

GLOSARIUM

CDM (consep data model) adalah tampilan bisnis terstruktur dari data yang diperlukan untuk mendukung proses bisnis, dan melacak ukuran kinerja terkait.

PDM Merupakan sebuah model yang menggunakan beberapa tabel untuk menggambarkan sebuah data serta hubungan antara data-data yang ada.

PERTEMUAN 13

PENGANTAR NORMALISASI

A. TUJUAN PEMBELAJARAN

Setelah mempelajari materi yang diberikan ini, diharapkan seluruh mahasiswa mampu memahami materi tentang normalisasi, terutama pengertian dari normalisasi bentuk-bentuk normal dan kelebihanannya.

B. URAIAN MATERI

1. Normalisasi

Normalisasi adalah teknik desain *database* yang mengurangi redundansi data dan menghilangkan karakteristik yang tidak diinginkan seperti Penyisipan, Pembaruan, dan Anomali Penghapusan. Aturan normalisasi membagi tabel yang lebih besar menjadi tabel yang lebih kecil dan menautkannya menggunakan hubungan. Menurut Thomas Connolly dan Carolyn Begg didalam bukunya yang berjudul *Database Systems : A Practical Approach to Design, Implementation, and Management*, pada tahun 2005. Normalisasi yaitu teknik desain sebuah *database*, yang awali dengan memeriksa sebuah hubungan (disebut sebuah dependensi fungsional) antara atribut. Atribut menggambarkan beberapa properti data atau hubungan antara data yang penting bagi perusahaan. Normalisasi menggunakan serangkaian tes (dijelaskan sebagai bentuk normal) untuk membantu mengidentifikasi pengelompokan yang optimal untuk atribut-atribut ini untuk akhirnya mengidentifikasi sekumpulan relasi yang sesuai yang mendukung persyaratan data perusahaan. Tujuan dari Normalisasi dalam SQL adalah untuk menghilangkan data yang berlebihan (berulang) dan memastikan data disimpan secara logis.

Penemu model relasional Edgar Codd mengajukan teori normalisasi dengan pengenalan Bentuk Normal Pertama, dan ia melanjutkan memperluas teori dengan Bentuk Normal Kedua dan Ketiga. Kemudian bergabung dengan Raymond F. Boyce untuk mengembangkan teori Bentuk Normal Boyce-Codd.

Terdapat Tiga bentuk normal awalnya diusulkan disebut Bentuk Normal Pertama atau disebut juga 1NF, Bentuk Normal Kedua atau 2NF, dan Bentuk Normal Ketiga atau 3NF. Dan selanjutnya, R. Boyce dan E.F. Codd memperkenalkan juga definisi yang lebih kuat daripada bentuk normal ketiga yang disebut Bentuk Normal Boyce-Codd atau BCNF dari Codd, 1974. Menggunakan pengecualian 1NF, semua bentuk normalnya didasarkan pada suatu bentuk ketergantungan fungsional yang di antara atribut pada suatu relasi (Maier, 1983). Bentuk normal lebih tinggi bisa dikatakan Bentuk Normal Keempat (4NF) dan selanjutnya Bentuk Normal Kelima (5NF) (Fagin, 1977, 1979). Namun, bentuk normal yang terakhir muncul sangat jarang digunakan. Ada beberapa tujuan dari sebuah bentuk normalisasi, yaitu :

- a. Dapat untuk menghilangkan sebuah redudansi data.
- b. Dapat untuk mengurangi sebuah kompleksitas, dan
- c. Dapat juga untuk mempermudah pemodifikasian sebuah data.

Berikut adalah yang termasuk manfaat dari proses normalisasi tetapi juga tidak terbatas pada hal-hal berikut ini :

- a. Mengurangi anomali yang terkait dengan modifikasi data.
- b. Pencarian indeks lebih cepat dengan indeks yang lebih pendek dan lebih kecil.
- c. Membuat *database* lebih tersusun rapi dengan sedikit atau tanpa nilai null dan lebih sedikit data yang mengulang atau redudan.
- d. Memicu eksekusi perintah dengan cara yang lebih cepat dan lebih efisien selagi mempertahankan data yang mengulang atau berlebihan.
- e. Menciptakan ruang untuk penggunaan segmen yang lebih baik yang dapat dimanfaatkan dalam mengontrol penempatan data fisik.
- f. Ekspresi indeks yang lebih sedikit per tabelnya untuk menciptakan ruang untuk dapat melakukan pengaturan yang lebih fleksibel.

