Achmad Bisri

Pengantar dan Konsep Dasar Basis Data

APA ITU BASIS DATA ?

- Basis data (*database*) dapat digambarkan seperti lemari arsip. Apabila kita memiliki sebuah lemari arsip dan bertugas untuk mengelolanya, maka kemungkinan besar kita akan melakukan hal-hal seperti :
  - Memberi sampul (map) pada kumpulan (bundel) arsip yang akan disimpan.
  - Menentukan kelompok/jenis arsip
  - Memberi penomoran dengan pola tertentu yang nilainya unik pada setiap sampul (map).
  - Menempatkan arsip-arsip tersebut dengan cara/urutan tertentu didalam lemari.

DEFINISI

- Basis data terdiri atas 2 kata, yaitu Basis dan Data. Basis dapat diartikan sebagai kumpulan, tempat, markas, gudang.
- Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti:
  - Manusia (pegawai, mahasiswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, angka, huruf, simbol, teks, gambar, bunyi dan sebagainya.

Arsip/Data

KEMUNCULAN BASIS DATA

- Menurut sejarah, sistem pemrosesan basis data terbentuk setelah masa sistem pemrosesan manual dan sistem pemrosesan berkas.
- Sistem pemrosesan manual (berbasis kertas) merupakan bentuk pemrosesan yang menggunakan dasar berupa setumpuk rekaman yang disimpan pada rak-rak berkas. Jika sesuatu berkas diperlukan, berkas tersebut harus dicari pada rak-rak tersebut. Bentuk seperti ini dapat dijumpai dalam kehidupan sehari-hari.

Lemari Arsip

BASIS DATA & LEMARI ARSIP (1)

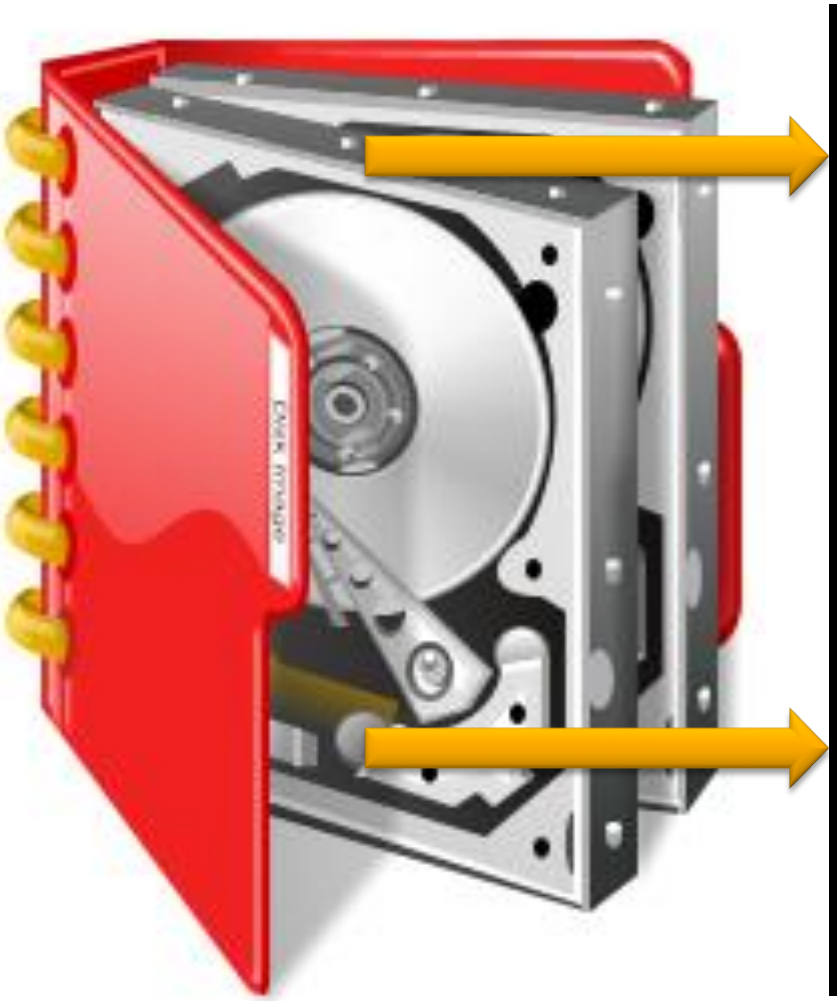
- Basis Data dan lemari arsip sesungguhnya memiliki prinsip kerja dan tujuan yang sama. Prinsip utamanya adalah pengaturan data/arsip. Dan tujuan utamanya adalah **kemudahan dan kecepatan dalam pengambilan kembali data/arsip**. Perbedaannya hanya terletak pada **media penyimpanan** yang digunakan. Jika lemari arsip menggunakan lemari dari besi atau kayu sebagai media penyimpanan, maka basis data menggunakan media penyimpanan elektronis seperti disk (*flashdisk* atau *harddisk*).

BASIS DATA & LEMARI ARSIP (2)

- Hal ini merupakan konsekuensi yang logis, karena lemari arsip langsung dikelola/ditangani oleh manusia, sementara basis data dikelola/ditangani melalui perantaraan alat/mesin pintar elektronis (yang kita kenal sebagai komputer). Perbedaan media ini yang selanjutnya melahirkan perbedaan-perbedaan lain yang menyangkut jumlah dan jenis metoda/cara yang dapat digunakan dalam upaya penyimpanan.

BASIS DATA & LEMARI ARSIP

BASIS DATA & LEMARI ARSIP



BASIS DATA (*DATABASE*)

- **Menurut Gordon C. Everest** :

  Database adalah koleksi atau kumpulan data yang mekanis, terbagi/shared, terdefinisi secara formal dan dikontrol terpusat pada organisasi.

- **Menurut C.J. Date :**

  Database adalah koleksi "data operasional" yang tersimpan dan dipakai oleh sistem aplikasi dari suatu organisasi.

  - Data input adalah data yang masuk dari luar sistem
  - Data output adalah data yang dihasilkan sistem
  - Data operasional adalah data yang tersimpan pada sistem

- **Menurut Toni Fabbri :**

  Database adalah sebuah sistem file-file yang terintegrasi yang mempunyai minimal primary key untuk pengulangan data.

- **Menurut S. Attre :**

  Database adalah koleksi data-data yang saling berhubungan mengenai suatu organisasi / enterprise dengan macam-macam pemakaiannya.

SUDUT PANDANG BASIS DATA

- Basis Data sendiri dapat didefinisikan dalam sejumlah sudut pandang, seperti :
  - Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
  - Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
  - Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.
  - Kumpulan data yang disusun sedemikian rupa sehingga menghasilkan informasi yang berguna

METADATA

- **Metadata** adalah data yang menggambarkan sifat dan konteks dari data pengguna atau informasi yang tersimpan mengenai database, biasanya disimpan dalam sebuah kamus data atau katalog. Kamus data dapat berisi informasi user, hak istimewa (*privileges*) dan struktur internal database.

APA BEDANYA DATA DENGAN INFORMASI ?

- **Data** adalah fakta – fakta yang dapat disimpan dan mempunyai arti.
- **Informasi** adalah data yang dapat diolah sehingga menghasilkan sesuatu yang bermanfaat bagi orang yang menggunakannya / menerimanya.

Data Elektronis = Basis Data ?

CONTOH KASUS

Setiap data elektronis = Basis data ?

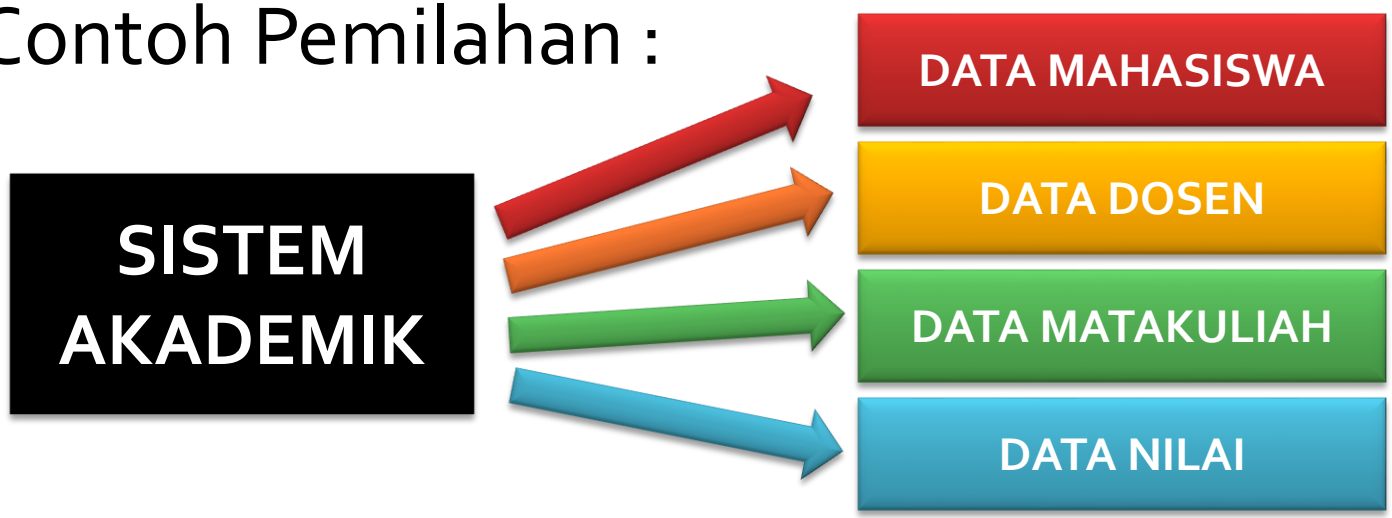
Bagian administrasi akademik pada Perguruan Tinggi Universitas Pamulang selalu menggunakan komputer dengan aplikasi MS. Excel untuk mengolah data mahasiswa, dosen, matakuliah, nilai, dll.

Apakah dapat dikatakan PT UNPAM telah menerapkan basis data ?

JAWABAN

Belum tentu, karena di dalam pengelolaannya belum tentu terdapat pemilahan dan pengelompokan data sesuai jenis / fungsi data.

Contoh Pemilahan :

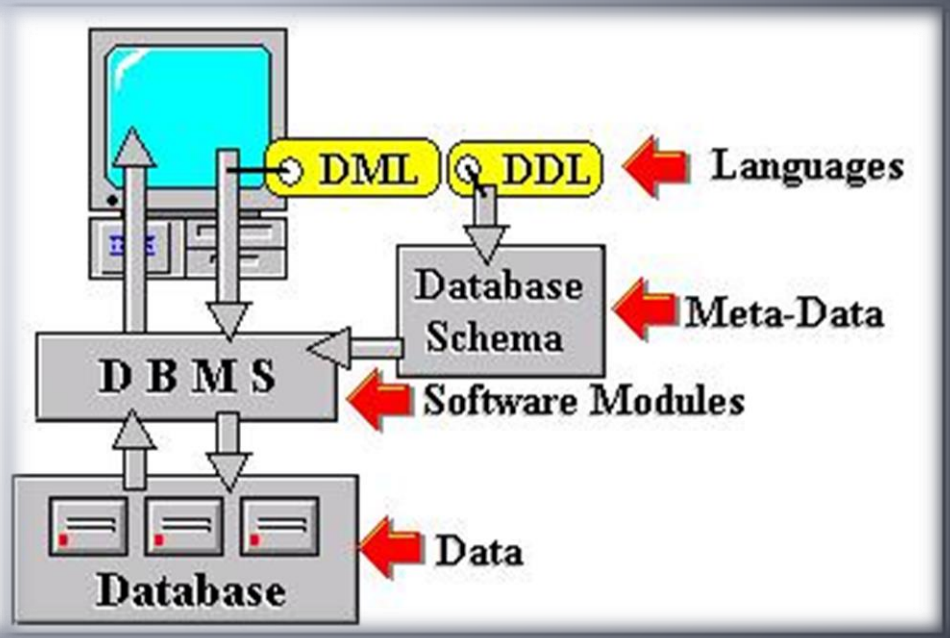


- Hal penting yang harus diperhatikan,
  - Bahwa basis data bukan hanya sekedar penyimpanan data elektronis (dengan bantuan komputer). Artinya, **tidak semua bentuk penyimpanan data elektronis bisa disebut basis data**.
- Contoh
  - Penyimpanan dokumen berisi data dalam bentuk file teks (dengan program pengolah kata), file spreedsheet, dll.

- Contoh diatas tidak bisa disebut sebagai basis data, karena didalamnya tidak ada **pemilihan** dan **pengelompokan** data sesuai jenis/fungsi data, sehingga akan menyulitkan pencarian data.
- Yang sangat ditonjolkan dalam basis data adalah **pengaturan, pemilahan, pengelompokan, pengorganisasian** data yang akan disimpan sesuai fungsi dan jenisnya.

