



Topik

Contoh Penerapan Normalisasi Data 1NF hingga 3NF

Tujuan

Mahasiswa memahami:

1. Konsep normalisasi skema relasional ke dalam bentuk yang diinginkan
2. Ciri-ciri tahapan normalisasi 1NF hingga 3NF

Pendahuluan

Pengertian normalisasi

Normalisasi diartikan sebagai suatu teknik yang menstrukturkan/ mendekomposisi/ memecah data dalam cara-cara tertentu untuk mencegah timbulnya permasalahan pengolahan data dalam basis data. Permasalahan yang dimaksud adalah berkaitan dengan penyimpangan-penyimpangan (*anomalies*) yang terjadi akibat adanya kerangkapan data dalam relasi dan inefisiensi pengolahan.

Proses normalisasi akan menghasilkan relasi yang optimal, yaitu :

1. Memiliki struktur *record* yang mudah untuk dimengerti.
2. Memiliki struktur *record* yang sederhana dalam pemeliharaan.
3. Memiliki struktur *record* yang mudah untuk ditampilkan kembali untuk memenuhi kebutuhan pemakai.
4. Minimalisasi kerangkapan data guna meningkatkan kinerja sistem.

Dalam pendekatan normalisasi, perancangan basis data bertitik tolak dari situasi nyata. Ia telah memiliki item-item data yang siap ditempatkan dalam baris dan kolom pada tabel-tabel relasional. Demikian juga dengan sejumlah aturan tentang keterhubungan antara item-item data tersebut. Sementara pendekatan model data ER lebih tepat dilakukan jika yang diketahui baru prinsip sistem secara keseluruhan.

Atribut Tabel

Atribut yang sebenarnya identik dengan pemakaian istilah kolom data. Istilah atribut ini lebih umum digunakan dalam perancangan basis data, karena istilah itu lebih impresif dalam menunjukkan fungsinya sebagai pembentuk karakteristik (sifat-sifat) yang melekat pada sebuah table.

1. Atribut Sederhana (*Simple Attribute*) dan Atribut Komposit (*Composite Attribute*)

Atribut sederhana adalah atribut atomik yang tidak dapat dipilah lagi. Sementara atribut komposit merupakan atribut yang masih dapat diuraikan lagi menjadi sub-sub atribut yang masing-masing memiliki makna.

2. Atribut Bernilai Tunggal (*Single Valued Attribute*) dan Atribut Bernilai Banyak (*Multi Valued Attribute*)

Atribut bernilai tunggal ditujukan pada atribut-atribut yang memiliki paling banyak satu nilai untuk setiap basis data. Pada data mahasiswa, semua atribut (nim, nama_mhs, alamat_mhs dan tgl_lahir) merupakan atribut bernilai tunggal, karena atribut-atribut tersebut hanya dapat

berisi 1 (satu) nilai. Jika seorang mahasiswa yang memiliki 2 (dua) tempat tinggal, maka hanya salah satu saja yang boleh diisikan ke dalam atribut `alamat_mhs`.

3. Atribut Harus Bernilai (*Mandatory Attribute*) dan Nilai Null

Ada sejumlah atribut pada sebuah tabel yang kita tetapkan harus berisi data. Jadi nilainya tidak boleh kosong. Atribut semacam ini disebut *Mandatory Attribute*. Pada tabel mahasiswa, misalnya atribut `nim` dan `nama_mhs` dapat kita golongkan sebagai *Mandatory Attribute*, karena setiap mahasiswa yang datanya ingin disimpan ke tabel tersebut, paling tidak harus diketahui dengan pasti `nim` dan namanya. Sebaliknya pula atribut–atribut lain suatu tabel yang nilainya boleh dikosongkan (*Non Mandatory Attribute*). Nilai Null digunakan untuk menyatakan/ mengisi atribut–atribut demikian yang nilainya memang belum siap atau tidak ada.

