

DENORMALISASI



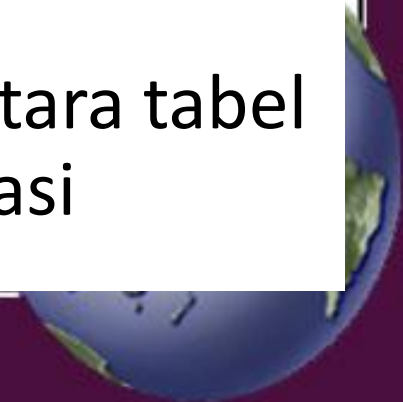
Denormalisasi

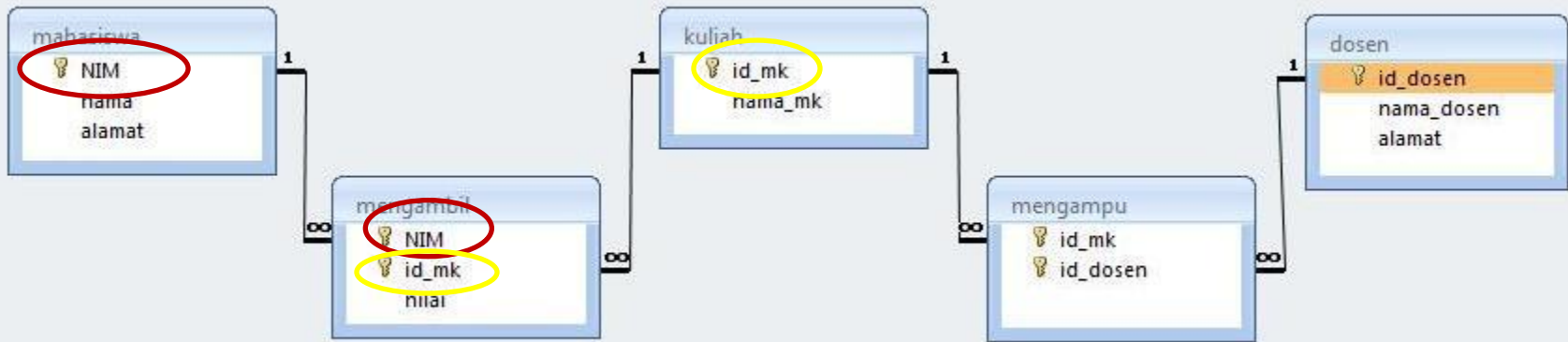
- Pelanggaran Normalisasi basis data disebut dengan Denormalisasi.
- Satu-satunya alasan mengapa kita boleh melakukan denormalisasi adalah pertimbangan performansi.
- Jika performansi yang kita peroleh bisa menjadi jauh lebih baik, maka normalisasi yang telah dilakukan cukup beralasan untuk dilanggar.



Normalisasi Vs Denormalisasi

- Normalisasi akan meningkatkan *data integrity* tetapi juga akan meningkatkan *query complexity*.
- Denormalisasi akan mengurangi *data integrity* dan juga akan mengurangi *query complexity*
- Pada basis data relational, redundansi tidak bisa dihilangkan sama sekali khususnya redundansi pada atribut-atribut yang berfungsi sebagai key primer.
- Karena dengan inilah keterhubungan antara tabel satu dengan yang lain dapat terakomodasi





- Relasi antar tabel tidak akan ada jika redundansi dihilangkan sama sekali
- Akan tetapi redundansi juga harus diminimalisir karena berpotensi mengganggu integritas basis data.

- Khususnya pada saat terjadi operasi perubahan data yang tidak dijalankan ke tabel-tabel lain yang berhubungan
- Performansi dapat ditingkatkan dengan mengendalikan redundansi untuk mengurangi perhitungan, kompleksitas perintah dan jumlah tabel yang harus dilibatkan (join).
- Untuk itu digunakan Denormalisasi basis data.



