

**LAPORAN PRAKTIKUM
BASIS DATA I**

ENTITY RELATIONSHIP DIAGRAM



**Oleh :
Mariana
A1317041**

**PROGRAM STUDI TEKNIK INFORMATIKA
POLITEKNIK NEGERI TANAH LAUT
PELAIHARI
2018**

ERD

(ENTITY RELATIONSHIP DIAGRAM)

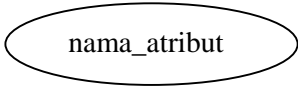
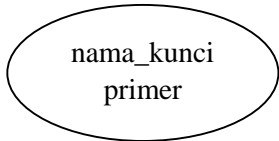
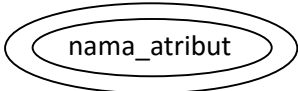
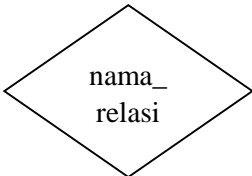

1.1 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak. *Entity Relationship Diagram* merupakan model jaringan yang menekankan pada struktur dan hubungan antardata. *Entity Relationship Diagram* juga memperlihatkan hubungan antardata *store* pada *Data Flow Diagram*. *Entity Relationship Diagram* atau lebih dikenal dengan *E-R* adalah notasi grafik dari sebuah model data atau sebuah model jaringan yang menjelaskan tentang data yang disimpan (*storage data*) dalam sistem secara abstrak. *Entity Relationship Diagram* tidak menyatakan bagaimana memanfaatkan data, membuat data, mengubah data dan menghapus data.

Entity Relationship Diagram (ERD) dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow's Foot dan beberapa notasi lain. Namun yang banyak digunakan adalah notasi dari Chen. Berikut simbol-simbol yang digunakan pada ERD dengan notasi Chen:

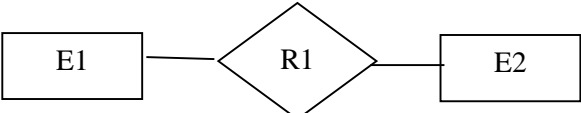
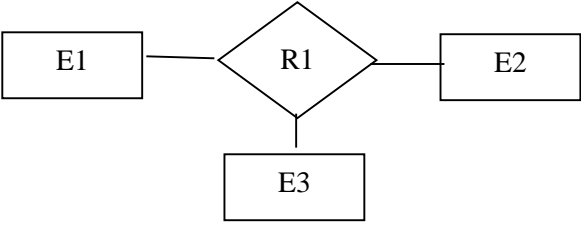
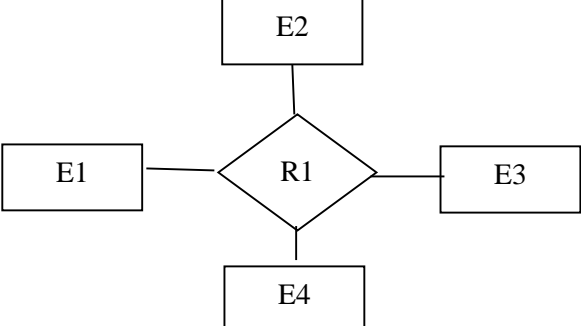
Tabel 1.1 Simbol-simbol Entity Relationship Diagram

Nama simbol	Simbol	Deskripsi
Entitas / entity	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">Nama entitas</div>	Entitas merupakan data inti yang akan disimpan; bakal table pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi computer; penamaan

		entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.
Atribut		Field atau kolom data yang butuh disimpan dalam suatu entitas.
Atribut kunci primer		Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama)
Atribut multi nilai / multivalue		Field atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu. Misalnya riwayat pendidikan, nomer handphone, email dan lain sebagainya.
Relasi		Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja
Asosiasi / association		Penghubung antara relasi dan antitas di mana kedua ujungnya memiliki multiplicity kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan one to many menghubungkan entitas A dengan entitas B.

ERD biasanya memiliki hubungan binary (satu relasi menghubungkan dua buah entitas). Beberapa metode perancangan ERD menoleransi hubungan relasi ternary (satu relasi menghubungkan tiga buah relasi) atau N-ary (satu relasi menghubungkan banyak entitas), tapi banyak metode perancangan ERD yang tidak mengizinkan hubungan ternary atau N-ary. Berikut adalah contoh bentuk hubungan relasi dalam ERD.