Database Management System (DBMS)

Achmad Bisri



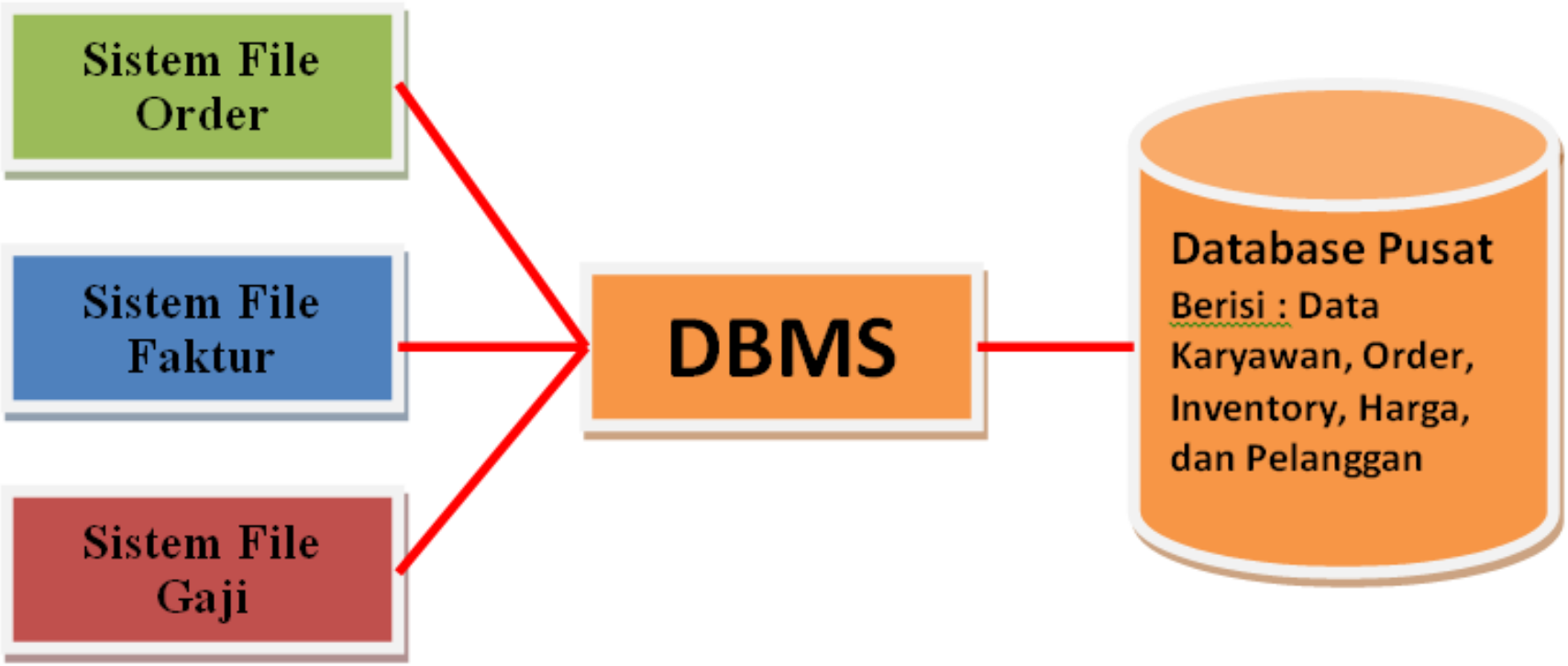
Database Management System (DBMS)

- Menutut C.J. Date : DBMS adalah merupakan *software* yang menghandel seluruh akses pada *database* untuk melayani kebutuhan user.
- Menurut S, Attre : DBMS adalah *software*, *hardware*, *firmware* dan *procedure-procedure* yang memanage *database*. *Firmware* adalah *software* yang telah menjadi modul yang tertanam pada hardware (ROM).
- Menurut Gordon C. Everest : DBMS adalah manajemen yang efektif untuk mengorganisasi sumber daya data.

Database Management System (DBMS)

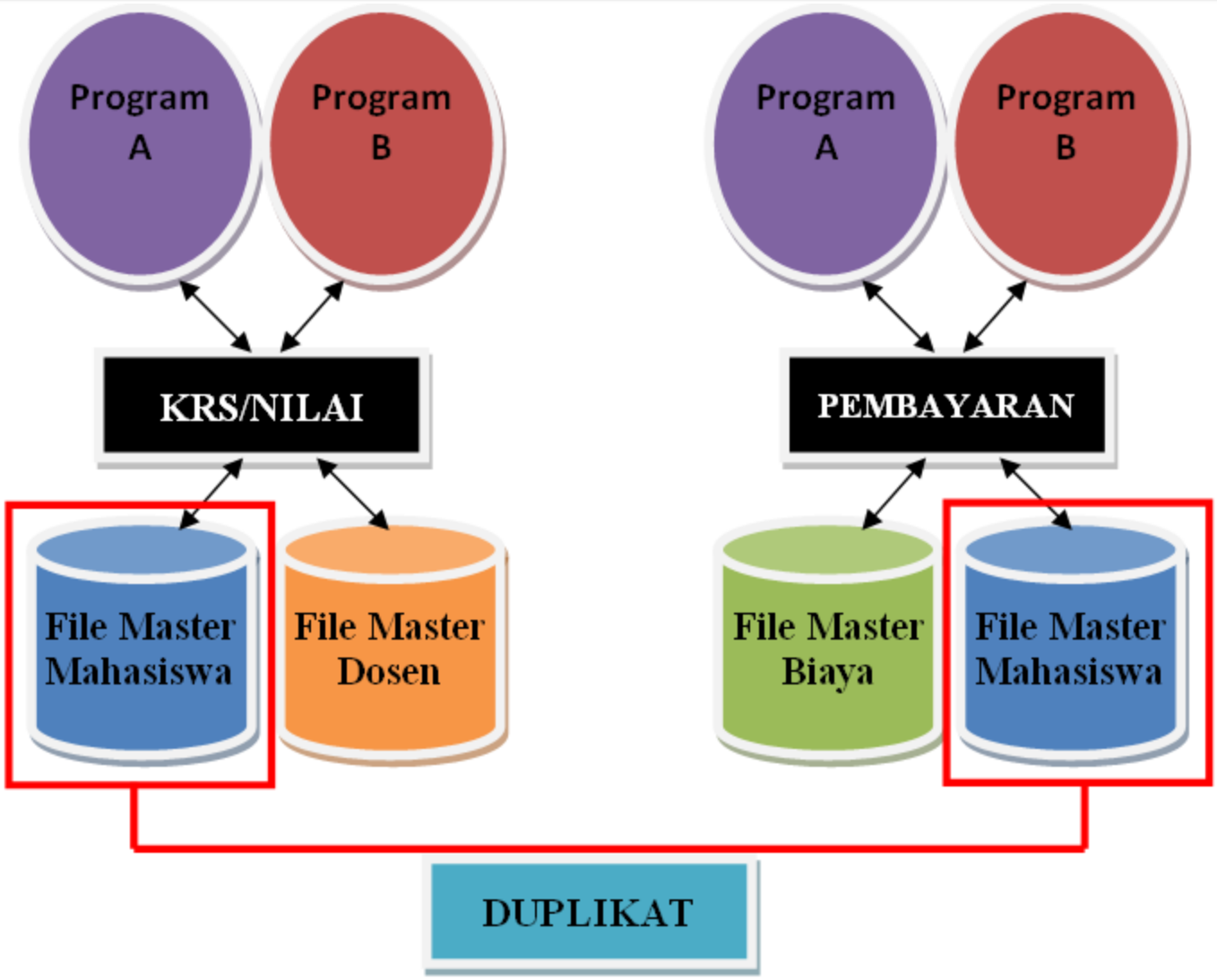
- Dapat disimpulkan bahwa DBMS merupakan suatu sistem perangkat lunak yang memungkinkan pengguna (*user*) untuk mendefinisikan, membuat, dan memelihara *database* maupun menyediakan akses yang terkontrol terhadap data.
- Istilah database kerap digunakan sebagai acuan terhadap data itu sendiri, namun demikian ada sejumlah komponen tambahan lainnya yang juga menjadi bagian dari suatu sistem manajemen database yang utuh.

Database Management System (DBMS)



Pendekatan Dengan Basis Data

Masalah Dengan Ketergantungan Data



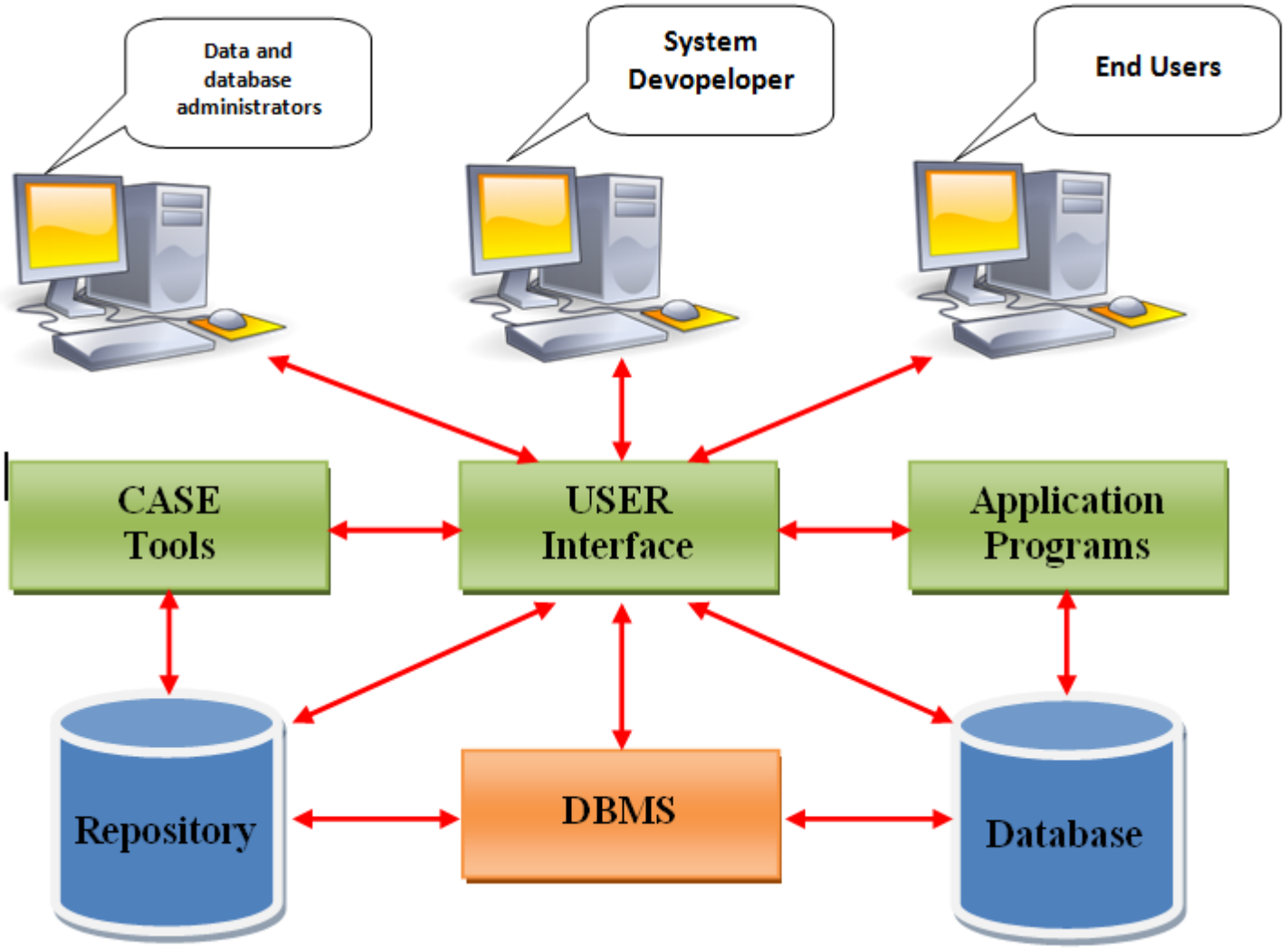
Masalah Dengan Data Redundansi

- Pemborosan ruang untuk memiliki duplikat data
- Sulit untuk pemeliharaannya
- Perubahan data dalam satu file dapat menyebabkan inkonsistensi
- Penyelesaian dalam integritas data

Solusi Dengan Pendekatan Database

- Repositori pusat data bersama (*Central repository of shared data*)
- Data dikelola oleh agen pengendalian (*Data is managed by a controlling agent*)
- Disimpan dalam bentuk, standar nyaman (*Stored in a standardized, convenient form*)
- Membutuhkan Sistem Manajemen Database (*Requires a Database Management System*)

Komponen Lingkungan Basis Data



Komponen Lingkungan Basis Data

1. Perangkat lunak (*software*/aplikasi)
2. Perangkat keras (*hardware*)
3. Pengguna (*user*)
4. Data

1. Perangkat Lunak (*Software*)

- Merupakan interface antara user dengan data fisik pada database seperti : *Database Management System* (DBMS), program aplikasi / prosedur – prosedur.

2. Perangkat Keras (*Hardware*)

- Terdiri dari peralatan perangkat keras komputer yang berfungsi untuk mengelola sistem database seperti : peralatan penyimpanan (*disk*), peralatan input/output, peralatan komunikasi data, dll.

3. Pengguna (*User*)

Pengguna dibagi menjadi tiga bagian yaitu :

- *Database Administrator* (*DBA*), orang yang bertugas / bertanggungjawab terhadap pengelolaan sistem database.
- *Programmer* : orang yang bertugas / bertanggungjawab terhadap pembuatan aplikasi dan mengakses database dengan menggunakan bahasa pemrograman.
- *End User* : orang yang mengakses database dengan menggunakan bahasa query atau program aplikasi yang telah dibuat oleh programmer.

Tugas/Tanggungjawab DBA

- Mendefinisikan basis data
- DBA menentukan isi basis data
- Menentukan sekuritas basis data
  - Setiap pengguna diberi hak akses terhadap basis data secara tersendiri. Tidak semua pengguna bisa menggunakan data yang bersifat sensitif. Penentuan hak akses disesuaikan dengan wewenang pengguna dalam organisasi.
- Memantau kinerja sistem
  - Secara periodik DBA memantau kinerja DBMS. Termasuk dalam hal ini adalah pemantauan waktu tanggapan selama beban puncak. Informasi yang diperoleh dapat digunakan untuk menentukan perlu tidaknya pengembangan sistem perangkat keras dimasa mendatang ataupun melakukan perubahan organisasi didalam basis data.

Tugas/Tanggungjawab DBA

- Merencanakan backup dan recovery
  - DBA sebagai pembuat panduan, prosedur serta standar untuk melakukan pencadangan data (backup) terhadap basis data. Begitu juga proses pemulihan data (recovery) bila terjadi kerusakan data pada sistem.
- Mengikuti perkembangan produk
  - DBA juga bertanggungjawab terhadap perkembangan produk (Versi DBMS terbaru, tool, dan perangkat pendukung) sehingga dapat memberikan usulan kepada organisasi untuk melakukan hal-hal yang dipandang perlua, seperti melakukan pelatihan kepada pemrogram aplikasi.

Program Utility DBA

- ***Create Routine*** : Untuk membuat database baru
- ***Reorganization Routine*** : Untuk menyusun kembali database (misal : untuk menghapus tempat-tempat kososng dari record-record yang sudah tidak berlaku)
- ***Journalizing / Logging Routine*** : Untuk mencatat semua operasi yang telah dikerjakan,siapa usernya.
- ***Recovery Routine*** : Memperbaiki kerusakan database pada posisi sebelum kerusakan.
- ***Statistical Analisys Routine*** : Untuk memonitor hasil-hasil database.

