

PERANCANGAN SISTEM TERINCI DATABASE

1. PERANCANGAN DATABASE

Perancangan Database adalah proses untuk menentukan isi dan pengaturan data yang dibutuhkan untuk mendukung berbagai rancangan sistem.

Perancangan sistem terjadi pada dua tingkat , yaitu :

Pada tingkat pertama, perencanaan sistem, analisis dan rancangan umum dilaksanakan untuk menetapkan kebutuhan pemakai. Tingkat perancangan database ini melibatkan tahap *front-end*, bebas dari perancangan database tertentu atau Database Management System (DBMS).

Pada tingkat kedua, rancangan umum, seperti diagram entitas relasi tingkat tinggi, ditransformasikan (atau didekomposisikan) ke dalam perancangan database rinci untuk sebuah DBMS tertentu yang akan digunakan untuk mengimplementasikan sistem total.

Tiga model database yang cukup dikenal adalah :

- § Model Hierarkikal
- § Model Jaringan
- § Model Relasional

Pada masa lalu banyak penjual (*vendors*) menawarkan Database Management Systems (DBMS) yang berdasarkan pada Model Hierarkikal dan Model Jaringan. Saat ini Model Relasional adalah dominan. Karena itu hampir semua penjual perangkat lunak database menawarkan produk perangkat lunak Relational Database Management Systems (RDBMS).

RDBMS dibuat dengan struktur tiga skema , yaitu :

- § Eksternal
- § Konseptual
- § Internal

Lihat Gambar 1. Struktur tiga-skema dari sebuah RDBMS

Struktur lapisan ini mendefinisikan data perusahaan pada tingkat yang berbeda.

Skema Eksternal mendefinisikan bagaimana pemakai mengakses dan melihat output dari RDBMS, bebas dari bagaimana data disimpan atau diakses secara fisik. Akses dan manipulasi seperti ini dilaksanakan oleh pemakai dengan memperkerjakan bahasa prosedural, seperti COBOL atau bahasa query, seperti Structured Query Language (SQL), bahasa standar yang diakui untuk RDBMS.

Skema Konseptual yang mendefinisikan model database relasional terdiri dari sekumpulan tabel yang dinormalisasi. Skema konseptual adalah rancangan dari database yang merupakan subyek utama dari bab ini.

Skema Internal terdiri dari organisasi fisik dari data (mis. sekuensial, indeks sekuensial, langsung) dalam hal struktur fisik data dan metode-metode pengaksesan dari sistem operasi komputer.

APAKAH DATABASE RELASIONAL ITU ?

Model relasional berdasarkan teori himpunan matematik. Struktur didefinisikan dengan Tabel. Dalam istilah matematika, tabel disebut sebagai Relasi. Profesional sistem sering menggunakan istilah “tabel” dan “relasi” secara bergantian.

Tiap tabel dalam model relasional dikomposisikan dari baris dan kolom. Kolom disebut Atribut. Nilai untuk sebuah atribut harus dipilih dari sekelompok nilai yang dinamakan Domain. Karena banyak kolom dalam tabel yang sama dapat diidentifikasi atas domain yang sama, maka nama atribut didefinisikan untuk tiap kolom. Tiap nama atribut dalam sebuah relasi harus unik. Urutan kiri ke kanan dari kolom tidak penting. Urutan dari baris juga tidak penting. Perpotongan dari suatu baris dan kolom berisi sebuah nilai tunggal.

Sifat-sifat Tabel :

1. Duplikasi baris tidak diperbolehkan. Untuk melaksanakan sifat ini, harus terdapat paling sedikit satu atribut atau kombinasi beberapa atribut yang mengidentifikasi secara unik tiap baris dari tabel. Atribut atau kombinasi beberapa atribut yang melaksanakan tugas ini disebut **Kunci Primer** (*Primary Key*). Contoh : Nomor_Mahasiswa, adalah kunci primer yang mengidentifikasi tiap mahasiswa secara unik.

