

1. Langkah-langkah normalisasi data :

Normalisasi data menjadi bentuk 1NF

Menggunakan data dari studi kasus, dapat dilihat bahwa terdapat beberapa kolom yang berulang-ulang, yaitu No. Anggota, Nama, Status, KodeBuku, Judul, Pengarang, Penerbit, dan ISBN.

Selain itu, kolom Jumlah dan JmlHari juga dapat dijadikan satu kolom saja, yaitu JumlahHalaman. Oleh karena itu, data dapat dinormalisasi menjadi bentuk 1NF dengan membuat dua tabel, yaitu Tabel Anggota dan Tabel Peminjaman.

Tabel Anggota :

No. Anggota	Nama	Status
A-001	Budi	Tetap
A-002	Aulia	Tidak Tetap
A-003	Susan	Tidak Tetap

Tabel Peminjaman :

No. Trans.	No. Anggota	KodeBuku	Judul	Pengarang	Penerbit	ISBN	JumlahHalaman	TglPinjam	TglKembali	JmlHari
12001	A-001	B-001	Delphi	Antoni	Informatika Bandung	978-979-769-513-2	1	10/12/2016	15/12/2016	5
12001	A-001	B-002	Visual Basic	Joni	Erlangga	978-979-518-852-0	1	10/12/2016	15/12/2016	5
12001	A-001	B-003	Foxpro	Andrea	Andi	978-979-8340-17-8	1	10/12/2016	15/12/2016	5
12002	A-002	B-001	Delphi	Antoni	Informatika Bandung	978-979-769-513-2	1	11/12/2016	14/12/2016	3
12002	A-002	B-002	Visual Basic	Joni	Erlangga	978-979-518-852-0	1	11/12/2016	14/12/2016	3
12003	A-003	B-001	Delphi	Antoni	Informatika Bandung	978-979-769-513-2	1	12/12/2016	17/12/2016	5
12003	A-003	B-004	Pascal	Abdul	Modula	978-979-692-046-4	1	12/12/2016	17/12/2016	5

Lakukan pengecekan (Lossless-Join Decomposition) pada tabel baru hasil dekomposisi tersebut!

Tabel Anggota dan Tabel Peminjaman sudah berada dalam bentuk normalisasi 1NF dan tidak dapat didekomposisikan lagi. Oleh karena itu, tidak perlu dilakukan pengecekan Lossless-Join Decomposition pada kedua tabel tersebut.

Namun, pada tabel Peminjaman terdapat beberapa FD yang perlu diperiksa, yaitu:

- No. Trans -> No. Anggota, Nama, Status
- KodeBuku -> Judul, Pengarang, Penerbit, ISBN
- (No. Trans, KodeBuku) -> Jml, TglPinjam, TglKembali, JmlHari

Untuk memastikan bahwa tabel Peminjaman sudah berada dalam bentuk normalisasi 1NF dengan baik, maka perlu dilakukan pengecekan FD pada tabel tersebut.

Setelah dilakukan pengecekan, ditemukan bahwa tabel Peminjaman sudah memenuhi kriteria bentuk normal 1NF dengan baik dan tidak ada anomali data yang muncul setelah dilakukan dekomposisi.

Oleh karena itu, tidak perlu dilakukan normalisasi lebih lanjut ke bentuk normal 2NF dan 3NF. Meskipun demikian, tetap perlu diperhatikan bahwa normalisasi tidak hanya tentang membagi tabel menjadi beberapa tabel yang lebih kecil, melainkan juga memastikan bahwa tabel-tabel

tersebut tidak mengandung anomali data dan bergantung pada key secara fungsional. Oleh karena itu, meskipun tabel Peminjaman sudah berada dalam bentuk normalisasi 1NF dengan baik, perlu diperiksa kembali apakah tabel tersebut sudah memenuhi kriteria bentuk normal 2NF dan 3NF. Dalam kasus ini, terdapat beberapa FD yang perlu diperiksa kembali, yaitu:

- No. Trans -> No. Anggota, Nama, Status
- KodeBuku -> Judul, Pengarang, Penerbit, ISBN
- (No. Trans, KodeBuku) -> Jml, TglPinjam, TglKembali, JmlHari

FD pertama dan kedua sudah memenuhi kriteria bentuk normal 2NF, karena tidak terdapat atribut non-key yang bergantung pada subset key. Sedangkan FD ketiga baru memenuhi kriteria bentuk normal 2NF jika tabel Peminjaman dibagi menjadi dua tabel, yaitu:

- Tabel Peminjaman1 (No. Trans, No. Anggota, TglPinjam, TglKembali)
- Tabel Peminjaman2 (No. Trans, KodeBuku, Jml, JmlHari)

Dengan pemecahan tersebut, semua atribut pada tabel Peminjaman1 dan Peminjaman2 sudah bergantung pada key secara fungsional dan tidak terdapat anomali data.

Namun, FD ketiga belum memenuhi kriteria bentuk normal 3NF, karena terdapat atribut non-key (Jml, JmlHari) yang bergantung pada atribut non-key lain (KodeBuku). Oleh karena itu, perlu dibuat tabel baru, yaitu:

- Tabel Buku (KodeBuku, Judul, Pengarang, Penerbit, ISBN)

Dengan pemecahan tersebut, tabel Peminjaman2 dapat diubah menjadi:

- Tabel Peminjaman2 (No. Trans, KodeBuku, TglPinjam, TglKembali)
- Tabel DetailPeminjaman (No. Trans, KodeBuku, Jml, JmlHari)

Sehingga semua atribut pada tabel Peminjaman1, Peminjaman2, dan DetailPeminjaman sudah bergantung pada key secara fungsional dan tidak terdapat anomali data.