Tabel 1.1 Hubungan relasi dalam Entity Relationship Diagram

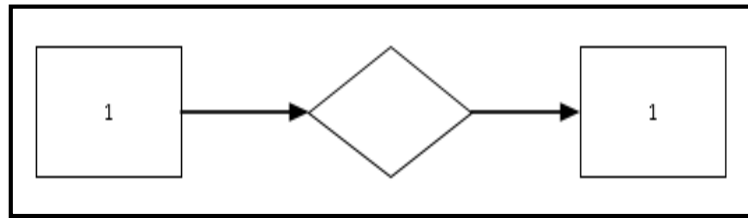
Nama	Gambar
Binary	
Ternary	
N-ary	

Relationship mempunyai tiga tipe yang mana tiap tipe menunjukkan jumlah *record* dari setiap tabel yang direlasikan ke *record* pada tabel lain. Ketiga tipe tersebut adalah sebagai berikut :

1. Hubungan satu ke satu (*One to one relationship*)

Hubungan antara file pertama dan file kedua berbanding satu. Dalam hubungan ini, tiap *record* dalam tabel A hanya memiliki satu *record* yang cocok dalam tabel B dan tiap *record* dalam tabel B hanya memiliki satu

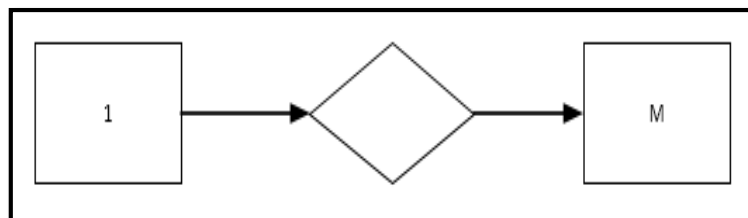
record dalam tabel A. Logika penalaran matematika dari *one to one relationship* adalah pemetaan dengan “perkawanan satu-satu” atau sering disebut dengan korespondensi satu-satu. Contoh *One to one relationship* adalah satu pasien mempunyai satu tempat tidur.



Gambar 1.1 *One to one relationship*

2. Hubungan satu ke banyak (*One to many relationship*)

Hubungan antara file pertama dengan file kedua adalah satu berbanding banyak. Dalam hubungan ini tiap *record* dalam tabel A memiliki beberapa *record* yang cocok dalam tabel B. Logika penalaran matematika dari *one to many relationship* adalah “Perkawanan satu ke banyak”. Contoh *One to many relationship* adalah satu dosen mengajar banyak mata kuliah.

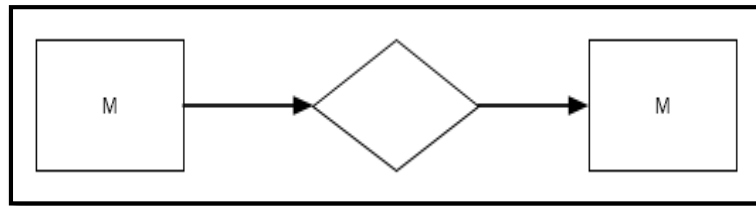


Gambar 1.2 *One to many relationship*

3. Hubungan banyak ke banyak (*Many to many relationship*)

Hubungan antara file pertama dengan file kedua adalah banyak berbanding banyak. Dalam hubungan ini tiap *record* dalam tabel A memiliki beberapa *record* yang cocok dalam tabel B dan tiap *record* dalam tabel B hanya memiliki satu *record* yang cocok dalam tabel A. Logika penalaran matematika dari *many to many relationship* adalah pemetaan “Perkawanan banyak ke banyak”. Contoh *many to many relationship* adalah banyak mahasiswa memiliki banyak mata kuliah dan banyak mata kuliah memiliki banyak mahasiswa. Hubungan *many to many* tidak dapat diimplementasikan

ke dalam *database* relationship sehingga hubungan ini harus dipecah menjadi hubungan *one to many*.



Gambar 1.3 *Many to Many relationship*

1.2 Studi Kasus

Dalam modul ini akan membahas sebuah studi kasus tentang Sistem Informasi Permintaan Barang dalam suatu perusahaan.

1. Judul

Sistem Informasi Permintaan Barang.

2. Deskripsi

Sistem Informasi Permintaan Barang merupakan suatu sistem informasi untuk melakukan proses permintaan barang di gudang suatu perusahaan. Yang meliputi permintaan barang oleh mandor, yang diproses oleh kepala gudang kemudian akan diterima oleh mandor yang meminta.

3. Aturan

Aturan yang harus diatasi dalam sistem informasi yang akan dimodelkan ini adalah sebagai berikut:

- a. Permintaan yang dilakukan mandor dengan menulis pada bon permintaan.
- b. Setiap mandor dapat menulis pada bon permintaan dalam satu waktu yang sama dan boleh lebih dari satu barang.
- c. Satu barang akan disimpan sebagai satu data dengan id yang unik.
- d. Pada bon_permintaan terdapat keterangan digunakannya dari barang yang diminta.

