

Mata Kuliah	:	Sistem Basis Data
Bobot Sks	:	3 sks
Dosen Pengembang	:	Cian Ramadhona Hassolthine, S.Kom., M.Kom
Tutor	:	Cian Ramadhona Hassolthine, S.Kom., M.Kom
Capaian Pembelajaran Mata Kuliah	:	Mahasiswa dapat memodelkan data dengan menggunakan konsep EER (1) dan tugas 2
Kompetensi Akhir Di Setiap Tahap (Sub-Cpmk)	:	Mahasiswa dapat memodelkan Konsep EER, Superclass/subclass Relationship dan Derajat Relationship
Minggu Perkuliahan Online Ke-	:	Sesi 10

Normalization 4NF, 5NF

- Penyimpangan dalam modifikasi
- Penyimpangan dalam proses modifikasi data disebut anomalies

Ada 3 bentuk penyimpangan :

1. Delete anomalies

Adalah proses penghapusan suatu entity logik yang mengakibatkan hilangnya informasi tentang entity yang tidak direlasikan secara logik

Contoh :

Nomhs	Nama	Kode Mtk	SKS
123456	Ali baba	INA 101	3
123457	Pipiyot	TFD 234	2
123467	Nirmala	INA 201	3
123445	Lala	INA 101	3

Apabila “Ali baba” membatalkan mengambil matakuliah “INA 101”, maka apabila record tersebut dihapus akan menyebabkan seluruh informasi tentang ‘Ali baba” akan ikut terhapus.

2. Insert anomalies

Adalah proses penyisipan entity logik yang memerlukan penyisipan entity logik yang lain

3. Update anomalies

Adalah proses mengupdate data pada suatu entity logik yang mengakibatkan perubahan pada lebih dari satu tempat dalam suatu relasi

Contoh : Perubahan SKS pada “INA 101” tidak hanya dilakukan pada satu record saja, tetapi pada record dan relasi lain yang memuat data tersebut.

Keharusan Menghilangkan Masalah-Masalah Akibat Ketergantungan

Yang harus dilakukan adalah jika struktur data dalam relasi dirancang sedemikian rupa sehingga atribut-atribut bukan kunci hanya tergantung pada atribut kunci dan tidak pada atribut lain

Ada 3 ketergantungan :

1. Functional Dependence (FD)

FD akan muncul diantara dua rinci data dalam suatu struktur data jika nilai salah satu rinci data mengimplikasikan nilai pada rinci data kedua

Atau rinci data pertama menentukan (determines) rinci data kedua

Contoh :

Matakuliah (Kode, Nama, SKS, Semester)

FD = Matakuliah.Kode \rightarrow (Matakuliah.Nama, Matakuliah.Semester)

Matakuliah.nama \rightarrow (Matakuliah.Kode, Matakuliah.Semester)

2. Full Functional Dependence (FFD)

Suatu rinci data dikatakan FFD pada suatu kombinasi rinci data jika FD pada kombinasi rinci data dan tidak FD pada bagian lain dari kombinasi rinci data

Contoh : SKS pada tabel matakuliah hanya bergantung pada kode matakuliah, dan tidak ditentukan oleh siapa yang mengambil matakuliah tersebut

3. Transitive Dependence (TD)

Muncul jika suatu nilai pada rinci data pertama menentukan nilai pada rinci data kedua

yang bukan Candidate Key (CK), dan nilai pada rinci data kedua menentukan nilai pada rinci data ketiga

Jadi TD terjadi jika suatu nilai rinci data mempunyai ketergantungan pada dua nilai rinci data

EFEK-EFEK NORMALISASI

Akibat yang muncul dalam proses normalisasi :

1. Masalah kekangan dalam basis data
 - a. Duplikasi rinci data
 - b. Adanya Integritas referensial yang harus terjaga dan nilai-nilai pada Attribute Key (AK) tidak boleh null maka proses dekomposisi akan menghasilkan suatu set yang inheren pada batasan integritas referensial
2. Ketidakefisienan dalam menampilkan kembali data tersebut

ATRIBUT TABEL

Atribut adalah karakteristik atau sifat yang melekat pada sebuah tabel, atau disebut juga kolom data

PENGELOMPOKAN ATRIBUT :

1. **Atribut Key** Adalah satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris data dalam tabel secara unik (tidak boleh ada dua atau lebih baris data dengan nilai yang sama untuk atribut tertentu).