4. Atribut Turunan

Atribut turunan adalah atribut yang nilai–nilainya diperbolehkan dari pengolahan atau dapat diturunkan dari atribut atau tabel lain yang berhubungan. Atribut demikian sebetulnya dapat ditiadakan dari sebuah tabel, karena nilai–nilainya bergantung pada nilai yang ada di atribut lainnya. Penambahan atribut angkatan dan IP (Indeks Prestasi) pada tabel mahasiswa berikut merupakan contoh atribut turunan

Domain dan Tipe Data

- Penetapan tipe data pada setiap atribut (kolom) untuk keperluan penentuan struktur setiap tabel. Penetapan tipe data ini akan berimplikasi pada adanya batas–batas nilai yang mungkin disimpan/diisikan kesetiap atribut tersebut. Jika kita menetapkan bahwa tipe data untuk sebuah atribut adalah integer, maka kita hanya mungkin untuk menyimpan data angka yang bulat diantara -32.768 hingga 32.768 . Kita tidak mungkin untuk memasukkan data diluar batas nilai tersebut, data pecahan apalagi data berupa string/text.
- Domain memiliki banyak kesamaan pengertian dengan fungsi tipe data tersebut. Akan tetapi, tipe data merujuk pada kemampuan penyimpanan data yang mungkin bagi suatu atribut secara fisik, tanpa melihat layak tidaknya data tersebut bila dilihat dari kenyataannya pemakaiannya. Sementara domain nilai lebih ditetapkan pada batas–batas nilai yang diperbolehkan bagi suatu atribut, dilihat dari kenyataannya yang ada.
- Contoh : pada tabel kuliah, ditetapkan tipe data untuk atribut `sks` adalah integer. Dengan begitu secara fisik kita dapat menyimpan nilai -1 , 0 atau 100 untuk atribut `sks`. Tetapi kita mengetahui dengan pasti, bahwa nilai–nilai tersebut tidak pantas (*invalid*) untuk menjadi data pada atribut `sks`. Lalu nilai–nilai yang boleh (*valid*) untuk atribut `sks` adalah 1 , 2 , 3 , 4 dan 6 , maka dapat dikatakan, domain nilai untuk atribut `sks` adalah 1 , 2 , 3 , 4 dan 6 .

Ketergantungan Fungsional (*Functional Dependency*)

Diberikan sebuah tabel T berisi paling sedikit 2 buah atribut, yaitu A dan B. Kita dapat menyatakan notasi berikut ini :

$$A \rightarrow B$$

Yang berarti A secara fungsional menentukan B atau B secara fungsional tergantung pada A, jika dan hanya jika setiap kumpulan baris (*row*) yang ada di tabel T, pasti ada 2 baris data (*row*) di tabel dengan nilai A yang sama, maka nilai B pasti juga sama. Definisi yang paling formal untuk itu adalah :

Diberikan 2 *row* r_1 dan r_2 dalam tabel T dimana $A \rightarrow B$.

jika $r_1(A) = r_2(A)$ maka $r_1(B) = r_2(B)$

Dengan melihat data di atas dan dengan pertimbangan intuisi kita, maka ketergantungan fungsional yang dapat kita ajukan adalah :

– $nim \rightarrow nama_mhs$

yang berarti bahwa atribut `nama_mhs` hanya tergantung pada atribut `nim`. Hal ini dibuktikan dari fakta : untuk setiap nilai `nim` yang sama maka pasti nilai `nama_mhs`nya juga sama.

– $nama_kul, nim \rightarrow indeks_nilai$

yang berarti bahwa atribut *indeks_nilai* tergantung pada atribut *nama_kul* dan *nim* secara bersama-sama, memang kita tidak dapat menunjukkan fakta, bahwa untuk setiap nilai *nama_kul* dan *nim* yang sama, maka nilai *indeks_nilainya* juga sama, karena *nama_kul*, *nim* merupakan *key* (sehingga bersifat unik) untuk tabel tersebut. Tetapi, ketergantungan fungsional tersebut sesuai dengan pengertian bahwa setiap *indeks_nilai* diperuntukkan pada mahasiswa tertentu untuk mata kuliah tertentu yang diambilnya.