4. Data

- Terdiri dari tiga jenis data yaitu :
  - Data operasional : data informasi yang disimpan pada database
  - Data masukan : data yang dimasukan melalui peralatan input (keyboard) yang dapat menambah, merubah, dan menghapus data
  - Data keluaran : data/informasi sebagai hasil dari dalam sistem database

Biaya & Resiko (Kerugian) Pendekatan Basisdata

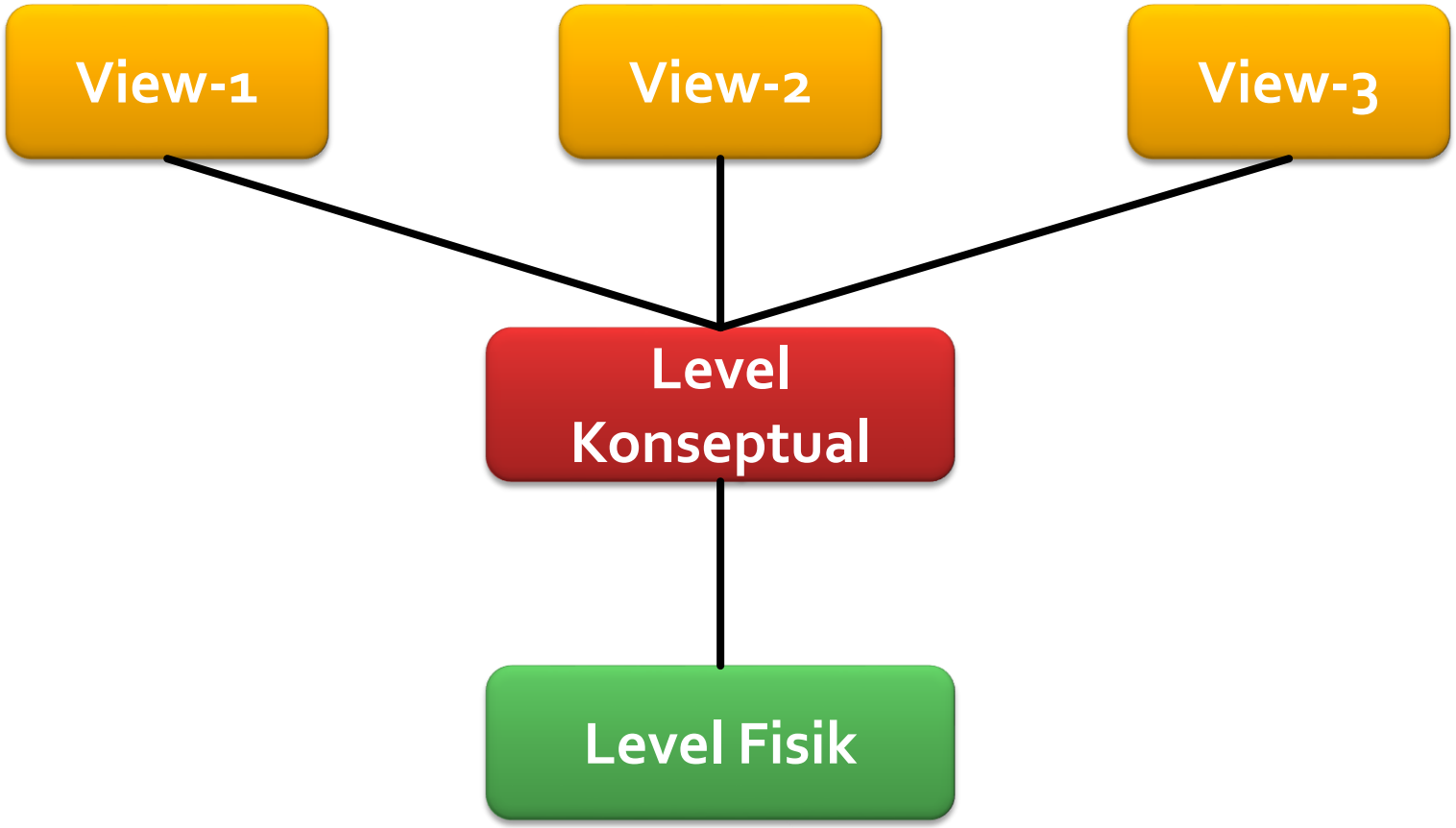
- Hal yang baru bagi user yang belum pernah menggunakan database
- Biaya untuk instalasi dan manajemen serta kompleksitas
- Biaya untuk konversi dari yang lama (konvensional) kepada database
- Membutuhkan backup dan recovery data secara eksplisit
- Adanya konflik dalam organisasi/perusahaan dengan menggunakan database

Abstraksi Data

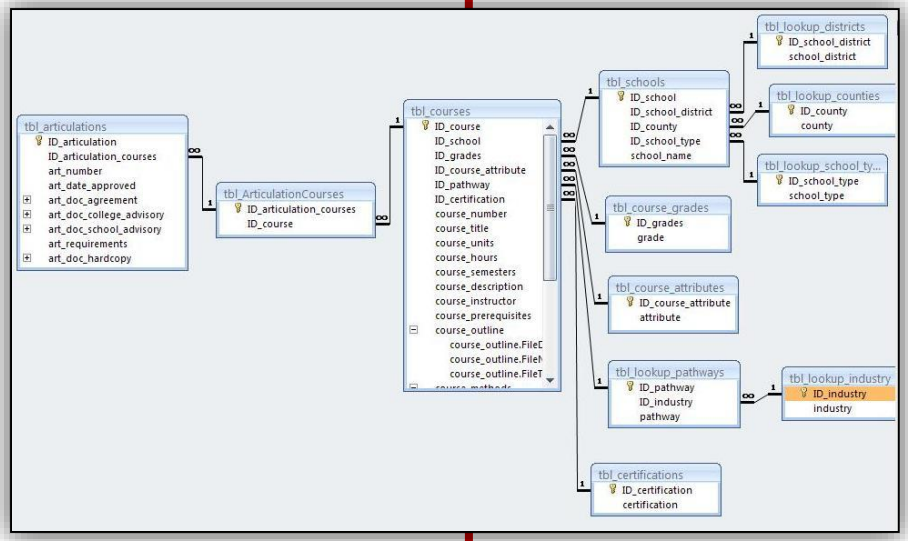
ABSTRAKSI DATA

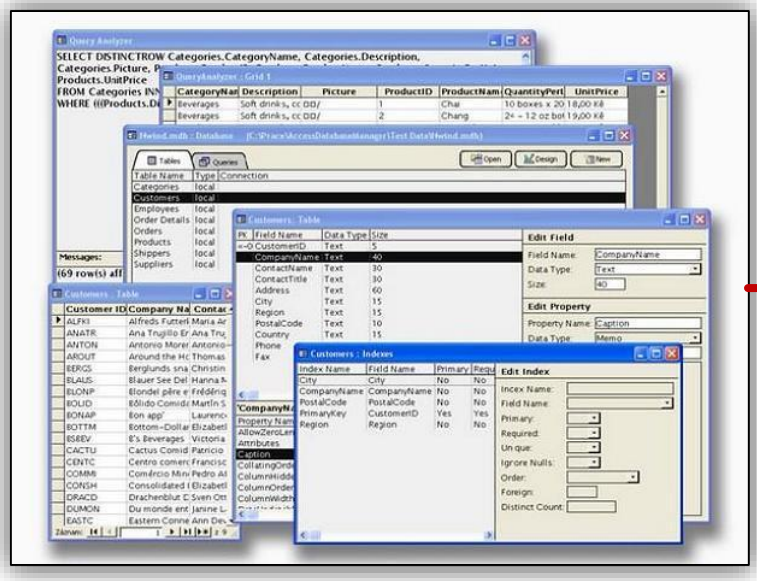
- Sebuah sistem biasanya akan menyembunyikan detail tentang bagaimana data itu disimpan dan dipelihara. Karena itu seringkali data yang terlihat oleh pemakai berbeda dengan yang tersimpan secara fisik.
- Abstraksi data merupakan tingkatan (*level*) dalam, bagaimana melihat data dalam sebuah sistem basis data.

Tiga Level Abstraksi Data




VISA
DEBIT CARD
VISA


tbl_articulations
ID_articulation
ID_articulation_courses
art_number
art_date_approved
art_doc_agreement
art_doc_college_advisory
art_doc_school_advisory
art_requirements
art_doc_hardcopy
tbl_ArticulationCourses
ID_articulation_courses
ID_course
tbl_courses
ID_course
ID_school
ID_grades
ID_course_attribute
ID_pathway
ID_certification
course_number
course_title
course_units
course_hours
course_semesters
course_description
course_instructor
course_prerequisites
course_outline
course_outline.FileE
course_outline.FileN
course_outline.FileT
tbl_schools
ID_school
ID_school_district
ID_county
ID_school_type
school_name
tbl_lookup_districts
ID_school_district
school_district
tbl_lookup_counties
ID_county
county
tbl_lookup_school_ty...
ID_school_type
school_type
tbl_course_grades
ID_grades
grade
tbl_course_attributes
ID_course_attribute
attribute
tbl_lookup_pathways
ID_pathway
ID_industry
pathway
tbl_lookup_industry
ID_industry
industry
tbl_certifications
ID_certification
certification


Query Analyzer
SELECT DISTINCTROW Categories.CategoryName, Categories.Description,
Categories.Picture,
Products.UnitPrice
FROM Categories INN
WHERE (((Products.D
QueryAnalyzer : Grid 1
CategoryNar
Description
Picture
ProductID
ProductNam
QuantityPert
UnitPrice
Beverages
Soft drinks, cc DD/
10 boxes x 20 18,00 Kč
24 - 12 oz bot 19,00 Kč
Chang
Messages:
(69 row(s) aff
Tables
Queries
Open
Design
New
Table Name
Type
Connection
Categories
Customers
Employees
Order Details
Orders
Products
Shippers
Suppliers
local
Customers : Table
Field Name
Data Type
Size
CustomerID
CompanyName
ContactName
ContactTitle
Address
City
Region
PostalCode
Country
Phone
Fax
Text
Edit Field
Field Name:
Data Type:
Size:
Edit Property
Property Name:
Caption
Memo
Customers : Indexes
Index Name
Field Name
Primary
Edit Index
Incex Name:
Field Name:
Primary:
Required:
Un que:
Ignore Nulls:
Order:
Foreign:
Distinct Count:

Level Penampakan (*View level*)

- Merupakan level abstraksi data yang hanya menunjukan sebagian dari basis data. Banyak *user* dalam sistem basis data tidak akan terlibat (*concern*) dengan semua data/informasi yang ada/disimpan.

Level Konseptual (*Conseptual level*)

- Merupakan abstraksi data yang menggambarkan data apa yang sebenarnya (secara fungsional) disimpan dalam basis data dan hubungannya dengan data yang lain.
- Misalnya, mengetahui bahwa data pegawai disimpan/direpresentasikan dalam beberapa file/tabel.

Level Fisik (*Physical level*)

- Merupakan level terendah dalam abstraksi, yang menunjukan bagaimana sesungguhnya suatu data disimpan.
- Pemakai melihat data sebagai gabungan struktur dan datanya sendiri.
- Pada level ini pemakai berkompeten dalam mengetahui bagaimana representasi fisik dari penyimpanan/pengorganisasian data

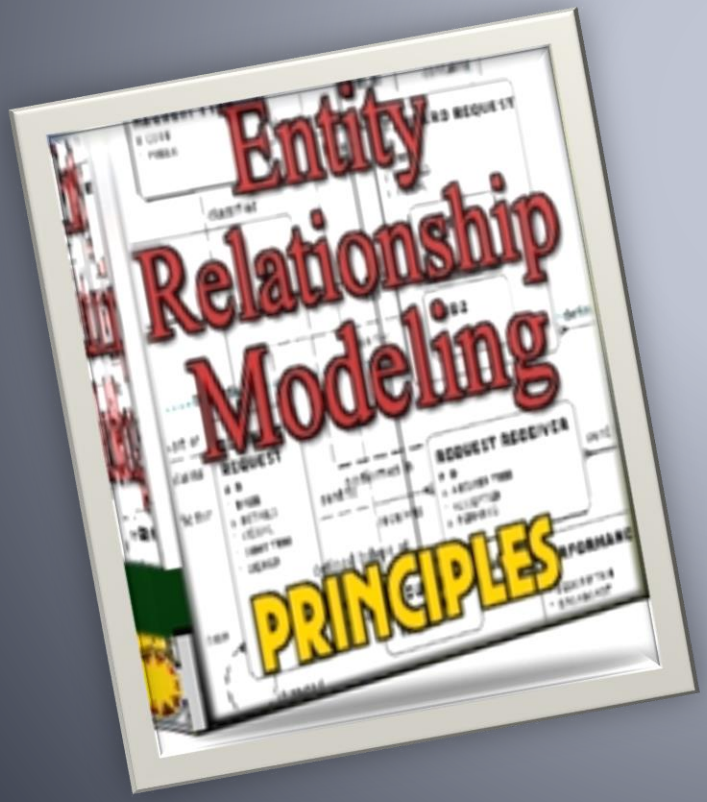
Bahasa Basis Data (*Database Language*)

- *Database Management System* (DBMS) merupakan perantara bagi pemakai dengan basis data dalam media penyimpanan (*disk*).
- Cara berinteraksi/berkomunikasi antara pemakai dengan basis data tersebut diatur dalam suatu bahasa khusus yang ditetapkan oleh perusahaan pembuat DBMS.

Bahasa Basis Data (*Database Language*)

- Bahasa itu dapat kita sebut sebagai bahasa basis data yang terdiri atas sejumlah perintah (*statement*) yang diformulasikan dan dapat diberikan user dan dikenali/diproses oleh DBMS untuk melakukan suatu aksi/pekerjaan tertentu.
- Sebuah bahasa basis data biasanya dapat dipilah kedalam beberapa bentuk :
  - *Data Definition Language* (DDL)
  - *Data Manipulation Language* (DML)
  - *Data Control Language* (DCL)

Model Data
Entity Relationship Diagram (ERD)



ER-Diagram

- ER-Diagram merupakan sebuah konsep untuk merancang basis data secara logis yang terdiri dari entitas, atribut, dan relasi sehingga dapat mendeskripsikan struktur basis data.