2. Database relasional adalah nilai *Primary Key* tidak boleh mempunyai duplikat atau NIL (NULL, yaitu nilai tidak diketahui).
3. Keterhubungan (*relationship*) antara dua tabel. Jika Tabel R2 mempunyai sebuah **Kunci Asing** (*Foreign Key*) yang cocok dengan kunci primer dari Tabel R1, maka untuk setiap nilai *Foreign Key* harus terdapat sebuah nilai kecocokan dari *Primary Key*, atau nilai *Foreign Key* harus nil.

Menggunakan Structures Query Language (SQL)

SQL adalah bahasa standar database yang digunakan untuk query, manipulasi dan memperbaiki RDBMS. Karena semakin banyak organisasi yang memutuskan untuk mengkonsolidasikan database mereka ke dalam sistem seluas usaha, pengetahuan mengenai SQL akan menjadi kebutuhan untuk para perancang database.

2. ENTITY RELATIONSHIP DIAGRAM (ERD)

Model Entity Relationship

Adalah suatu penyajian data dengan menggunakan Entity dan Relationship.

Entity

§ Entity adalah obyek yang dapat dibedakan dalam dunia nyata

§ Entity set adalah kumpulan dari entity yang sejenis

§ Entity set dapat berupa :

- Obyek secara fisik : Rumah, Kendaraan, Peralatan
- Obyek secara konsep : Pekerjaan , Perusahaan, Rencana

Relationship

§ Relationship adalah hubungan yang terjadi antara satu atau lebih entity.

§ Relationship set adalah kumpulan relationship yang sejenis.



Atribut

§ Atribut adalah karakteristik dari entity atau relationship, yang menyediakan penjelasan detail tentang entity atau relationship tersebut.

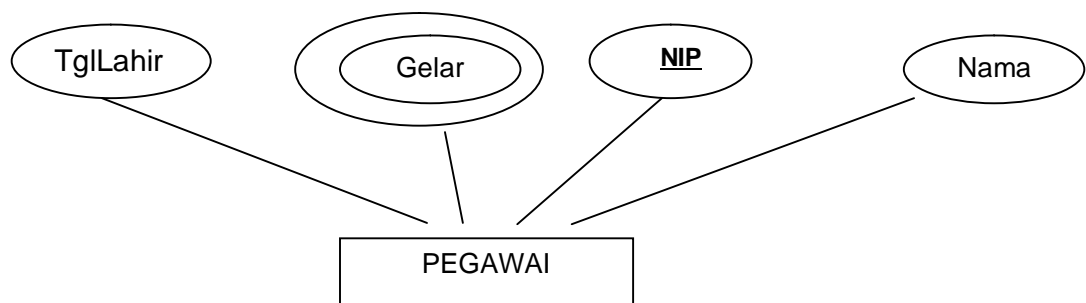
§ Nilai Atribut merupakan suatu data aktual atau informasi yang disimpan pada suatu atribut di dalam suatu entity atau relationship.

Jenis-jenis atribut :

§ Key
Atribut yang digunakan untuk menentukan suatu entity secara unik.

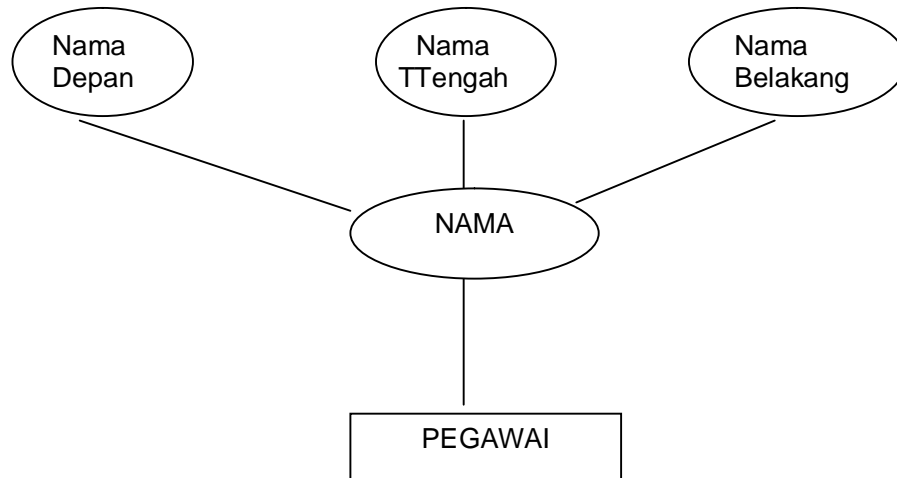
§ Atribut Simple
Atribut yang bernilai tunggal.