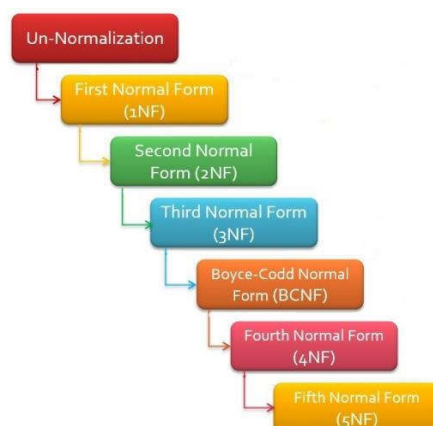
Kemampuan untuk melakukan analisis data dengan lebih mudah adalah alasan yang cukup bagi Sebagian organisasi untuk menggunakan normalisasi data. Namun, ada lebih banyak alasan untuk melakukan proses normalisasi ini, semuanya sangat bermanfaat. Salah satu yang paling terlihat adalah fakta bahwa normalisasi data berarti *database* menggunakan ruang yang lebih sedikit. Perhatian utama dalam mengumpulkan dan menggunakan data yang besar atau banyak adalah besarnya jumlah memori yang dibutuhkan untuk

menyimpannya. Meskipun pilihan penyimpanan menjadi lebih besar dan lebih efisien dengan kemajuan teknologi, dengan sekarang berada dimasa penyimpanan sudah mencapai terabyte dan yang lebih besar pun sudah tidak dapat mengakomodirnya lagi. Dengan demikian, menemukan cara untuk mengurangi ruang penyimpanan adalah sebuah prioritas, dan normalisasi data dapat melakukannya.

Menggunakan ruang penyimpanan yang lebih sedikit adalah hal yang bagus, dengan begitu juga memiliki efek yang dapat meningkatkan kinerja dari *database* tersebut. *Database* yang berjalan lancar oleh banyak informasi yang tidak perlu berarti analisis data dapat dilakukan dengan lebih cepat dan efisien. Jika kesulitan dengan analisis data, maka melakukan normalisasi data adalah hal yang wajib dilakukan untuk sebuah *database*. Manfaat normalisasi data melampaui ruang penyimpanan dan efek yang terkaitnya. Dengan menggunakan proses normalisasi data ini, kita akan lebih mudah mengubah dan memperbarui data dalam *database*. Karena redundansi dan kesalahan tidak ada, maka datanya akan jauh lebih bersih dan tidak akan kesulitan Ketika merubah informasi dari data tersebut.

2. Bentuk-Bentuk Normalisasi

Normalisasi memiliki banyak bentuk, dan dari setiap bentuk normalisasi masing-masing memiliki kelebihan dan aturannya. Ada beberapa tahapan dalam normalisasi.



Gambar 0.1 Tahapan Normalisasi

a. Un-Normalization

Pada bentuk un-normalization, semua atribut dan data pada dokumen dasar dimasukkan dalam satu himpunan. Bentuk di bawah ini dikatakan un-Normalization, dikarenakan adanya :

- 1) Nilai yang kosong (*null*)
- 2) Mereduksi/menanggulangi redudansi data
- 3) Memiliki atribut yang bernilai jamak

NIM	NAMA	JURUSAN	SERTIFIKAT
20120101	Achmad	Teknik Informatika	Oracle
	Null		Java
			Cisco
20120205	Shinta	Sistem Informasi	MS. Office
	Null		Linux
20120304	Ristie	Teknik Komputer	Java
	Null		Linux

Gambar 0.2 Un-normalization

Berikut adalah atribut yang bernilai jamak.

NIM	NAMA	JURUSAN	SERTIFIKAT
20120101	Achmad	Teknik Informatika	Oracle, Java, Cisco
20120205	Shinta	Sistem Informasi	MS. Office, Linux
20120304	Ristie	Teknik Komputer	Java, Linux

Gambar 0.3 Atribut yang bernilai jamak

Pada sebuah perancangan *database* dapat diawali dari sebuah dokumen dasar yang biasanya dipakai dalam *system* sesuai dengan yang ada di lingkup *system* yang akan dibuat rancangan *datasenya* tersebut. Berikut ini adalah contoh dokumen mengenai sebuah faktur pembelian sebuah barang pada PT Revanda Jaya.