- ER-Diagram dikembangkan oleh Chen (1976)

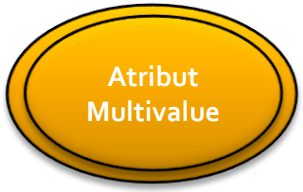
Komponen Dalam ER-Diagram

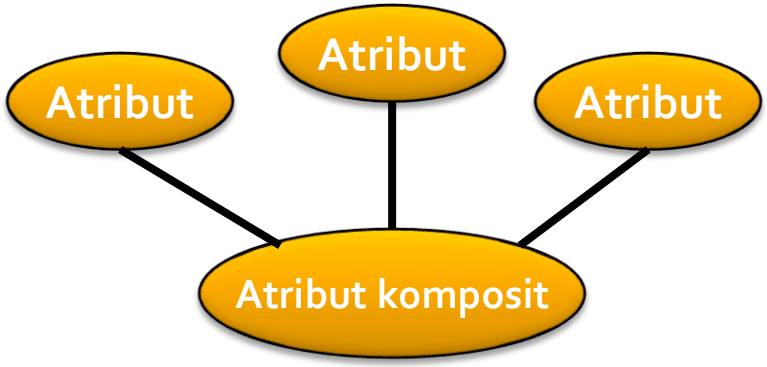
- Entitas (*entity*): objek dalam dunia nyata baik fisik maupun konsep.
  - Contoh: Mahasiswa, Dosen, Matakuliah.
- Atribut (*attribute*): sifat atau karakteristik dari suatu entitas.
  - Contoh: entitas → Mahasiswa

    memiliki atribut → NIM, Nama, Tgl_Lahir, Alamat
- Relasi (*relationship*): hubungan antara entitas yang satu dengan entitas lainnya.
  - Contoh: entitas Mahasiswa berhubungan dengan entitas Matakuliah.

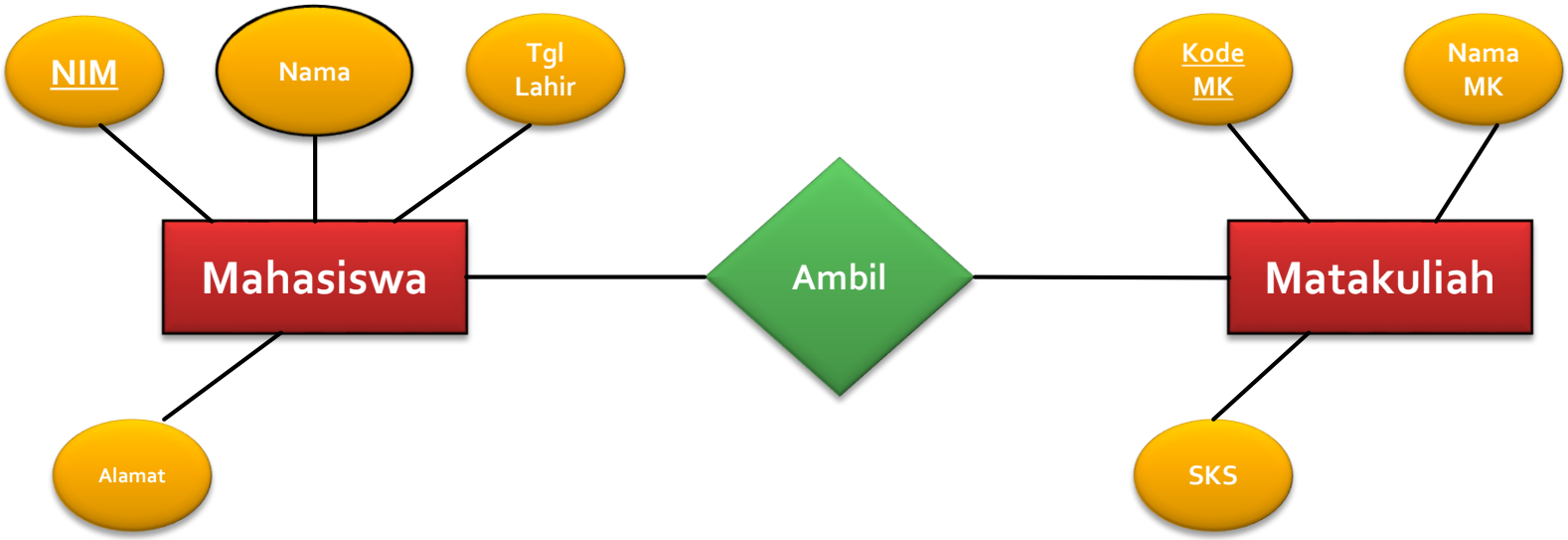
    Hubungannya yaitu Mahasiswa mengambil Matakuliah

Notasi ER-Diagram





Contoh ER-Diagram (Entitas, Atribut, dan Relasi)



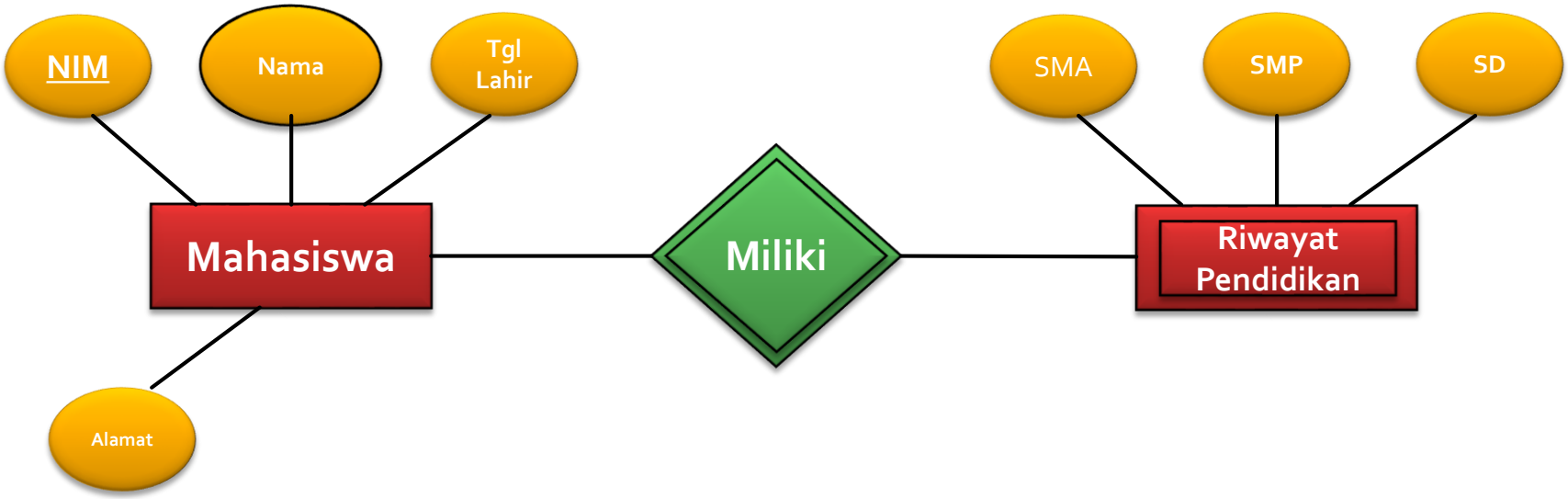
Entitas

- Entitas kuat: entitas yang berdiri sendiri
  - Contoh: Mahasiswa, Dosen, Matakuliah

- Entitas lemah: entitas yang keberadaannya bergantung pada entitas kuat
  - Contoh: RiwayatPendidikan

    RiwayatPendidikan bergantung pada entitas Mahasiswa

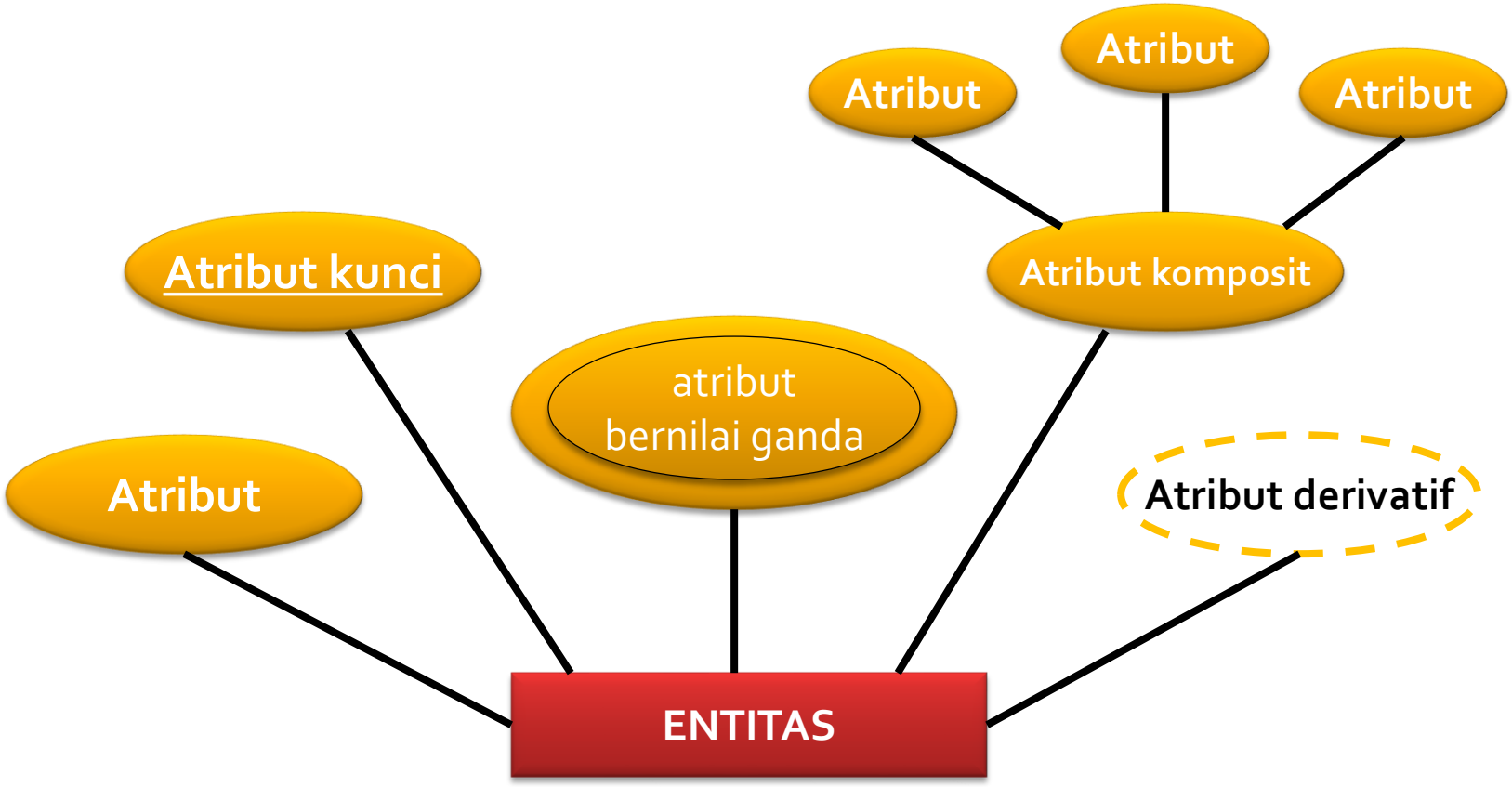
Contoh Entitas Kuat dan Lemah



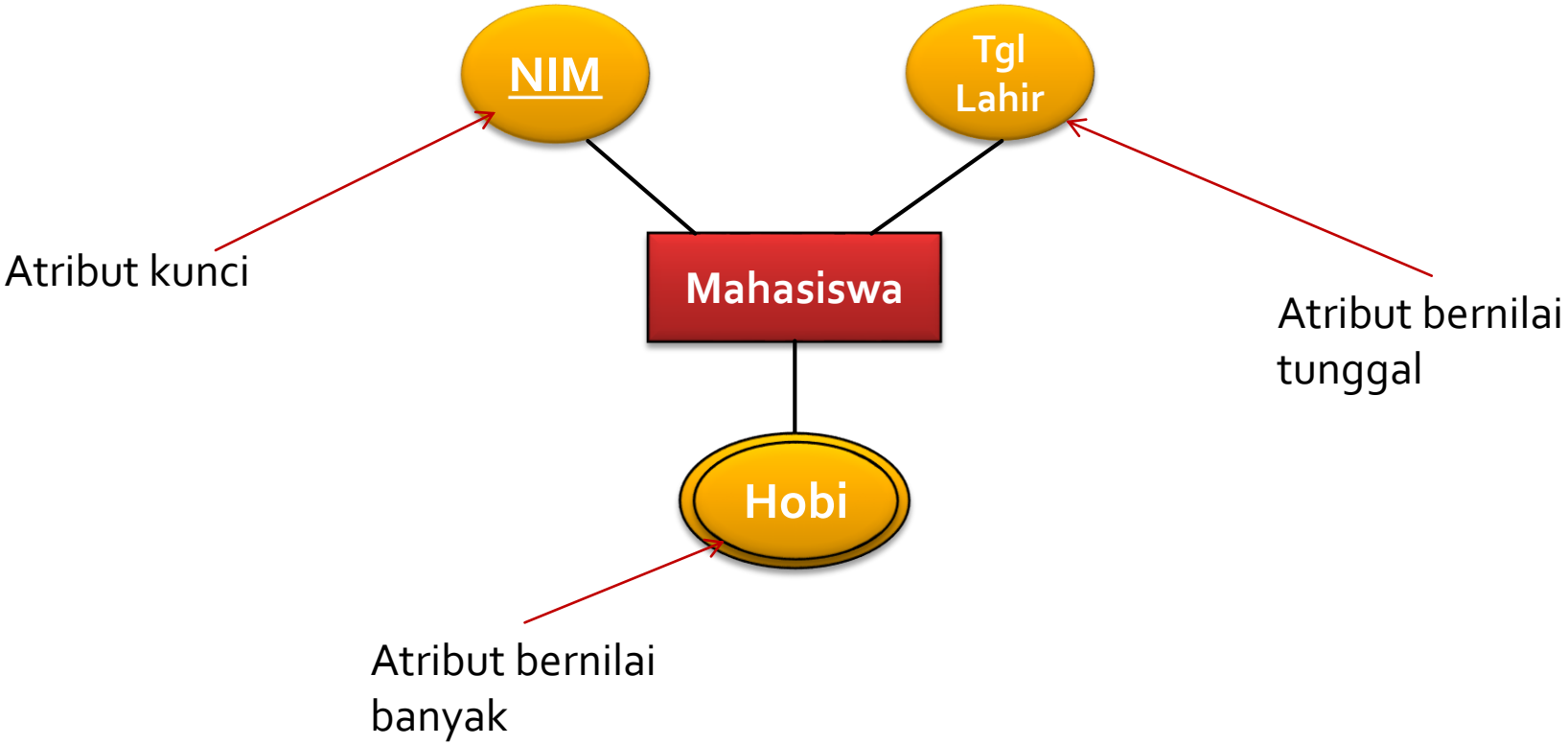
Macam Atribut

- Atribut kunci (*key attribute*): atribut yang bernilai tunggal dan unik.
- Atribut bernilai tunggal (*single valuen attribute*): atribut yang memiliki hanya satu nilai
- Atribut bernilai banyak (*multivalue attribute*): atribut yang memiliki sekelompok nilai
- Atribut komposit (*composite attribute*): atribut yang dapat dipecah/dijadikan beberapa atribut lain
- Atribut derivatif (*derivative attribute*): atribut turunan yang diperoleh dari atribut lain.