§ Atribut Multivalue
Atribut yang memiliki sekelompok nilai untuk setiap instan entity.



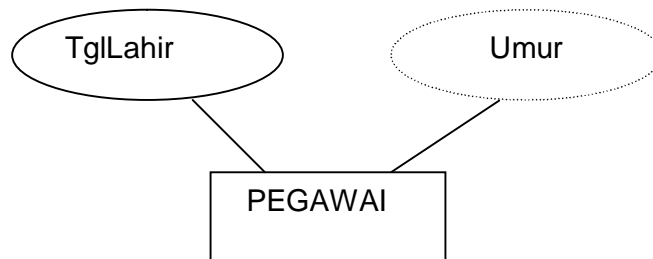
§ Atribut Composite

Suatu atribut yang terdiri dari beberapa atribut yang lebih kecil yang mempunyai arti tertentu.



§ Atribut Derivatif

Suatu atribut yang dihasilkan dari atribut yang lain.



Derajat dari relationship

Menjelaskan jumlah entity yang berpartisipasi dalam suatu relationship.

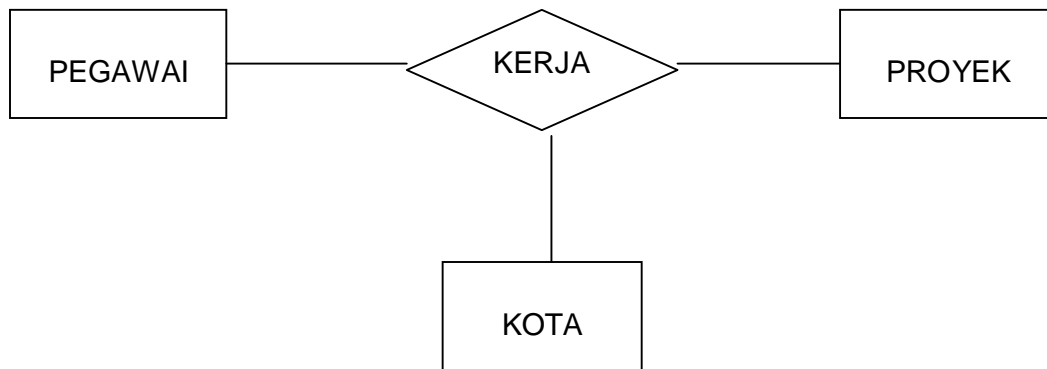
Unary Degree (Derjat Satu)



Binary Degree (Derajat Dua)



Ternary Degree (Derajat Tiga)



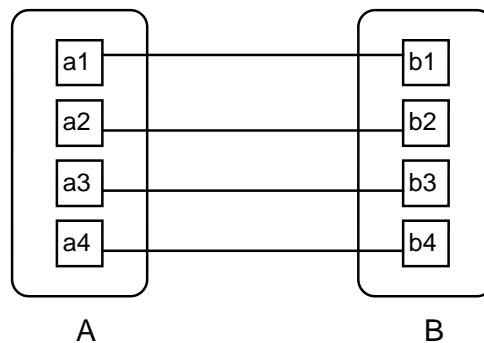
Cardinality Ratio Constraint

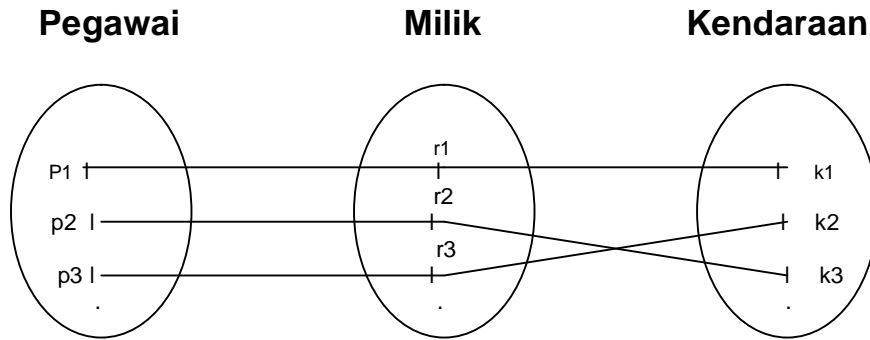
§ Menjelaskan batasan jumlah keterhubungan satu entity dengan entity lainnya.