Tanpa memperhatikan pengertian ketergantungan secara alamiah terhadap tabel tersebut, kita juga dapat mengajukan sejumlah ketidaktergantungan (*non KF*) dengan hanya melihat fakta yang ada, yaitu :

– $\text{nama_kul} \not\rightarrow \text{nim}$

yang artinya atribut *nim* tidak tergantung pada atribut *nama_kul*. Buktinya terlihat pada *row 1* dan *row 2* : dengan nilai *nama_kul* yang sama, tapi nilai *nim*nya berbeda.

– $\text{nim} \not\rightarrow \text{indeks_nilai}$

yang artinya atribut *indeks_nilai* tidak bergantung pada atribut *nim*. Buktinya terlihat pada *row 1* dan *row 3* : dengan nilai *nim* yang sama, tapi nilai *indeks_nilai* berbeda.

Normalisasi dan Ketergantungan Fungsional

Dalam perspektif normalisasi, sebuah basis data dapat dikatakan baik, jika setiap tabel yang menjadi unsur pembentuk basis data tersebut juga telah berada dalam keadaan baik atau normal. Selanjutnya, sebuah tabel dapat dikategorikan baik (*efisien*) atau normal, jika telah memenuhi 3 (tiga) kriteria berikut :

1. Jika ada *dekomposisi* (penguraian) tabel, maka dekomposisinya harus dijamin aman (*Lossless-Join Decomposition*).
2. Terpeliharanya ketergantungan fungsional pada saat perubahan data (*Dependency Preservation*).
3. Tidak melanggar *Boyce-Code Normal Form* (BCNF)

Lossless-Join Decomposition

Dekomposisi memang merupakan upaya untuk mendapatkan tabel yang baik. Tetapi jika tidak hati-hati, upaya ini justru dapat menghasilkan kesalahan. Dekomposisi yang benar terjadi jika tabel-tabel hasil dekomposisi kita gabungkan kembali dapat menghasilkan tabel awal sebelum didekomposisi. Dekomposisi yang benar semacam ini disebut *Lossless-Join Decomposition* atau *Lossless Decomposition* (dekomposisi aman).

Di bawah ini contoh abstrak yang menghasilkan dekomposisi tidak aman (*Lossy-Join Decomposition*) :

Diasumsikan ada table XYZ yang didefinisikan dua buah ketergantungan fungsional $X \rightarrow Y$ dan $Y \rightarrow Z$. Kedua ketergantungan fungsional tersebut diperoleh dari pengamatan terhadap data yang kurang memadai atau karena asumsi yang kurang tepat.

Dependency Preservation

Dependency Preservation (pemeliharaan ketergantungan) merupakan kriteria kedua yang harus dapat dicapai untuk mendapatkan tabel dan basis data yang baik. Ketika kita melakukan perubahan data, maka harus bisa dijamin agar perubahan tersebut tidak menghasilkan *inkonsistensi* data yang mengakibatkan ketergantungan fungsional yang sudah benar menjadi tidak terpenuhi. Akan tetapi, dalam upaya untuk memelihara ketergantungan fungsional yang ada untuk tetap terpenuhi tersebut, prosesnya harus dapat dilakukan dengan efisien.

Bentuk-Bentuk Normalisasi

Normalisasi data adalah proses yang berkaitan dengan model data relasional untuk mengorganisasi himpunan data dengan ketergantungan dan keterkaitan yang tinggi/ erat. Hasil dari proses normalisasi adalah tabel-tabel data dalam bentuk normal (*normal form*), yaitu tabel-tabel data yang terhindar dari :

- Pengulangan informasi.
- Potensi *inkonsistensi* data pada operasi perubahan.

Bentuk Normal Tahap Pertama (1st Normal Form)

Suatu tabel memenuhi 1stNF jika dan hanya jika tabel tersebut tidak memiliki atribut bernilai banyak atau lebih dari satu atribut dengan domain nilai yang sama.

Bentuk Normal Tahap Kedua (2nd Normal Form)

Suatu tabel memenuhi 2ndNF jika dan hanya jika :

1. Tabel tersebut memenuhi 1stNF
2. Setiap atribut yang bukan kunci utama (*primary key*) tergantung secara fungsional terhadap semua atribut kunci utama dan bukan tergantung secara fungsional hanya pada sebagian atribut kunci utama.

