

BAB III

MODEL DATA

1. Pengertian Model Data

Sebagaimana telah disebutkan di bab sebelumnya, upaya perancangan basis data dapat juga kita tempuh dengan secara langsung membuat sebuah model dari awal sekali. Langkah ini biasanya ditempuh dari kelangkaan data/fakta yang kita miliki. Di sisi lain, sebuah model dinyatakan dalam bentuk diagram awal akan lebih mudah untuk dievaluasi/dianalisis untuk kemudian dilakukan perbaikan-perbaikan untuk mendapatkan sebuah model data yang lebih permanen dan lebih mendekati kenyataan sesungguhnya.

Menurut (Fathansyah, 2012), model data dapat didefinisikan sebagai kumpulan perangkat konseptual untuk menggambarkan data, hubungan data, semantic (makna) data dan batasan data. Oleh karena yang ingin ditunjukkan adalah makna dari data dan keterhubungannya dengan data lain, maka model data ini lebih tepat jika disebut Model Data Logik.

2. Jenis-jenis Model Data

a. Model Data Berdasarkan Object

Menurut (Ladjamudin, 2005), Model data berbasis objek menggunakan konsep entitas, atribut dan hubungan antar entitas (relationship). Entity adalah sesuatu apa saja yang ada didalam sistem, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data. Entitas diberi nama

dengan kata benda dan dapat dikelompokkan dalam empat jenis nama, yaitu nama orang, benda, lokasi, kejadian (terdapat unsur waktu didalamnya), sedangkan atribut merupakan relasi fungsional dari satu objek set ke objek set yang lain. Relationship adalah hubungan alamiah yang terjadi antara entitas. Model data berbasis objek memiliki beberapa bentuk, sebagai berikut :

1) Model Keterhubungan Entitas (Entity-Relationship Model)

a) Pengertian Model Keterhubungan Entitas (Entity Relationship Model / ER Model)

Model Keterhubungan Entitas (Entity-Relationship Model) merupakan model yang paling populer digunakan dalam perancangan basis data. Entity-Relationship Model merupakan Model untuk menjelaskan hubungan antar data dalam basis data berdasarkan suatu persepsi bahwa *real world* terdiri dari objek-objek dasar yang mempunyai hubungan atau relasi antara objek-objek tersebut. Komponen utama pembentuk Model *Entity-Relationship*, yaitu: Entitas (*Entity*) dan Relasi (*Relation*). Kedua komponen ini dideskripsikan lebih lanjut melalui sejumlah Atribut/Properti.

Tabel Buku			Tabel Anggota	
Kode Buku	Judul	Stok Buku	Kode Anggota	Nama
B01	Pemrograman C++	10	A01	Surya
B02	Membuat Aplikasi 30 Menit	15	A02	Fitri
B03	Cooking is Easy	15	A03	Syahrur

Tabel Peminjaman					
Kode pinjam	Tgl pinjam	Kode buku	Kode anggota	jumlah	Tgl kembali
PJ01	10-01-2019	B01	A01	1	13-01-2019
PJ01	10-01-2019	B02	A01	1	13-01-2019
PJ01	10-01-2019	B03	A01	1	13-01-2019
PJ02	12-01-2019	B02	A02	1	14-01-2019
PJ02	12-01-2019	B03	A02	1	14-01-2019

Dari tabel diatas kita dapat menentukan :

Entitas : Buku, Anggota, Peminjaman.



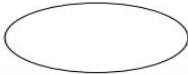
Atribut : Tabel Buku (kode buku, judul, stok buku), Tabel Anggota (kode anggota, nama) dan Tabel Peminjaman (kode pinjam, tgl pinjam, kode buku, kode anggota, jumlah, tgl kembali).

Relasi : hubungan antara kode buku di tabel buku dengan kode buku di tabel peminjaman. Begitu pula dengan kode anggota.