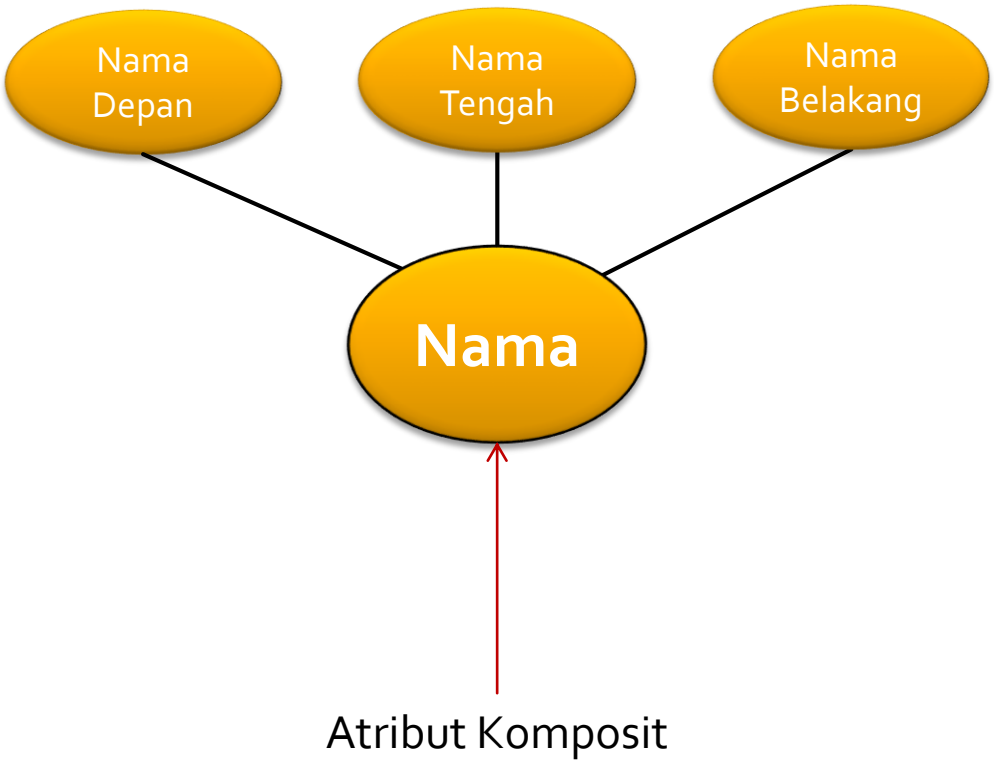
Notasi Atribut



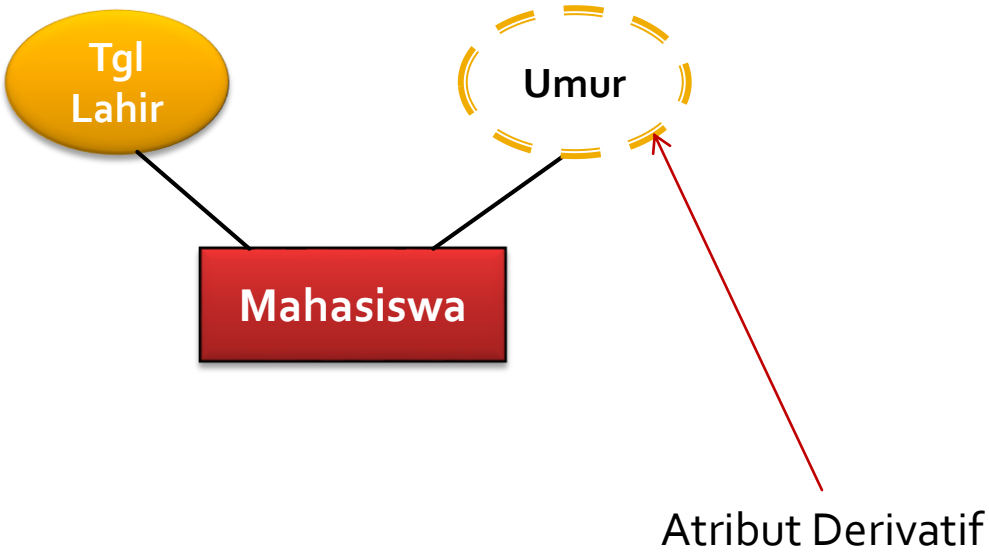
Atribut kunci, tunggal, banyak



Atribut Komposit



Atribut Derivatif

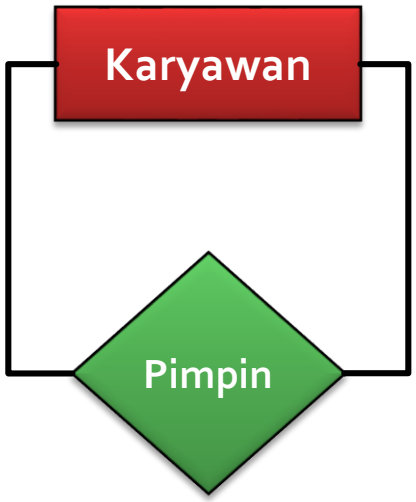


Derajat dari Relationship

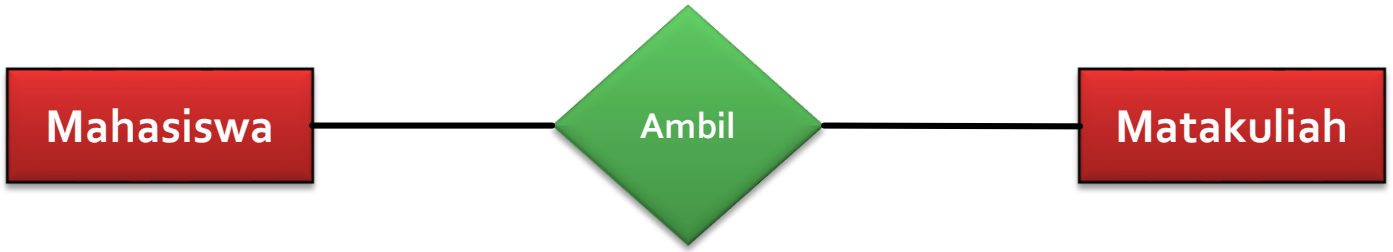
- Derajat dari *relationship* menjelaskan jumlah entitas yang berpartisipasi dalam suatu *relationship*

- Ada tiga derajat dari *relationship*:
  - Berderajat satu (*unary degree*): sebuah entitas berelasi dengan dirinya.
  - Berderajat dua (*binary degree*): dua buah entitas yang saling berhubungan.
  - Berderajat tiga (*ternary degree*): tiga buah entitas atau lebih yang saling berhubngan.