Ada 3 key :

- a. Superkey Merupakan satu atau kumpulan atribut yang dapat membedakan setiap baris data dalam sebuah tabel secara unik

Contoh : superkey di tabel mahasiswa

(nomhs, nama, alamat, tgllahir)

(nomhs, nama, tgllahir)

(nomhs, nama)

(nomhs)

- b. Candidate key Merupakan kumpulan atribut minimal yang dapat membedakan setiap

baris data dalam sebuah tabel secara unik Sebuah CK pasti superkey, tapi belum tentu sebaliknya

Contoh : pada tabel mahasiswa

(nomhs)

(nama)

- c. **Primary key** Dari beberapa CK dapat dipilih satu untuk dijadikan PK, yang memiliki keunikan paling baik

Contoh : dari tabel mahasiswa, yang layak dijadikan PK adalah nomhs

- 2. **Atribut deskriptif** Merupakan atribut yang bukan merupakan anggota dari PK

- 3. **Atribut sederhana** Adalah atribut atomik yang tidak dapat dipilah lagi

Contoh : Nomhs, Nama

- 4. **Atribut komposit** Adalah atribut yang masih bisa diuraikan lagi menjadi sub-atribut yang masing-masing memiliki makna.

Contoh : Alamat → Alamat, Kota, Propinsi, Kode Pos

- 5. **Atribut bernilai tunggal** Ditujukan pada atribut-atribut yang memiliki paling banyak satu nilai untuk setiap baris data

Contoh : Nomhs, Nama, Tanggal lahir → hanya dapat berisi satu nilai untuk seorang mahasiswa

- 6. **Atribut bernilai banyak** Ditujukan pada atribut-atribut yang dapat diisi dengan lebih dari satu nilai, tapi jenisnya sama

Contoh : pada tabel mahasiswa dapat ditambah atribut HOBBY, karena seorang mahasiswa dapat memiliki beberapa hobby

- 7. **Atribut harus bernilai (mandatory)** Adalah atribut yang nilainya tidak boleh kosong, atau harus ada nilainya. Misalnya data Nomhs dan Nama mahasiswa

- a. Nilai NULL digunakan untuk mengisi atribut yang nilainya belum siap atau tidak ada
- b. NULL (karakter ke 0) tidak sama dengan SPASI (karakter ke 32)

Domain Dan Tipe Data

- **Domain**, memiliki pengertian yang 5ttrib sama dengan tipe data, namun domain lebih ditekankan pada batas-batas nilai yang diperbolehkan pada suatu atribut
Contoh : data SKS bertipe integer. Namun dalam kenyataan tidak 5ttribu yang bernilai 5ttribut. Berarti domain nilai sks adalah integer > 0
- **Tipe data** merujuk pada kemampuan penyimpanan data yang mungkin bagi suatu atribut secara fisik, tanpa melihat kelayakan data tersebut bila dilihat dari kenyataan pemakaiannya

Bentuk-Bentuk Normal

Normalisasi merupakan sebuah 5ttrib dalam logical desain sebuah basis data, 5ttrib pengelompokan atribut dari suatu relasi sehingga membentuk struktur relasi yang baik (tanpa redundansi)

1. First Normal Form (1NF)

Tidak diperkenankan atribut dengan nilai jamak, komposit, dan segala kombinasinya yang akan menyebabkan redundansi

Contoh:

- SISWA {NRP, NAMA, KELAS, JENIS_KELAMIN}
- Jika penulisan JENIS_KELAMIN misalkan "Laki-laki" atau "Pria" atau "Perempuan" atau "Wanita" bisa menimbulkan masalah konsistensi
- Kalau penulisan menggunakan "L"/"P" atau "1"/"2" masih dianggap benar
- Normalisasi 1NF, JENIS_KELAMIN dijadikan 5ttrib baru dengan atribut KODE sebagai key dan JENIS_KELAMIN

Catatan:

- a. Perbedaan antara redundansi dan tidak unik → lihat penjelasan mengenai mapping pada atribut jamak
- b. Masalah penambahan kunci baru pada 5ttrib yang tidak memiliki kunci (dianggap banyak data yang kembar atau tidak unik) □ lihat penjelasan pada Catatan saat dilakukan mapping

Data yang “kebetulan kembar” bukan masalah, yang tidak diperbolehkan adalah data yang “benar-benar kembar” □ lihat penjelasan mengenai perbedaan redundansi dan tidak unik.