Model Entity Relationship yang berisi komponen himpunan entitas, relasi, yang dilengkapi atribut-atribut, dapat digambarkan menggunakan Diagram Entity Relationship (Diagram E-R).

b) Simbol dasar yang digunakan :

Tabel III.1
Simbol Dasar Entity Relationship Diagram

	Menyatakan Himpunan Entitas
	Menunjukkan Himpunan Relasi
	Menyatakan Atribut (Atribut key digaris bawah)

—————	Penghubung/Link
-------	-----------------

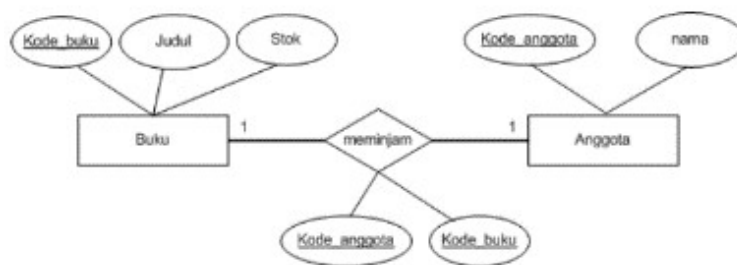
c) Mapping Kardinalitas

Dalam Diagram E-R aturan terpenting adalah Kardinalitas relasi/ *Mapping Cardinalities* yang menentukan jumlah entity yang dapat dikaitkan dengan entity lainnya melalui relationship-set.

Jenis *Mapping Cardinalities*:

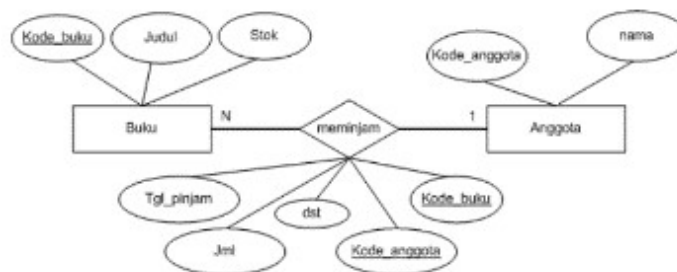
(1) Relasi satu ke satu (*one-to-one*)

Contoh :



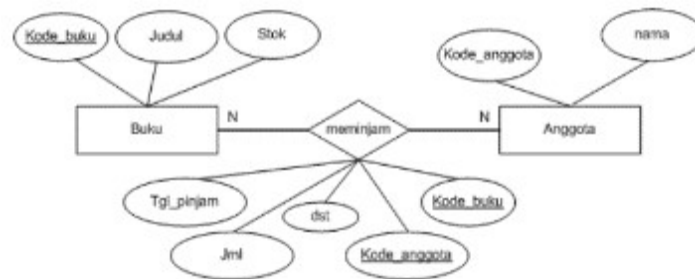
Gambar III.1
Relasi *one to one*

(2) Relasi satu ke banyak (*one-to-Many*)



Gambar III.2
Relasi *one to many*

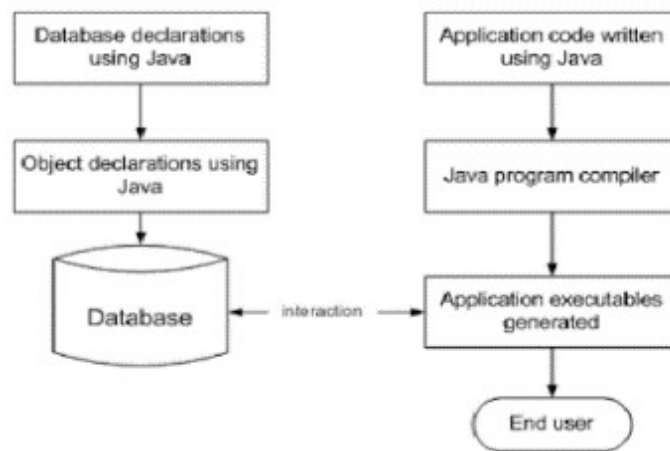
(3) Relasi banyak ke banyak (*many-to-many*)



Gambar III.3

Relasi *many to many*

2) Model Berorientasi Object (Object-Oriented Model)



Gambar III.4

Model Berorientasi Object

Penggambaran model berbasis objek menggunakan UML. UML Digambarkan dengan 2 Jenis, menurut (Sukamto & Shalahuddin, 2018):

a) *Structural Diagram*(1) *Class Diagram*

Menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

(2) *Object Diagram*

Menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem.

