

Model Data Relasional Lanjutan

- Varian Entitas
- Varian Relasi
- Spesialisasi dan Generalisasi
- Agregasi
- Proses Lanjutan

Proses perancangan database

- **Tahap 1**, Pengumpulan data dan analisis.
- **Tahap 2, Perancangan database** secara konseptual.
- **Tahap 3**, Pemilihan DBMS.
- **Tahap 4, Perancangan database** secara logika (data model mapping)
- **Tahap 5, Perancangan database** secara fisik.
- **Tahap 6**, Implementasi Sistem **database**.

Tahap 1, Pengumpulan data dan analisis

- Mengetahui dan melakukan analisis apa yang diinginkan dari **user** atau orang-orang yang akan menggunakan aplikasi yang kita buat, terutama bagian yang berinteraksi dengan **sytem database**, baik yang lama maupun yang baru.
- Kebutuhan yang ada kemudian dikumpulkan dan dianalisis.
- Kegiatan yang dilakukan pada tahapan awal adalah sebagai berikut.
 - Menentukan kelompok pemakai dan bidang-bidang aplikasinya
 - Peninjauan dokumentasi yang ada
 - Analisa lingkungan operasi dan pemrosesan data
 - Daftar pertanyaan dan wawancara

Tahap 2, Perancangan database secara konseptual

- Perancangan skema konseptual merupakan tahapan berikutnya.
- Tahapan tersebut adalah perancangan skema konseptual untuk menghasilkan **database** yang tidak bergantung pada sistem manajemen **database**.
- Bisa dengan menggunakan diagram ER atau dengan proses normalisasi

Tahap 3, pemilihan DBMS

- Pemilihan sistem manajemen database merupakan tahapan pemilihan sistem manajemen database yang ditentukan oleh tipe model data, struktur penyimpanan database, programmer, dan tipe bahasa query yang akan digunakan.
- Selain itu, juga penentuan biaya, baik pembiayaan hardware, software, biaya personalia, biaya pembuatan database, biaya pelatihan, maupun biaya pemeliharaan dan juga faktor organisasi yang dilihat dari struktur data yang akan digunakan.
- Faktor-faktor ekonomi dan organisasi yang mempengaruhi satu sama lain dalam pemilihan DBMS :
 - Struktur data
Jika data yang disimpan dalam *database* mengikuti struktur hirarki, maka suatu jenis hirarki dari DBMS harus dipikirkan.
 - Personal yang telah terbiasa dengan suatu sistem
Jika staf *programmer* dalam suatu organisasi sudah terbiasa dengan suatu DBMS, maka hal ini dapat mengurangi biaya latihan dan waktu belajar.
 - Tersedianya layanan penjual
Keberadaan fasilitas pelayanan penjual sangat dibutuhkan untuk membantu memecahkan beberapa masalah sistem.

Tahap 4, Perancangan database secara logika (data model mapping)

- Tahap selanjutnya adalah membuat sebuah skema konseptual dan skema eksternal pada model data dari DBMS yang terpilih.
- Tahap ini dilakukan oleh pemetaan skema konseptual dan skema eksternal yang dihasilkan pada tahap 2. Pada tahap ini, skema konseptual ditransformasikan dari model data tingkat tinggi yang digunakan pada tahap 2 ke dalam model data dari model data dari DBMS yang dipilih pada tahap 3.
- Pemetaan tersebut dapat diproses dalam 2 tingkat:
 - Pemetaan *system-independent*
Pemetaan ke dalam model data DBMS dengan tidak mempertimbangkan karakteristik atau hal-hal yang khusus yang berlaku pada implementasi DBMS dari model data tersebut.
 - Penyesuaian skema ke DBMS yang spesifik
Mengatur skema yang dihasilkan pada langkah 1 untuk disesuaikan pada implementasi yang khusus di masa yang akan datang dari suatu model data yang digunakan pada DBMS yang dipilih. Hasil dari tahap ini memakai perintah-perintah DDL (*Data Definition Language*) dalam bahasa DBMS yang dipilih yang menentukan tingkat skema konseptual dan eksternal dari sistem *database*. Tetapi dalam beberapa hal, perintah-perintah DDL memasukkan parameter-parameter rancangan fisik sehingga DDL yang lengkap harus menunggu sampai tahap perancangan *database* secara fisik telah lengkap. Tahap ini dapat dimulai setelah pemilihan sebuah implementasi model data sambil menunggu DBMS yang spesifik yang akan dipilih. Contoh: jika memutuskan untuk menggunakan beberapa *relational* DBMS tetapi belum memutuskan suatu relasi yang utama. Rancangan dari skema eksternal untuk aplikasi-aplikasi yang spesifik seringkali sudah selesai selama proses ini.