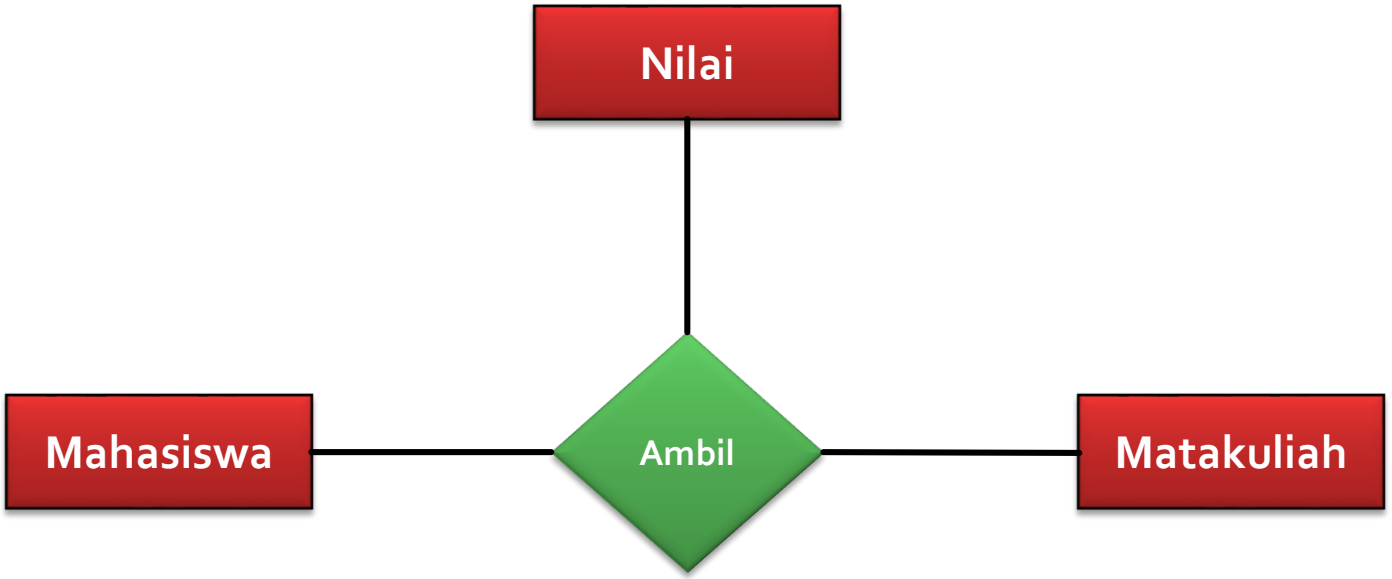
Contoh *Unary Degree*



Contoh *Binary Degree*



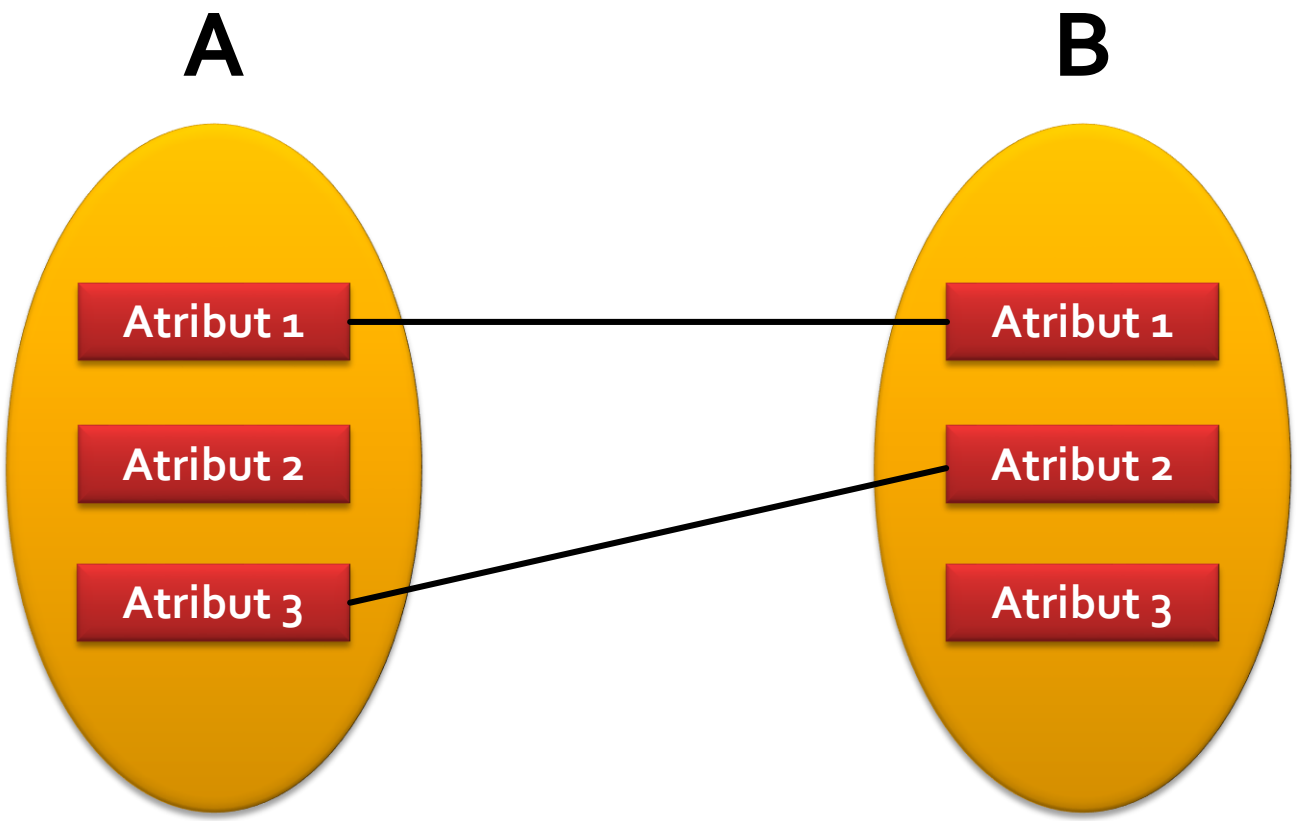
Contoh *Ternary Degree*



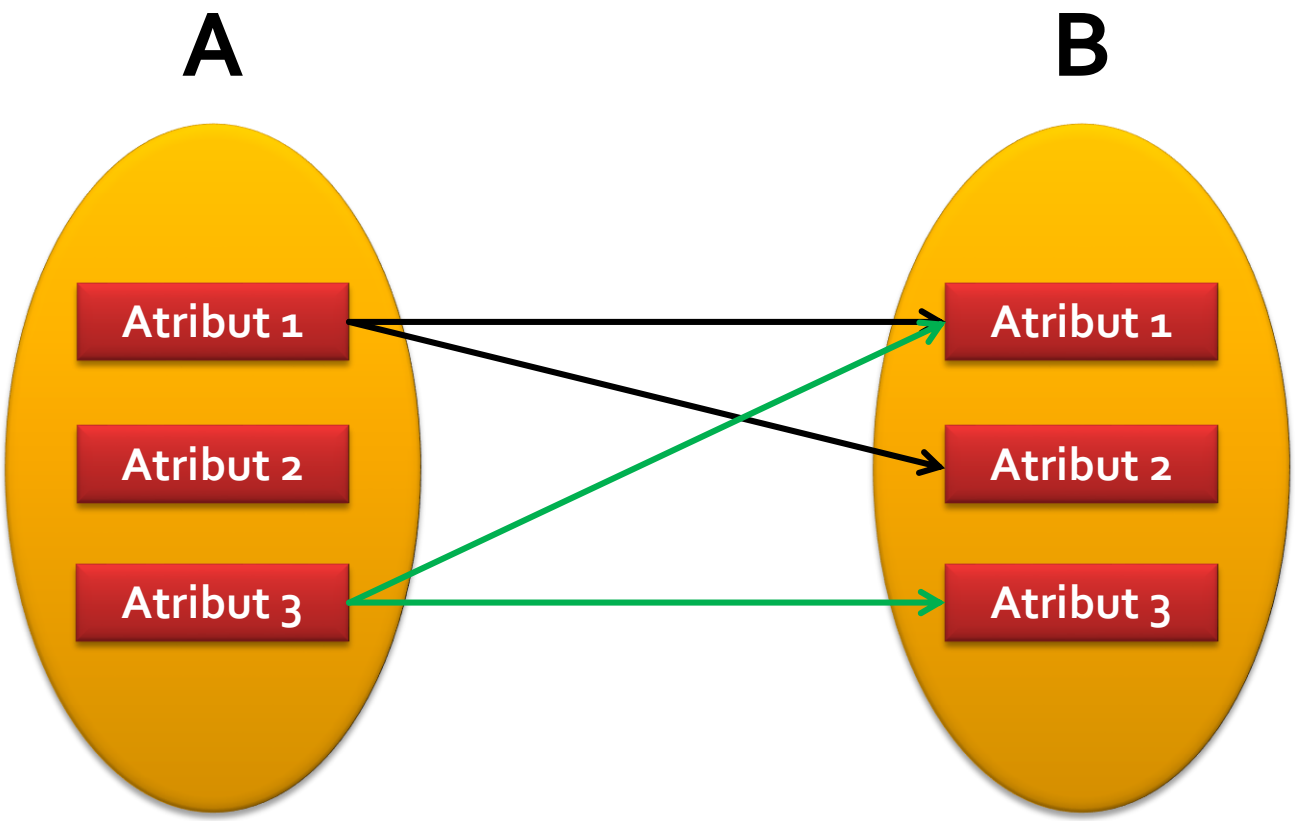
Rasio Kardinalitas (derajat relasi)

- Satu ke satu (*one to one*) → 1 - 1
  - Satu atribut yang berada pada entitas A hanya dapat berhubungan dengan satu atribut yang ada pada entitas B.
- Satu ke banyak (*one to many*) → 1 - N
  - Satu atribut yang berada pada entitas A dapat berhubungan dengan banyak atribut yang ada pada entitas B.
- Banyak ke banyak (*many to many*) → N - M
  - Banyak atribut yang berada pada entitas A dapat berhubungan dengan banyak atribut yang ada pada entitas B.

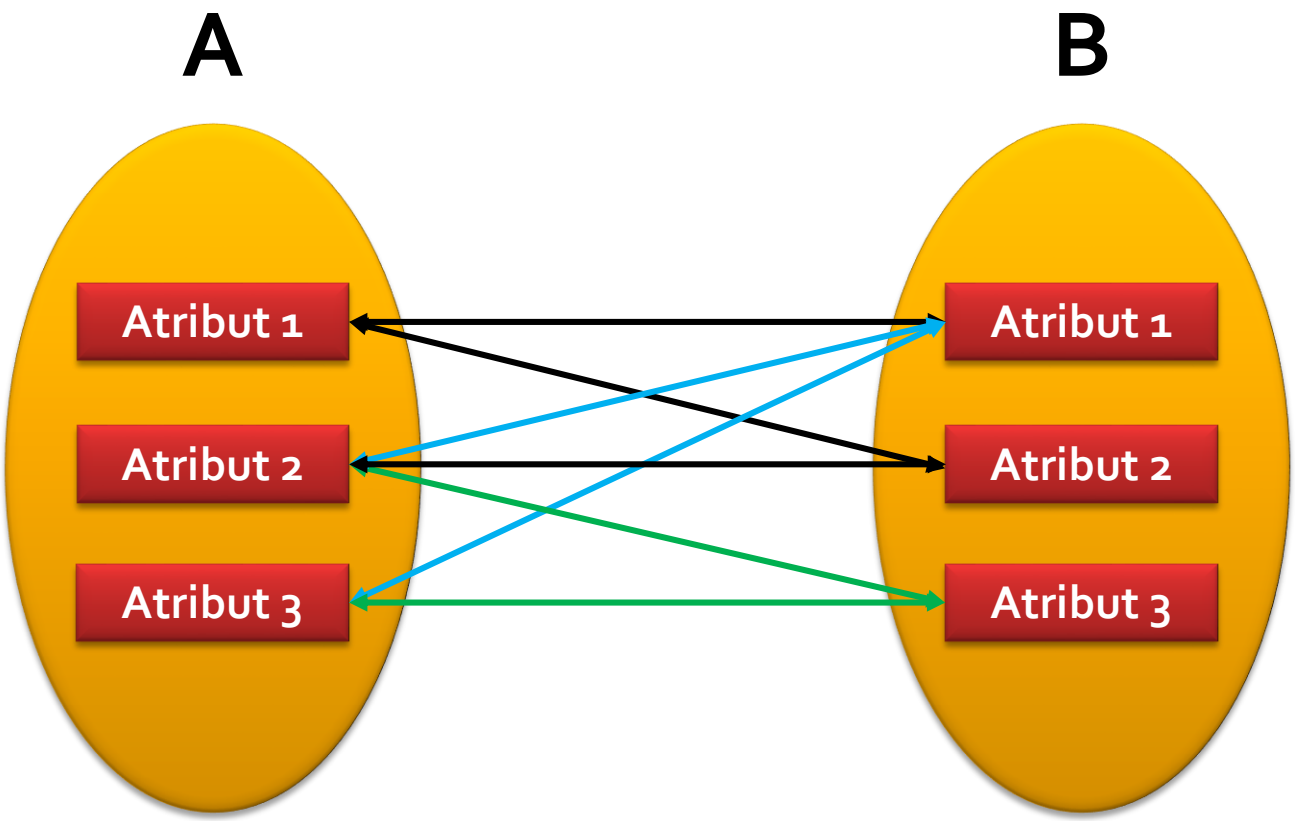
Contoh: satu ke satu



Contoh: satu ke banyak



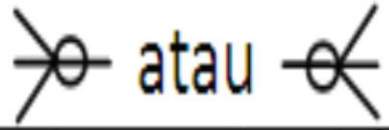
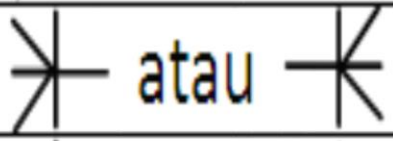
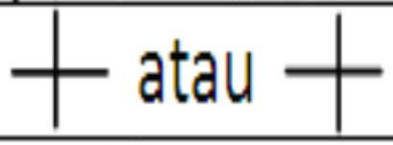
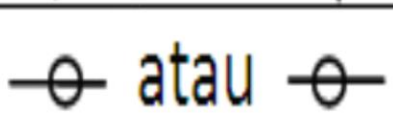
Contoh: banyak ke banyak



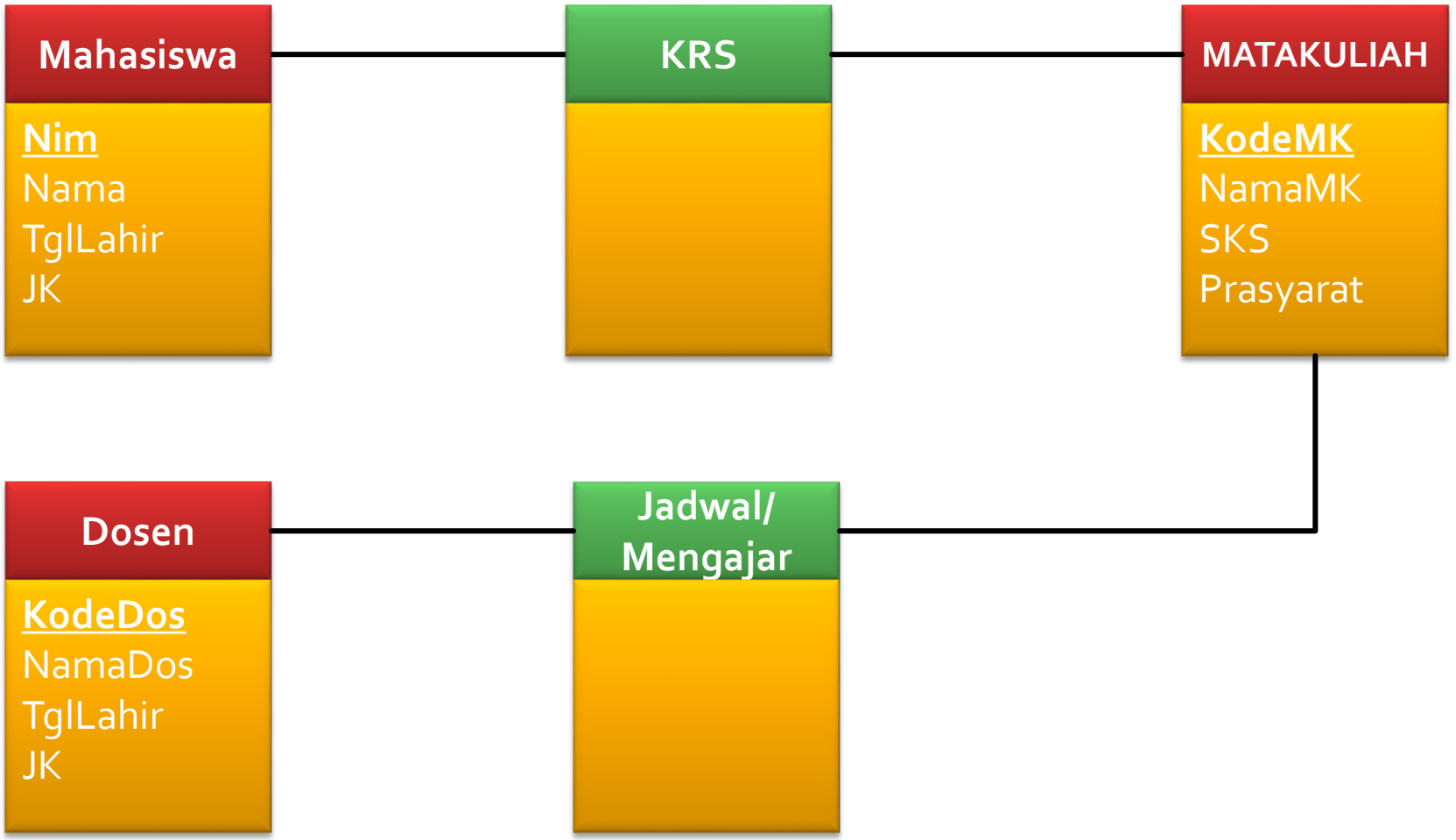
Kunci (*Key*)

- *Candidate key (CK)*: atribut yang dapat dijadikan sebagai calon kunci untuk menentukan kunci utama yang memiliki nilai keunikan.
- Primary key (PK): *candidate key* yang dipilih untuk mengidentifikasikan record secara unik dalam suatu tabel atau relasi.
- *Alternate key*: *candidate key* yang tidak dipilih sebagai primary key
- *Foreign key (FK)*: atribut kunci utama yang keberadaannya pada tabel/relasi lain dan dinyatakan sebagai kunci tamu (asing).
- *Super key (SK)*: kumpulan atribut yang mengidentifikasi sebuah record secara unik.

Rasio Derajat Relasi Minimum-Maksimum

Notasi	Derajat Relasi Minimum-Maksimum
 atau	(0, N)
 atau	(1, N)
 atau	(1, 1)
 atau	(0, 1)

Contoh

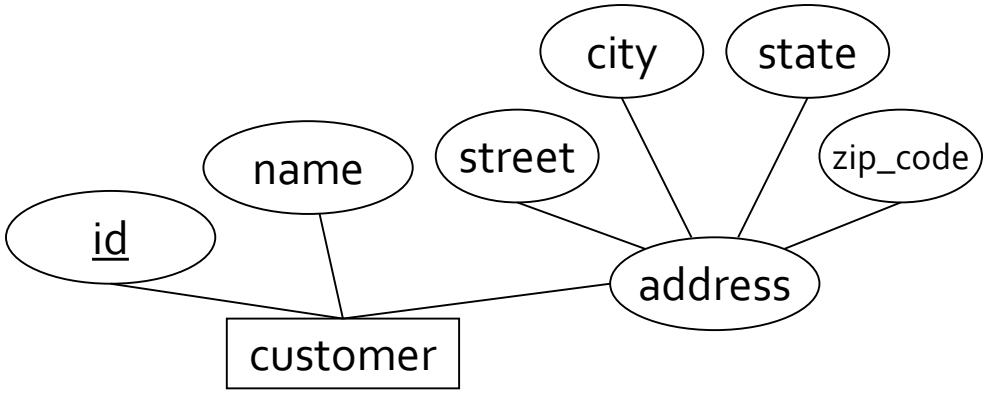

Mahasiswa
Nim
Nama
TglLahir
JK
KRS
MATAKULIAH
KodeMK
NamaMK
SKS
Prasyarat
Dosen
KodeDos
NamaDos
TglLahir
JK
Jadwal/
Mengajar

Transformasi Model Data

Transformasi E-R Diagram kedalam Basis Data Relasional

Tahap-Tahap Transformasi :

1. Entity-Relationship Diagram menjadi basis data.
2. Entity menjadi tabel dan atribut menjadi kolom/field dari tabel.
3. Entitas lemah → key dari "owner" (entitas kuat) ke tabel entitas lemah.
4. Setiap tipe entity dibuat suatu tabel yang memuat semua atribut simple, sedangkan untuk atribut komposit hanya dimuat komponen-komponennya saja. Contoh :