FAKTUR PEMBELIAN BARANG				
PT REVANDA JAYA Jl. Bekasi Timur No. 2 Bekasi Timur				
Kode Supplier : G01 Nama Supplier : Gobel Nustra			Tanggal : 07/02/2001 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 Nama Supplier : Hitachi			Tanggal : 02/02/2001 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				

Gambar 0.4 Contoh Faktur pembelian barang

Untuk membuat un-Normalized form atau bentuk tidak normal, caranya dengan mengidentifikasi dan mencantumkan keseluruhan item data yang ada pada faktur diatas. Contohnya menjadi seperti dibawah ini.

No Fac	Kode Supp	Nama Supp	Kode Brg	Nama Barang	Tanggal	Jatuh Tempo	Qty	Harga	Jumlah	Total
779	S02	Hitachi	R02	Rice Chocker C3	02/02/01	09/03/01	10	150000	1500000	1500000
998	G01	Gobel Nustra	A01	AC Split ½ PK	07/02/01	09/03.01	10	135000	13500000	33500000
			A02	AC Split 1 PK			10	2000000	20000000	

Gambar 0.5 Contoh Un-normalized

b. First Normal Form (1NF)

Normalisasi struktur tabel akan mengurangi redundansi data. Jika grup bersulang memang ada, grup tersebut harus dihilangkan dengan memastikan bahwa setiap baris mendefinisikan satu *instance* entitas dan bahwa setiap persimpangan kolom baris hanya memiliki satu nilai. Selain itu, ketergantungan harus diidentifikasi untuk mendiagnosis bentuk normal. Identifikasi bentuk normal yang memungkinkan Anda untuk mengetahui di manakah Anda berada dalam proses normalisasi. kita harus mendefinisikan konsep kunci dalam konteks tabel yang dinormalisasi. Superkey adalah

sekumpulan satu dan lebih atribut, yang bila diambil secara kolektif, memungkinkan kita dapat mengidentifikasi secara unik suatu entitas dan tabel. Subset apa pun dari atribut superkey yang juga adalah superkey, dan tidak dapat direduksi menjadi superkey lain, disebut kunci kandidat. *Primary key* dipilih dari kumpulan kunci kandidat yang akan digunakan dalam indeks untuk tabel itu. Suatu relasi/tabel dapat memenuhi 1NF yang hanya terdapat di setiap atribut dari setiap relasi/tabelnya hanya memiliki satu nilai dalam satu baris atau *record*. Oleh sebab tidak boleh ada nilai kosong (*null value*) dari sebuah grup yang berulang.

Berikut adalah hasil dari 1NF.

No Fac	Kode Supp	Nama Supp	Kode Brg	Nama Barang	Tanggal	Jatuh Tempo	Qty	Harga
779	S02	Hitachi	R02	Rice Chocker C3	02/02/01	09/03/01	10	150000
998	G01	Gobel Nustra	A01	AC Split ½ PK	07/02/01	09/03/01	10	135000
998	G01	Gobel Nustra	A02	AC Split 1 PK	07/02/01	09/03/01	10	2000000

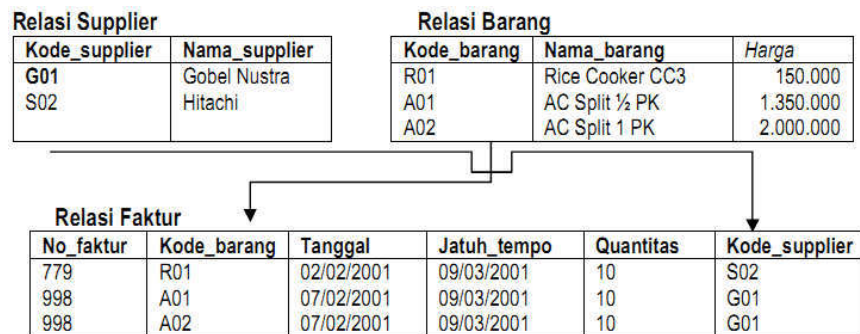
Gambar 0.6 Contoh first normal form

c. Second Normal Form (2NF)

Sebuah relasi atau tabel dapat memenuhi kriteria 2NF jika sudah memenuhi kriteria dari 1NF. Atribut yang bukan kata kunci ketergantungan secara fungsional kepada kunci utamanya dan bukan sebagai sebuah atribut. Adapun caranya dalam membuat 2NF, yaitu:

- 1) Tentukan atribut yang akan digunakan sebagai kunci utama.
- 2) Dekomposisi atribut-atribut yang bergantung terhadap kunci utamanya.
- 3) Hilangkan data yang redudan atau mengulang pada atribut kunci.
- 4) Abaikan atau hilangkan derived attribute (sebuah atribut yang nilainya didapat dari atribut lainnya).
- 5) Identifikasi atribut sebuah kunci yang akan menjadi suatu relasi.