Bentuk Normal Tahap Ketiga (3rd Normal Form)

Suatu tabel memenuhi bentuk normal 3rdNF jika dan hanya jika :

1. Tabel tersebut memenuhi 2ndNF.
2. Tidak ada atribut bukan kunci tergantung secara *transitive* pada kunci utama.

Boyce Code Normal Form (BCNF)

Kriteria berikutnya untuk mendapatkan tabel yang baik adalah dengan menerapkan BCNF. Sebuah tabel dikatakan memenuhi BCNF jika untuk semua ketergantungan fungsional dengan notasi $X \rightarrow Y$, maka X harus merupakan *candidate key* pada tabel tersebut. Jika tidak demikian, maka tabel tersebut harus didekomposisi berdasarkan ketergantungan fungsional yang ada, sedemikian hingga X menjadi *candidate key* dari tabel–tabel hasil dekomposisi.

Bentuk Normal Tahap Keempat (4th Normal Form)

Suatu tabel memenuhi bentuk normal 4rdNF jika dan hanya jika :

1. Memenuhi kriteria BCNF
2. Setiap atribut di dalamnya tidak mengalami ketergantungan pada banyak nilai atau dengan kalimat lain, bahwa semua atribut yang mengalami ketergantungan pada banyak nilai adalah bergantung secara fungsional (*functionally dependency*)

Bentuk Normal Tahap Kelima (5th Normal Form)

Suatu tabel memenuhi bentuk normal 5rdNF jika dan hanya jika :

Kerelasiaan antar data dalam relasi tersebut tidak dapat direkonstruksi dari struktur relasi yang memuat atribut yang lebih sedikit.

Langkah-langkah praktikum

Langkah 1

- 1) Bukalah berkas spreadsheet bernama '**DataSoftwareHouseAustria.XLSX**' yang dilampirkan bersama jobsheet ini.
- 2) File tersebut berisi data pegawai software house di Austria beserta atribut lain yang mendukung project yang dikerjakan di perusahaan tersebut.
- 3) Teliti setiap atribut yang tertulis pada data tersebut dan amati ketidak sesuaian atribut yang ada sehingga mempersulit proses insert, update, delete dan modifikasi.
- 4) Pertimbangkan pula bagaimana bentuk yang baik dan sesuai dengan aturan normalisasi yang efisien.

num_project	name_project	num_employee	name_employee	street_address	city	postal_code	num_jobclass	name_jobclass	chr_per_hour	hours_billed
15	Evergreen	103	June E. Arbough	Pazmaniteng 24-9	Vienna	A-1020	E01	Elec. Engineering	€ 84,50	23,8
		101	John G. News	Weihburggasse 26	Vienna	A-1020	E02	Database Designer	€ 105,00	19,4
		105	Alice K. Johnson	Pazmaniteng 25-19	Vienna	A-1020	E02	Database Designer	€ 105,00	35,7
		106	William Smithfield	Gartenweg 8	Rafing	A-1010	E03	Programmer	€ 35,75	12,6
		102	David H. Senior	Wiener Hauptstraße 10/A/4/15	Wien	A-1014	E04	System Analyst	€ 96,75	23,8
18	Amber Wave	114	Annelise Jones	Pazmaniteng 1-1	Vienna	A-1020	E05	Application Designer	€ 48,10	24,6
		118	James J. Frommer	Gartenweg 20	Rafing	A-1010	E06	General Support	€ 18,36	45,3
		104	Anne K. Ramoras	Sterzinger Straße 1	Innsbruck	A-6020	E04	System Analyst	€ 96,75	32,4
		112	Darlene M. Smithson	Liebiggasse 5/1/5	Wien	A-1014	E07	DSS Analyst	€ 45,95	44
22	Einsten	105	Alice K. Johnson	Pazmaniteng 25-19	Vienna	A-1020	E02	Database Designer	€ 105,00	64,7
		104	Anne K. Ramoras	Sterzinger Straße 1	Innsbruck	A-6020	E04	System Analyst	€ 96,75	48,4
		113	Delbert K. Joenbrood	Liebiggasse 3/1/10	Wien	A-1014	E05	Application Designer	€ 48,10	23,6
		111	Geoff B. Wabash	Krakaudorf 32	Krakaudorf	A-8854	E08	Clerical Support	€ 26,87	22
		106	William Smithfield	Gartenweg 8	Rafing	A-1010	E03	Programmer	€ 35,75	12,8
25	Phoenix	107	Maria D. Alonzo	Krakaudorf 11	Krakaudorf	A-8854	E03	Programmer	€ 35,75	24,6
		115	Travis B. Bawangi	Liebiggasse 3/1/10	Wien	A-1014	E04	System Analyst	€ 96,75	45,8
		101	John G. News	Weihburggasse 26	Vienna	A-1020	E02	Database Designer	€ 105,00	56,3
		114	Annelise Jones	Pazmaniteng 1-1	Vienna	A-1020	E05	Application Designer	€ 48,10	33,1
		108	Ralph B. Washington	Krakaudorf 22	Krakaudorf	A-8854	E04	System Analyst	€ 96,75	23,6
		118	James J. Frommer	Gartenweg 20	Rafing	A-1010	E06	General Support	€ 18,36	30,5
		112	Darlene M. Smithson	Liebiggasse 5/1/5	Wien	A-1014	E07	DSS Analyst	€ 45,95	41,4