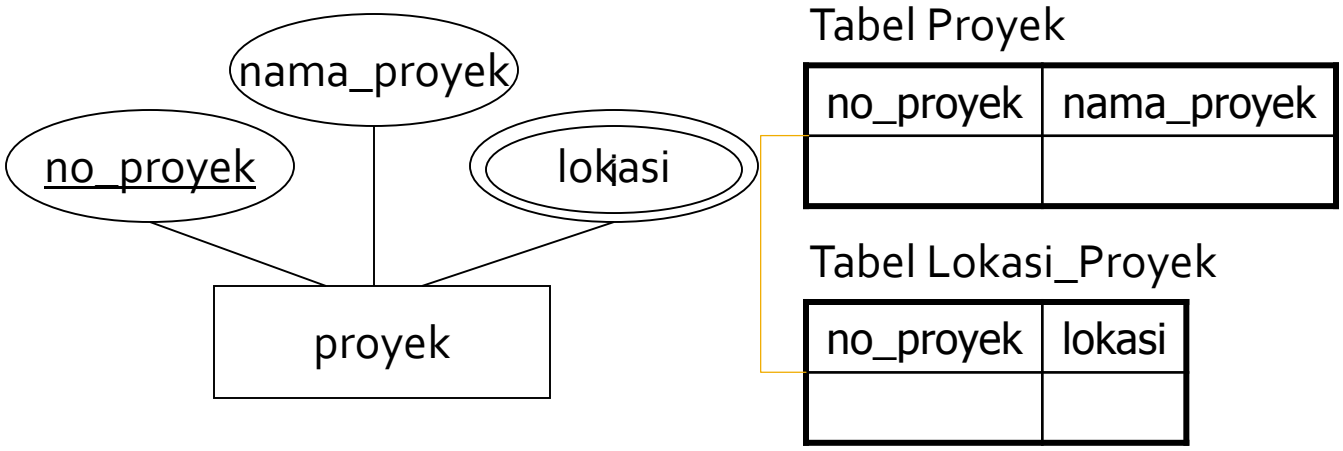


Tabel Customer

id	name	street	city	state	zip_code

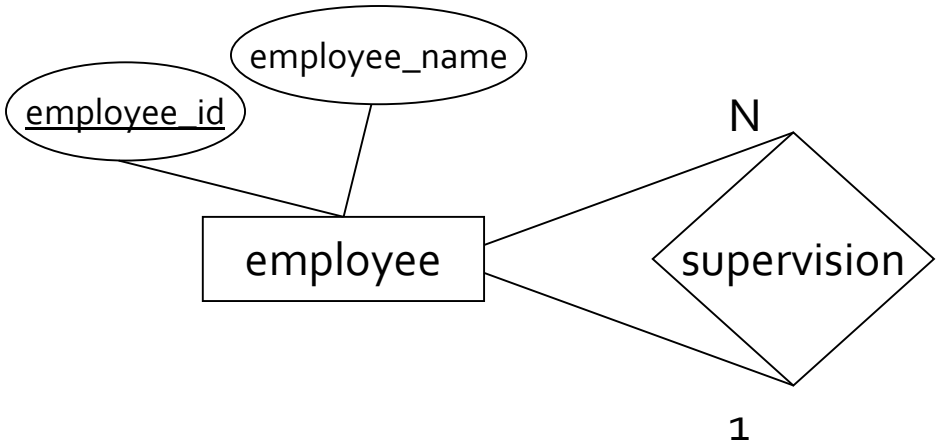
Transformasi E-R Diagram ke Basis Data Relational (lanj)

5. Setiap tabel yang mempunyai atribut multivalue, buatlah tabel baru dimana primary key-nya merupakan gabungan dari primary key dari tabel tersebut dengan atribut multivalue.



Transformasi E-R Diagram ke Basis Data Relational (lanj)

6. Setiap unary relationship 1:N, selain membuat tabel baru berdasarkan entity, buat juga tabel baru berdasarkan relationship-nya dengan atribut kunci tamu (foreign key) berdasarkan atribut kunci dari entity tersebut dan atribut kunci alternatif sebagai primary key-nya.



Tabel Employee

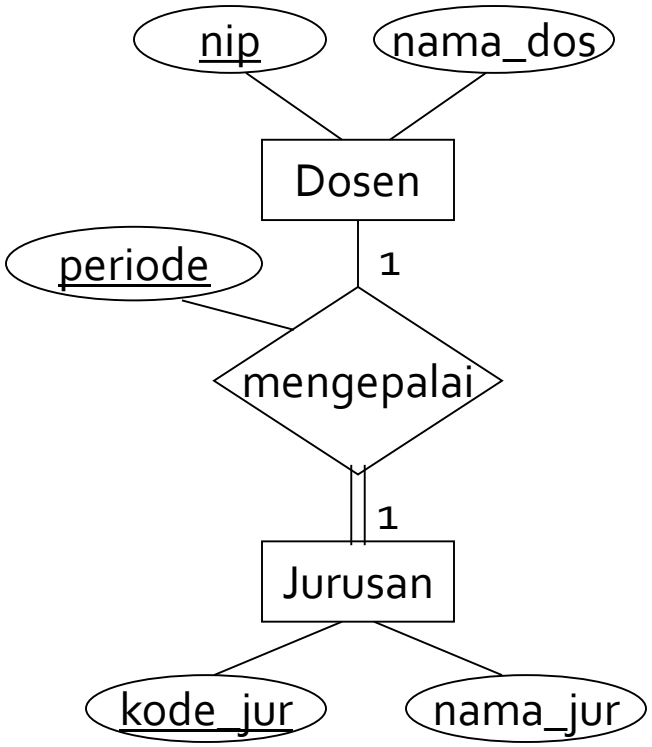
employee_id	employee_name

Tabel Supervision

supervisor_id	employee_id

Transformasi ER-Diagram ke Basis Data Relasional (lanj)

7. Untuk CR 1:1 dengan atau tanpa total participation maka akan dibuat tabel baru berdasarkan relationship, dimana kolom-kolomnya terdiri dari alternate key, dan primary key dari masing-masing entity.



Tabel Dosen

nip	Nama_dos

Tabel kaprodi

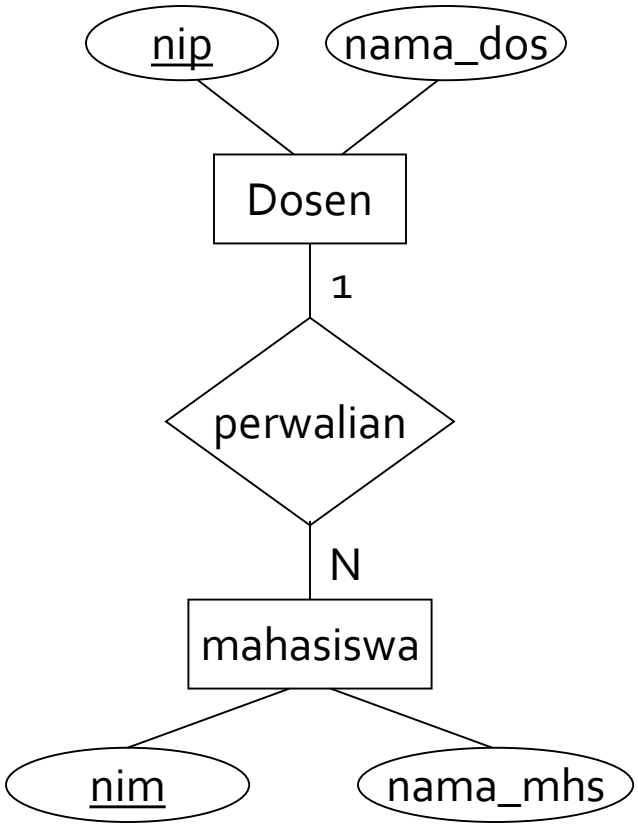
periode	kode_jur	nip

Tabel Jurusan

kode_jur	nama_jur

Transformasi ER-Diagram ke Basis Data Relasional (lanj)

8. Untuk CR 1:N dengan atau tanpa total participation maka primary key dari sisi 1 masuk ke sisi N.



Tabel Dosen

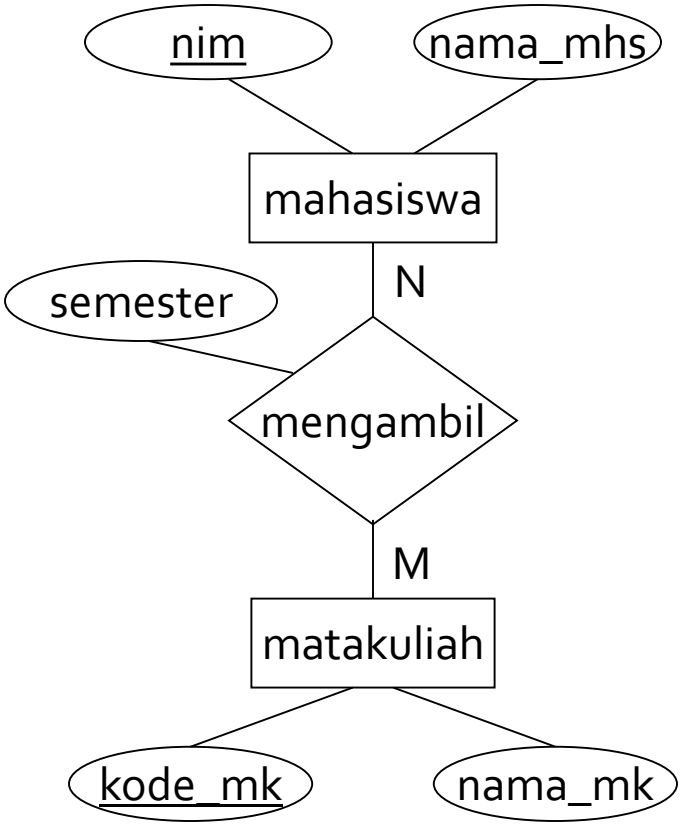
nip	nama_dos

Tabel Mahasiswa

nim	nama_mhs	nip

Transformasi ER-Diagram ke Basis Data Relasional (lanj)

9. Untuk CR M:N → dibuat tabel tersendiri berdasarkan relationshipnya dengan kolom-kolomnya terdiri dari alternate key dan primary key dari masing-masing entity.



Tabel Mahasiswa

nim	nama_mhs

Tabel KRS

semester	nim	kode_mk

Tabel Matakuliah

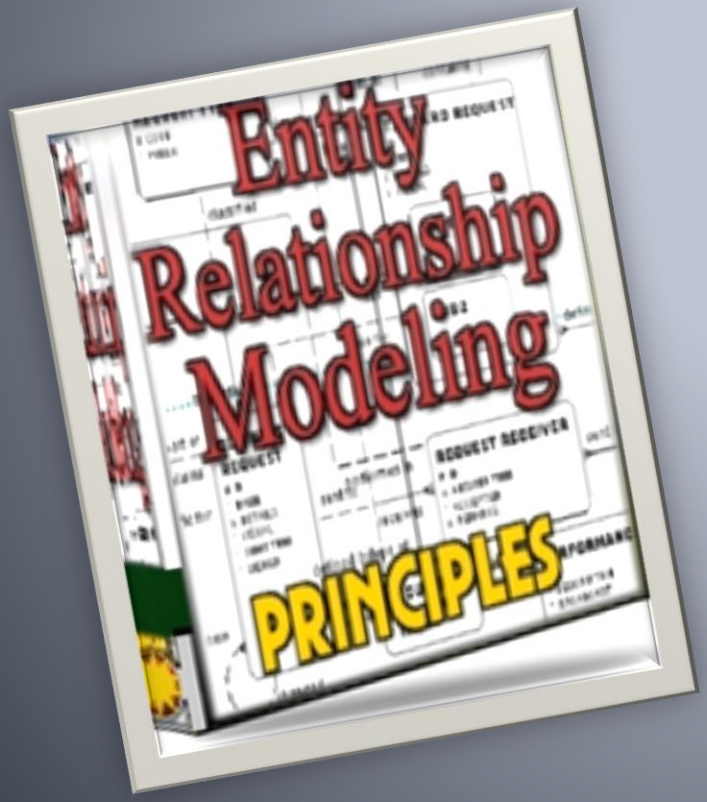
kode_mk	nama_mk

Tahapan ER-Diagram

- Mengidentifikasikan dan menetapkan seluruh entitas yang akan terlibat
- Menentukan atribut-atribut dari setiap entitas
- Menentukan atribut kunci utama (*primary key*) dari setiap entitas
- Menentukan relasi antar entitas
- Menentukan atribut-atribut dari setiap relasi
- Menentukan rasio kardinalitas

Latihan Desain ER-Diagram

Achmad Bisri



Achmad Bisri

Normalisasi Data

Pokok Bahasan

1. Definisi Normalisasi
2. Tujuan Normalisasai
3. Anomali
4. Dependensi
5. Diagram Dependensi Fungsional
6. Bentuk Normalisasi

Definisi Normalisasi

- Normalisasi pertama kali diperkenalkan oleh E.F.Codd pada tahun 1972.
  - Menurut E.F Codd, Normalisasi dipakai untuk membuat struktur tabel (relasi) dalam basis data (mengurangi kemubaziran data).
  - Normalisasi basisdata adalah proses pengorganisasian *field-field* dan tabel-tabel pada sebuah basisdata relasional untuk meminimalisasikan redundansi dan ketergantungan.

Definisi Normalisasi (lanj…)

- Menurut Kroenke, Normalisasi sebagai proses untuk mengubah suatu relasi yang memiliki masalah tertentu ke dalam dua buah relasi atau lebih yang tidak memiliki masalah tersebut (*anomaly*).

Definisi Normalisasi (lanj...)

- Normalisasi adalah 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, 2002: 52).
- Normalisasi adalah proses pembentukan struktur basis data sehingga sebagian besar *ambiguitas* (ketidakjelasan) bisa dihilangkan.
- Normalisasi merupakan metode yang digunakan dalam merancang basis data relasional yang logis melalui himpunan data dengan tingkat ketergantungan fungsional dan keterkaitan sehingga menghasilkan struktur tabel yang normal.

Tujuan Normalisasi

- Mengapa normalisasi perlu dilakukan?
  - Untuk optimalisasi struktur tabel
  - Untuk meminimalisasi redundansi data
  - Untuk meminimalisasi ketergantungan data
  - Untuk mempermudah pemodifikasian data
  - Untuk meningkatkan integritas data
  - Untuk menghindari anomali

Anomali

- Anomali adalah proses pada basis data yang memberikan efek samping yang tidak diharapkan (misalnya menyebabkan ketidakonsistenan data).