(3) Component Diagram

Dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem.

(4) Deployment Diagram

Menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi.

b) Behaviour Diagram

(1) Use case Diagram

Merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat.

(2) Sequence Diagram

Mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek.

(3) Communication Diagram

Penyederhanaan dari diagram kolaborasi (*collaboration diagram*). *Collaboration diagram* sudah tidak muncul pada UML versi 2.x.

(4) Statechart Diagram

Menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem atau objek.



(5) *Activity Diagram*

Menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.

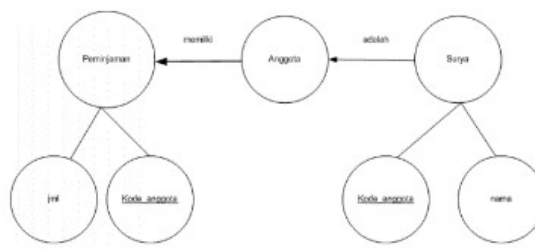
3) Model Data Semantik (*Semantic Data Model*)

Hampir sama dengan *Entity Relationship model* dimana relasi antara objek dasar tidak dinyatakan dengan simbol tetapi menggunakan kata-kata (*Semantic*). Sebagai contoh, dengan masih menggunakan relasi pada Bank X sebagaimana contoh sebelumnya, dalam semantic model adalah seperti terlihat pada gambar di atas. Tanda-tanda yang menggunakan dalam *semantic model* adalah:

Tabel III.2
Simbol Model Data Semantik

	Menunjukkan adanya relasi
	Menunjukkan atribut

Contoh :



Gambar III.5
Contoh Model Data Semantik

4) Model Data Fungsional (*Functional Data Model*)

b. Model Data Berdasarkan *Record*

Model ini berdasarkan pada record untuk menjelaskan kepada user tentang hubungan logic antar data dalam basis data.

Perbedaan dengan model data berbasis objek adalah *pada record based data model* disamping digunakan untuk menguraikan struktur logika keseluruhan dari suatu database, juga digunakan untuk menguraikan implementasi dari sistem database (*higher level description of implementation*)

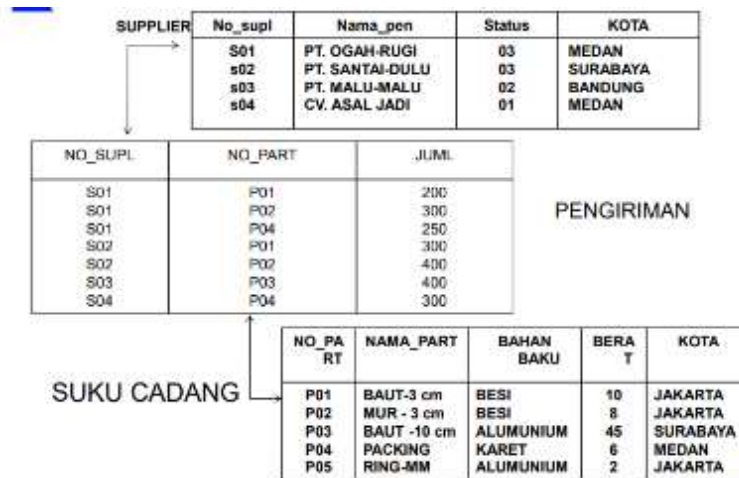
Jenis-Jenis Model Data Berbasis Record :

1) *Model Relational*

Dimana data serta hubungan antar data direpresentasikan oleh sejumlah tabel dan masingmasing tabel terdiri dari beberapa kolom yang namanya *unique*. Model ini berdasarkan notasi teori himpunan (*set theory*), yaitu relation.

