

BASIS DATA I

ENTITY RELATIONSHIP DIAGRAM (ERD)



Oleh. Masbahah, S.Pd., M.Pd

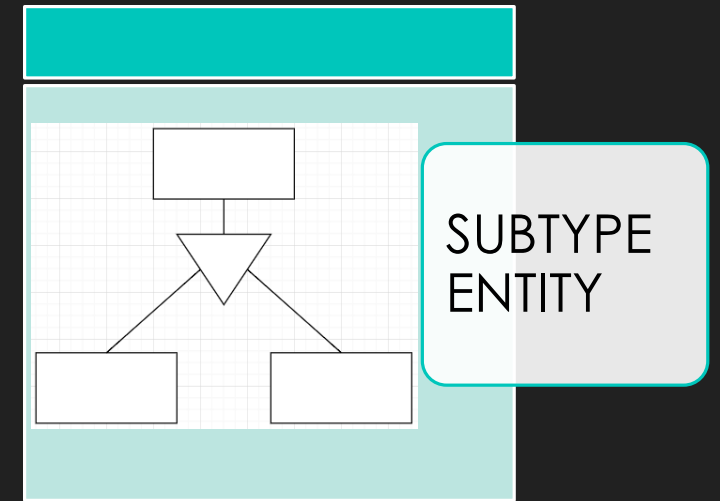
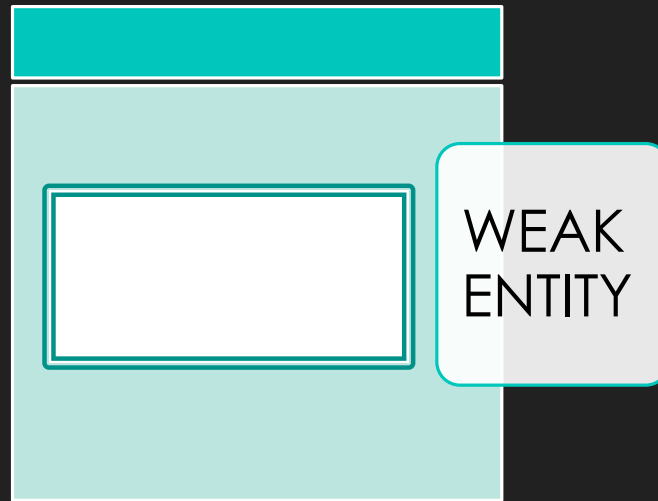
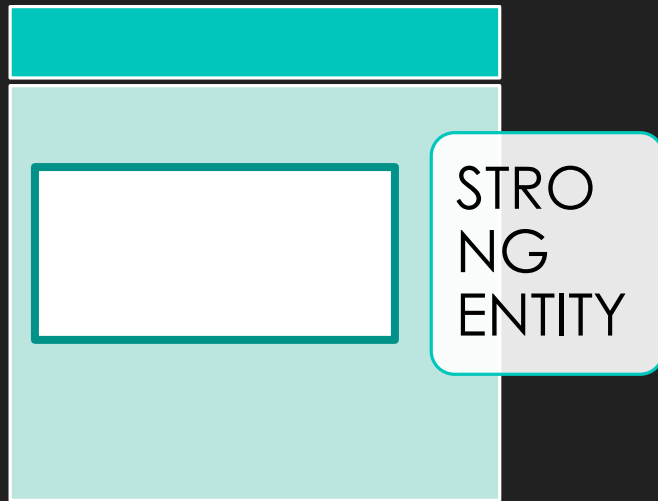
D3 Teknik Informatika

Sekolah Vokasi UNS

TOPIK

- VARIAN ENTITAS
- VARIAN ATRIBUT
- VARIAN RELASI
- AGREGASI
- PROSES LANJUTAN / FINAL DESIGN

VARIAN ENTITAS



STRONG & WEAK ENTITY

- Strong entity merupakan entitas yang mandiri, yang keberadaannya tidak bergantung pada keberadaan entitas yang lainnya.
- Weak entity merupakan entitas yang kemunculannya tergantung pada eksistensinya dalam sebuah relasi terhadap entitas lain (Strong Entity)



