**MATERI DESAIN BASIS DATA**

**NORMALISASI DAN DENORMALISASI DATA**

**Dosen Pengampu :**

**Ainatul Mardhiyah, S.Kom, M.Cs**



**By:**

Hari Robiansyah (11650099)

Dwi umi wahyuni (13650036)

Riza Anugrah (13650038)

Faisal Briliansyah (13650056)

Kienna Candra S. (13650067)

Aulia Setianti W (13650084)

**JURUSAN TEKNIK INFORMATIKA**

**UIN MALIKI MALANG**

**2014**

**NORMALISASI**

* Memahami Normalisasi

Normalisasi merupakan suatu proses untuk mengubah suatu tabel yang memiliki masalah tertentu ke dalam dua buah tabel atau lebih yang tidak lagi memiliki masalah tersebut (Abdul Kadir, 2008). Ada juga yang berpendapat normalisasi adalah proses penyusunan tabel – tabel yang tidak redundan (dobel), yang dapat menyebabkan anomali yaitu proses basis data yang memberikan efek samping yang tidak diharapkan (Misalnya menyebabkan data tidak konsisten atau membuat data hilang ketika data yang lain dihapus).

Normalisasi digunakan sebagai teknik analisis data pada database, sehingga dapat diketahui apakah pembuatan tabel – tabel yang terelasi dalam database itu sudah baik. Kondisi sudah baik yaitu suatu kondisi pada saat proses insert, update, delete dan modifikasi pada satu atau beberapa atribut suatu tabel tidak berpengaruh terhadap integritas data yang lain dalam satu hubungan relasi database.

* Langkah – langkah Normalisasi

Langkah – langkah Normalisasi terdiri dari :

*1. Unnormalization Form*

Bentuk yang tidak normal dimaksudkan suatu kumpulan data yang akan diolah yang diperoleh dari format – format yang beraneka ragam, masih terdapat duplikasi, bisa saja tidak sempurna atau tidak lengkap, dan sesuai fakta lapangan.

Bentuk ini didapat dari dokumen yang ada dilapangan atau manual dengan atribut bukan nilai sederhana.

2. *First Normal Form* (1NF)

Suatu tabel dianggap normal ke satu (1NF) jika :

- Tidak terdapat baris yang bernilai ganda atau duplikat.

- Masing - masing baris bernilai tunggal dan tidak bernilai null. Langkah – langkah :

- Isikan setiap data bernilai tunggal dan tidak null

- Membuang perulangan data dalam satu baris dengan baris yang lain.

3. *Second Normal Form* (2NF)

Bentuk normal kedua (2NF) terpenuhi jika :

- Harus telah berbentuk normal pertama (1NF).

- pada sebuah tabel semua atribut yang tidak termasuk dalam primary key memiliki ketergantungan fungsional pada primary key secara utuh. Suatu atribut dikatakan ketergantungan fungsional jika harga pada atribut tersebut menentukan harga dari atribut yang lain. Misalnya, nim → mhs\_nama.

Langkah – langkah :

- Jika terdapat atribut yang bergantung terhadap atribut bukan kunci utama dan merupakan atribut kunci maka pecah menjadi table baru.

4. *Third Normal Form* (3NF)

Bentuk normal ketiga (3NF) terpenuhi jika :

- Harus telah berbentuk normal kedua (2NF).

- Tidak terdapat anomali – anomali hasil dari ketergantungan transitif. ketergantungan transitif adalah ketergantungan fungsional antara 2 atau lebih atribut bukan kunci.

Langkah – langkah :

- Pastikan semua atribut non kunci bergantung penuh terhadap atribut kunci.

- Pisahkan menjadi tabel baru jika menemukan ketergantungan transitif dalam tabel tersebut.

5. *Boyce Codd Normal Form* (BCNF)

Secara praktis tujuan analisis database cukup sampai pada 3NF, Akan tetapi dalam suatu kasus tertentu lebih baik bila dapat mencapat BCNF. Beberapa pemikir menyamakan antara 3NF dengan BCNF. Bentuk normal BCNF terpenuhi jika :

- Masing-masing atribut utama bergantung fungsional penuh pada masing kunci dimana kunci tersebut bukan bagiannya.