Tahap 5, Perancangan database secara fisik

- Perancangan *database* secara fisik merupakan proses pemilihan struktur-struktur penyimpanan dan jalur-jalur akses pada *file-file database* untuk mencapai penampilan yang terbaik pada bermacam-macam aplikasi. Selama fase ini, dirancang spesifikasi-spesifikasi untuk *database* yang disimpan yang berhubungan dengan struktur-struktur penyimpanan fisik, penempatan record dan jalur akses. Berhubungan dengan *internal schema* (pada istilah 3 level arsitektur DBMS).
- Beberapa petunjuk dalam pemilihan perancangan *database* secara fisik :
 - *Response time*
Waktu yang telah berlalu dari suatu transaksi *database* yang diajukan untuk menjalankan suatu tanggapan. Pengaruh utama pada *response time* adalah di bawah pengawasan DBMS yaitu : waktu akses *database* untuk data item yang ditunjuk oleh suatu transaksi. *Response time* juga dipengaruhi oleh beberapa faktor yang tidak berada di bawah pengawasan DBMS, seperti penjadwalan sistem operasi atau penundaan komunikasi.
 - *Space utility*
Jumlah ruang penyimpanan yang digunakan oleh *file-file database* dan struktur-struktur jalur akses.
 - *Transaction throughput*
Rata-rata jumlah transaksi yang dapat diproses per menit oleh sistem *database*, dan merupakan parameter kritis dari sistem transaksi (misal : digunakan pada pemesanan tempat di pesawat, bank, dll). Hasil dari fase ini adalah penentuan awal dari struktur penyimpanan dan jalur akses untuk *file-file database*.

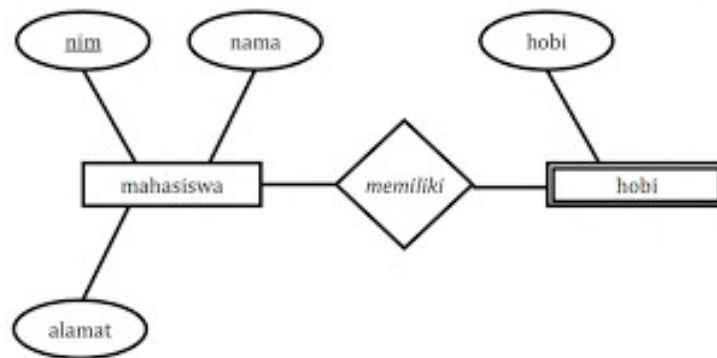
Tahap 6, Implementasi sistem database

- Setelah perancangan secara logika dan secara fisik lengkap, kita dapat melaksanakan implementasi sistem *database*.
- Perintah-perintah dalam DDL dan SDL(*Storage Definition Language*) dari DBMS yang dipilih, dihimpun dan digunakan untuk membuat skema *database* dan *file-file database* (yang kosong).
- Sekarang *database* tersebut dimuat (disatukan) dengan datanya. Jika data harus dirubah dari sistem komputer sebelumnya, perubahan-perubahan yang rutin mungkin diperlukan untuk format ulang datanya yang kemudian dimasukkan ke *database* yang baru.
- Transaksi-transaksi *database* sekarang harus dilaksanakan oleh para programmer aplikasi. Spesifikasi secara konseptual diuji dan dihubungkan dengan kode program dengan perintah-perintah dari *embedded DML* yang telah ditulis dan diuji.
- Suatu saat transaksi-transaksi tersebut telah siap dan data telah dimasukkan ke dalam *database*, maka tahap perancangan dan implementasi telah selesai, dan kemudian tahap operasional dari sistem *database* dimulai.