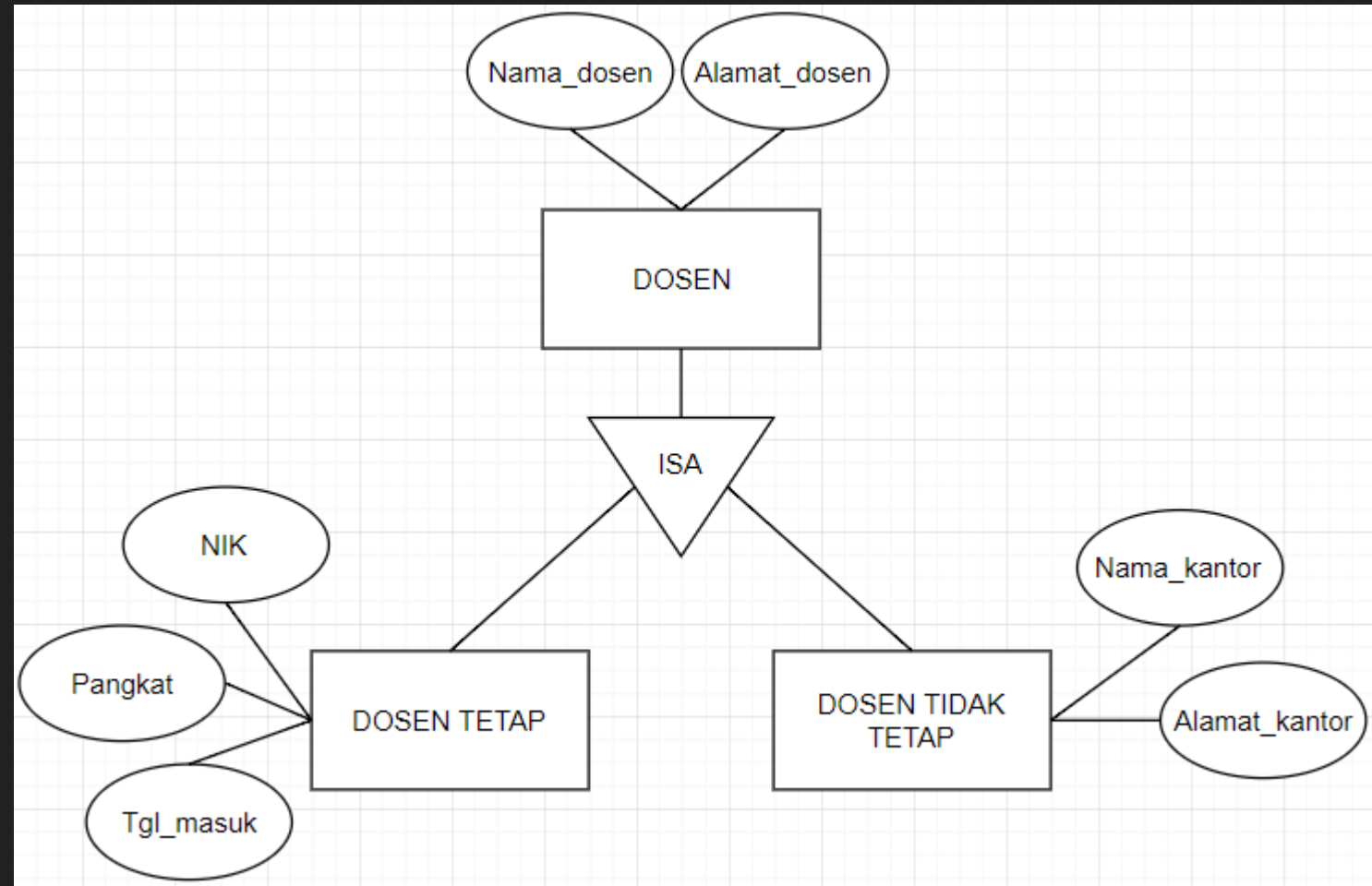
SUBTYPE ENTITY

- Entitas yang beranggotakan entitas-entitas yang merupakan bagian dari himpunan entitas yang lebih superior/utama

1. Spesialisasi
2. Generalisasi

SUBTYPE ENTITY

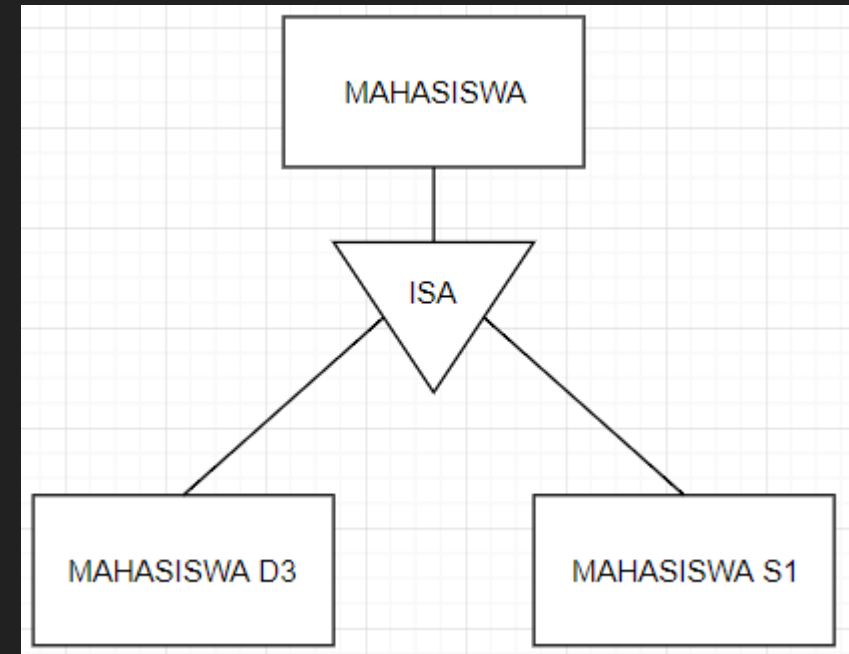
Spesialisasi (Top – Down)
Hasil dekomposisi
himpunan entitas
berdasarkan
pengelompokan tertentu



SUBTYPE ENTITY

Generalisasi (Bottom – up)

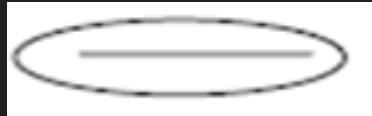
Atribut masing-masing entitas tidak memiliki perbedaan yang jelas/tegas sehingga disatukan



VARIAN ATRIBUT

- Key Atribut

Atribut yang digunakan untuk menentukan suatu entitas secara unik



- Atribut deskriptif

Seluruh atribut yang bukan merupakan anggota dari key atribut

VARIAN ATRIBUT

- Atribut Simple/Atomic Attribute

Atribut yang tidak dapat dibagi-bagi lagi menjadi atribut lebih mendasar

- Composite Attribute

Atribut yang terdiri dari beberapa atribut yang lebih mendasar.

Contoh: atribut Alamat, terdiri atas atribut Jalan, Kota, Kode_Pos.

Simbol:



VARIAN ATRIBUT

- Single-Valued Attribute

Atribut yang hanya memiliki satu harga/nilai.

Contoh: atribut Umur, Nomor_Pegawai

- Multi-Valued Attribute

Atribut yang memiliki isi lebih dari satu nilai. Contoh:

atribut Pendidikan_Tinggi pada entitas Pegawai, dapat berisi lebih dari satu nilai, yaitu: SMP, SMU, Perguruan Tinggi (Sarjana), Doktor, dll.

Atribut Hobi pada entitas Siswa, dapat memiliki lebih dari satu nilai, yaitu: sepak bola, menyanyi, menari, tenis, dll.

Simbol :



VARIAN ATRIBUT

- Null Values Attribute

Atribut dari entitas yang tidak memiliki nilai.

Contoh: atribut Pendidika tinggi untuk tamatan SMP

- Mandatory atribut

Atribut yang harus memiliki nilai

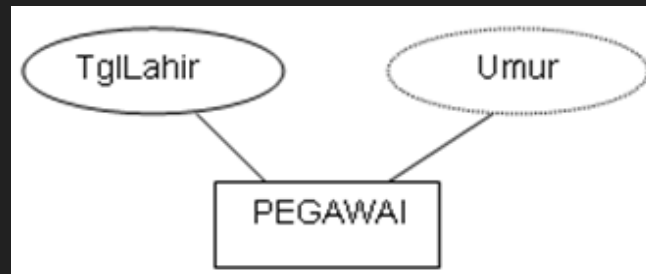
Contoh: NIM, Nama pada entitas mahasiswa

VARIAN ATRIBUT

○ Derived Attribute

Atribut yang nilainya dapat diisi atau diturunkan dari perhitungan atau algoritma tertentu.