Setelah dilakukan normalisasi tersebut, perlu dilakukan pengecekan Lossless-Join Decomposition dan Dependency Preservation. Setelah dilakukan pengecekan, ditemukan bahwa semua tabel sudah memenuhi kriteria Lossless-Join Decomposition dan Dependency Preservation.

Oleh karena itu, normalisasi yang dilakukan pada tabel Peminjaman meliputi pemecahan menjadi beberapa tabel baru, yaitu Peminjaman1, Peminjaman2, DetailPeminjaman, dan Buku, sehingga semua tabel memenuhi kriteria bentuk normal 2NF dan 3NF dengan baik.

Setelah dilakukan normalisasi pada tabel Peminjaman, hasilnya terdiri dari empat tabel baru, yaitu:

- Tabel Peminjaman1, terdiri dari atribut No\_Trans, No\_Anggota, Nama, dan Status. Tabel ini berisi informasi mengenai transaksi peminjaman buku, termasuk nomor transaksi, nomor anggota yang meminjam, nama anggota, dan status keanggotaan.
- Tabel Peminjaman2, terdiri dari atribut No\_Trans, KodeBuku, Jml, TglPinjam, TglKembali, dan JmlHari. Tabel ini berisi informasi mengenai transaksi peminjaman buku, termasuk nomor transaksi, kode buku yang dipinjam, jumlah buku yang dipinjam, tanggal peminjaman, tanggal pengembalian, dan jumlah hari peminjaman.
- Tabel DetailPeminjaman, terdiri dari atribut No\_Trans dan KodeBuku. Tabel ini berisi informasi detail mengenai buku yang dipinjam dalam suatu transaksi peminjaman.
- Tabel Buku, terdiri dari atribut KodeBuku, Judul, Pengarang, Penerbit, dan ISBN. Tabel ini berisi informasi mengenai buku yang tersedia di perpustakaan, termasuk kode buku, judul buku, pengarang, penerbit, dan ISBN.

Setelah melakukan normalisasi ke bentuk normal 2NF dan 3NF, dilakukan pengecekan Lossless-Join Decomposition dan dependency preservation pada setiap tabel. Dengan normalisasi tersebut, data pada tabel Peminjaman lebih mudah dikelola dan terdapat minimal redundant data, sehingga menghasilkan database yang lebih efisien dan terstruktur.

## 2. Langkah-langkah normalisasi:

Bentuk 1NF:

- Setiap kolom harus memiliki nilai atomik, yaitu tidak dapat dibagi lagi menjadi bagian yang lebih kecil
- Tidak ada kolom yang berisi nilai yang berulang
- Setiap baris pada tabel memiliki identifikasi yang unik (primary key)

Dalam studi kasus ini, kita sudah memiliki satu tabel. Namun, terdapat beberapa kolom yang dapat dipecah lagi, yaitu FROM dan TO. Kita dapat membuat dua tabel terpisah, yaitu:

- Tabel 1: flight\_schedule
- Tabel 2: airport (dengan atribut IATA\_code, city)

Tabel flight\_schedule:

No	FLIGHT_NO	ETA	ETD	DAY_OF_SERVICES	FLEET	REMARKS	from_airport	to_airport
----	-----------	-----	-----	-----------------	-------	---------	--------------	------------

Tabel airport:

IATA_code	city
-----------	------

Kita juga dapat menambahkan primary key pada setiap tabel.

Bentuk 2NF:

- Tabel sudah berada pada bentuk 1NF
- Setiap kolom yang bukan primary key bergantung pada seluruh primary key, bukan hanya sebagian saja

Dalam studi kasus ini, tidak ada masalah pada bentuk 2NF.

Bentuk 3NF:

- Tabel sudah berada pada bentuk 2NF
- Tidak ada ketergantungan transitif, yaitu suatu atribut tidak bergantung pada atribut yang bukan primary key

Dalam studi kasus ini, terdapat beberapa ketergantungan transitif pada tabel flight\_schedule, yaitu:

- Kolom from\_airport dan to\_airport bergantung pada kolom FROM dan TO
- Kolom city bergantung pada kolom IATA\_code

Oleh karena itu, kita dapat memisahkan tabel airport menjadi dua tabel terpisah, yaitu:

- Tabel 2: airport (dengan atribut IATA\_code, city)
- Tabel 3: airport\_route (dengan atribut from\_airport, to\_airport)

Tabel flight\_schedule:

No	FLIGHT_NO	ETA	ETD	DAY_OF_SERVICES	FLEET	REMARKS	from_airport	to_airport
----	-----------	-----	-----	-----------------	-------	---------	--------------	------------

Tabel airport:

IATA_code	city
-----------	------

Tabel airport\_route:

From_airpot	to_airport
-------------	------------

Kita juga dapat menambahkan primary key pada setiap tabel.

Dengan demikian, bentuk normalisasi yang baik adalah:

- Tabel flight\_schedule (berada pada bentuk 2NF dan 3NF)
- Tabel airport (berada pada bentuk 2NF)
- Tabel airport\_route (berada pada bentuk 2NF dan 3NF)