Contoh lain:

- Anggap ada suatu relasi MELANGGAR antara SISWA dan GURU dengan rasio N:M sehingga dalam mapping-nya harus dibuatkan 6ttrib baru
- Pada 6ttrib baru tersebut memiliki atribut NRP, NIP, TANGGAL dan PELANGGARAN
- Apakah atribut NRP, NIP dan TANGGAL tidak unik ? Ya, karena memang banyak data yang kembar
- Apakah atribut tersebut redundant ? Ya, karena pada atribut tersebut digunakan untuk menyimpan data yang memang maksudnya sama. Contoh NRP 1235 disimpan berulang-ulang
- Apakah 6ttrib tersebut perlu dibuatkan kunci yang bersifat unik ? Tidak perlu, selama 6ttrib PELANGGARAN dianggap memang tidak memerlukan kunci.
Kalau memang diinginkan suatu kunci ? Harus sudah dibuat sejak dirancang spesifikasi atau ERD, misalkan NO_PELANGGARAN
- Apakah 6ttrib tersebut perlu dinormalisasi (dipisahkan NRP, NIP dan TANGGAL menjadi 6ttrib baru) ? Tidak
 - ✓ Mengapa ? Karena atribut tersebut sudah berupa data yang dianggap sederhana dan mungkin tidak dapat disederhanakan lagi.
 - ✓ Contoh, apakah ada kunci lain sebagai pengganti NRP ? Kalau ada, kunci ini seharusnya sudah digunakan saat merancang spesifikasi atau ERD
 - ✓ Contoh, apakah ada data lain yang lebih sederhana untuk menggantikan tanggal ?
- Apakah atribut NRP, NIP, TANGGAL dan PELANGGARAN tidak unik ? Ya, karena memang banyak data yang kembar
- Khusus untuk NRP, NIP dan TANGGAL sudah dibahas pada penjelasan sebelumnya

- Apakah atribut PELANGGARAN berisi data yang redundant ? Ya, karena berisi data yang maksudnya sama
- Apakah 7ttrib tersebut perlu dinormalisasi dengan memisah atribut PELANGGARAN menjadi 7ttrib baru ? Bisa Ya, bisa Tidak.
- Jika data-data pada atribut PELANGGARAN dimaksudkan berupa data yang bersifat mandiri, berupa keterangan yang boleh ditulis bebas, tidak mengapa menuliskan dengan cara yang berbeda meskipun maksudnya sama, maka tidak perlu dilakukan normalisasi.
Contoh, tidak mengapa menuliskan pelanggaran, misalkan, "Memakai sandal", dan kadang ditulis "Memakai sandal jepit".
- Namun, jika diinginkan untuk jenis-jenis pelanggaran yang sama harus persis dituliskan sama, dan kelak ingin dilakukan analisa dengan cara mengelompokkan jenis pelanggaran, maka ini harus dilakukan normalisasi

2. Second Normal Form (2NF)

Ketergantungan fungsional secara penuh

Suatu atribut harus bergantung sepenuhnya dengan kunci utama, tidak boleh dengan kombinasi antara kunci utama dengan atribut lain

Contoh

Kuliah {Hari, Jam, Ruang MataKuliah}

Misalkan, ruang kuliah selalu sama pada satu hari, tetapi mata kuliah berbeda untuk jam kuliah tertentu

Ruang bergantung Hari (bergantung penuh)

MK bergantung Hari dan Jam (bergantung 7ttribut/tidak penuh/parsial)

3. Third Normal Form (3NF)

Ketergantungan fungsional langsung

Suatu atribut harus bergantung secara langsung dengan kunci utama, tanpa melalui perantara atribut lain (bergantung tidak langsung)

Contoh

Data Pegawai {NIP, NAMA, ALAMAT, LULUSAN, JURUSAN, KAJUR, SEKJUR}

Nama, Alamat, Lulusan, Jurusan bergantung langsung dengan NIP

Kajur dan Sekjur bergantung langsung dengan Jurusan dan bergantung tidak langsung dengan NIP

Catatan:

- Jika perancangan ERD dilakukan secara detil (segala aspek diperhitungkan), biasanya yang sering dilakukan hanya sampai NF1 (membuang redundansi), Sedangkan NF2 dan NF3 hampir tidak perlu dilakukan
- Jangan menambahkan atribut baru, misalkan atribut kunci, dengan alasan tidak ada kunci atau alasan lainnya. Lakukan penambahan kunci dan sebagainya pada tahap masih di ERD atau di spesifikasinya. Pada saat Pemetaan (mapping atau pembuatan skema), tidak diperkenankan menambahkan atribut baru, kecuali 8trib baru hasil dari relasi N:M atau relasi orde lebih dari 2 atau hasil dari Normalisasi.

