

Pada level ini, setiap kolom dalam tabel harus bergantung sepenuhnya pada kunci utama tabel. Dalam kata lain, setiap nonkunci kolom harus bergantung pada kunci utama tabel, dan tidak boleh bergantung pada kolom non-kunci lainnya.

3. Third Normal Form (3NF)

Pada level ini, setiap kolom non-kunci dalam tabel harus bergantung pada kunci utama tabel. Selain itu, tidak boleh ada ketergantungan transitive antara kolom-kolom dalam tabel. Ketergantungan transitive terjadi ketika sebuah kolom bergantung pada kolom non-kunci lainnya.

4. Boyce-Codd Normal Form (BCNF)

Pada level ini, setiap ketergantungan fungsional dalam tabel harus bersifat trivial atau trivial. Dalam kata lain, kunci utama tabel harus dapat menentukan semua kolom non-kunci dalam tabel.

5. Fourth Normal Form (4NF)

Pada level ini, tabel harus bebas dari multi-nilai dependensi, yaitu ketergantungan fungsional yang melibatkan beberapa nilai dalam satu kolom.

6. Fifth Normal Form (5NF) atau Normal Form Domain/ Projection-Join.

Pada level ini, setiap ketergantungan fungsional antara kunci utama dan non-kunci harus dijaga di seluruh relasi database. Relasi ini mencakup semua informasi yang relevan dan tidak ada informasi yang redundan atau ambigu.

C. CONTOH

CONTOH 1:

Berikut ini adalah contoh normalisasi database struk penjualan.

kode_faktur	tanggal	kode_barang	nama_barang	harga	qty
KF001	10/03/2023	KB001	Rinso Matic	60000	1
		KB002	Pepsodent	25000	1
		KB003	Tissue Paseo	15000	1
		KB004	Sikat Gigi	10000	1
KF002	12/03/2023	KB005	Minya Bimoli	40000	1
		KB006	Indome Kari	2900	10
KF003	15/03/2023	KB007	Nugget Fiesta	50000	2

Contoh data di atas merupakan data yang belum dinormalisasi, selanjutnya menuju tahap normalisasi 1NF.

1. 1NF

Suatu tabel dikatakan 1NF jika dan hanya jika setiap atribut dari data tersebut hanya memiliki nilai tunggal dalam satu baris. Jadi, tabel yang belum dinormalisasi tadi perlu diubah, sehingga bentuk 1NF menjadi seperti ini:

kode_faktur	tanggal	kode_barang	nama_barang	harga	qty
KF001	10/03/2023	KB001	Rinso Matic	60000	1
KF001	10/03/2023	KB002	Pepsodent	25000	1
KF001	10/03/2023	KB003	Tissue Paseo	15000	1
KF001	10/03/2023	KB004	Sikat Gigi	10000	1
KF002	12/03/2023	KB005	Minya Bimoli	40000	1
KF002	12/03/2023	KB006	Indome Kari	2900	10
KF003	15/03/2023	KB007	Nugget Fiesta	50000	2

Inti dari normalisasi 1NF adalah tidak boleh ada grouping data ataupun duplikasi data. Sekarang lanjut pada tahap normalisasi 2NF.

2. 2NF

Syarat 2NF adalah tidak diperkenankan adanya partial “functional dependency” kepada primary key dalam sebuah tabel. Apa itu “functional dependency”? Functional dependency adalah setiap atribut yang bukan kunci (non key) bergantung secara fungsional terhadap primary key. Intinya adalah pada tahap normalisasi 2NF ini tabel tersebut harus dipecah berdasarkan primary key. Sehingga bentuk normalisasi 2NF dari tabel tersebut adalah sebagai berikut:

• Tabel Barang

kode_barang	nama_barang	harga
KB001	Rinso Matic	60000
KB002	Pepsodent	25000
KB003	Tissue Paseo	15000
KB004	Sikat Gigi	10000
KB005	Minya Bimoli	40000
KB006	Indome Kari	2900
KB007	Nugget Fiesta	50000