Contoh: atribut Umur, dapat dihitung dari atribut Tgl_Lahir



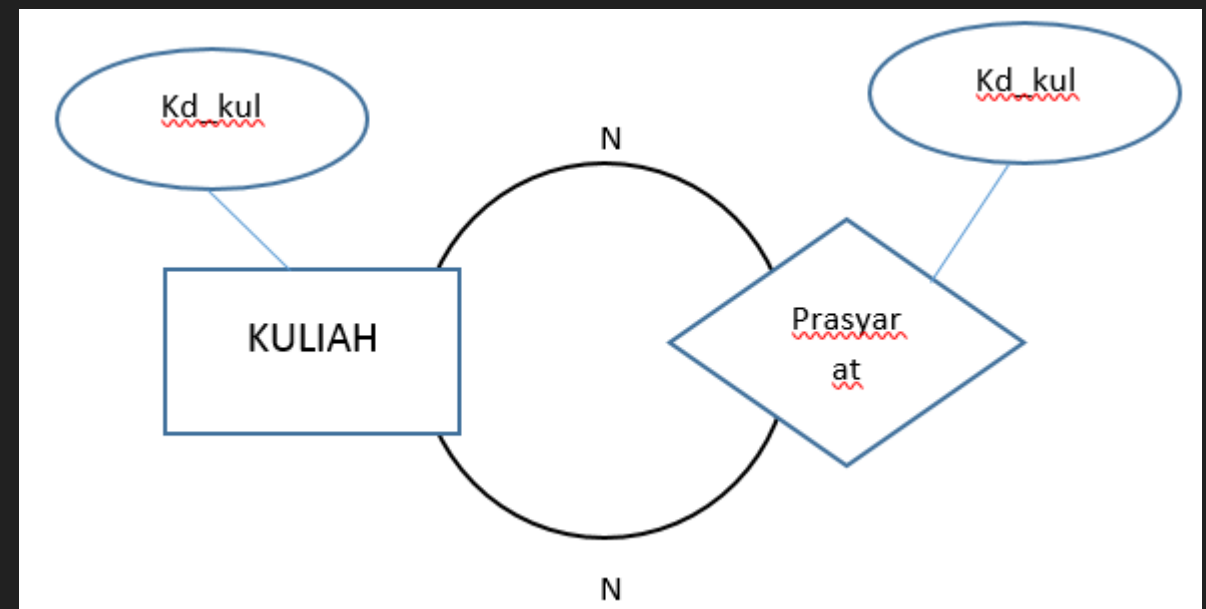
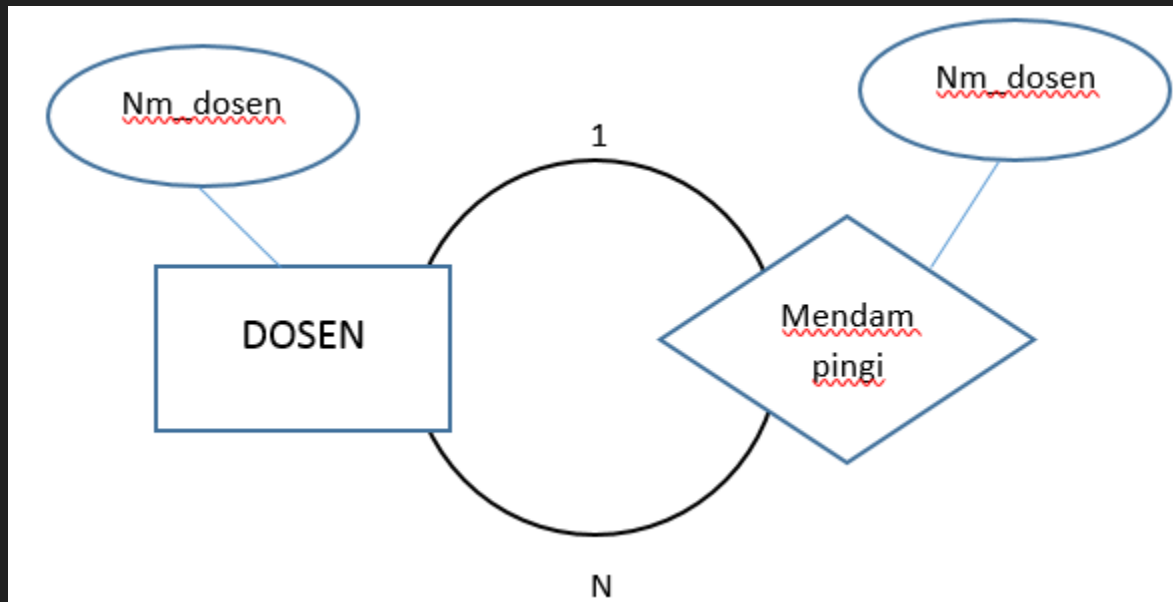
Gambar. Derived Attribute

VARIAN RELASI

- Relasi Tunggal (Unary Relation)
- Relasi Multi-Entitas (N-ary Relation)
- Relasi Ganda (Redudant Relation)