4. Boyce-Codd Normal Form (BCNF)

- Bentuk BCNF terpenuhi dalam sebuah tabel, jika untuk setiap functional dependency terhadap setiap atribut atau gabungan atribut dalam bentuk:
 $X \rightarrow Y$ maka X adalah super key
- tabel tersebut harus di-dekomposisi berdasarkan functional dependency yang ada, sehingga X menjadi super key dari tabel-tabel hasil dekomposisi
- Setiap tabel dalam BCNF merupakan 3NF. Akan tetapi setiap 3NF belum tentu termasuk BCNF . Perbedaannya, untuk functional dependency $X \square A$, BCNF tidak membolehkan A sebagai bagian dari primary key.

5. Four Normal Form (4NF)

Bentuk normal keempat (4NF) adalah tingkat normalisasi basis data yang memastikan bahwa tabel tidak memiliki ketergantungan multi-nilai. Ketergantungan multi-nilai terjadi ketika satu atribut menentukan lebih dari satu atribut lainnya, dan atribut ini tidak tergantung satu sama lain. Misalnya, jika sebuah tabel menyimpan hobi dan keterampilan karyawan, dan setiap

karyawan dapat memiliki beberapa hobi dan keterampilan, maka terdapat ketergantungan multivalued antara ID karyawan dan hobi dan keterampilan tersebut. Untuk mencapai 4NF, kita perlu membagi tabel menjadi dua tabel terpisah, satu untuk hobi dan satu lagi untuk keterampilan, dan menautkannya dengan kunci asing. Manfaat 4NF adalah menghilangkan kemungkinan data yang tidak konsisten atau duplikat, dan membuat database lebih fleksibel dan efisien. Kelemahan 4NF adalah dapat menambah jumlah tabel dan gabungan,

Bentuk normal 4NF terpenuhi dalam sebuah tabel jika telah memenuhi bentuk BCNF, dan tabel tersebut tidak boleh memiliki lebih dari sebuah multivalued attribute. Untuk setiap multivalued dependencies (MVD) juga harus merupakan functional dependencies

6. Five Normal Form (5NF)

Bentuk normal kelima (5NF) adalah tingkat normalisasi database yang memastikan bahwa tabel tidak memiliki dependensi gabungan. Ketergantungan gabungan terjadi ketika tabel dapat didekomposisi menjadi dua atau lebih tabel yang lebih kecil, lalu direkonstruksi dengan menggabungkannya pada kunci utama, tanpa kehilangan informasi apa pun. Misalnya, jika tabel menyimpan produk, warna, dan ukuran yang dijual perusahaan, dan setiap produk dapat memiliki beberapa warna dan ukuran, maka ada ketergantungan gabungan antara ID produk, warna, dan ukuran. Untuk mencapai 5NF, kita perlu membagi tabel menjadi tiga tabel terpisah, satu untuk produk, satu untuk warna, dan satu untuk ukuran, dan menautkannya dengan kunci komposit. Manfaat 5NF adalah memastikan bahwa database sepenuhnya dinormalisasi dan tidak memiliki redundansi atau anomali.

Bentuk normal 5NF terpenuhi jika tidak dapat memiliki sebuah lossless decomposition menjadi tabel-tabel yg lebih kecil. Jika 4 bentuk normal sebelumnya dibentuk berdasarkan functional dependency, 5NF dibentuk berdasarkan konsep join dependence. Yakni apabila sebuah tabel telah di-dekomposisi menjadi tabel-tabel lebih kecil, harus bisa digabungkan lagi (join) untuk membentuk tabel semula.

Kapan menggunakan 4NF dan 5NF?