4. Analisa

Permintaan barang pada suatu perusahaan meliputi fungsi-fungsi sebagai berikut:

- a. Validasi users
 - Login
 - Logout

- b. Mengelola data users yang memiliki hak akses untuk mengelola data users, meliputi:
 - Menambah data users baru
 - Mengubah data users
 - Mencari data users
 - Melihat data users
 - Menghapus data users
- c. Mengelola data bon_permintaan, meliputi:
 - Menambah data bon_permintaan
 - Mengubah data bon_permintaan
 - Mencari data bon_permintaan
 - Melihat data bon_permintaan
 - Menghapus data bon_permintaan
- d. Mengelola data barang, meliputi:
 - Menambah data barang.
 - Mengubah data barang.
 - Mencari data barang.
 - Melihat data barang
 - Menghapus data barang
- e. Mengelola data kelompok
 - Menambah data kelompok
 - Mengubah data kelompok
 - Mencari data kelompok
 - Melihat data kelompok
 - Menghapus data kelompok
- f. Mengelola data nomor_akun
 - Menambah data nomor_akun
 - Merubah data nomor_akun
 - Mencari data nomor_akun
 - Melihat data nomor_akun
 - Menghapus nomor_akun

g. Mengelola data pegawai

- Menambah data pegawai
- Mengubah data pegawai
- Mencari data pegawai
- Melihat data pegawai
- Menghapus data pegawai

1.3 ERD

1. Definisi Entitas dan Atribut

No.	Entitas	Atribut
1.	users entitas yang menyimpan data users yang berhak login ke aplikasi untuk mengelola data	username atribut untuk melakukan proses login password merupakan kata sandi dari masing-masing users untuk login pada sistem hak_akses atribut untuk mengetahui hak akses users yang berhak mengelola data pada sistem atau tidak (biasanya disebut sebagai admin/administrator/yang mengurus administrasi)
2.	pegawai entitas yang menyimpan data mandor	id_pegawai merupakan suatu atribut yang menjadi identitas pegawai nama_pegawai merupakan atribut dari nama pegawai alamat atribut alamat pegawai no_telp atribut nomer telepon pegawai

3.	jabatan entitas yang menyimpan data jabatan	kode_jab merupakan atribut yang menjadi identitas dari jabatan.
		jabatan merupakan atribut keterangan jabatan
4.	barang entitas yang menyimpan data- data barang	kode_barang merupakan suatu atribut untuk menjadi identitas dari barang
		nama_barang merupakan atribut nama barang
		satuan merupakan atribut satuan dari barang.
		stok merupakan atribut untuk keterangan stok barang.
5.	kelompok entitas yang menyimpan data kelompok barang	kode_kel merupakan suatu atribut untuk menjadi identitas dari kelompok
		kelompok Merupakan suatu atribut untuk mneyimpan keterangan dari kelompok
6.	nomor_akun entitas yang menyimpan nomor akun	no_akun atribut yang menjadi identitas dari nomor_akun
		deskripsi atribut yang menjadi keterangan dari deskripsi nomor_akun
7.	bon_permintaan entitas yang menyimpan permintaan barang dari mandor	id_bon atribut yang menjadi identitas dari bon_permintaan
		id_pegawai atribut <i>foreign key</i> dari tabel pegawai

		kode_barang atribut <i>foreign key</i> dari tabel barang
		no_akun atribut <i>foreign key</i> dari tabel nomor_akun
		tgl_minta atribut tanggal permintaan
		jumlah_barang atribut dari jumlah barang yang akan diminta
		keterangan Atribut dari keterangan digunakannya bahan, dari bon_permintaan
		status merupakan atribut dari status bon_permintaan

2. Definisi Relasi

No.	Relasi	Atribut
1.	mengelola dalam bon_permintaan	Merupakan relasi antara entitas pegawai dan entitas bon_permintaan dimana mengelola memiliki makna bahwa bon_permintaan dikelola oleh pegawai yang disimpan pada entitas bon_permintaan. Kardinalitas antara entitas pegawai dan entitas bon_permintaan adalah one to many karena seorang pegawai dapat terlibat dengan banyak bon_permintaan.

2.	mempunyai dalam bon_permintaan	<p>Merupakan relasi antara barang dengan bon_permintaan. Dimana maksud dari mempunyai pada bon_permintaan adalah bon_permintaan dimiliki oleh barang yang disimpan pada entitas bon_permintaan.</p> <p>Kardinalitas dari barang dengan bon_permintaan adalah one to many karena suatu barang dapat terlibat dengan banyak bon_permintaan.</p>
3.	memiliki dalam bon_permintaan	<p>Merupakan relasi antara nomor_akun dengan bon_permintaan. Dimana maksud dari memiliki pada bon_permintaan adalah bon_permintaan dimiliki oleh nomor_akun yang disimpan pada entitas bon_permintaan.</p> <p>Kardinalitas dari nomor_akun dengan bon_permintaan adalah one to many karena suatu nomor_akun dapat terlibat dengan banyak bon_permintaan.</p>
4.	mempunyai dalam jabatan	<p>Merupakan relasi antara pegawai dengan jabatan. Dimana maksud dari mempunyai pada jabatan adalah pegawai mempunyai jabatan yang disimpan pada entitas pegawai.</p> <p>Kardinalitas dari pegawai dengan jabatan adalah many to one karena banyak pegawai dapat terlibat dengan satu jabatan.</p>

5.	terdapat dalam barang	<p>Merupakan relasi antara kelompok dengan pegawai. Dimana maksud dari terdapat pada barang adalah kelompok mempunyai barang-barang yang disimpan pada entitas barang.</p> <p>Kardinalitas dari barang dengan kelompok adalah many to one karena banyak barang dapat terlibat dengan satu kelompok.</p>
----	-----------------------	---

1.3.3 Diagram ER