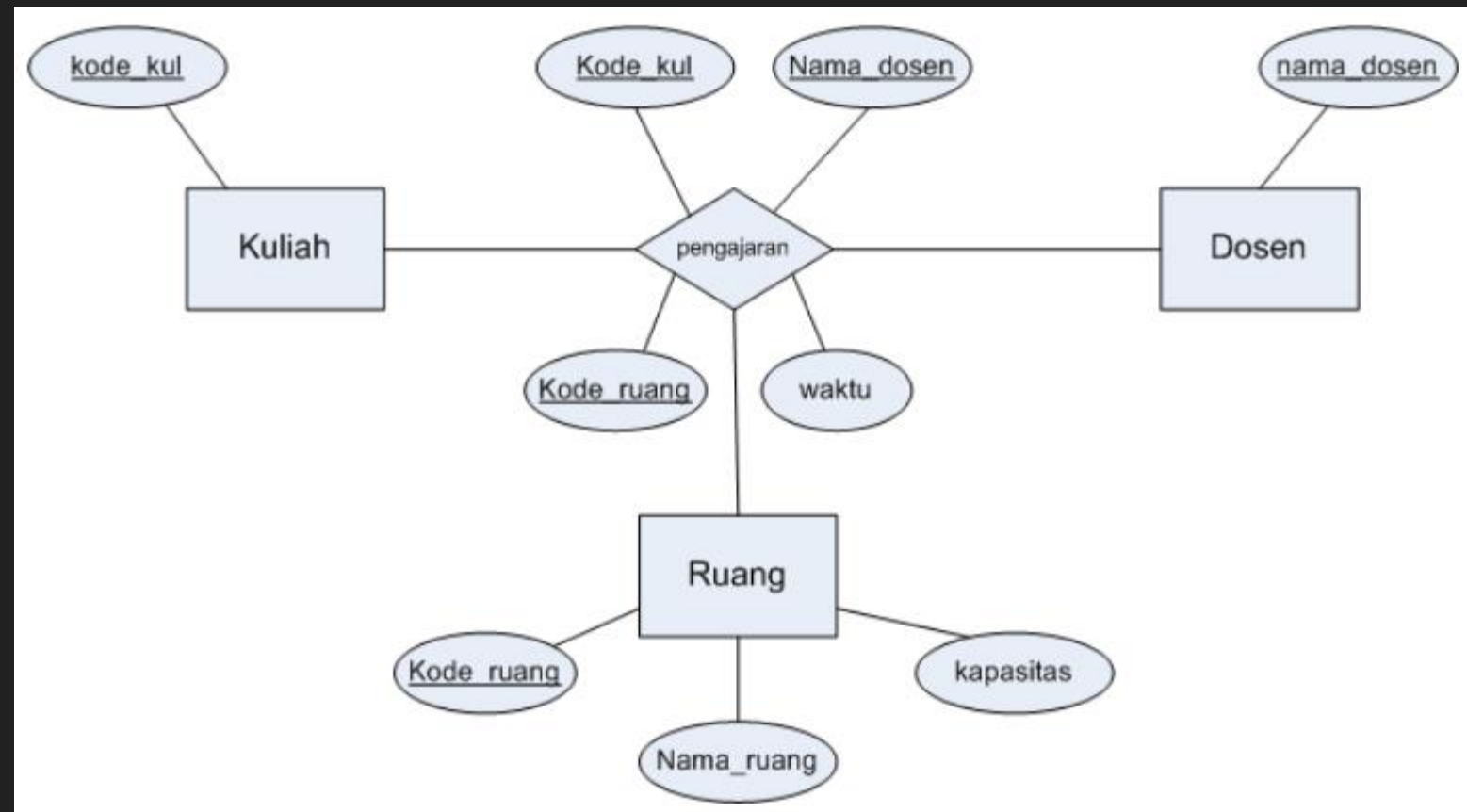
VARIAN RELASI

- Relasi Tunggal (Unary Relation)
- Merupakan relasi yang terjadi dari sebuah himpunan entitas ke himpunan entitas yang sama



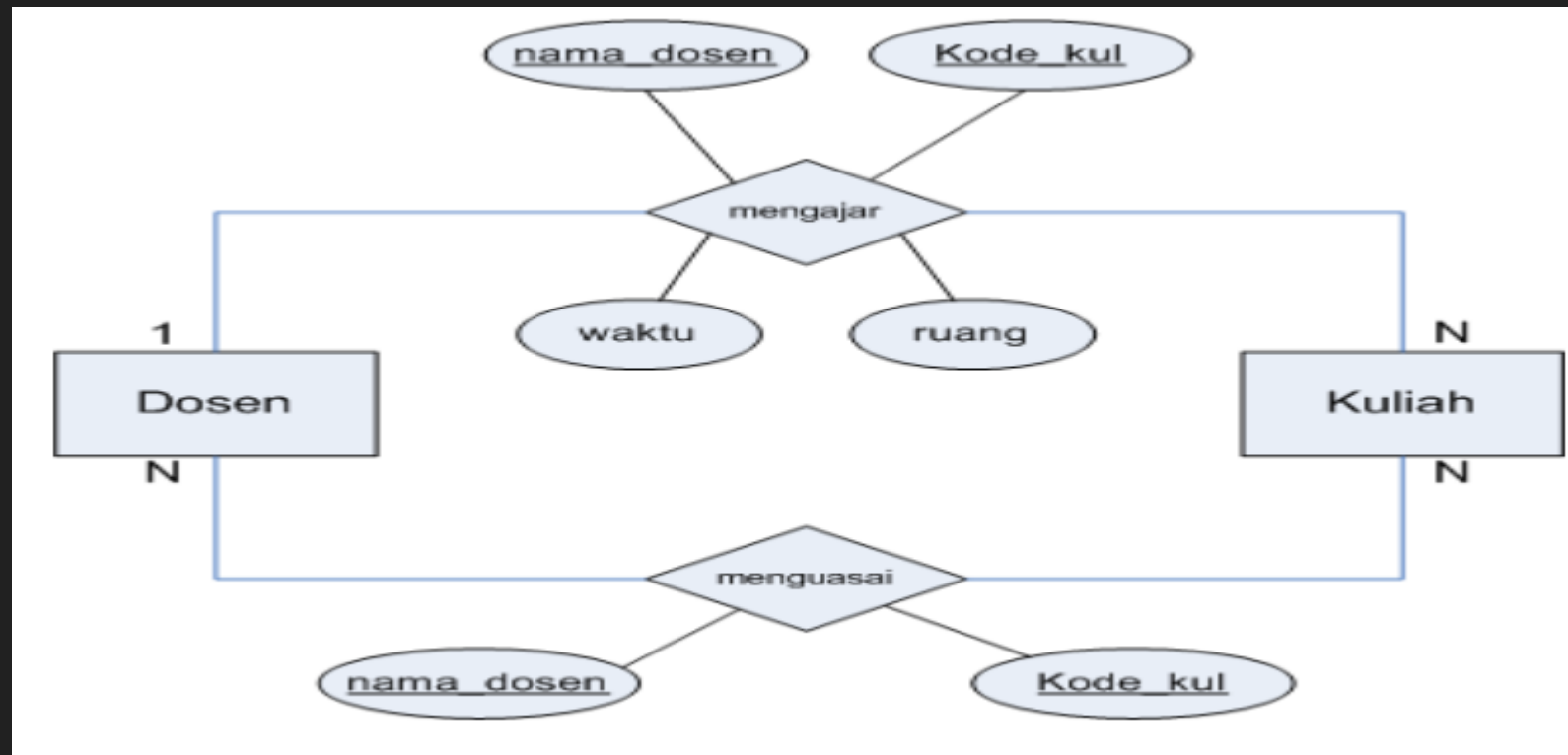
VARIAN RELASI

- Relasi Multi-Entitas (N-ary Relation)
- Merupakan relasi dari 3 himpunan entitas atau lebih



VARIAN RELASI

- Relasi Ganda (Redudant Relation)
- Relasi yang muncul antara dua himpunan entitas tidak hanya 1 relasi, tetapi lebih dari satu.