Keputusan untuk menggunakan 4NF dan 5NF bergantung pada sifat dan kompleksitas data serta persyaratan aplikasi. Secara umum, 4NF dan 5NF tidak terlalu umum dalam praktiknya, karena sebagian besar database dapat mencapai tingkat normalisasi yang memuaskan dengan menerapkan tiga bentuk normal pertama (1NF, 2NF, dan 3NF). Namun, mungkin ada beberapa kasus di mana 4NF dan 5NF berguna atau diperlukan, seperti saat menangani data yang memiliki banyak dimensi, atribut, atau nilai, atau saat memastikan tingkat kualitas dan konsistensi data tertinggi. Dalam kasus seperti itu, 4NF dan

5NF dapat membantu menghindari anomali data, meningkatkan kinerja kueri, dan menyederhanakan manipulasi data. Namun, 4NF dan 5NF juga memiliki beberapa kompromi, seperti menambah jumlah tabel dan gabungan, memperumit desain dan pemeliharaan database, dan membutuhkan lebih banyak ruang penyimpanan dan daya pemrosesan.

NORMALISASI DATA

Normalisasi merupakan sebuah teknik dalam logical desain sebuah basis data yang mengelompokkan atribut dari suatu relasi sehingga membentuk struktur relasi yang baik (tanpa redundansi).

- Normalisasi adalah proses pembentukan struktur basis data sehingga sebagian besar *ambiguity* bisa dihilangkan.

Tujuan Normalisasi

- Untuk mencegah kerangkapan data.
- Untuk mengurangi kompleksitas.
- Untuk mempermudah pemodifikasian data.

Proses Normalisasi

- Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat.
- Apabila tabel yang diuji belum memenuhi persyaratan tertentu, maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal.

Tahapan Normalisasi

- Tahapan Normalisasi dimulai dari tahap paling ringan (1NF) hingga yang paling ketat (5NF).
- Biasanya hanya sampai tingkat 3NF atau BCNF karena sudah cukup memadai untuk menghasilkan tabel-tabel yang berkualitas baik.
- Urutan : 1NF, 2NF, 3NF, BCNF, 4NF, 5NF.



Sebuah tabel dikatakan baik (efisien) atau normal jika memenuhi 3 kriteria sebagai berikut :

1. Jika ada dekomposisi (penguraian) tabel, maka dekomposisinya harus dijamin aman (*Lossless-Join Decomposition*). Artinya, setelah tabel tersebut diuraikan / didekomposisi menjadi tabel-tabel baru, tabel-tabel baru tersebut bisa menghasilkan tabel semula dengan sama persis.
2. Terpeliharanya ketergantungan fungsional pada saat perubahan data (*Dependency Preservation*).
3. Tidak melanggar Boyce-Codd Normal Form (BCNF).

Jika kriteria ke tiga (BCNF) tidak dapat terpenuhi, maka paling tidak tabel tersebut tidak melanggar bentuk normal tahap ketiga (3rd Normal Form / 3NF).

relasi tidak normal (Tabel Personil)

ID_Personil	Tanggal Lahir	Karakteristik
I102	17 Januari 1970	Tinggi 162
		Berat 50
		Rambut hitam
A212	12 Desember 1966	Tinggi 170
		Berat 64
		Rambut hitam

Relasi Normal (Tabel Personil)

ID Personil	Tanggal lahir	Tinggi	Berat	Warna Rambut
I102	17 Januari 1970	162	50	Hitam
A212	12 Desember 1966	170	64	Hitam

Tabel Universal

Tabel Universal (*Universal / Star Table*) merupakan sebuah tabel yang merangkum semua kelompok data yang saling berhubungan, bukan merupakan tabel yang baik. Misalnya :

No-Mhs	Nm-Mhs	Jurusan	Kd-MK	Nama-MK	Kd-Dosen	Nm_Dosen	Nilai
2683	Welli	MI	MI350	Manajemen Basis Data	B104	Ati	A
			MI465	Analisis Pre. Sistem	B317	Dita	B
5432	Bakri	AK	MI350	Manajemen Basis Data	B104	Ati	C
			AKN201	Akuntansi Keuangan	D310	Lia	B
			MKT300	Dasar Pemasaran	B212	Lola	A

Functional Dependency

- Notasi : A -> B
A dan B adalah atribut dari sebuah tabel. Berarti secara fungsional A menentukan B atau B tergantung pada A, jika dan hanya jika ada 2 baris data dengan nilai A yang sama, maka nilai B juga sama.
- Notasi : A x-> B
Adalah kebalikan dari notasi sebelumnya.