§ Jenis Cardinality Ratio

1 : 1 (One-To-One)

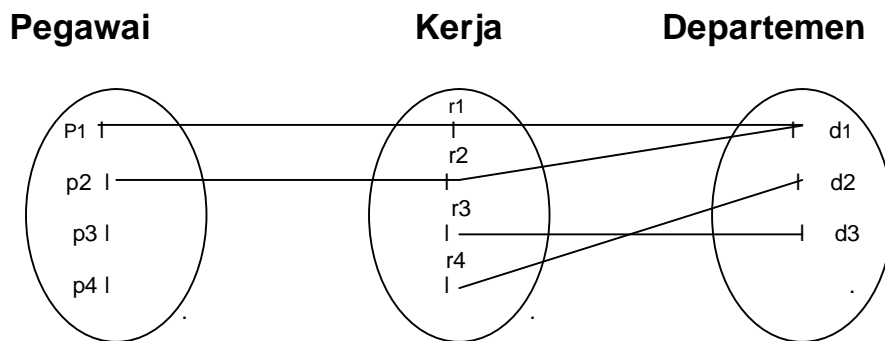
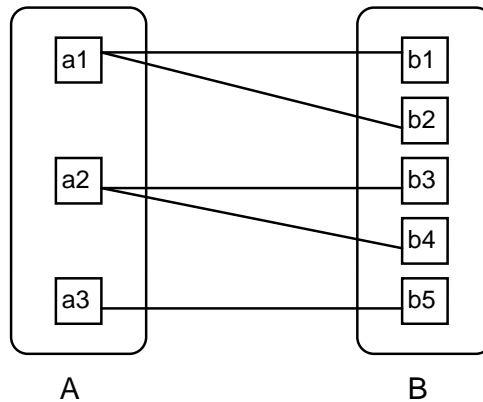
Sebuah entity A diasosiasikan pada sebuah entity B, dan sebuah entity B diasosiasikan dengan paling banyak sebuah entity A.

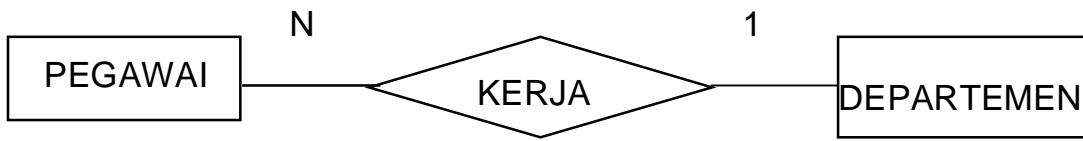




1 : N (One-To-Many)

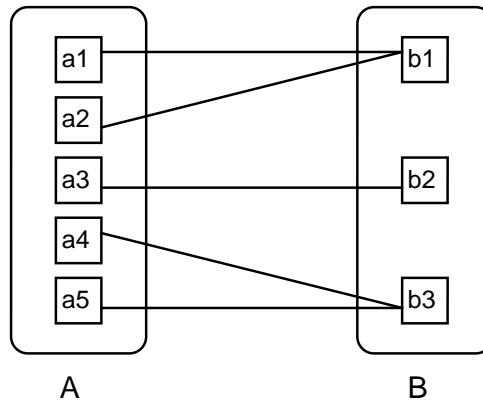
Sebuah entity A diasosiasikan dengan sejumlah entity B, tetapi entity B dapat diasosiasikan paling banyak satu entity A.