• Tabel Transaksi

kode_faktur	tanggal	kode_barang	qty
KF001	10/03/2023	KB001	1
KF001	10/03/2023	KB002	1
KF001	10/03/2023	KB003	1
KF001	10/03/2023	KB004	1
KF002	12/03/2023	KB005	1
KF002	12/03/2023	KB006	10
KF003	15/03/2023	KB007	2

3. 3NF

Pada 3NF tidak diperkenankan adanya partial “transitive dependency” dalam sebuah tabel. Apa itu “transitive dependency”? Transitive dependency biasanya terjadi pada tabel hasil relasi, atau kondisi dimana terdapat tiga atribut A, B, C. Kondisinya adalah $A \Rightarrow B$ dan $B \Rightarrow C$. Maka C dikatakan sebagai transitive dependency terhadap A melalui B. Intinya pada 3NF ini, 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 ada pada atribut qty, kolom tersebut tidak bergantung langsung pada primary key kode_faktur melainkan bergantung pada kolom kode_barang. Jadi setelah dinormalisasi 3NF akan menghasilkan tabel berikut:

- Tabel Barang

kode_barang	nama_barang	harga
KB001	Rinso Matic	60000
KB002	Pepsodent	25000
KB003	Tissue Paseo	15000
KB004	Sikat Gigi	10000
KB005	Minya Bimoli	40000
KB006	Indome Kari	2900
KB007	Nugget Fiesta	50000

- Tabel Transaksi

kode_faktur	tanggal
KF001	10/03/2023
KF002	12/03/2023
KF003	15/03/2023

- Tabel Detail Barang

kode_faktur	kode_barang	qty	harga
KF001	KB001	1	60000
KF001	KB002	1	25000
KF001	KB003	1	15000
KF001	KB004	1	10000
KF002	KB005	1	40000
KF002	KB006	10	2900
KF003	KB007	2	50000

Dari tabel di atas dapat dilihat pada tahap normalisasi 3NF menghasilkan 1 tabel baru dari hasil pemecahan tabel transaksi yaitu tabel detail barang yang isinya menampung barang-barang yang dibeli.

Kenapa di tabel detail barang terdapat kolom harga lagi? padahal kolom harga sudah ada di tabel barang. Karena kolom harga pada tabel detail barang digunakan untuk menyimpan harga barang pada saat proses transaksi. Jadi, meskipun kolom harga pada tabel barang berubah (naik/turun), harga barang yang ada pada tabel detail barang tidak ikut berubah (fixed). Bayangkan jika kita tidak menambahkan kolom harga pada tabel detail barang, maka yang terjadi total invoice dari transaksi akan berubah seiring berubahnya harga barang.

CONTOH 2:

Kita ambil contoh menggunakan data perfilman yang belum dinormalisasi atau Unnormalized Form (UNF) seperti ini:

Table director			
nik	name	origin	phone_no
123	Kimo	DKI Jakarta	08555
321	Timo	Papua	08333
999	Hanung	DKI Jakarta	08111,08222

1. 1NF

Kriteria untuk memenuhi 1NF adalah tidak ada Multivalue pada kolom. Pada table director, kolom phone_no dengan NIK 999 Multivalue karena ada lebih dari satu nomor telepon sehingga melanggar 1NF. Masalahnya adalah ribet saat data dengan NIK 999 mengganti salah satu nomornya. Misalkan dari 08222 ke 08221 maka kita harus update kolom tersebut dengan memasukkan salah satu nomor lama, lalu tambahkan nomor baru di belakangnya secara comma-separated. Untuk itu kita perlu normalisasi seperti berikut:

Table director			
nik	name	origin	phone_no
123	Kimo	DKI Jakarta	08555
321	Timo	Papua	08333
999	Hanung	DKI Jakarta	08111
999	Hanung	DKI Jakarta	08222

Sekarang ga ada lagi kolom yang multivalue karena datanya sudah dipisah. Tabel di atas sudah memenuhi kriteria 1NF. Jika sutradara dengan NIK 999 mau mengganti nomornya dari 08222 ke 08221 maka cukup update satu row saja yang menyimpan value 08222.