Contoh tabel nilai :

<u>Namakul</u>	<u>Nrp</u>	<u>namaMhs</u>	<u>NiHuruf</u>
Struktur Data	980001	Ali Akbar	A
Struktur Data	980004	Indah Susanti	B
Basis Data	980001	Ali Akbar	
Basis Data	980002	Budi Haryanto	
Basis Data	980004	Indah Susanti	
Bahasa Indonesia	980001	Ali Akbar	B
Matematika I	980002	Budi Haryanto	C

Functional Dependency dari tabel nilai :

- Nrp -> namaMhs
Karena untuk setiap nilai nrp yang sama, maka nilai namaMhs juga sama.
- {Namakul, nrp} -> NiHuruf
Karena atribut NiHuruf tergantung pada Namakul dan nrp secara bersama-sama. Dalam arti lain, untuk Namakul dan nrp yang sama, maka NiHuruf juga sama, karena Namakul dan nrp merupakan key (bersifat unik).
- Namakul x-> nrp
- Nrp x-> NiHuruf

Contoh FD 1

- Andaikan ada tabel :
NILAI (NIM, Nm-mk, Semester, Nilai).
- Atribut kunci : NIM, Nm-mk, Semester.
- Maka Functional Dependency :
NIM, Nm-mk, Semester -> Nilai

Bentuk Bentuk Normal

1. Bentuk Normal Tahap Pertama (1st Normal Form / 1NF).
2. Bentuk Normal Tahap Kedua (2nd Normal Form / 2NF).
3. Bentuk Normal Tahap Ketiga (3rd Normal Form / 3NF).
4. Boyce-Codd Normal Form (BCNF).
5. Bentuk Normal Tahap Keempat (4th Normal Form / 4NF).
6. Bentuk Normal Tahap Kelima (5th Normal Form / 5NF).

Normal Pertama (1st Normal Form)

Aturan :

- Tidak adanya **atribut multi-value**, **atribut komposit** atau kombinasinya.

- Mendefinisikan atribut kunci.
- Setiap atribut dalam tabel tersebut harus bernilai *atomic* (tidak dapat dibagi-bagi lagi).

Contoh 1 (atribut multi-value)

Misal data mahasiswa sbb:

Nrp	nama	Hobi
12020001	Heri Susanto	Sepakbola, membaca komik, berenang
12020013	Siti Zulaiha	Memasak, mrogram komputer
12020015	Dini Susanti	Menjahit, membuat roti

Atau:

Nrp	nama	hobi1	hobi2	Hobi3
12020001	Heri Susanto	Sepak Bola	Membaca komik	berenang
12020013	Siti Zulaiha	Memasak	mrogram komputer	
12020015	Dini Susanti	Menjahit	membuat kue	

Tabel-tabel di atas tidak memenuhi syarat 1NF

Didekomposisi menjadi:

➤ **Tabel Mahasiswa**

Nrp	Nama
12020001	Heri Susanto
12020013	Siti Zulaiha
12020015	Dini Susanti

➤ **Tabel Hobi**

Nrp	Hobi
12020001	Sepakbola
12020001	membaca komik
12020001	Berenang
12020013	Memasak
12020013	mrogram komputer
12020015	Menjahit
12020015	membuat roti

Contoh 2 (Composite)

- Dimana nilai pada atribut jadwal berisi gabungan antara Hari dan Jam.
- Jika asumsi hari dan jam memegang peranan penting dalam sistem basis data, maka atribut Jadwal perlu dipisah sehingga menjadi JadwalHari dan JadwalJam sbb:

Kodekul	NamaKul	Dosen	Kelas	JadwalHari	JadwalJam
---------	---------	-------	-------	------------	-----------

Normalisasi	Kedua	(2nd	Normal	Form)
Aturan				:

- Sudah memenuhi dalam bentuk normal kesatu (1NF).
- Semua atribut bukan kunci hanya boleh tergantung (functional dependency) pada atribut kunci.
- Jika ada **ketergantungan parsial** maka atribut tersebut harus dipisah pada tabel yang lain.
- Perlu ada tabel penghubung ataupun kehadiran foreign key bagi atribut-atribut yang telah dipisah tadi.