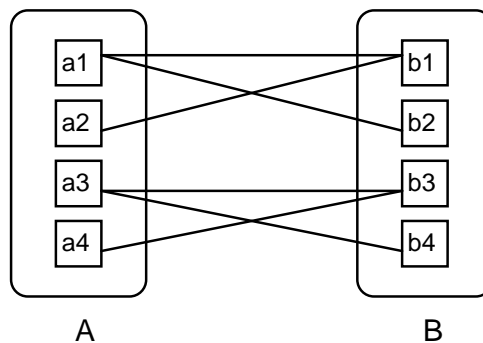
N : 1 (Many-To-One)

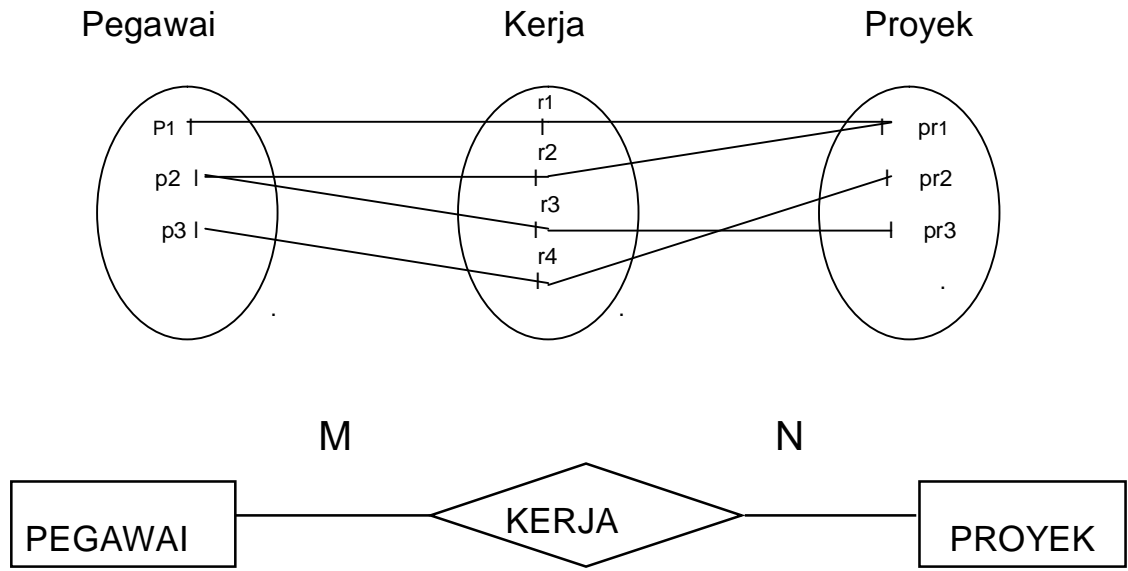
Suatu entity A dapat diasosiasikan dengan paling banyak sebuah entity B, tetapi entity B dapat diasosiasikan dengan sejumlah entity di A.



M : N (Many-To-Many)

Suatu entity A dapat diasosiasikan dengan sejumlah entity B dan entity B dapat diasosiasikan dengan sejumlah entity di A.





Participation Constraint

Menjelaskan apakah keberadaan suatu entity tergantung pada hubungannya dengan entity lain .

Terdapat 2 macam Participation Constraint :

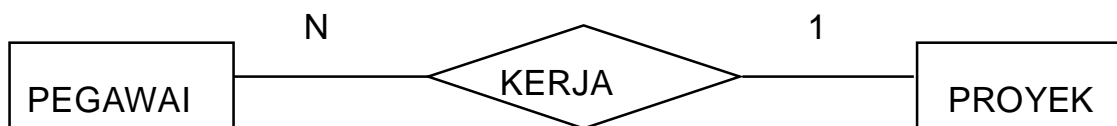
Total Participation

Keberadaan suatu entity tergantung pada hubungannya dengan entity lain.



Partial Participation

Keberadaan suatu entity tidak tergantung pada hubungannya dengan entity lain.