Key dalam Database:

1. Candidate Key

Candidate Key adalah satu atau kombinasi beberapa kolom unik yang menjadi penanda bahwa data tersebut merupakan satu kesatuan data yang mewakili kolom-kolom lainnya. Candidate Key dari tabel di atas ada 2, yaitu {nik} dan {phone_no}, karena nik adalah kolom unik yang mewakili kolom name & origin. Lewat nik kita bisa tahu nama dan asal sutradara karena satu NIK sudah pasti memiliki satu nama dan daerah asal.

Kolom phone_no juga kolom unik yang mewakili kolom nik. Lewat phone_no kita bisa tahu ini nomor siapa karena satu nomor sudah pasti memiliki satu NIK. Kolom name bukan Candidate Key karena bisa saja ada orang yang namanya sama. Begitu juga dengan origin. Candidate Key fungsinya sebagai Identifier sebuah data secara bisnis.

2. Primary Key

Primary Key adalah kolom unik yang menjadi pembeda antar row yang bisa dijadikan referensi dari data tersebut dan tidak boleh null. Primary Key fungsinya sebagai Identifier sebuah data secara teknis. Candidate Key boleh dijadikan Primary Key.

Tapi dalam hal ini Candidate Key tidak dianjurkan menjadi Primary Key karena bisa berubah nilainya. Misalkan pada tabel di atas kolom nik dijadikan Primary Key. Kemudian data tersebut berelasi dengan tabel lain. Lalu ternyata ada typo saat input NIK di awal dan harus diganti. Tentu tidak semudah itu mengganti Primary Key kalau datanya sudah berelasi.

Meskipun ada fitur cascades, tetap saja ga worth it mengubah beberapa data dan tabel. Oleh karena itu, kita akan pakai auto-increment value sebagai Primary Key pada kasus ini, sedangkan Candidate Key cukup diberi constraint Unique Key.

3. Unique Key

Unique Key adalah kolom yang nilainya unik. Bedanya dengan Primary Key adalah Unique Key membolehkan nilainya nullable maupun non-nullable.

4. Foreign Key

Foreign Key adalah kolom yang menjadi penghubung relasi antar tabel sebagai referensi data dari satu tabel ke tabel lainnya.

2. 2NF

Syarat untuk memenuhi kriteria 2NF adalah memenuhi syarat 1NF dan tidak ada Partial/Functional Dependency. Partial/Functional Dependency adalah ketika sebuah tabel memiliki lebih dari satu Candidate Key dan ada kolom yang bergantung pada salah satu Candidate Key saja.

Contohnya pada data di atas Candidate Key-nya {nik} dan {phone_no}. Kita bisa tahu nama dan daerah asal sutradara lewat NIK karena datanya diwakili NIK. Tapi kita tidak bisa tahu nama dan daerah asal sutradara lewat nomor telepon karena itu bukan bagian dari nomor telepon. Jadi kolom name dan origin hanya bergantung pada Candidate Key nik saja.

Masalahnya, akan terjadi pengulangan data pada kolom Non-Candidate Key jika sutradara tersebut memiliki nomor telepon lebih dari satu sehingga ribet dimaintain. Contohnya jika sutradara dengan NIK 999 yang memiliki lebih dari satu nomor telepon pada data di atas ganti nama atau ganti daerah asal, maka butuh update lebih dari satu row.