Contoh : database penjualan terdiri dari 3 tabel, sebagai berikut :

- (a) Tabel Supllier
- (b) Tabel Suku Cadang
- (c) Tabel Pengiriman

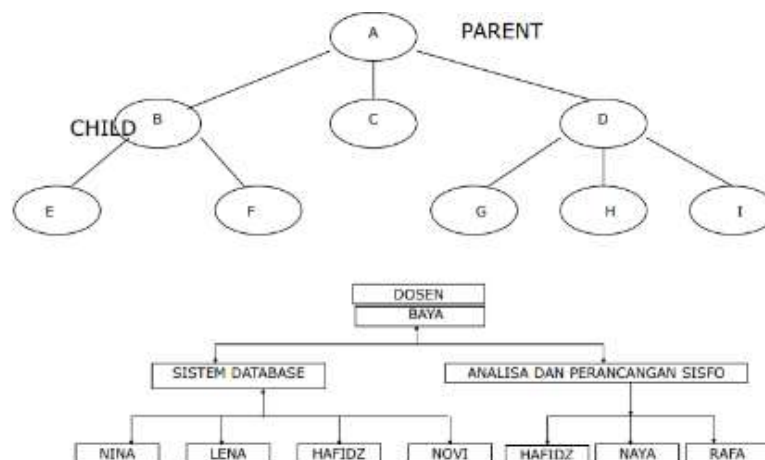


Gambar III.6
Contoh Model Data Relational

2) Model Hirarki

Dimana data serta hubungan antar data direpresentasikan dengan record dan link (pointer), dimana record-record tersebut disusun dalam bentuk tree (pohon), dan masing-masing node pada tree tersebut merupakan record/grup data elemen dan memiliki hubungan kardinalitas 1:1 dan 1:M.

Contoh :

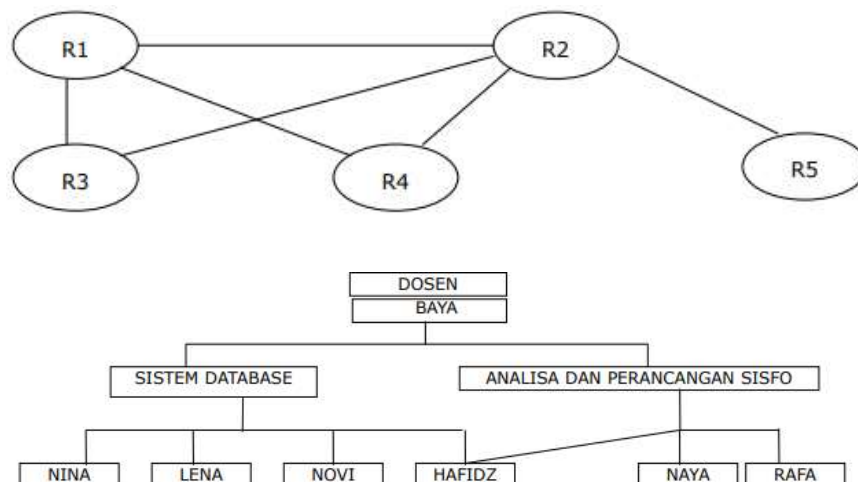


Gambar III.7
Contoh Model Data Hirarki

3) Model Jaringan

Distandarisasi tahun 1971 oleh Database Task Group (DBTG) atau disebut juga model CODASYL (Conference on Data System Language), mirip dengan hirarkikal model dimana data dan hubungan antar data direpresentasikan dengan record dan links. Perbedaannya terletak pada susunan record dan linknya yaitu network model menyusun record-record dalam bentuk graph dan menyatakan hubungan cardinalitas 1:1, 1:M dan N:M.

Contoh :



Gambar III.8
Contoh Model Data Jaringan