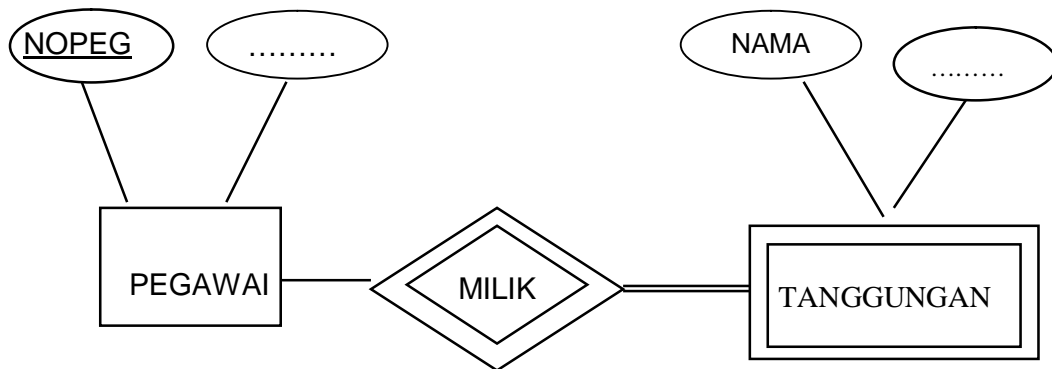


Weak Entity


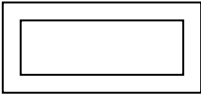
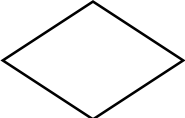
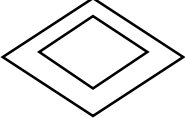

§ Weak Entity adalah suatu Entity dimana keberadaan dari entity tersebut tergantung dari keberadaan entity lain.

§ Entity yang merupakan induknya disebut *Identifying Owner* dan relationshipnya disebut *Identifying Relationship*.

§ Weak Entity selalu mempunyai Total Participation constraint dengan Identifying Owner.



Simbol-simbol ER-Diagram

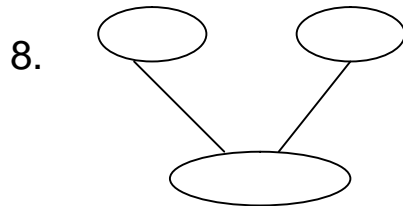
<u>Notasi</u>	<u>Arti</u>
1. 	1. Entity
2. 	2. Weak Entity
3. 	3. Relationship
4. 	4. Identifying Relationship
5. 	5. Atribut



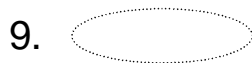
6. Atribut Primary Key



7. Atribut Multivalued



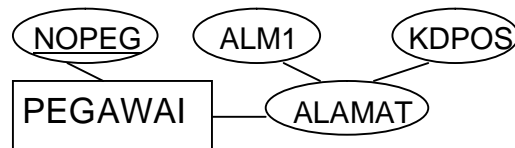
8. Atribut Composite



9. Atribut Derivatif

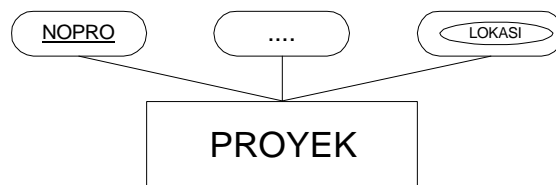
Transformasi dari ERD ke Database Relasional

1. Setiap tipe Entity dibuat suatu relasi yang memuat semua atribut simple, sedangkan untuk atribut composite hanya dimuat komponen-komponennya saja.



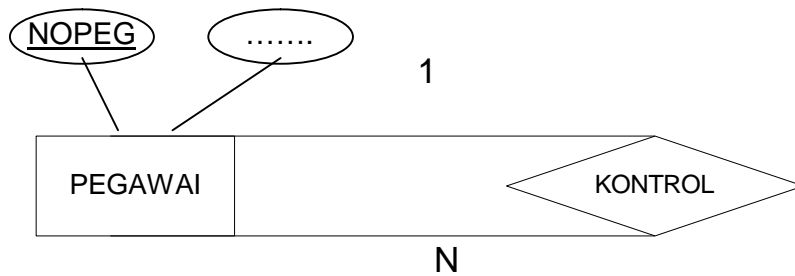
PEGAWAI (NOPEG, ALM1, KDPOS,

2. Setiap relasi yang mempunyai atribut multivalued, buatlah relasi baru dimana Primary Keynya merupakan gabungan dari Primary Key dari relasi tersebut dengan atribut multivalued.