- Setiap determinan atribut-atribut relasi adalah kunci relasi atau kandidat kunci.

- BCNF dapat memiliki lebih dari satu kunci.

- BCNF hampir sama dengan 3NF atau bisa dibilang sama dengan 3NF. Langkah – langkah :

- Hilangkan dependensi pada bukan kunci kandidat.

6. *Fourth Normal Form* (4NF)

Dilakukan jika terdapat anomali pada (3NF)

7. *Fifth Normal Form* (5NF)

Langkah ini untuk memecah relasi menjadi dua sehingga relasi tersebut tidak digabungkan kembali manjadi satu dan jika terdapat anomali pada (5NF)

* Contoh permasalahan dalam Normalisasi

Terdapat suatu hasil kartu studi yang bersifat manual dan akan kita anilisis menggunakan teknik Normalisasi sebagai berikut :

**KARTU HASIL STUDI**

|  |  |  |  |
| --- | --- | --- | --- |
| Nomor Induk | : 055 | Jurusan | : Teknik Elektro |
| Nama | : Fuad | Fakultas | : SAINTEK |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Kode** | **Nama Matakuliah** | **Kode Dosen** | **Ruang** | **Nilai** |
| 1. | 021 | Pemrograman | D01 | B202 | B |
| 2. | 022 | Basis Data | D02 | B304 | A |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **KARTU HASIL STUDI**  Nomor Induk : 057 Jurusan : Teknik Informatika  Nama : Neneng Fakultas : SAINTEK | | | | | | | |
|  | **No.** | **Kode** | **Nama Matakuliah** | **Kode Dosen** | **Ruang** | **Nilai** |  |
| 1. | 021 | Pemrograman | D01 | B201 | A |
| 2. | 022 | Basis Data | D02 | B305 | C |
| 3. | 026 | Prak. Digital | D05 | B307 | B |

Tahap anilisis dengan Normalisasi :

- **Tahap *Unnormalization Form***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| NIM | Mah\_nama | Jurusan | Fakultas | Mat\_kode | Mat\_nama | Dos\_kode | Ruang | Nilai |
| 055 | Fuad | Sistem Informasi | SAINTEK | 021 | Pemrograman | D01 | B202 | B |
|  |  |  |  | 022 | Basis Data | D02 | B304 | A |
| 057 | Neneng | Teknik Informatika | SAINTEK | 021 | Pemrograman | D01 | B201 | A |
|  |  |  |  | 022 | Basis data | D02 | B305 | C |
|  |  |  |  | 026 | Prak. Digital | D05 | B307 | B |

- **Tahap *First Normal Form* (1NF)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **NIM** | **Nama\_mhs** | **Jurusan** | **Fakultas** | **Kode\_mk** | **Nama\_mk** | **Kode\_dsn** | **Ruang** | **Nilai** |
| **055** | Fuad | Sistem Informasi | SAINTEK | 021 | Pemrograman | D01 | B202 | B |
| **055** | Fuad | Sistem Informasi | SAINTEK | 022 | Basis Data | D02 | B304 | A |
| **057** | Neneng | Teknik Informatika | SAINTEK | 021 | Pemrograman | D01 | B201 | A |
| **057** | Neneng | Teknik Informatika | SAINTEK | 022 | Basis data | D02 | B305 | C |
| **057** | Neneng | Teknik Informatika | SAINTEK | 026 | Prak. Digital | D05 | B307 | B |

- **Tahap *Second Normal Form* (2NF)**

Tabel Mahasiswa

|  |  |  |  |
| --- | --- | --- | --- |
| NIM\* | Nama\_mhs | Jurusan | Fakultas |
| 055 | Fuad | Sistem Informasi | SAINTEK |
| 057 | Neneng | Teknik Informatika | SAINTEK |

Tabel Matakuliah Tabel Nilai

|  |  |  |  |
| --- | --- | --- | --- |
| NIM (fk) | Mat\_kode (fk) | Ruang | Nilai |
| 055 | 021 | B202 | B |
| 055 | 022 | B304 | A |
| 057 | 021 | B201 | A |
| 057 | 022 | B305 | C |
| 057 | 026 | B307 | B |

|  |  |  |
| --- | --- | --- |
| Mat\_kode\* | Mat\_nama | Dos\_kode |
| 021 | Pemrograman | D01 |
| 022 | Basis Data | D02 |
| 026 | Prak. Digital | D05 |