Langkah 2

- 1) Normalisasi data sesuai bentuk normal kesatu karena data pada langkah 1 tidak memenuhi bentuk normal data bagian satu (1NF). Maka akan menghasilkan dekomposisi tabel sebagai berikut :

num_project	name_project	num_employee	name_employee	street_address	city	postal_code	num_jobclass	name_jobclass	chr_per_hour	hours_billed
15	Evergreen	103	June E. Arbough	Pazmaniteng 24-9	Vienna	A-1020	E01	Elec. Engineering	€ 84,50	23,8
15	Evergreen	101	John G. News	Weihburggasse 26	Vienna	A-1020	E02	Database Designer	€ 105,00	19,4
15	Evergreen	105	Alice K. Johnson	Pazmaniteng 25-19	Vienna	A-1020	E02	Database Designer	€ 105,00	35,7
15	Evergreen	106	William Smithfield	Gartenweg 8	Rafing	A-1010	E03	Programmer	€ 35,75	12,6
15	Evergreen	102	David H. Senior	Wiener Hauptstraße 10/A/4/15	Wien	A-1014	E04	System Analyst	€ 96,75	23,8
18	Amber Wave	114	Annelise Jones	Pazmaniteng 1-1	Vienna	A-1020	E05	Application Designer	€ 48,10	24,6
18	Amber Wave	118	James J. Frommer	Gartenweg 20	Rafing	A-1010	E06	General Support	€ 18,36	45,3
18	Amber Wave	104	Anne K. Ramoras	Sterzinger Straße 1	Innsbruck	A-6020	E04	System Analyst	€ 96,75	32,4
18	Amber Wave	112	Darlene M. Smithson	Liebiggasse 5/1/5	Wien	A-1014	E07	DSS Analyst	€ 45,95	44
22	Einsten	105	Alice K. Johnson	Pazmaniteng 25-19	Vienna	A-1020	E02	Database Designer	€ 105,00	64,7
22	Einsten	104	Anne K. Ramoras	Sterzinger Straße 1	Innsbruck	A-6020	E04	System Analyst	€ 96,75	48,4
22	Einsten	113	Delbert K. Joenbrood	Liebiggasse 3/1/10	Wien	A-1014	E05	Application Designer	€ 48,10	23,6
22	Einsten	111	Geoff B. Wabash	Krakaudorf 32	Krakaudorf	A-8854	E08	Clerical Support	€ 26,87	22
22	Einsten	106	William Smithfield	Gartenweg 8	Rafing	A-1010	E03	Programmer	€ 35,75	12,8
25	Phoenix	107	Maria D. Alonzo	Krakaudorf 11	Krakaudorf	A-8854	E03	Programmer	€ 35,75	24,6
25	Phoenix	115	Travis B. Bawangi	Liebiggasse 3/1/10	Wien	A-1014	E04	System Analyst	€ 96,75	45,8
25	Phoenix	101	John G. News	Weihburggasse 26	Vienna	A-1020	E02	Database Designer	€ 105,00	56,3
25	Phoenix	114	Annelise Jones	Pazmaniteng 1-1	Vienna	A-1020	E05	Application Designer	€ 48,10	33,1
25	Phoenix	108	Ralph B. Washington	Krakaudorf 22	Krakaudorf	A-8854	E04	System Analyst	€ 96,75	23,6
25	Phoenix	118	James J. Frommer	Gartenweg 20	Rafing	A-1010	E06	General Support	€ 18,36	30,5
25	Phoenix	112	Darlene M. Smithson	Liebiggasse 5/1/5	Wien	A-1014	E07	DSS Analyst	€ 45,95	41,4