Berikut adalah hasil dari 2NF.

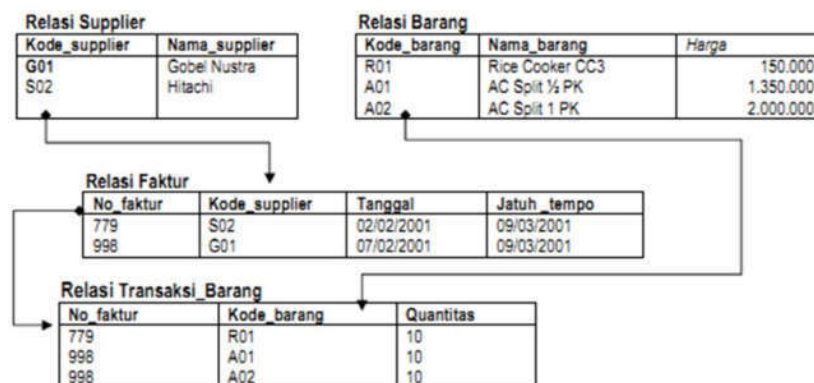


Gambar 0.7 Contoh second normal form

d. Third Normal Form (3NF)

Sebuah relasi atau tabel memenuhi kriteria 3NF jika sudah memenuhi 2NF. Atribut yang bukan sebuah kunci tidak dapat memiliki ketergantungan secara transitif kepada kunci utamanya (*primary key*) atau mentiadakan ketergantungan transitif. Berikut adalah cara dalam memenuhi 3NF yaitu:

- 1) Pisahkan atribut yang ketergantungan secara transitif.
- 2) Relasikan atribut kunci dari ketergantungan transitif dari relasi dekomposisi.



Gambar 0.8 Contoh third normal form

e. Boyce-Codd Normal Form (BCNF)

Bentuk Normal Boyce–Codd (BCNF) dasarnya adalah dependensi fungsional yang memperhitungkan semua kandidat kunci dalam sebuah bentuk relasi. Namun, BCNF juga mempunyai beberapa kendala tambahan dibandingkan definisi umum dari 3NF. Perbedaan dari 3NF dan BCNF yaitu

bahwa terdapat ketergantungan fungsional $A \rightarrow B$, 3NF dapat menjadi relasi jika atribut B adalah atribut kunci-primer dan atribut A bukan kunci kandidat, sedangkan BCNF bersikeras untuk ketergantungan ini tetap di dalam Relasi A, atribut A harus menjadi sebuah kunci kandidat. Oleh sebab itu, Bentuk Normal Boyce-Codd adalah bentuk 3NF lebih kuat, sehingga tiap relasi dalam BCNF juga terdapat didalam 3NF. Namun, relasi di 3NF tidak harus ada di BCNF. Kita bisa mendekomposisi relasi yang tidak ada di BCNF menjadi BCNF. Namun, mungkin tidak selalu diinginkan atau dibutuhkan untuk mengubah relasi menjadi BCNF; misalnya, jika ada ketergantungan fungsional yang tidak dipertahankan saat kita melakukan dekomposisi (yaitu, determinan dan atribut yang ditentukan ditempatkan dalam relasi yang berbeda). Dalam situasi ini, sulit untuk menegakkan ketergantungan fungsional dalam relasi, dan kendala penting hilang. Berikut adalah contoh dari BCNF :

Contoh :

Tabel $R=(A, B, C)$ dengan FD : $A \rightarrow B$ dan $B \rightarrow C$

maka R bukan bagian BCNF, sebab :

- 1) A merupakan superkey ?

$A \rightarrow B$ (diketahui)

$A \rightarrow B$ dan $B \rightarrow C$ maka $A \rightarrow C$ (transitif)

$A \rightarrow A$ (refleksif)

Sehingga $A \rightarrow (A, B, C)$ atau $A \rightarrow R$. Jadi A adalah superkey.

- 2) B merupakan superkey ?