- **Tahap *Third Normal Form* (3NF)**

Tabel Matakuliah

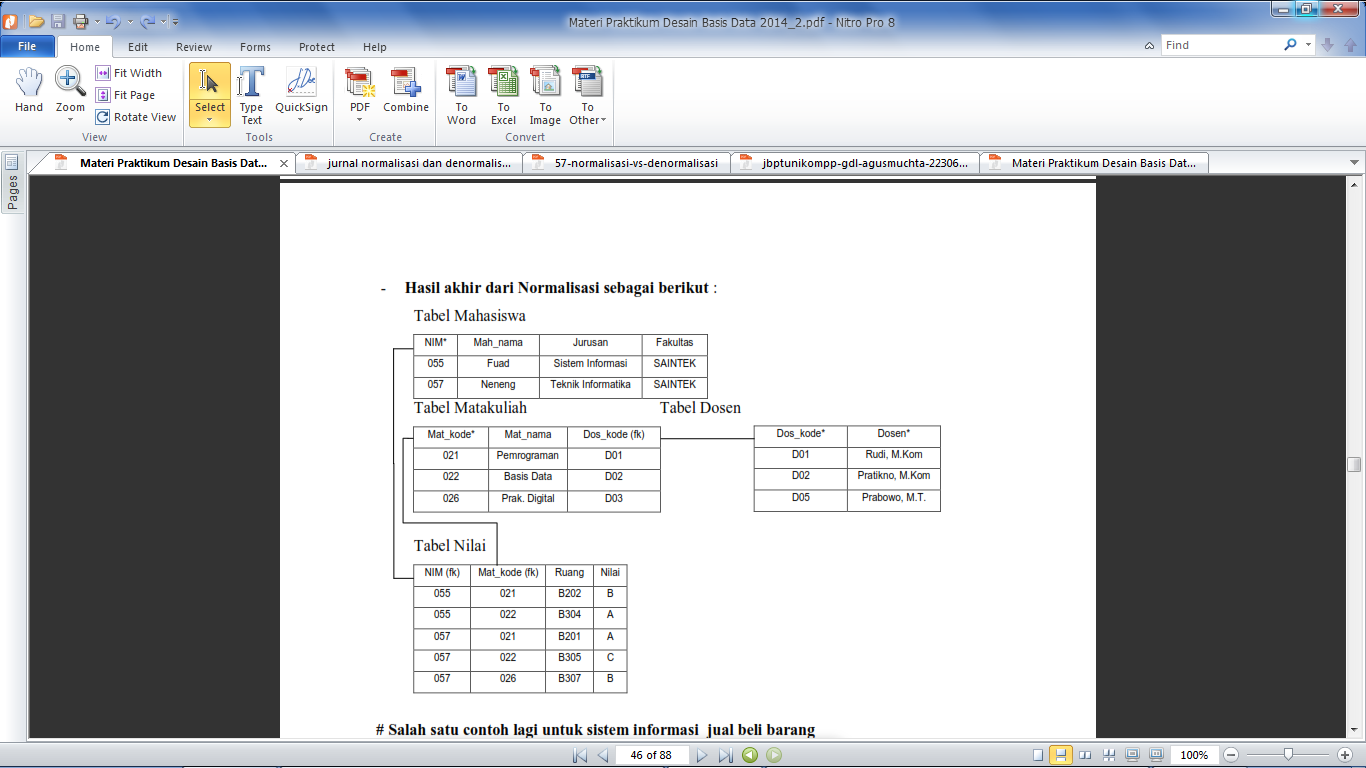
|  |  |  |
| --- | --- | --- |
| Mat\_kode\* | Mat\_nama | Dos\_kode (fk) |
| 021 | Pemrograman | D01 |
| 022 | Basis Data | D02 |
| 026 | Prak. Digital | D03 |

Tabel Dosen

|  |  |
| --- | --- |
| Dos\_kode\* | Dosen |
| D01 | Rudi, M.Kom |
| D02 | Pratikno, M.Kom |
| D03 | Prabowo, M.T. |

Biasanya ketika mencapai pada tahap *Third Normal Form* (3NF) sudah memenuhi syarat untuk digunakan dalam pembuatan aplikasi database. Namun apabila masih terdapat anomali maka bisa dilanjutkan ke tahap berikutnya sampai tidak terdapat anomali.