project {num_project, name_project,
num_employee, name_employee,
street_address, city, postal_code,
num_jobclass, name_jobclass,
chr_per_hour, hours_billed}

project{num_project, name_project}

employee{num_project, num_employee,
name_employee, street_address, city,
postal_code, num_jobclass, name_jobclass,
chr_per_hour, hours_billed}

- 2) Lakukan pengecekan (*Lossless-Join Decomposition*) pada tabel baru hasil dekomposisi tersebut! <SOAL>

Langkah 3

- 1) Pengecekan FD pada tabel hasil langkah 2 akan menghasilkan data FD sebagai berikut :
Pada tabel project
num_project → name_project

Pada tabel employee

- {num_project, num_employee, num_jobclass} → name_employee
- {num_project, num_employee, num_jobclass} → street_address
- {num_project, num_employee, num_jobclass} → city
- {num_project, num_employee, num_jobclass} → postal_code
- {num_project, num_employee, num_jobclass} → name_jobclass
- {num_project, num_employee, num_jobclass} → chr_per_hour
- {num_project, num_employee, num_jobclass} → hours_billed

FD

num_project → name_project

no_employee → {name_employee, street_address, city, postal_code}

no_jobclass → {name_jobclass, chr_per_hours}

{num_project, num_employee, num_jobclass} → hours_billed

- 2) Normalisasi tabel menjadi bentuk 2NF. Berdasarkan informasi FD maka dekomposisi tabel yang semula 2 tabel menjadi 4 tabel sebagai berikut :
 - project{num_project, name_project}
 - employee{no_employee, name_employee, street_address, city, postal_code}
 - jobclass{no_jobclass, name_jobclass, chr_per_hours}
 - hours{num_project, num_employee, num_jobclass, hours_billed}
- 3) Lakukan pengecekan (*Lossless-Join Decomposition*) dan (*dependency preservation*) pada tabel baru hasil dekomposisi tersebut! **<SOAL>**

Langkah 4

- 1) Mencari hubungan syarat 3NF dengan mempertimbangkan $X \rightarrow A$ sesuai dengan ketentuan 3NF Terdapat beberapa non key yang memiliki ketergantungan dengan non key lain, sesuai FD baru sebagai berikut :
 - {city, street_address} → postal_code dimana X adalah superkey
 - name_project → {name_employee, name_jobclass} dimana X adalah superkey
 - {name_employee, name_jobclass} → chr_per_hours dimana X adalah superkey
- 2) Berdasarkan FD tersebut maka tabel pada langkah 3 diperbaharui dekomposisinya menjadi :
 - project{num_project, name_project}
 - employee{no_employee, name_employee, street_address }
 - jobclass{no_jobclass, name_jobclass, chr_per_hours}
 - hours{num_project, num_employee, num_jobclass, hours_billed}
 - address{city, street_address, postal_code}
 - jobclass_detail{name_project, name_employee, name_jobclass}
 - charge{name_employee, name_jobclass, chr_per_hours}
- 3) Lakukan pengecekan (*Lossless-Join Decomposition*) dan (*dependency preservation*) pada tabel baru hasil dekomposisi tersebut! **<SOAL>**