Contoh :

Tabel berikut memenuhi 1NF tapi tidak termasuk 2NF:

Mhs_nrp	mhs_nama	mhs_alamat	mk_kode	mk_nama	mk_sks	nihuruf
---------	----------	------------	---------	---------	--------	---------

- Tidak memenuhi 2NF, karena {Mhs_nrp, mk_kode} yang dianggap sebagai primary key sedangkan:
{Mhs_nrp, mk_kode} ↗ mhs_nama
{Mhs_nrp, mk_kode} ↗ mhs_alamat
{Mhs_nrp, mk_kode} ↗ mk_nama
{Mhs_nrp, mk_kode} ↗ mk_sks
{Mhs_nrp, mk_kode} → nihuruf
- Tabel di atas perlu didekomposisi menjadi beberapa tabel yang memenuhi syarat 2NF

- Setiap tabel dalam BCNF merupakan 3NF. Akan tetapi setiap 3NF belum tentu termasuk BCNF. Perbedaanannya, untuk functional dependency $X \rightarrow A$, BCNF tidak membolehkan A sebagai bagian dari primary key

Bentuk Normal Tahap Keempat (4th Normal Form / 4NF)

- Bentuk normal 4NF terpenuhi dalam sebuah tabel jika memenuhi bentuk BCNF, dan tabel tersebut tidak boleh memiliki lebih dari sebuah *multivalued attribute*.
- Untuk setiap *multivalued attribute* (MVD) juga harus merupakan *functional dependencies*.

Contoh :

Misal, tabel berikut tidak memenuhi 4NF:

Employee	Project	Skill
Jim	11	Program
Mary	5	Design
Mary	NULL	Analysis

Setiap employee dapat bekerja di lebih dari project dan dapat memiliki lebih dari satu skill. Untuk kasus seperti ini tabel tersebut harus di-dekomposisi menjadi:

(Employee, Project)
(Employee, Skill)

Bentuk Normal Tahap Kelima (5th Normal Form / 5NF)

- Bentuk normal 5NF terpenuhi jika tidak dapat memiliki *lossless decomposition* menjadi tabel-tabel yang lebih kecil.
- Jika 4 bentuk normal sebelumnya dibentuk berdasarkan *functional dependencies*, 5NF dibentuk berdasarkan konsep *join dependencies*. Yakni apabila sebuah tabel telah di-dekomposisi menjadi tabel-tabel lebih kecil, harus bisa digabungkan lagi (join) untuk membentuk tabel semula.

Studi Kasus Normalisasi Data

NoProyek	NamaProyek	NoPegawai	NamaPegawai	Golongan	BesarGaji
NP001	BBR	Peg01	Anton	A	1.000.000
		Peg02	Paula	B	900.000
		Peg06	Koko	C	750.000
NP002	PEMDA	Peg01	Anton	A	1.000.000
		Peg12	Sita	B	900.000
		Peg14	Yusni	B	900.000

Untuk mendapatkan hasil yang paling normal, maka proses normalisasi dimulai dari normal pertama. Field-field di atas yang merupakan group berulang : NoPegawai, NamaPegawai, Golongan, BesarGaji.

Normalisasi Pertama :

Solusinya hilangkan duplikasi dengan mencari ketergantungan parsial. Menjadikan field-field menjadi tergantung pada satu atau beberapa field. Karena yang dapat dijadikan kunci adalah *NoProyek* dan *NoPegawai*, maka langkah kemudian dicari field-field mana yang tergantung pada *NoProyek* dan mana yang tergantung pada *NoPegawai*.

<i>NoProyek</i>	NamaProyek	<i>NoPegawai</i>	NamaPegawai	Golongan	BesarGaji
NP001	BRR	Peg01	Anton	A	1.000.000
NP001	BRR	Peg02	Paula	B	900.000
NP001	BRR	Peg06	Koko	C	750.000
NP002	PEMDA	Peg01	Anton	A	1.000.000
NP002	PEMDA	Peg12	Sita	B	900.000
NP002	PEMDA	Peg14	Yusni	B	900.000

Normalisasi Kedua :

Field-field yang tergantung pada satu field haruslah dipisah dengan tepat, misalnya *NoProyek* menjelaskan *NamaProyek* dan *NoPegawai* menjelaskan *NamaPegawai*, *Golongan* dan *BesarGaji*.