- **Hasil akhir dari Normalisasi sebagai berikut** :

Tabel Mahasiswa

|  |  |  |  |
| --- | --- | --- | --- |
| NIM\* | Mah\_nama | Jurusan | Fakultas |
| 055 | Fuad | Sistem Informasi | SAINTEK |
| 057 | Neneng | Teknik Informatika | SAINTEK |

Tabel Matakuliah Tabel Dosen

Mat\_kode\* Mat\_nama Dos\_kode (fk)

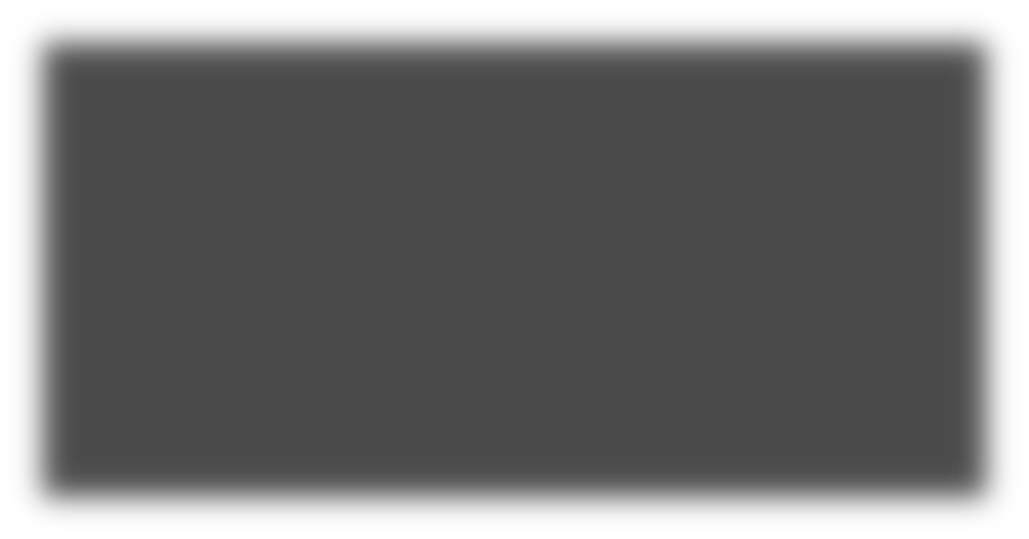
021 Pemrograman D01

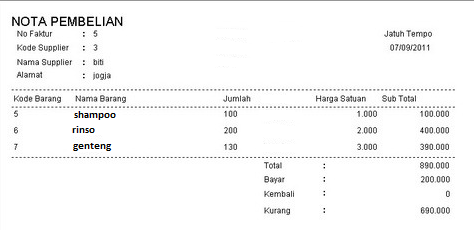
022 Basis Data D02

026 Prak. Digital D03

Dos\_kode\* Dosen\* D01 Rudi, M.Kom D02 Pratikno, M.Kom

D05 Prabowo, M.T.





**# Salah satu contoh lagi untuk sistem informasi jual beli barang**

**Dari nota diatas dibentuk menjadi *Unnormalization Form***

Tabel Nilai

|  |  |  |  |
| --- | --- | --- | --- |
| NIM (fk) | Mat\_kode (fk) | Ruang | Nilai |
| 055 | 021 | B202 | B |
| 055 | 022 | B304 | A |
| 057 | 021 | B201 | A |
| 057 | 022 | B305 | C |
| 057 | 026 | B307 | B |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No\_faktur** | **Kode\_supp** | **Nama\_supp** | **Kota** | **Kode\_brg** | **Nama\_brg** | **tanggal** | **jml** | **Satuan** | **Sub\_total** | **Total** |
| **5** | **3** | **biti** | **Malang** | **5** | **shampoo** | **07/09/2011** | **100** | **1000** | **100000** | **890000** |
|  |  |  |  | **6** | **Rinso** | **07/09/2011** | **200** | **2000** | **400000** | **890000** |
|  |  |  |  | **7** | **genteng** | **07/09/2011** | **130** | **3000** | **390000** | **890000** |
| **6** | **4** | **rodi** | **nganjuk** | **9** | **Paku** | **09/10/2013** | **100** | **100** | **10000** | **10000** |