Langkah 5

- 1) Mencari hubungan syarat BCNF dengan mempertimbangkan $X \rightarrow A$ sesuai dengan ketentuan 3NF Terdapat beberapa non key yang memiliki ketergantungan dengan non key lain, sesuai FD baru sebagai berikut :
 - street_address → postal_code dimana X adalah superkey bagian primary key
 - postal_code → city dimana A adalah superkey dan bagian primary key
- 4) Berdasarkan FD tersebut maka tabel pada langkah 4 diperbaharui dekomposisinya menjadi :
 - project{num_project, name_project}
 - employee{no_employee, name_employee, street_address }

- jobclass{no_jobclass, name_jobclass, chr_per_hours}
- hours{num_project, num_employee, num_jobclass, hours_billed}
- address1{street_address, postal_code}
- address2{postal_code, city}
- jobclass_detail{name_project, name_employee, name_jobclass}
- charge{name_employee, name_jobclass, chr_per_hours}

2) Lakukan pengecekan (*Lossless-Join Decomposition*) dan (*dependency preservation*) pada tabel baru hasil dekomposisi tersebut! **<SOAL>**

Tugas

1. Jawablah **<SOAL>** dengan terlebih dahulu melakukan langkah-langkah kegiatan diatas!
2. Lakukan normalisasi sesuai dengan langkah-langkah percobaan diatas sekaligus lakukan pengecekan mulai bentuk 1NF sampai dengan 3NF yang sesuai dengan 3 kriteria bentuk normal yang baik pada studi kasus di bawah ini!

Sebuah data peminjaman perpustakaan di Politeknik Negeri Balikpapan berikut memiliki keistimewaan pada jumlah hari peminjaman buku. Semakin lama jumlah hari peminjaman diartikan semakin banyak jumlah halaman buku yang dipinjam.

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

3. Lakukan normalisasi pada studi kasus di bawah ini mulai bentuk 1NF sampai dengan 3NF, sekaligus lakukan pengecekan yang sesuai dengan 3 kriteria bentuk normal yang baik! (Bentuk normalisasi tidak dibatasi, silahkan disesuaikan dengan kondisi studi kasus)

Indonesia Air memiliki data penerbangan domestik yang rutin dilaksanakan dalam periode tertentu.



PESONA PENERBANGAN INDONESIA

No	FROM	TO	FLIGHT NO.	ETA	ETD	DAY OF SERVICES	FLEET	REMARKS
1	BANDUNG	PADANG	18 6420	06:00	07:40	DAILY	A320	DIRECT
		PALEMBANG	18 6440	10:10	11:20	DAILY	A320	DIRECT
		MEDAN	18 6440	10:10	13:20	DAILY	A320	VIA PALEMBANG
		PEKANBARU	18 6450	17:30	19:10	DAILY	A320	DIRECT
2	MEDAN	PALEMBANG	18 6441	13:50	15:30	DAILY	A320	DIRECT
		BANDUNG	18 6441	13:50	17:05	DAILY	A320	VIA PALEMBANG
3	PADANG	BANDUNG	18 6251	08:05	09:45	DAILY	A320	DIRECT
4	PEKAN BARU	BANDUNG	18 6451	19:35	21:15	DAILY	A320	DIRECT
5	PALEMBANG	BANDUNG	18 6441	15:55	17:05	DAILY	A320	DIRECT
		MEDAN	18 6440	11:45	13:02	DAILY	A320	DIRECT
6	PONTIANAK	SINTANG	18 200	08:00	08:45	FRIDAY & SUNDAY	ATR42	DIRECT
		PANGKALANBUN	18 206	10:15	11:15	MON, FRI, SAT, SUN	ATR42	DIRECT
		KETAPANG	18 202	08:10	08:45	MONDAY & SATURDAY	ATR42	DIRECT
7	SINTANG	PONTIANAK	18 201	09:00	09:45	FRIDAY & SUNDAY	ATR42	DIRECT
8	PANGKALAN BUN	PONTIANAK	18 207	15:30	16:30	MON, FRI, SAT, SUN	ATR42	DIRECT
9	KETAPANG	PONTIANAK	18 203	09:00	09:35	MONDAY & SATURDAY	ATR42	DIRECT

4. Dokumentasikan jawaban pertanyaan 1 sampai 3 dalam sebuah laporan dengan format **PDF!**

-- Selamat Mengerjakan --