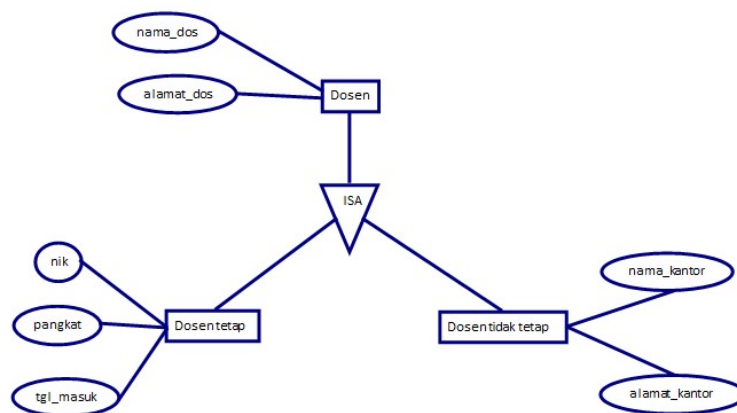
Varian Entitas

- Himpunan Entitas Lemah (Weak Entity Sets)
 - Entitas yang kemunculannya tergantung pada eksistensinya dalam sebuah relasi terhadap entitas lain (Strong Entity)
 - Tidak memiliki atribut sebagai key
- Sub entitas (Subtype Entities)
 - Merupakan himpunan entitas yang beranggotakan entitas-entitas yang merupakan bagian dari himpunan entitas yang lebih utama
 - Merupakan hasil dekomposisi (spesialisasi) himpunan entitas berdasarkan pengelompokan tertentu.

Contoh Entitas Lemah



Sub Entitas (*Subtype Entities*)



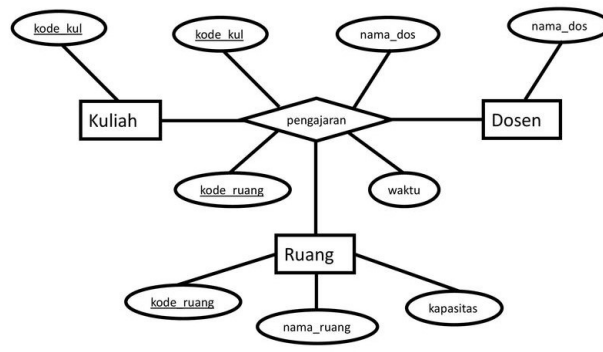
Varian Relasi

- Relasi Tunggal (Unary Relation)
 - Relasi yang terjadi dari sebuah himpunan entitas ke himpunan entitas yang sama
- Relasi Multi Entitas (N-ary Relation)
 - Relasi dari tiga himpunan entitas atau lebih.
 - Sebaiknya dihindari karena mengaburkan derajat (kardinalitas) relasi
- Relasi Ganda (Redundant Relation)
 - Muncul antara dua himpunan entitas tidak hanya satu relasi tetapi lebih dari satu relasi

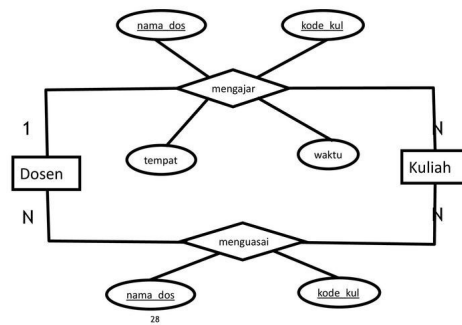
Relasi Tunggal



Relasi Multi Entitas



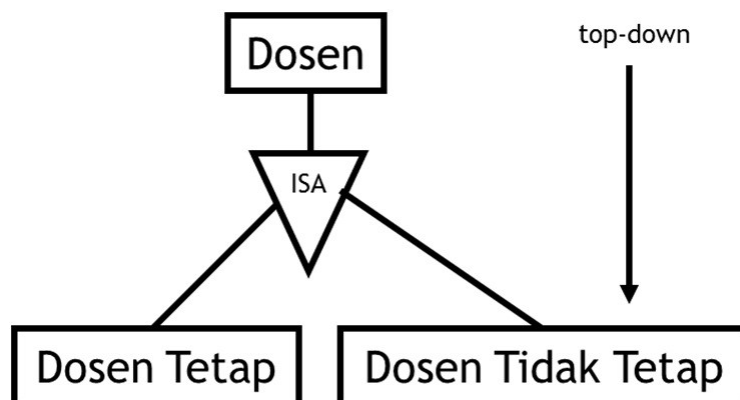
Relasi Ganda



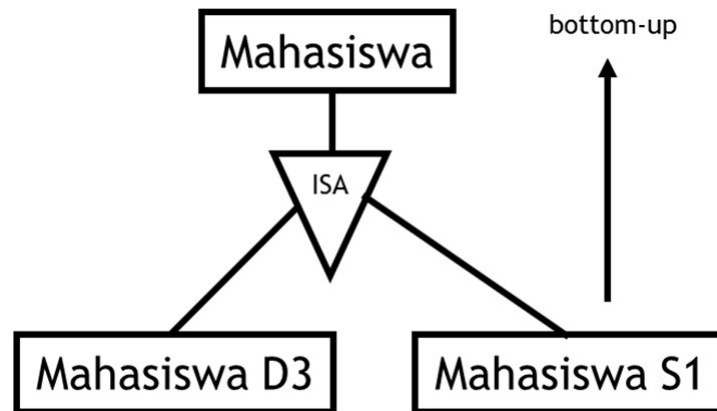
Spesialisasi dan Generalisasi

- Spesialisasi
 - Melakukan pengelompokan dari sebuah entitas menjadi entitas-entitas baru yang lebih spesial (khusus)
 - Jika kita memulai dari sebuah himpunan entitas lalu kemudian melakukan pengelompokan yang melahirkan himpunan entitas baru (proses top down)
 - Yang ditekankan adalah perbedaan masing-masing entitas
- Generalisasi
 - Menyatukan beberapa entitas yang hampir sama menjadi sebuah kelompok baru
 - Yang ditekankan adalah persamaan masing-masing entitas