AGREGASI

- Sebuah himpunan relasi yang secara langsung menghubungkan sebuah Entitas dengan sebuah himpunan relasi dalam Diagram-ER.
- Sebuah relasi terbentuk tidak hanya dari himpunan entitas tetapi juga mengandung unsur dari relasi lain.
- Sebenarnya kondisi ini tidak tepat bahkan ada yang dengan tegas tidak memperbolehkan.

AGREGASI

- Mengikuti praktikum dilakukan setelah (jika ada relasi) mempelajari kuliah

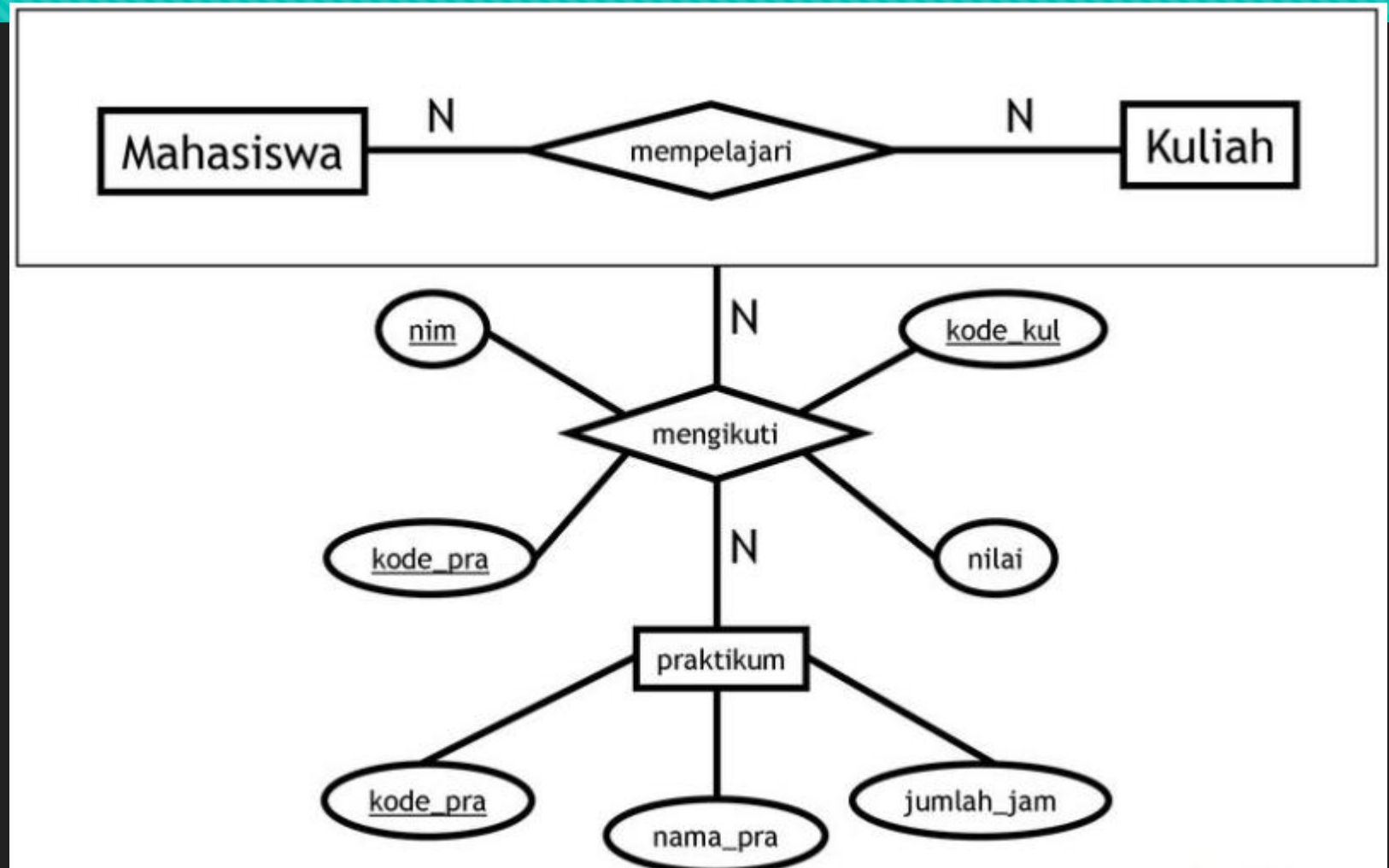
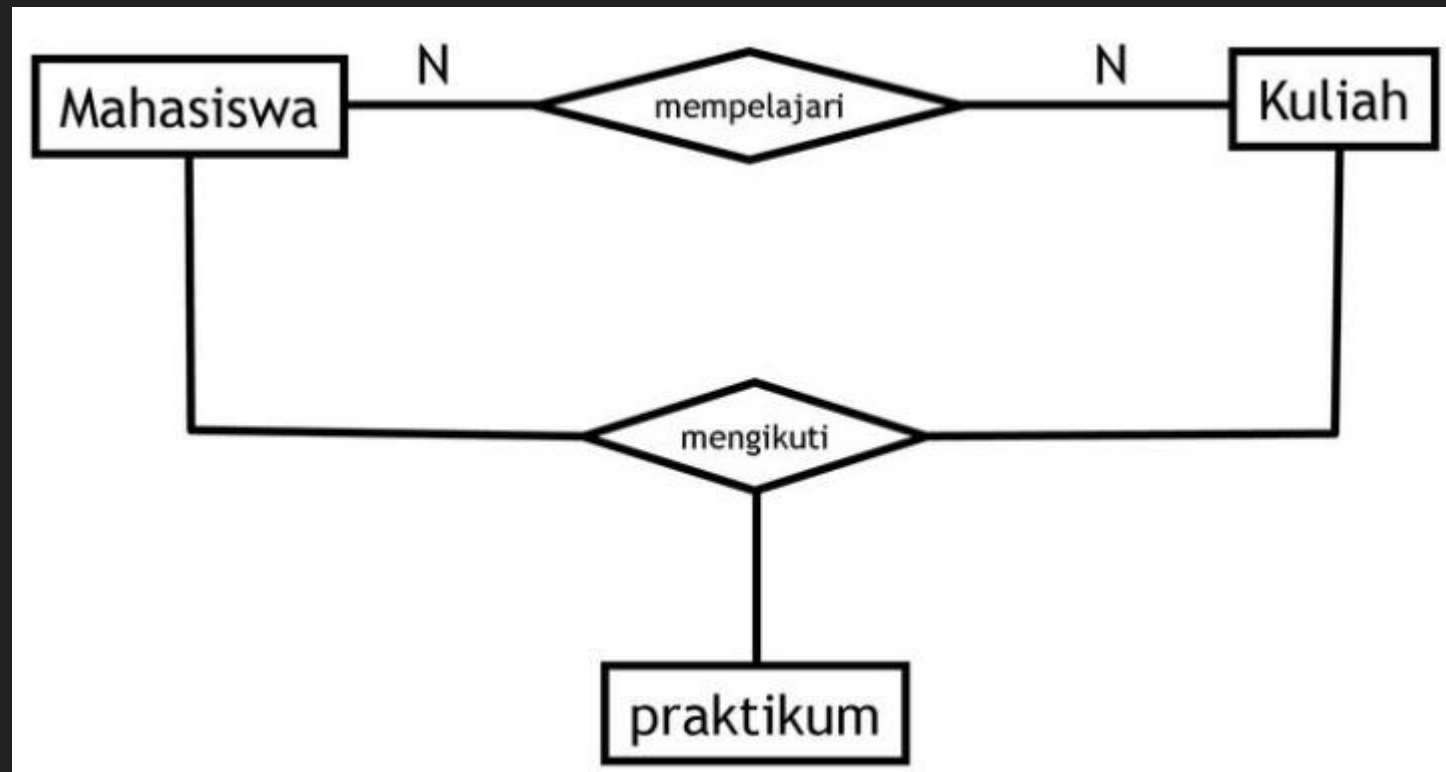