- **Tahap *First Normal Form* (1NF)**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No\_faktur** | **Kode\_supp** | **Nama\_supp** | **Kota** | **Kode\_brg** | **Nama\_brg** | **tanggal** | **jml** | **Satuan** | **Sub\_total** | **Total** |
| **5** | **3** | **biti** | **Malang** | **5** | **shampoo** | **07/09/2011** | **100** | **1000** | **100000** | **890000** |
| **5** | **3** | **biti** | **Malang** | **6** | **Rinso** | **07/09/2011** | **200** | **2000** | **400000** | **890000** |
| **5** | **3** | **biti** | **Malang** | **7** | **genteng** | **07/09/2011** | **130** | **3000** | **390000** | **890000** |
| **6** | **4** | **rodi** | **nganjuk** | **9** | **Paku** | **09/10/2013** | **100** | **100** | **10000** | **10000** |

Adapun kelemahan dari 1NF ini sebagai berikut :

1. Inserting (Memasukkan data)

Kita tidak dapat memasukkan kode dan nama supplier saja tanpa ada transaksi pembelian, sehingga supplier baru dapat masuk bila ada transaksi pembelian.

2. Deleting (Menghapus data)

Bila satu baris data transaksi dihapus, maka akan berakibat menghapus pada data supplier. Padahal data suplier masih diperlukan pada transaksi berikutnya. Misalkan transaksi di hapus berdasarkan no\_faktur =6, maka akan menghapus supplier=4(rodi).

3. Updating (Mengubah data)

Kode dan nama pada supplier terlihat tertulis berulang-ulang 3(biti). Jika kita ingin merubah nama supplier tersebut maka harus mengganti disemua baris yang mengandung supplier

3(biti), jika ada yang terlewat data tidak konsisten lagi.

Karena masih terdapat kesalahan maka perlu dilakukan tahap berikutnya.

- **Tahap *Second Normal Form* (2NF)**

|  |
| --- |
| Supplier |
| kode\_supp\* |
| nama\_supp |
| kota |

Pembentukan bentuk normal kedua ini dilakukan dengan mencari kunci-kunci field yang dapat dipakai sebagai patokan dalam pencarian dan sifatnya unik. Berdasarkan kondisi di atas dapat diambil kunci kandidat yaitu : No\_faktur, Kode\_supplier, dan kode\_barang.

Barang kode\_brg\* nama\_brg

|  |
| --- |
| Transaksi |
| No\_faktur |
| Jatuh\_tempo |
| Jml |
| Satuan |
| Total |
| kode\_supp\*\* |
| kode\_brg\*\* |