TABEL PROYEK

<i>NoProyek</i>	NamaProyek
NP001	BRR
NP002	PEMDA

TABEL PEGAWAI

<i>NoPegawai</i>	NamaPegawai	Golongan	BesarGaji
Peg01	Anton	A	1.000.000
Peg02	Paula	B	900.000
Peg06	Koko	C	750.000
Peg12	Sita	B	900.000
Peg14	Yusni	B	900.000

Untuk membuat hubungan antara dua tabel, dibuat suatu tabel yang berisi key-key dari tabel yang lain.

TABEL PROYEKPEGAWAI

<i>NoProyek</i>	<i>NoPegawai</i>
NP001	Peg01
NP001	Peg02
NP001	Peg06
NP002	Peg01
NP002	Peg12
NP002	Peg14

Normalisasi Ketiga :

- Pada tabel di atas masih terdapat masalah, bahwa *BesarGaji* tergantung pada *Golongan* nya. Padahal di sini *Golongan* bukan merupakan field kunci.
- Artinya kita harus memisahkan field non-kunci *Golongan* dan *BesarGaji* yang tadinya tergantung secara parsial kepada field kunci *NoPegawai*, untuk menghilangkan ketergantungan transitif.

TABEL PROYEK

<i>NoProyek</i>	<i>NamaProyek</i>
NP001	BRR
NP002	PEMDA

TABEL PROYEKPEGAWAI

<i>NoProyek</i>	<i>NoPegawai</i>
NP001	Peg01
NP001	Peg02
NP001	Peg06
NP002	Peg01
NP002	Peg12
NP002	Peg14

TABEL PEGAWAI

<i>NoPegawai</i>	<i>NamaPegawai</i>	<i>Golongan</i>
Peg01	Anton	A
Peg02	Paula	B
Peg06	Koko	C
Peg12	Sita	B
Peg14	Yusni	B

TABEL GOLONGAN

<i>Golongan</i>	<i>BesarGaji</i>
A	1.000.000
B	900.000
C	750.000

Contoh Lain Studi Kasus Normalisasi

<u>No-Mhs</u>	<u>Nm-Mhs</u>	<u>Jurusan</u>	<u>Kd-MK</u>	<u>Nama-MK</u>	<u>Kd-Dosen</u>	<u>Nm Dosen</u>	<u>Nilai</u>
2683	Weti	MI	MI350	Manajemen Basis Data	B104	Ati	A
			MI465	Analisis Pro. Sistem	B317	Data	B
5432	Bakri	AK	MI350	Manajemen Basis Data	B104	Ati	C
			AKN201	Akuntansi Keuangan	D310	Lia	B
			MKT300	Dasar Pemasaran	B212	Lola	A

No-Mhs	Nm-Mhs	Jurusan	Kd-MK	Nama-MK	Kd-Dosen	Nm. Dosen	Nilai
2683	Weli	MI	MI350	Manajemen Basis Data	B104	Ati	A
2683	Weli	MI	MI465	Analisis Proc. Sistem	B317	Dita	B
5432	Bakri	AK	MI350	Manajemen Basis Data	B104	Ati	C
5432	Bakri	AK	AKN201	Akuntansi Keuangan	D310	Lia	B
5432	Bakri	AK	MKT300	Dasar Pemasaran	B212	Lola	A

2 NF

No-Mhs	Nama-Mhs	Jurusan
2683	Weli	MI
5432	Bakri	AK

Tabel Mahasiswa

Kode-MK	Nama-MK	Kode-Dosen	Nama-Dosen
MI350	Manajemen Basis Data	B104	Ati
MI465	Analisis Proc. Sistem	B317	Dita
AKN201	Akuntansi Keuangan	D310	Lia
MKT300	Dasar Pemasaran	B212	Lola

Tabel Kuliah

No-Mhs	Kode MK	Nilai
2683	MI350	A
2683	MI465	B
5432	MI350	C
5432	AKN201	B
5432	MKT300	A

Tabel Nilai

3 NF

<u>Kode-MK</u>	Nama-MK	Kode-Dosen
MI350	Manajemen Basis Data	B104
MI465	Analisis Proc. Sistem	B317
AKN201	Akuntansi Keuangan	D310
MKT300	Dasar Pemasaran	B212

Tabel Mata Kuliah

<u>Kode-Dosen</u>	Nama-Dosen
B104	Ati
B317	Dita
D310	Lia
B212	Lola

Tabel Dosen