Oleh karena itu kolom-kolom yang memiliki Partial/Functional Dependency dipisah menjadi tabel sendiri agar memenuhi syarat 2NF seperti berikut:

Table director			
id (PK)	nik (UK)	name	origin
1	123	Kimo	DKI Jakarta
2	321	Timo	Papua
3	999	Hanung	DKI Jakarta

Table phone		
id (PK)	phone_no (UK)	director_id (FK)
1	08111	3
2	08222	3
3	08555	1
4	08333	2

Sekarang tidak ada lagi pengulangan data pada kolom Non-Candidate Key pada tabel di atas. Kita juga memisahkan Primary Key-nya menggunakan auto-increment, sedangkan Candidate Key menggunakan constraint Unique Key. Kedua tabel tersebut dihubungkan oleh kolom director_id di tabel phone. Itulah yang disebut Foreign Key seperti penjelasan sebelumnya. Sekarang kalau mau ganti nama sutradara cukup update satu row saja.

3. 3NF

Misalkan tabel director dibuat sebagai berikut:

Table director					
id	nik	name	origin_code	origin_province	origin_capital
1	123	Kimo	JKT	DKI Jakarta	Jakarta Pusat
2	321	Timo	PAP	Papua	Jayapura
3	999	Hanung	JKT	DKI Jakarta	Jakarta Pusat

Kriteria yang memenuhi 3NF adalah sudah memenuhi 2NF dan tidak ada Transitive Dependency. Transitive Dependency adalah ketika kolom A bergantung pada kolom B yang bukan Candidate Key, dan kolom B bergantung pada kolom C yang menjadi Candidate Key, tetapi kolom C tidak bergantung pada kolom B. Jadi, ada kolom yang bergantung pada kolom lain yang bukan Candidate Key. Biar ga pusing kita ambil contoh di atas.

Tabel di atas sudah memenuhi 2NF karena tidak ada Partial/Functional Dependency, Candidate Key-nya hanya nik karena cuma NIK yang unik. Namun itu tidak memenuhi 3NF. Kolom origin_province & origin_capital bergantung ke kolom origin_code yang bukan Candidate Key, dan kolom origin_code bergantung ke kolom nik. Jadi, dari NIK kita bisa tahu sutradara tersebut kode asalnya apa, dan dari kode tersebut kita bisa tahu itu provinsi apa dan ibu kotanya apa.

Tapi dari kode tersebut kita tidak bisa tahu itu NIK siapa karena satu daerah bisa terdapat lebih dari satu sutradara. Masalahnya, jika data pada kolom origin_province, misalkan sebelumnya dengan value "DKI Jakarta" diubah jadi "Jakarta", maka perlu update semua row yang memiliki value "DKI Jakarta".

Kalau data sutradara yang berasal dari DKI Jakarta sangat banyak tentu ga efisien. Sekarang kita ubah menjadi 3NF:

Table director			
id	nik	name	origin_id
1	123	Kimo	1
2	321	Timo	2
3	999	Hanung	1

Table origin			
id	origin_code	province	capital
1	JKT	DKI Jakarta	Jakarta Pusat
2	PAP	Papua	Jayapura

Tabel di atas sudah memenuhi 3NF. Ga ada lagi Transitive Dependency. Saat melakukan perubahan nama provinsi, cukup ganti satu row pada tabel origin saja, tabel director hanya mereferensikan datanya lewat origin_id.

4. BCNF

BCNF adalah singkatan dari Boyce-Codd Normalization Form. Ini sebenarnya mirip dengan 3NF dan disebut 3.5NF. Keduanya memiliki kriteria yang mirip, hanya saja pada BCNF tidak ada Functional Dependency sama sekali pada tabel yang memiliki kombinasi kolom Candidate Key untuk menghindari redundancy. Tagline-nya adalah "Nothing but the key". Jadi pada tabel relasi, isinya hanya Key yang berelasi saja. Contoh kasusnya seperti ini:

Table movie		
id	name	year
1	Rumah Dara	2010
2	Ratu Ilmu Hitam	2019
3	Sebelum Iblis Menjemput	2018

Table director			
id	nik	name	movie_id
1	123	Kimo	1
2	321	Timo	1
3	123	Kimo	2
4	321	Timo	3