DIAGRAM ER YANG TIDAK TEPAT

- Diagram E-R semacam ini akan mengaburkan faktor kronologis yang bertentangan dengan fakta sebenarnya



PROSES LANJUTAN / FINAL DESIGN

- Alternate Key
- Pengkodean internal
- Dekomposisi himpunan entitas dan normalisasi
- Fleksibilitas

PROSES LANJUTAN / FINAL DESIGN

Alternate Key

- Key himpunan entitas/relasi dipilih dari atribut yang dapat menjamin keunikan entitas
- Dikategorikan baik jika berukuran kecil dan sekuensial
- Misalnya kode_jadwal pada entitas jadwal. Kode_jadwal bukan merupakan fakta dari dunia nyata. Key semacam itu disebut key alternative

PROSES LANJUTAN / FINAL DESIGN

Pengkodean internal

- Pengkodean Merupakan cara untuk menyatakan suatu data (atribut) dalam bentuk lain yang bertujuan untuk efisiensi ruang penyimpanan. Pengkodean dapat dibedakan menjadi 2 yaitu :
 1. Pengkodean Eksternal (user-defined coding)

Pengkodean yang telah digunakan secara terbuka dan dikenal dengan baik oleh user (end-user). contoh : Atribut NIM dan Kode_kul.

PROSES LANJUTAN / FINAL DESIGN

2. Pengkodean Internal (system coding)

Pengkodean semacam ini tidak dikenal oleh end-user, melainkan oleh system (aplikasi maupun DBMS yang digunakan).

Ada 3 bentuk pengkodean yang dapat kita pilih yaitu :

- a. Sekuensial
- b. Mnemonic
- c. Blok

PROSES LANJUTAN / FINAL DESIGN

Sekuensial

- Pengkodean yang dilakukan dengan mengasosiasikan data dengan kode terurut (biasanya berupa bilangan asli atau abjad).
- Misalnya data indeks nilai ('Sempurna', 'Baik', 'Cukup', 'Kurang', 'Buruk') dikodekan dengan A, B, C, D, dan E.

PROSES LANJUTAN / FINAL DESIGN

Mnemonic

- Pengkodean yang dilakukan dengan membentuk suatu singkatan dari data yang ingin dikodekan misalnya data jenis kelamin (laki - laki dan perempuan) dikodekan dengan 'L', dan 'P'.

Blok

- Pengkodean yang dinyatakan dalam format tertentu misalnya NIM mahasiswa dengan format XXXX.YY.ZZZ yang terbentuk atas XXXX = angka tahun masuk, YY = kode jurusan, dan ZZZ = nomor urut mahasiswa.

PROSES LANJUTAN / FINAL DESIGN

Dekomposisi himpunan entitas dan normalisasi

- Sebuah himpunan entitas yang ada dalam sebuah Diagram E-R dapat didekomposisi menjadi beberapa himpunan entitas baru karena:
 - a. pertimbangan efisiensi ruang penyimpanan
 - b. pertimbangan kemudahan/kecepatan pengaksesan data.

PROSES LANJUTAN / FINAL DESIGN

- Upaya dekomposisi ini senantiasa akan menghasilkan :
 - a. satu himpunan entitas kuat (strong entity set)
 - b. satu atau beberapa himpunan entitas lemah atau sub entitas.
- Ada dua bentuk dekomposisi himpunan entitas, yaitu:
 - a. Dekomposisi Atribut/Vertikal
 - b. Dekomposisi Entitas/Horisontal

PROSES LANJUTAN / FINAL DESIGN

Dekomposisi Atribut/Vertikal

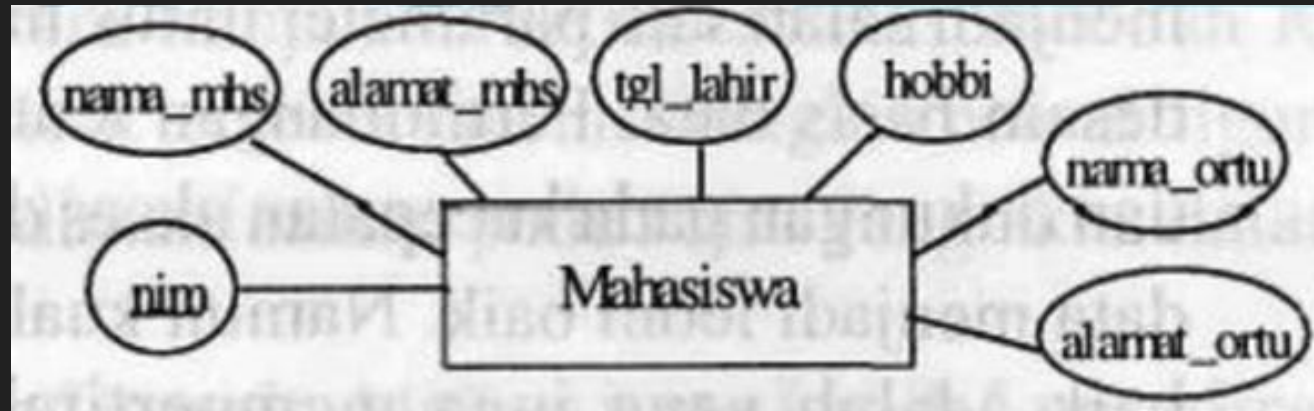
- Dekomposisi ini dilakukan dengan cara membagi sebuah himpunan entitas menjadi dua atau lebih dengan pemisahan atribut.