- Macam Anomali:
  - Anomali penyisipan (*insert*)
  - Anomali peremajaan (*update*)
  - Anomali penghapusan (*delete*)

Anomali Penyisipan (*insert*)

- Merupakan kesalahan (*error*) yang terjadi akibat dari operasi menyisipkan (*insert*) tuple/record pada sebuah relasi.
  - Relasi Penjualan

ID_Sales	Kode_Barang	Jumlah
S01	B01	10
S03	B02	15
S05	B04	12
S05	B08	17
S04		

  - Ada sales baru dengan ID_Sales **S04** yang akan memasarkan produknya tetapi belum mendapatkan penjualan, hal tersebut tidak bisa di ***insert*** relasi penjualan, sampai ada produk (kode_barang) yang terjual. **ID_Sales** dan **Kode_Barang** merupakan sebuah kunci utama dari relasi

Anomali Peremajaan (*update*)

- Anomali ini terjadi apabila ada perubahan pada sejumlah data yang berulang, tetapi tidak seluruhnya diubah.
  - Tabel Pesanan

Supplier	Kota	Barang
P128	Jakarta	Scanner
P256	Bandung	Laptop
P512	Surabaya	Printer
P256	Medan	PC Desktop

Anomali Penghapusan (*delete*)

- Anomali ini terjadi apabila dalam satu baris/tuple ada data yang akan dihapus sehingga akibatnya terdapat data lain yang hilang.
  - Relasi Penjualan

ID_Sales	Kode_Barang	Jumlah
S01	B01	10
S03	B02	15
S05	B04	12
S05	B08	17

- Sales dengan ID_Sales **S05** akan dihapus pada relasi penjualan karena batal dalam transaksi penjualan. Sehingga data yang behubungan dengan ID_Seles tersebut akan ikut terhapus.

4. *Dependency* (Ketergantungan)

- Macam *defendency*:
  a. *Functional dependency*
  b. *Trivial functional dependency*
  c. *Full functional dependency*
  d. *Transitive dependency*

a. Dependency functional

- Himpunan atribut ***Y*** dikatakan memiliki ketergantungan fungsional pada himpunan atribut ***X***. Jika dan hanya jika setiap nilai ***X*** pasti berhubungan dengan satu nilai ***Y***.
- Notasi (***X*** → ***Y***)
- Dibaca: ***X*** menentukan ***Y***, atau ***Y*** ketergantungan fungsional terhadap ***X***.
- Contoh: didalam **tabel mahasiswa** terdapat atribut **NIM** dan **Tgl_Lahir**.
- **NIM** → **Tgl_Lahir**

Keterangan

- Bagian yang terletak disebelah kiri tanda panah biasa disebut *determinant* (penentu) dan bagian yang terletak disebelah kanan panah disebut *dependency* (yang ketergantungan).
- Tanda { } biasanya digunakan untuk menentukan lebih dari satu atribut sebagai penentu atau sebagai yang ketergantungan.

b. *Trivial functional dependency*

- Adalah ketergantungan fungsional atribut pada superset dari dirinya sendiri.
- Notasi: *A* → *A* atau
- ***X*** → ***Y*** if ***Y*** adalah himpunan bagian dari ***X***

- Contoh:
  - {NIM, Alamat_MHS} → Alamat_MHS

c. *Full functional dependency*

- Atribut ***Y*** ketergantungan fungsional sepenuhnya terhadap atribut ***X*** jika:
  - ***Y*** mempunyai ketergantungan fungsional terhadap ***X*** dan
  - ***Y*** tidak memiliki ketergantungan terhadap bagian dari ***X***

Contoh

Supplier	Kota	Barang	Jumlah

- {Supplier, Barang} –> Jumlah
- Jumlah mempunyai ketergantungan fungsional terhadap Supplier dan Barang, tetapi tidak ketergantungan fungsional sepenuhnya terhadap Supplier. Karena Jumlah juga ketergantungan terhadap Barang.

d. *Transitive dependency*

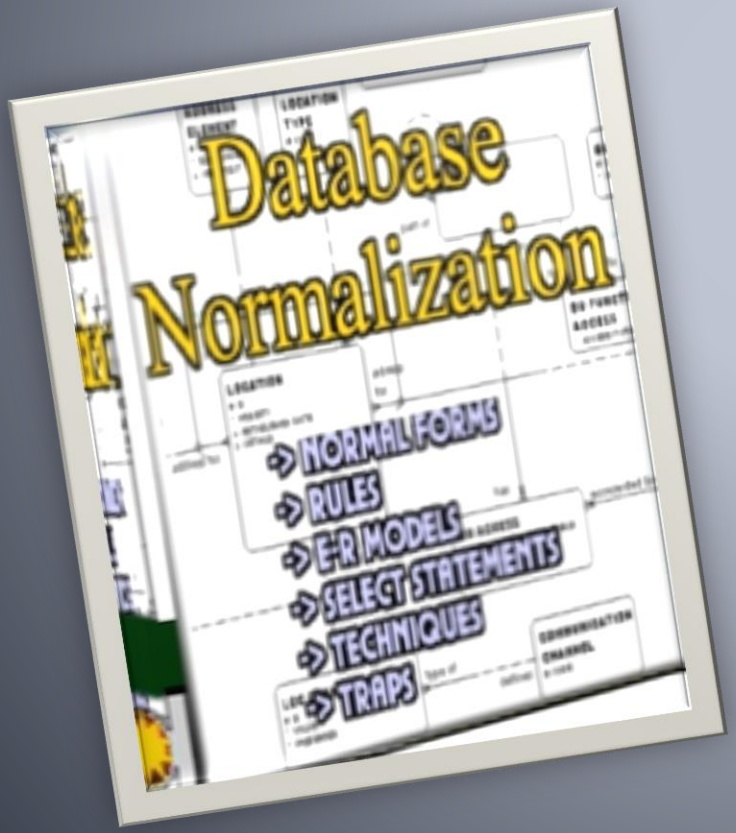
- Atribut *Z* mempunyai dependensi transitif terhadap *X*, jika:
  - *Y* memiliki ketergantungan fungsional terhadap *X*
  - *Z* memiliki ketergantungan fungsional terhadap *Y*

- Notasi : *X* → *Y* → *Z*

Kuliah	Ruang	Tempat	Waktu

Teknik Normalisasi

Achmad Bisri



Tahapan Normalisasi

Normal form	Defined by
First normal form (1NF)	Two versions: E.F. Codd (1970), C.J. Date (2003)[11]
Second normal form (2NF)	E.F. Codd (1971)[2]
Third normal form (3NF)	E.F. Codd (1971);[2] see +also Carlo Zaniolo's equivalent but differently-expressed definition (1982)[12]
Elementary Key Normal Form (EKNF)	C.Zaniolo (1982);[13]
Boyce–Codd normal form (BCNF)	Raymond F. Boyce and E.F. Codd (1974)[14]
Fourth normal form (4NF)	Ronald Fagin (1977)[15]
Fifth normal form (5NF)	Ronald Fagin (1979)[16]
Domain/key normal form (DKNF)	Ronald Fagin (1981)[17]
Sixth normal form (6NF)	C.J. Date, Hugh Darwen, and Nikos Lorentzos (2002)[4]

Normal form	Brief definition
First normal form (1NF)	Table faithfully represents a relation and has no *repeating groups*
Second normal form (2NF)	No non-prime attribute in the table is functionally dependent on a proper subset of any candidate key
Third normal form (3NF)	Every non-prime attribute is non-transitively dependent on every candidate key in the table
Elementary Key Normal Form (EKNF)	Every non-trivial functional dependency in the table is either the dependency of an elementary key attribute or a dependency on a superkey
Boyce–Codd normal form (BCNF)	Every non-trivial functional dependency in the table is a dependency on a superkey
Fourth normal form (4NF)	Every non-trivial multivalued dependency in the table is a dependency on a superkey
Fifth normal form (5NF)	Every non-trivial join dependency in the table is implied by the superkeys of the table
Domain/key normal form (DKNF)	Every constraint on the table is a logical consequence of the table's domain constraints and key constraints
Sixth normal form (6NF)	Table features no non-trivial join dependencies at all (with reference to generalized join operator)

Un-Normalization

NIM	NAMA	JURUSAN	SERTIFIKAT
20120101	Achmad	Teknik Informatika	Oracle
	Null		Java
			Cisco
20120205	Shinta	Sistem Informasi	MS. Office
	Null		Linux
20120304	Ristie	Teknik Komputer	Java
	Null		Linux

- Dikatakan un-normalization, karena adanya:
  - Nilai yang kosong (*null value*)
  - Mereduksi/menanggulangi redundansi data
  - Memiliki atribut yang benilai jamak

Atribut yang bernilai jamak

NIM	NAMA	JURUSAN	SERTIFIKAT
20120101	Achmad	Teknik Informatika	Oracle, Java, Cisco
20120205	Shinta	Sistem Informasi	MS. Office, Linux
20120304	Ristie	Teknik Komputer	Java, Linux

atau

NIM	NAMA	SERTIFIKAT-1	SERTIFIKAT-2	SERTIFIKAT-3
20120101	Achmad	Oracle	Java	Cisco
20120205	Shinta	MS. Office	Linux	
20120304	Ristie	Java	Linux	

Penerapan Bentuk Normalisasi

- Pada proses perancangan database dapat dimulai dari dokumen dasar yang dipakai dalam sistem sesuai dengan lingkup sistem yang akan dibuat rancangan databasenya.
- Berikut ini adalah contoh dokumen mengenai faktur pembelian barang pada PT. Revanda Jaya.

Contoh Normalisasi Faktur PT. Revenda Jaya

FAKTUR PEMBELIAN BARANG

PT REVANDA JAYA
Jl. Bekasi Timur No. 2
Bekasi Timur

Kode Supplier : G01
Nama Supplier : Gobel Nustra

Tanggal : 07/02/2001
Nomor : 998

Kode	Nama Barang	Qty	Harga	Jumlah
A01	AC Split ½ PK	10	1.350.000	13.500.000
A02	AC Split 1 PK	10	2.000.000	20.000.000
			Total Faktur	33.500.000

Jatuh Tempo Faktur : 09/03/2001

FAKTUR PEMBELIAN BARANG

PT REVANDA JAYA
Jl. Bekasi Timur No. 2
Bekasi Timur

Kode Supplier : S02
Nama Supplier : Hitachi

Tanggal : 02/02/2001
Nomor : 779

Kode	Nama Barang	Qty	Harga	Jumlah
R01	Rice Chocker C3	10	150.000	1.500.000
			Total Faktur	1.500.000

Jatuh Tempo Faktur : 09/03/2001

Un-Normalized Form

- Untuk membuat *un-normalized Form* (bentuk tidak normal) caranya dengan mengidentifikasi dan mencantukan seluruh item data yang ada.

No Fac	Kode Supp	Nama Supp	Kode Brg	Nama Barang	Tanggal	Jatuh Tempo	Qty	Harga	Jumlah	Total
779	S02	Hitachi	R02	Rice Chocker C3	02/02/01	09/03/01	10	150000	1500000	1500000
998	G01	Gobel	A01	AC Split ½ PK	07/02/01	09/03.01	10	135000	13500000	33500000
		Nustra	A02	AC Split 1 PK			10	2000000	20000000	

1NF (*First Normalized Form*)

- Suatu relasi/tabel memenuhi 1NF:
  - Jika dan hanya jika setiap atribut dari relasi/tabel tersebut hanya memiliki nilai tunggal dalam satu baris (*record*).
- Caranya:
  - Tidak boleh ada nilai yang kosong (*null value*) dari grup yang berulang (item data atau atribut yang memiliki nilai lebih dari satu dalam suatu baris data/record).

Hasil dari 1NF

No Fac	Kode Supp	Nama Supp	Kode Brg	Nama Barang	Tanggal	Jatuh Tempo	Qty	Harga
779	S02	Hitachi	R02	Rice Chocker C3	02/02/01	09/03/01	10	150000
998	G01	Gobel Nustra	A01	AC Split ½ PK	07/02/01	09/03/01	10	135000
998	G01	Gobel Nustra	A02	AC Split 1 PK	07/02/01	09/03/01	10	2000000

2NF (*Second Normalized Form*)

- Suatu relasi/tabel memenuhi 2NF:
  - Sudah memenuhi 1NF
  - Setiap atribut yang bukan kunci ketergantungan secara fungsional terhadap kunci utamanya dan bukan hanya sebagian atribut.
- Caranya:
  - Tentukan atribut yang sebagai kunci utama
  - Dekomposisi atribut-atribut yang bergantung terhadap kunci utamanya.
  - Hilangkan data yang redundan pada atribut kunci
  - Abaikan/hilangkan *derived attribute* (atribut yang nilainya dihasilkan dari atribut lain).
  - Identifikasi atribut kunci yang akan menjadi suatu relasi

Hasil dari 2NF

Relasi Supplier

Kode_supplier	Nama_supplier
G01	Gobel Nustra
S02	Hitachi

Relasi Barang

Kode_barang	Nama_barang	*Harga*
R01	Rice Cooker CC3	150.000
A01	AC Split ½ PK	1.350.000
A02	AC Split 1 PK	2.000.000

Relasi Faktur

No_faktur	Kode_barang	Tanggal	Jatuh_tempo	Quantitas	Kode_supplier
779	R01	02/02/2001	09/03/2001	10	S02
998	A01	07/02/2001	09/03/2001	10	G01
998	A02	07/02/2001	09/03/2001	10	G01

- Suatu relasi/tabel memenuhi 3NF:
  - Sudah memenuhi 2NF
  - Atribut yang bukan kunci tidak memiliki ketergantungan secara transitif terhadap kunci utama (*primery key*) atau menghilangkan ketergantungan transitif
- Caranya:
  - Pisahkan atribut yang ketergantungan secara transitif
  - Relasikan atribut kunci dari ketergantungan transitif dari relasi dekomposisi.

Hasil dari 3NF

Relasi Supplier

Kode_supplier	Nama_supplier
G01	Gobel Nustra
S02	Hitachi

Relasi Barang

Kode_barang	Nama_barang	*Harga*
R01	Rice Cooker CC3	150.000
A01	AC Split ½ PK	1.350.000
A02	AC Split 1 PK	2.000.000

Relasi Faktur

No_faktur	Kode_supplier	Tanggal	Jatuh _tempo
779	S02	02/02/2001	09/03/2001
998	G01	07/02/2001	09/03/2001

Relasi Transaksi_Barang

No_faktur	Kode_barang	Quantitas
779	R01	10
998	A01	10
998	A02	10

Diagram Dekomposisi