Tabel di atas memenuhi 3NF karena tidak ada Transitive Dependency. Kolom nik mewakili name & movie_id, sedangkan kolom movie_id hanya mewakili nik. Jadi di sini ada satu Candidate Key, yaitu kombinasi {nik, movie_id} karena keduanya saling bergantung. Tapi itu melanggar BCNF karena ada Functional Dependency, yaitu kolom name yang bergantung ke nik dan bukan Candidate Key. Permasalahannya adalah satu orang bisa menyutradai lebih dari satu film, dan satu film bisa disutradai lebih dari satu orang. Kimo menyutradai Rumah Dara & Ratu Ilmu Hitam, Timo menyutradai Rumah Dara & Sebelum Iblis Menjemput.

Makanya terjadi pengulangan data pada kolom name di atas. Next kita terapkan BCNF:

Table movie		
id	name	year
1	Rumah Dara	2010
2	Ratu Ilmu Hitam	2019
3	Sebelum Iblis Menjemput	2018

Table director		
id	nik	name
1	123	Kimo
2	321	Timo
3	999	Hanung

Table movie_director	
director_id	movie_id
1	1
1	2
2	1
2	3

Kita bikin tabel Composite khusus untuk mapping referensi data director dan movie yang berisi kombinasi Candidate Key saja tanpa kolom lain. Sekarang tidak ada lagi redundancy. Ini cocok untuk many-to-many relationship.

1NF hingga BCNF adalah normalisasi paling dasar dan paling sering ditemui saat desain skema. Kita lanjut ke part yang lebih ekstreme.

5. 4NF

Kriteria 4NF adalah memenuhi syarat BCNF dan tidak ada Multivalued Dependency. Multivalued Dependency adalah ketika terdapat lebih dari satu dependency pada tabel. Misalnya Candidate Key {A, B} saling bergantung dan Candidate Key {B, C} juga saling bergantung, sehingga terdapat lebih dari satu kombinasi Candidate Key pada tabel.

Contohnya seperti berikut:

Table director		
id	nik	name
1	123	Kimo
2	321	Timo
3	999	Hanung

Table movie		
id	name	year
1	Rumah Dara	2010
2	Ratu Ilmu Hitam	2019
3	Sebelum Iblis Menjemput	2018
4	Rudy Habibie	2016

Table genre	
id	name
1	Horror
2	Comedy
3	Biography
4	Drama
5	Action
6	Thriller

Table director_movie_genre		
director_id	movie_id	genre_id
1	1	1
1	1	6
2	1	1
2	1	6
3	4	3
3	4	4

Tabel di atas memenuhi BCNF karena hanya ada Candidate Key pada tabel relasi. Tapi itu Multivalued karena ada 2 Candidate Key, yaitu kombinasi {director_id, movie_id} dan kombinasi {movie_id, genre_id} sehingga melanggar 4NF.

Satu orang bisa menyutradai lebih dari satu film, dan satu film bisa disutradai lebih dari satu orang. Satu film bisa saja memiliki lebih dari satu genre, begitu juga satu genre bisa saja termasuk ke dalam lebih dari satu film. Jadi bisa disimpulkan bahwa, sutradara Kimo menyutradai film Rumah Dara, Rumah Dara adalah film Horror & Thriller, dan Kimo adalah sutradara film genre Horror & Thriller.

Masalahnya, akan terjadi redundancy pada tabel di setiap penambahan film baru dengan multi-genre atau multi-sutradara karena director_id secara tidak langsung ada dependency dengan kolom genre. Mulai kompleks kan? Solusinya kita akan pecah tabel director_movie_genre menjadi seperti berikut:

Table movie_director	
director_id	movie_id
1	1
2	1
3	4

Table movie_genre	
movie_id	genre_id
1	1
1	6
4	3
4	4

Setelah dipecah tidak ada lagi Multivalued Dependency karena masing-masing tabel sudah independent. Jadi tiap ada film baru, tinggal insert genre dan sutradaranya di masing-masing tabel relasi.