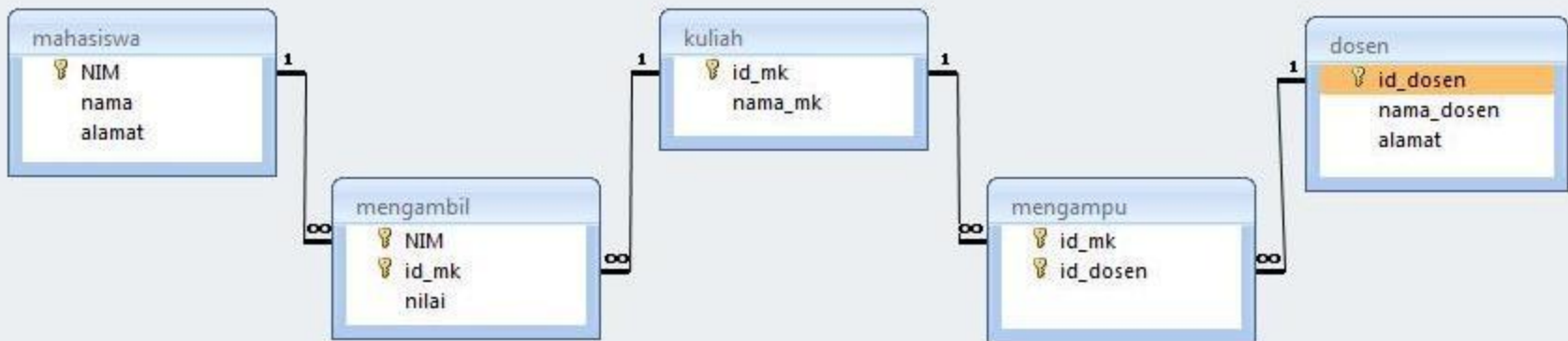
Bentuk-bentuk Denormalisasi

- 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



- Tampilkan berapa banyak matakuliah yang sudah diambil oleh mahasiswa tertentu !

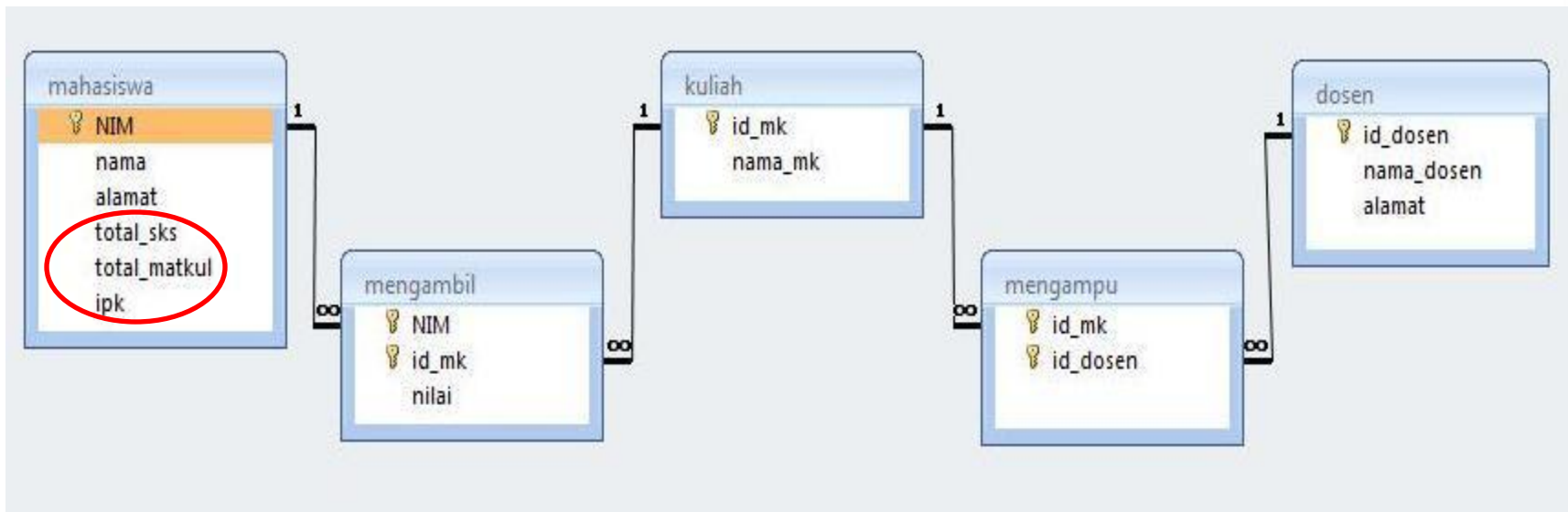
```
select      count (*)      from mengambil where  
NIM = '09.51.0062';
```

- Dari contoh sebelumnya, bagaimana jika data mahasiswa sangat banyak sedangkan proses untuk menampilkan jumlah matakuliah, jumlah sks ataupun IPK sering dilakukan?
- **Pasti membutuhkan waktu yang lama !**

Solusinya → denormalisasi



- Untuk itu perlu ditambahkan atribut baru pada tabel mahasiswa (**total_sks**, **total_matkul**, **ipk**)



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 tabelkuliah
- Kode kuliah SBD : **ST022**





- Tapi akan menjadi aneh jika kode matakuliah tidak mengikuti format aturan penulisan yang ada. Untuk itu kita bisa lakukan denormalisasi dengan tetap menuliskan kode matakuliah seperti di atas.

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

- NIM : 09.51.0062

Ket : angkatan.jurusan.no urut



- Sama halnya dengan atribut terkodekan, akan menjadi aneh jika nomor mahasiswa tersebut hanyalah nomor urut. Tentu hal ini akan membutuhkan informasi lebih tentang angkatan dan terdaftar di jurusan mana mahasiswa tersebut.