Dekomposisi Entitas/Horisontal (Spesialisasi)

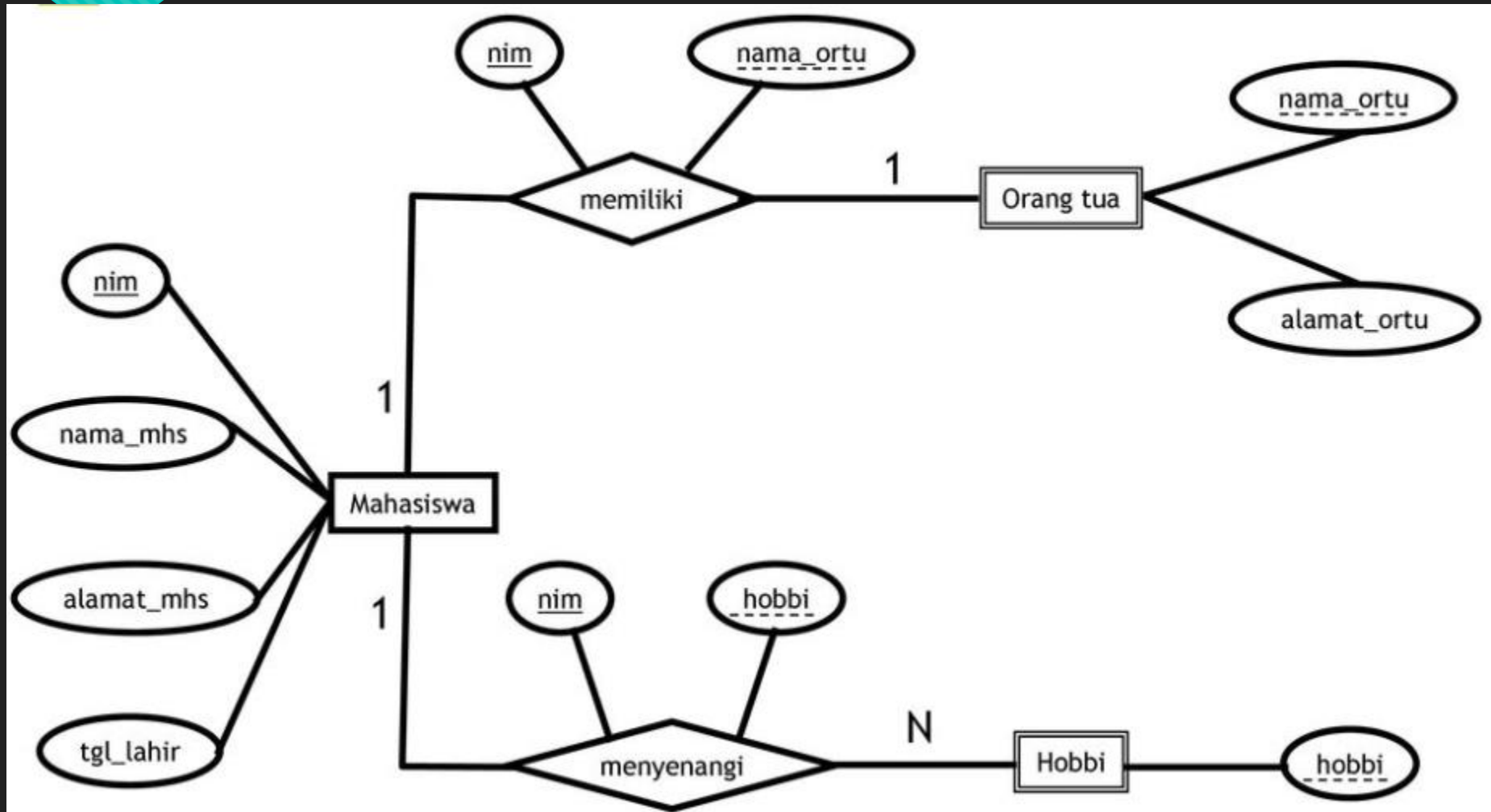
- Dekomposisi ini dilakukan dengan cara membagi sebuah himpunan entitas menjadi dua atau lebih dengan pemisahan entitas.