6. 5NF

Kriterianya adalah memenuhi syarat 4NF dan tidak ada Join Dependency pada tabel relasi. Join Dependency artinya terjadi redundancy saat Join dan menghasilkan data yang tidak diinginkan.

Misalnya Candidate Key {A, B} saling bergantung, lalu Candidate Key {B, C} di tabel lain juga saling bergantung, sehingga secara tidak langsung Candidate Key A ada dependency ke Candidate Key C melalui perantara Candidate Key B pada saat Join dan menimbulkan redundancy yang tidak diinginkan.

Biar lebih paham bisa dilihat skema di bawah:

Table genre		
id	name	
1	Horror	
2	Comedy	
3	Biography	
4	Drama	
5	Action	
6	Thriller	
Table movie		
id	name	year
1	Rumah Dara	2010
2	Ratu Ilmu Hitam	2019
3	Sebelum Iblis Menjemput	2018
4	Rudy Habibie	2016
Table subgenre		
id	name	
1	Slasher	
2	Gore	
3	Supernatural	
4	Psychological	
Table movie_genre		
movie_id		genre_id
1		1
1		6
4		3
4		4
Table genre_subgenre		
genre_id		subgenre_id
1		1

1	2
1	3
1	4

Satu film memiliki beberapa genre, satu genre bisa juga termasuk ke dalam beberapa film. Satu genre bisa memiliki beberapa subgenre, satu subgenre juga bisa termasuk ke dalam beberapa genre. Satu film bisa saja termasuk ke beberapa subgenre, dan satu subgenre juga bisa termasuk ke dalam beberapa film. Jadi bisa disimpulkan seperti film A memiliki genre B, dan genre B memiliki subgenre C & D, namun film A tersebut hanya termasuk ke dalam subgenre C saja. Makin kompleks dong.

Skema di atas sudah memenuhi 4NF karena tidak ada Multivalued Dependency. Tapi melanggar 5NF karena pada data di atas, film Rumah Dara bergenre Horror dan Thriller. Genre Horror memiliki subgenre seperti Slasher, Gore, Supernatural, dan Psychological. Akan tetapi film Rumah dara hanya termasuk ke dalam subgenre Slasher dan Gore saja.

Masalahnya, film Rumah Dara akan memiliki semua subgenre Horror saat semua table di-Join dan tidak sesuai kriteria bisnis karena akan terjadi redundancy subgenre yang tidak diinginkan seperti berikut:

Join Result		
movie	genre	subgenre
Rumah Dara	Horror	Slasher
Rumah Dara	Horror	Gore
Rumah Dara	Horror	Supernatural
Rumah Dara	Horror	Psychological

Kita harus memastikan bahwa subgenre dari film Rumah Dara hanya Slasher dan Gore. Maka dari itu kita normalize skema di atas dengan menambahkan tabel baru seperti berikut:

Tabel movie_subgenre	
movie_id	subgenre_id
1	1
1	2

Setelah dinormalisasi tidak ada lagi redundancy pada skema tersebut saat semua tabel di-Join dan sudah sesuai kriteria bisnis. Ini adalah normalisasi terakhir dimana tidak ada redundancy sama sekali.

Join Result		
movie	genre	subgenre
Rumah Dara	Horror	Slasher
Rumah Dara	Horror	Gore

D. LATIHAN

INSTRUKSI:

1. Tiap orang memilih 1 dari 19 basis data yang tersedia (list di grup).
2. Berikan deskripsi mengenai basis data tersebut!
3. Normalisasi sampai pada level tertentu (1NF, 2NF, 3NF, BCNF, 4NF, 5NF) tergantung pada kompleksitas tiap tabel!
4. Kondisi apa yang terpenuhi bahwa suatu basis data tersebut memenuhi normalisasi?