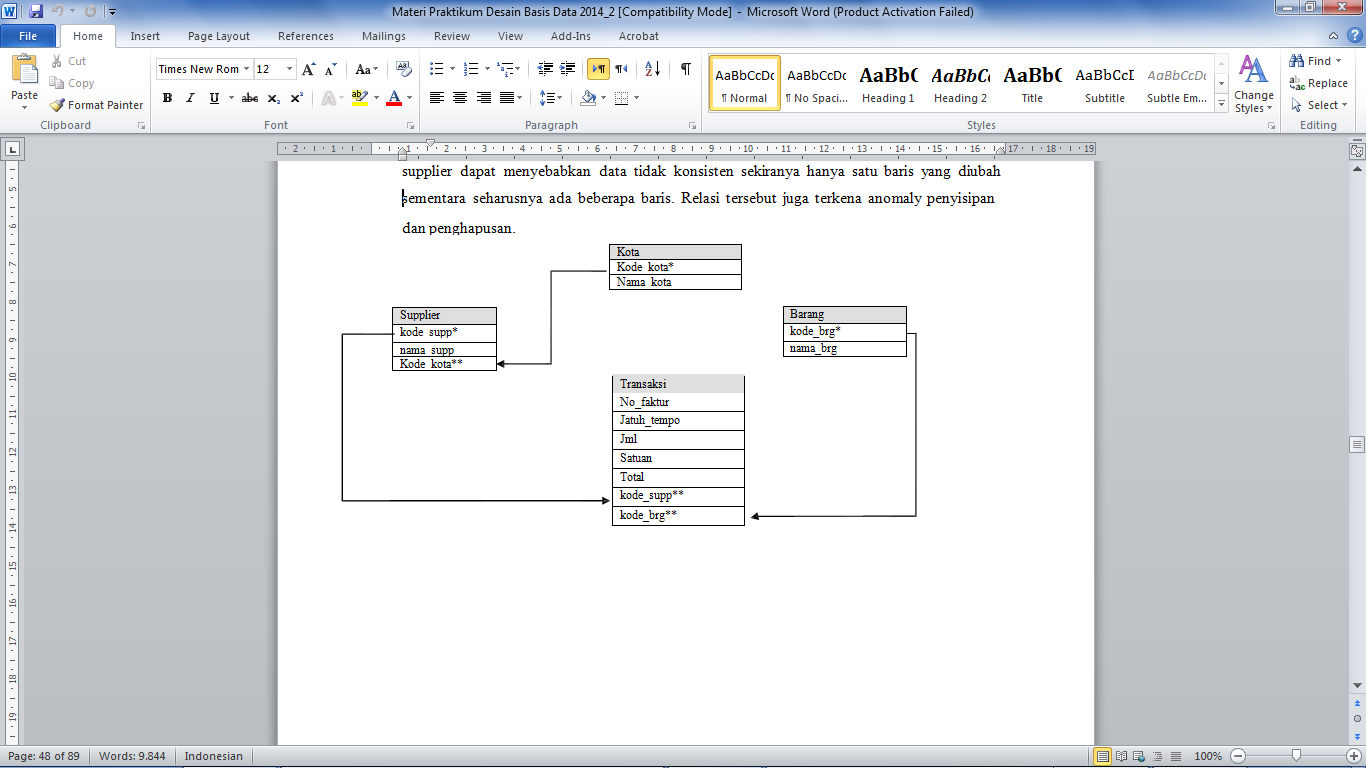
- **Tahap *Third Normal Form* (3NF)**

Syarat :

Setiap atribut yang bukan kunci harus bergantung hanya pada atribut kunci (primary key) secara menyeluruh. Hilangkan juga anomali – anomali yang masih mempunyai ketergantungan fungsional. Pada tabel supplier terdapat kolom kota, perubahan kota milik supplier dapat menyebabkan data tidak konsisten sekiranya hanya satu baris yang diubah sementara seharusnya ada beberapa baris. Relasi tersebut juga terkena anomaly penyisipan

dan penghapusan.

|  |
| --- |
| Supplier |
| kode\_supp\* |
| nama\_supp |
| Kode\_kota\*\* |



Sumber:  
Mater Praktikum Desain Basis Data Bab V Normalisasi

**Pengertian Denormalisasi Data**

Denormalisasi merupakan proses yang dilakukan pada database yang sudah dinormalisasi, dengan cara memodifikasi struktur tabel dan mengabaikan kerangkapan data (yang terkontrol) untuk meningkatkan kinerja database.

       Denormalisasi database adalah pelanggaran aturan normalisasi atau menjabarkan suatu tataan database yang telah normal untuk meningkatkan performa pengaksesan data pada database. Database yang telah normal disini dimaksudkan database yang redundansi datanya minim sehingga data yang disimpan tidak mengalami kerancuan dalam proses pengaksesan.

• Pelanggaran Normalisasi basis data disebut dengan Denormalisasi.

• Satu-satunya alasan mengapa kita boleh melakukan denormalisasi adalah pertimbangan performansi.

• Jika performansi yang kita peroleh bisa menjadi jauh lebih baik, maka normalisasi yang telah dilakukan cukup beralasan untuk dilanggar.

**Tahapan dalam Denormalisasi Data**

Denormalisasi dapat menggunakan dua cara yaitu :

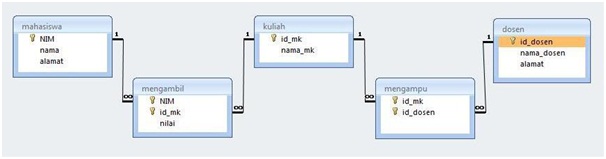
      Dengan pembuatan kolom baru pada tabel / mengabungkan kolom pada tabel satu dengan yang lain.