$B \rightarrow C$ (diketahui)

$B \rightarrow B$ (refleksif)

Tapi $B \nrightarrow A$. Sehingga $B \nrightarrow A, B, C$ atau B adalah bukan superkey.

Agar R memenuhi dari BCNF maka dapat didekomposisi menjadi sebagai berikut:

$R_1 = (A, B)$; FD : $A \rightarrow B$ dan $R_2 = (B, C)$; FD : $B \rightarrow C$.

Sehingga R_1 dan R_2 masing-masing dapat memenuhi BCNF. Karena A dan B semua sekarang menjadi sebuah superkey.

Students

sid	name	age
53666	Jones	18
53668	Smith	18
53669	Melissa	17
53670	Hilden	19

Gambar 0.9 Tabel Murid

Didalam tabel student terdapat sid, name dan age. Functional Depedency(FD) : $\text{sid} \rightarrow \text{name, age}$. Tabel diatas adalah BCNF karena sid adalah superkey.

Books

bid	title	year
B001	MySQL	2002
B002	Algorithm	2003
B003	Visual Foxpro 6.0	2003
B004	Visual basic 6.0	2005

Gambar 0.10 Tabel buku

Pada tabel books terdapat bid, title, dan year. Functional Depedency(FD) : $\text{bid} \rightarrow \text{title, year}$. Tabel diatas juga tabel BCNF karena bid adalah superkey.

Pinjam

idpinjam	sid	bid	date
P-01	53666	B002	10/11/2005
P-02	53668	B001	10/11/2005
P-03	53668	B004	11/12/2005
P-04	53670	B002	14/11/2005

Gambar 0.11 Tabel pinjam

Pada tabel pinjam terdapat (idpinjam, sid, bid, dan date). Functional Depedency (FD) : idpinjam \rightarrow bid, date. Tabel diatas bukanlah BCNF dikarenakan idpinjam bukan merupakan superkey, idpinjam \rightarrow sid. Tabel diatas dapat didekomposisi menjadi BCNF, contoh :

Tabel diatas didekomposisi menjadi :

Pinjam1

idpinjam	sid
P-01	53666
P-02	53668
P-03	53668
P-04	53670

Gambar 0.12 Tabel pinjam 1

FD trivial, maka menjadi BCNF

Pinjam2

idpinjam	bid	date
P-01	B002	10/11/2005
P-02	B001	10/11/2005
P-03	B004	11/12/2005
P-04	B002	14/11/2005

Gambar 0.13 Tabel pinjam 2

$\text{Idpinjam} \rightarrow \text{bid, date}$

Maka idpinjam adalah superkey. Maka tabel diatas menjadi sebuah BCNF.

f. Fourth Normal Form (4NF)

Fourth Normal Form (4NF) adalah bentuk normal yang lebih kuat daripada BCNF karena mencegah relasi mengandung MVD nontrivial, dan karenanya redundansi data (Fagin, 1977). Normalisasi relasi dari BCNF ke 4NF melibatkan penghapusan MVD dari sebuah relasi dengan menempatkan sebuah atribut didalam suatu relasi baru dengan salinan dari determinan.

g. Fifth Normal Form (5NF)

Fifth Normal Form (5NF) adalah suatu relasi yang tidak mengandung sebuah ketergantungan bentuk gabungan. Untuk sebuah relasi R dengan adanya himpunan bagian dari sebuah atribut R dilambangkan sebagai atribut berikut A, B, sampai dengan, Z, suatu relasi R dapat memenuhi dependensi gabungan jika dan hanya jika setiap atribut nilai legal R sama dengan gabungan proyeksinya pada A, B, sampai dengan, Z. Kapan saja kita mendekomposisi sebuah relasi menjadi dua relasi, relasi yang dihasilkan memiliki properti lossless-join. Properti ini mengacu pada fakta bahwa kita dapat menggabungkan kembali relasi yang dihasilkan untuk menghasilkan relasi asli. Namun, ada kasus dimana ada kebutuhan untuk mendekomposisi sebuah relasi menjadi lebih dari dua relasi. Meskipun jarang, kasus ini dikelola dengan join dependency dan fifth normal form (5NF). Lossless-join dependency adalah Properti dekomposisi, yang memastikan bahwa tidak ada struktur data palsu yang dihasilkan saat relasi disatukan kembali melalui operasi gabungan alami.