Spesialisasi



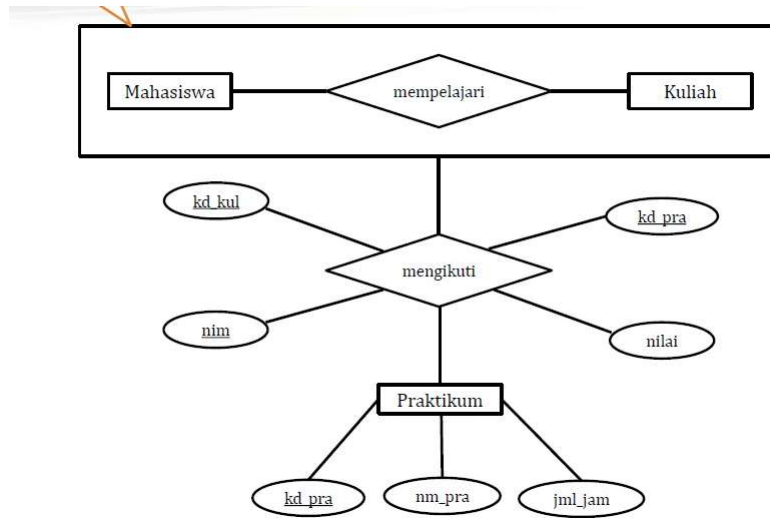
Generalisasi



Agregasi

- Relasi yang secara kronologis mensyaratkan telah adanya relasi lain
- Relasi terbentuk tidak hanya dari entitas tetapi juga dari relasi lain
- Agregasi ini sebaiknya tidak dilakukan dalam diagram ER

Agregasi



Proses Lanjutan

- Key Alternatif (Alternate Key)
- Pengkodean Internal
- Dekomposisi Himpunan Entitas dan Normalisasi (akan dibahas pada bab khusus)
- Fleksibilitas

Key Alternatif (Alternate Key)

- Adalah candidate **key** yang tidak terpilih.
- Misal dalam suatu entity terdapat dua field yang bisa dijadikan sebagai kunci.
- Sementara yang boleh dijadikan kunci hanya satu, maka anda harus memilih salah satu.

Key Alternatif

Primary Key Alternate Key Candidate Key

MHS

| NPM | NAMA | ALAMAT |
|------------|-------------|--------------------------|
| 10200123 | SULAEMAN | JL. SIRSAK 28 JAKARTA |
| 30100143 | DIANA | JL. STASIUN 23 BOGOR |
| 50100333 | SADIKIN | JL. MARGONDA RAYA 100 |
| 20100296 | THAMRIN | JL. JATIASIH 78 BEKASI |
| 10200928 | LINA | JL. NANGKA 4 JAKARTA |
| 50100375 | IRAWATI | JL. PEMUDA 382 TANGERANG |

Pengkodean Internal

- Pengkodean Eksternal (user-defined coding)
 - Mewakili pengkodean yang telah digunakan secara terbuka dan dikenal dengan baik oleh end-user.
 - Contoh : atribut no_bp, kode_kuliah, nip, dsb
- Pengkodean Internal (system coding)
 - Pengkodean yang hanya dipahami oleh database designer / programmer
 - Contoh : atribut kode_dosen
 - Dapat diterapkan pada atribut lain (non-key)

3 bentuk Pengkodean Internal

- Sekuensial
 - Dilakukan berdasarkan asosiasi data dengan kode terurut
 - Biasanya berupa bilangan asli / abjad
 - Misalnya data nilai kuliah. A, B, C, D dan E
- Mnemonic
 - Dibentuk oleh suatu singkatan
 - Misalnya L = Laki-laki dan P = Perempuan
- Blok
 - Pengkodean dinyatakan dalam format tertentu
 - Contoh : no_bp anda

Fleksibilitas

- Penambahan Atribut
- Pemilihan domain atribut yang lebih luas
- Generalisasi
- Perubahan struktur entitas dari berorientasi kolom menjadi berorientasi baris