PROSES LANJUTAN / FINAL DESIGN

○ Dekomposisi Atribut / Vertikal

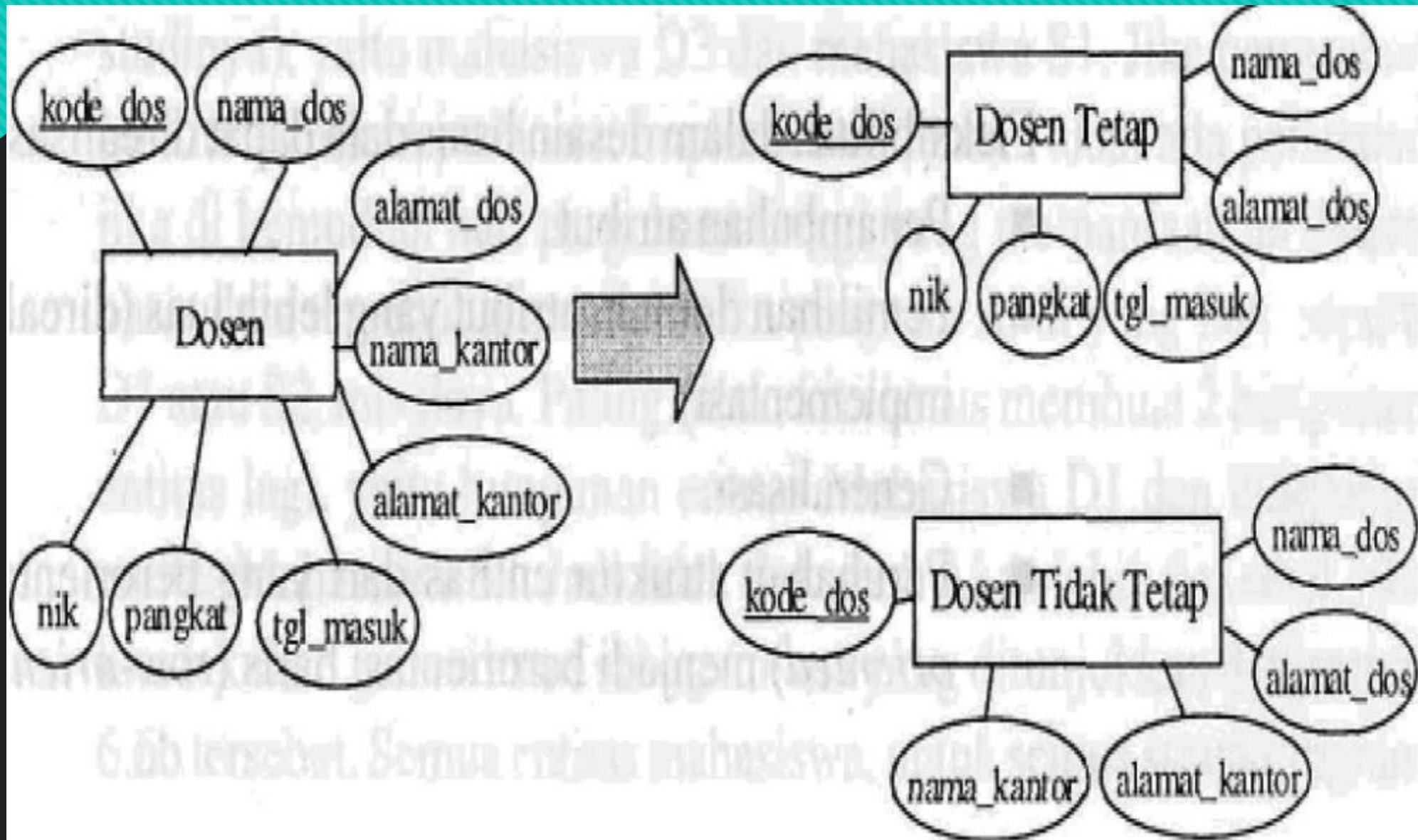


PROSES LANJUTAN / FINAL DESIGN



PROSES LANJUTAN / FINAL DESIGN

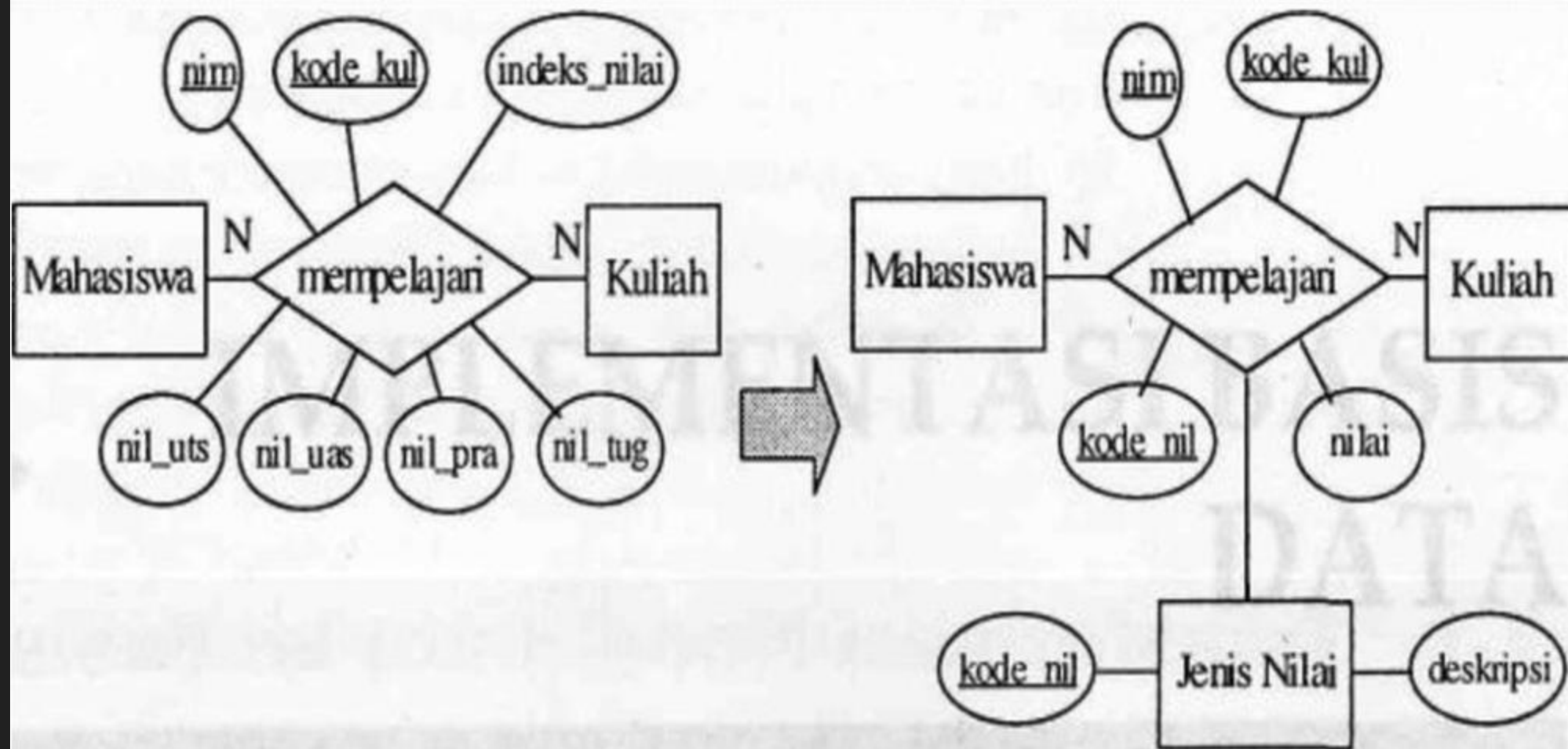
- Dekomposisi Entitas / Horizontal



PROSES LANJUTAN / FINAL DESIGN

Fleksibilitas

- Fleksibilitas dalam desain basis data dapat direalisasikan dalam bentuk:
 1. Penambahan atribut.
 2. Pemilihan domain atribut yang lebih luas (direalisasikan pada tahap implementasi).
 3. Generalisasi.
 4. Perubahan struktur entitas dari yang berorientasi kolom (column-oriented) menjadi berorientasi baris (row-oriented).



REFERENSI

- Fathansyah. 2012. Basis Data. Bandung: Informatika Bandung.

TERIMA KASIH