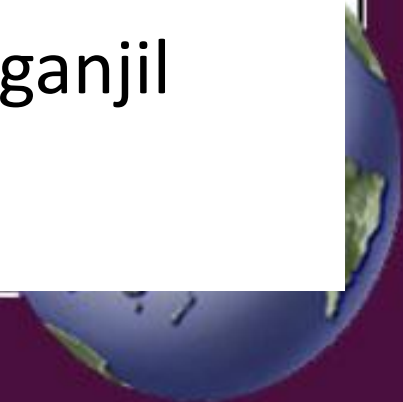


Atribut berlebihan → Atribut tumpang-tindih

- Atribut dengan nilai yang tidak sepenuhnya eksklusif

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 eksklusif).





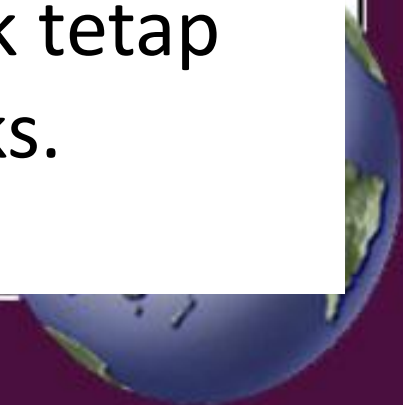
- Jadi sebaiknya matakuliah dimasukkan dua kali, jika itu dilaksanakan di semester ganjil dan genap.
- Tentu saja hal ini melanggar aturan normalisasi (redundansi), tapi untuk **performansi** hal ini dapat 'dilanggar'

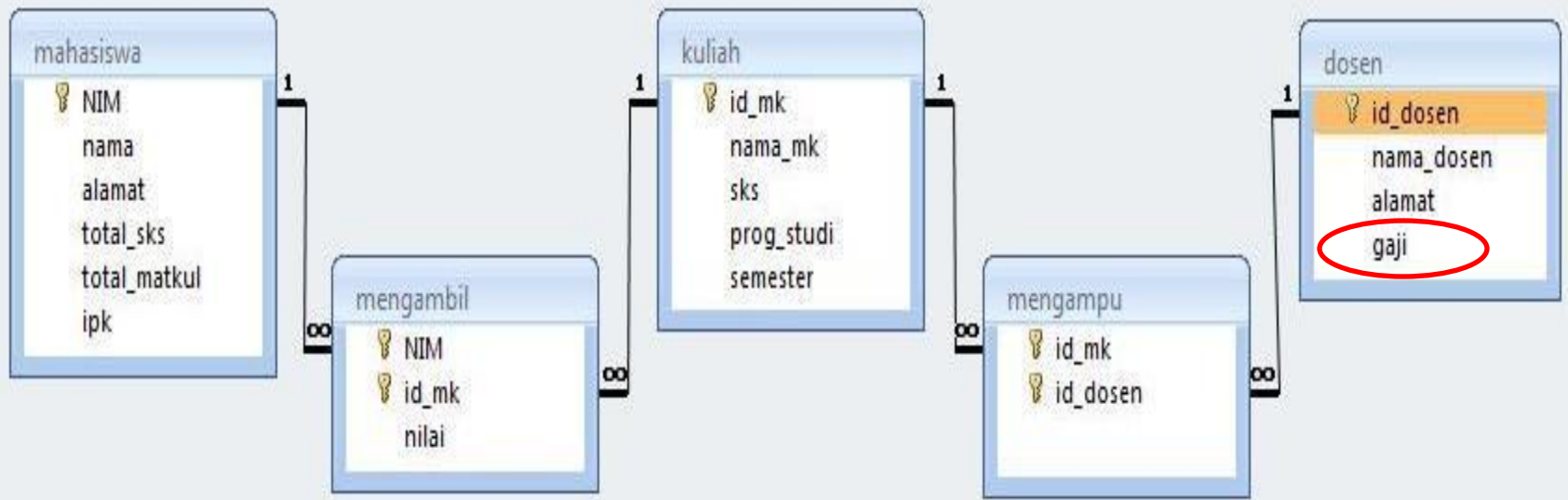
Atribut berlebihan : Atribut bermakna ganda

- 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.





- Jika gaji dosen harus dibedakan, maka harus disediakan 2 buah atribut yakni *gaji_tetap* dan *gaji_tidak_tetap*.
- Walaupun sama-sama berisi tentang jumlah gaji yang diterima dosen, hal ini tentu saja melanggar aturan normalisasi karena ada blok data yang kosong. Tetapi sekali lagi, denormalisasi dalam hal ini boleh dilakukan.

Tabel rekapitulasi

- Laporan hasil rekapitulasi akan selalu merupakan hasil 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.



- Pada contoh kasus sebelumnya, akan dibutuhkan waktu yang lama jika harus menghitung jumlah matakuliah, jumlah sks dan ipk mahasiswa yang pengolahannya berasal dari beberapa tabel.
- Untuk itu bisa dibuat tabel khusus, misal : *rekap_mahasiswa yang berisi data tentang jumlah matakuliah, jumlah sks, ipk.*
- Hal ini tentu saja akan menimbulkan redundansi, tapi dengan mempertimbangkan performansi, Denormalisasi pada kasus ini perlu dilakukan