Pilihan Basis Data:

1. Perpustakaan

ID_Peminjam	Nama_Anggota	Alamat	Judul_Buku	Pengarang	Tanggal_Pinjam	Tanggal_Kembali
P001	Ali	Jakarta	Matematika Dasar, Fisika Modern	Budi, Sari	2024-05-01	2024-05-10

2. Restoran

No_Transaksi	Nama_Pelanggan	Tanggal	Daftar_Menu	Harga_Total
TR001	Rina	2024-04-15	Nasi Goreng, Es Teh	25000

3. Sekolah

ID_Siswa	Nama	Alamat	Kelas	Mapel_Diikuti	Nama_Guru
S001	Didi	Bandung	X IPA 1	Matematika, Fisika	Bu Lilis, Pak Budi

4. Klinik

ID_Kunjungan	Nama_Pasien	Usia	Dokter	Diagnosa	Obat	Tanggal
K001	Andi	30	Dr. Siti	Flu, Demam	Paracetamol, Vitamin C	2024-03-10

5. Hotel

ID_Pemesanan	Nama_Tamu	Tipe_Kamar	Harga_Kamar	Lama_Menginap	Total
H001	Sarah	Deluxe	500000	2	1000000

6. Bioskop

No_Tiket	Nama_Penonton	Film	Waktu_Tayang	Seat	Harga
B001	Yusuf	Avatar, Dune	19:00, 21:00	A1, A2	70000, 80000

7. Marketplace

No_Transaksi	Pembeli	Produk	Jumlah	Harga_Satuan	Alamat_Kirim
M001	Ani	Mouse, Keyboard	1, 2	100000, 150000	Surabaya

8. Event Organizer

ID_Event	Nama_Klien	Jenis_Event	Vendor	Layanan	Tanggal
E001	Toko Jaya	Pernikahan	Dekorasi, Catering	Sari Decoration, Dapoer Enak	2024-06-20

9. Transportasi

ID_Transaksi	Nama_Penumpang	Trayek	Kendaraan	Supir	Biaya
T001	Lina	Jakarta – Bandung	Bus A	Pak Danu	150000

10. Universitas

ID_Mahasiswa	Nama	Fakultas	Mata_Kuliah	Dosen	Nilai
U001	Tono	Teknik	Kalkulus, Fisika	Bu Nina, Pak Taufik	A, B+

11. Toko Buku

No_Penjualan	Pembeli	Buku	Jumlah	Harga	Tanggal
TB001	Gita	Novel A, Buku B	1, 2	70000, 50000	2024-02-05

12. Gym/Fitness

ID_Member	Nama	Paket	Durasi_Bulan	Harga_Paket	Total
G001	Aldi	Silver	3	200000	600000

13. Music App

ID_User	Nama	Playlist	Lagu	Artis
MU001	Raka	Pagi Ceria	"Lagu A", "Lagu B"	Artis X, Artis Y

14. Pengiriman Barang

No_Resi	Pengirim	Penerima	Barang	Berat	Tujuan	Kurir
P001	Dika	Fina	Sepatu, Jaket	1kg, 2kg	Solo	Kurir A

15. Asuransi

ID_Klaim	Nasabah	Jenis_Asuransi	Nilai_Klaim	Tanggkal	Agensi
A001	Haris	Kesehatan, Mobil	5jt, 20jt	2024-01-10	Budi

16. Taman Hiburan

ID_Tiket	Pengunjung	Wahana	Harga_Tiket	Jumlah_Wahana	Total
TH001	Bayu	Roller Coaster, Bianglala	25000	2	50000

17. Rental Mobil

ID_Sewa	Nama_Pelanggan	Mobil	Plat	Tarif	Durasi	Total
R001	Lusi	Avanza	B1234CD	300000	3 hari	900000

18. Startup Tracker

ID_Startup	Nama	Pendiri	Investor	Industri	Kota
ST001	TechNova	Aldi, Risa	Alpha VC, Beta Capital	EdTech	Jakarta

19. Banking

No_Transaksi	Nama_Nasabah	Jenis_Transaksi	Nominal	Tanggal	Cabang
BNK001	Eko	Setor, Tarik	100000 0, 500000	2024- 04-01	Bandung