      Dengan pembuatan tabel baru

Cara yang pertama dilakukan apabila data yang didenormalisasi hanya kecil dan digunakan untuk mempermudah pengaksesan data apabila diakses dalam satu tabel. Sedangkan yang kedua dilakukan apabila data yang terdapat dalam tabel tersebut merupakan rangkuman / rekapitulasi dari satu atau beberapa tabel yang pengaksesannya terpisah dari tabel yang ada.

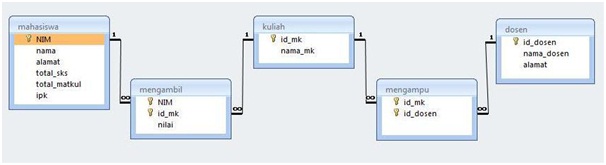
**Contoh Penerapan Denormalisasi Data**

Contoh Denormalisasi data

[](http://2.bp.blogspot.com/-Prqku4Bzy6I/UT3eocvNSoI/AAAAAAAAAHk/6DFQ-m-Bz40/s1600/1.jpg)Atribut turunan

Dari contoh diatas, bagaimana jika data mahasiswa sangat banyak sedangkan proses untuk menampilkan jumlah matakuliah, jumlah sks ataupun IPK sering dilakukan, akan memerlukan banyak waktu.

Untuk itu perlu ditambahkan atribut baru pada tabel mahasiswa (total\_sks,total\_matkul,ipk)

[](http://2.bp.blogspot.com/-HkmC5QQglJc/UT3fOBGHyxI/AAAAAAAAAHs/XiWqhAYaxog/s1600/1.jpg)

Bentuk-bentuk Denormalisasi

• Atribut yang terderivasi (atribut turunan)

• Atribut yang berlebihan

• Tabel rekapitulasi (summary table)

Atribut turunan

• Atribut yang nilainya bisa diperoleh dari nilai- nilai yang sudah ada pada atribut lain

Atribut turunan

Atribut yang berlebihan

• Atribut terkodekan (encoded attribute)

• Atribut gabungan (concatenated attribute)

• Atribut tumpang-tindih (overlapping attribute)

• Atribut bermakna ganda (alternate attribute)

Atribut berlebihan : atribut terkodekan

• Atribut yang memiliki kode tambahan yang menunjukkan beberapa kondisi lainnya

Contoh :

id\_mk di tabel kuliah yang didalamnya sudah terkandung data program studi. Data program studi ini sebenarnya tidak diperlukan lagi karena sudah ada atribut prog\_studi di tabel kuliah

Atribut dalam domain komposit

Atribut berlebihan : atribut gabungan

• Atribut dalam domain komposit

Contoh :

atribut nim di tabel mahasiswa merupakan gabungan dari tahun masuk/angkatan dengan program studi dan no urut mahasiswa. Dengan demikian atribut ini sebenarnya tidak atomik karena masih bisa dibagi lagi

Atribut berlebihan (atribut tumpang-tindih)

• Atribut dengan nilai yang tidak sepenuhnya ekslusif

Contoh :

atribut semester di tabel kuliah berisikan :

‘1’ -> matakuliah ganjil

‘2’ -> matakuliah genap

‘3’ -> matakuliah ganjil & genap

nilai ‘3’ mencakup semester genap dan ganjil sekaligus (jadi tidak ekslusif).

Atribut berlebihan : atribut bermakna ganda

• Atribut yang memiliki arti berbeda tergantung kelompok entitasnya

Contoh :

Di tabel dosen terdapat atribut gaji. Bagi dosen tetap atribut ini berisi gaji tetap perbulan, sedangkan bagi dosen tidak tetap gaji ini berisi insentif mengajar tiap sks.

Tabel rekapitulasi

• Laporan hasil rekapitulasi akan selalu merupakan hasi pengolahan dari semua tabel yang ada.

• Pengolahan tersebut melibatkan banyak tabel sehingga akan membutuhkan waktu yang lama.

• Jika hal tersebut sering diakses dan diperlukan, maka perlu dibuat tabel khusus untuk menyimpan data hasil rekapitulasi tersebut.