C. SOAL LATIHAN/TUGAS

Setelah mempelajari dan memahami materi tentang normalisasi diatas, berikut adalah soal dan Latihan untuk mengasah kemampuan.

1. Sebutkan apa saja manfaat dari melakukannya normalisasi *database*!
2. Bagaimanakah cara membuat 1NF, apa saja syarat-syarat yang harus dipenuhi Ketika membuat 1NF?
3. Bagaimanakah cara membuat 2NF dan apa saja syarat untuk memenuhi 2NF?
4. Bagaimana cara membuat 3NF dan apa saja syarat untuk memnuhi 3NF?
5. Berikut terdapat unnormalized form, buatlah bentuk normal 1NF, 2NF dan 3NF!

Id_pinjam	Nm_peminjam	Jm_hr_pj	Biaya_pinjam	Id_buku	Nm_buku	Penerbit	Thn_terbit
pj001	Boediyono	3	Rp.6000	Bk001	Matematika	Erlangga	2010
		1	Rp.2000	BK003	Visual basic	Gramedia	2012
pj002	Mulyadi	2	Rp.4000	Bk002	Basis data	Tinta emas	2009
pj003	Burhan	1	Rp.2000	Bk005	Struktur data	Smart book	2011
		5	Rp.9000	Bk001	Matematika	Erlangga	2010
pj004	Alex	3	Rp.6000	Bk004	Jaringan	Penebar swadaya	2008
pj005	Suparman	2	Rp.4000	Bk002	Basis data	Tinta emas	2009

6. Berikut terdapat normal form dari sebuah kuitansi pembelian barang otomotif, buatlah bentuk normal 1NF, 2NF dan 3NF.

" Honda Jaya Raya" AHASS 06488 Jatimulya - Bekasi Timur Telp. 021-82432162			No Faktur: 05103214 Tanggal : 25/10/2005		
BON PEMBELIAN					
No Polisi : B3117LB, Warna: Biru Merek : Supra X, Tahun: 2005 MekanikID : DDE, Nama: Djoko Dewanto					
Kode Parts	Nama Parts	Kuantum	Harga (*)	Discount	Jumlah Rp.
20W501000CC	Oli Top 1 1000cc	2	27,000	1,000	52,000
SERV001	Engine Tune Up	1	25,000	2,000	23,000
				Potongan	2,000
				Total Rp.	73,000

(*) Harga tersebut sudah termasuk PPN

Lembar ke-1 : Pelanggan

Lembar ke-2 : Accounting

D. REFERENSI

Connolly, T., & Begg, C. (2005). *Database System: A Practical Approach to Design, Implementation, and Management, Fourth Edition*. Harlow: Pearson Education Limited.

Coronel, C., & Morris, S. (2017). *Database System: Design, Implementation, & Management, 13th Edition*. Boston: Cengage Learning, Inc.

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2011). *Database System Concept ; Sixth Edition*. New York: McGraw-Hill.

teorey, t., lightstone, s., & nadeau, t. (2006). *Database Modeling & Design*. San Fransisco: Morgan Kaufmann Publisher

GLOSARIUM

Un-normalization adalah pada bentuk un-normalization, semua atribut dan data pada dokumen dasar dimasukkan dalam satu himpunan.

First normal form adalah Struktur tipikal pertama atau disebut juga 1NF memerlukan beberapa kondisi didalam sebuah *database* itu sendiri, yang artinya adalah elemen dari bentuk normal pertama ini.

Second normal form adalah Syarat untuk dapat menerapkan bentuk normalisasi kedua ini, dapat menjadi informasi yang telah dibentuk kedalam 1NF, yang artinya adalah beberapa kapasitas standard yang harus di penuhi 2NF.

Third Normal form Adalah Normalisasi sebuah *database* kedalam bentuk 3NF yang bertujuan untuk dapat membuang keseluruhan dari atribut atau *field* yang tidak dapat diidentifikasi dengan sebuah *primary key*. Dengan ini tidak akan ada ketergantungan yang transitif pada setiap sebuah kandidat key.

Boyce-codd normal form merupakan sebuah prosedur pada bentuk normalisasi *database* yang dapat juga disebut 3.5NF, yaitu memiliki hubungan yang sangat erat dengan sebuah bentuk dari 3NF.

Fourth normal form adalah bentuk normal yang lebih kuat daripada BCNF karena mencegah relasi mengandung MVD nontrivial.

Fifth normal form adalah relasi yang tidak mengandung ketergantungan gabungan.