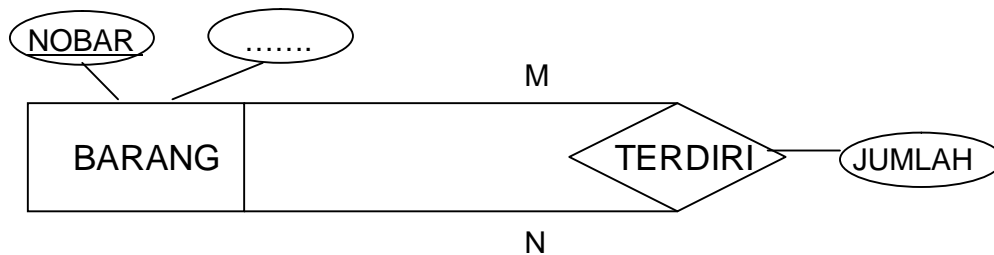
LOKPR(NOPRO, LOKASI)

3. Setiap Unary Relationship 1:N, pada relasi perlu ditambahkan suatu foreign key yang menunjuk ke nilai primary keynya.



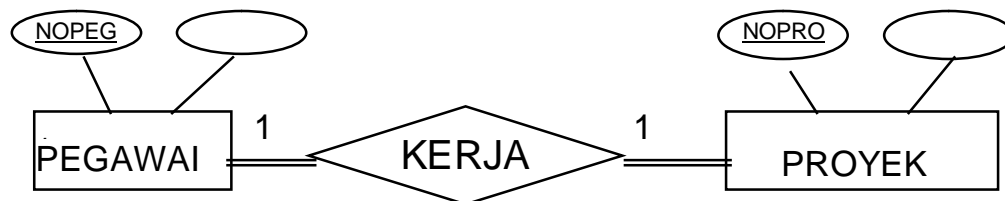
PEGAWAI (NOPEG,, SUPERVISOR-ID)

4. Setiap Unary Relationship M:N, buatlah relasi baru dimana primary keynya merupakan gabungan dari dua atribut dimana keduanya menunjuk ke primary key relasi awal dengan penamaan yang berbeda.



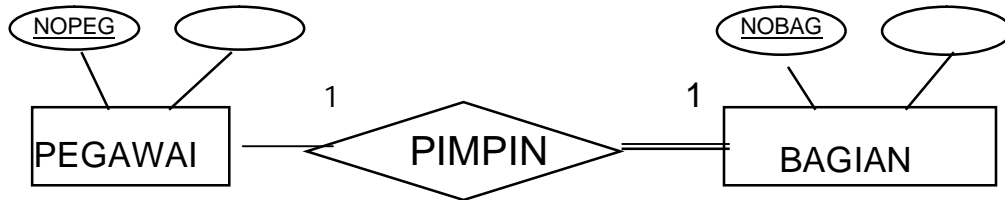
KOMBAR (NOBAR, NOKOMP , JUMLAH)

5. Setiap Binary Relationship 1:1, dimana Participation Constraint keduanya total, buatlah suatu relasi gabungan dimana Primary Keynya dapat dipilih salah satu.



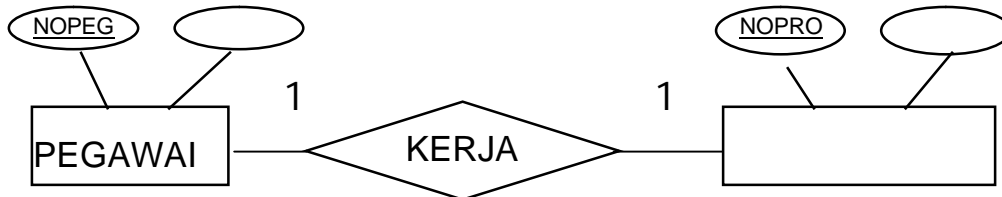
PEGAWAI (NOPEG, ... , NOPRO, ...).

6. Setiap Binary Relationship 1:1 dan salah satu Participation Constraintnya Total, maka Primary Key pada relasi yang Participation Constraintnya Partial menjadi Foreign Key pada relasi yang lainnya.



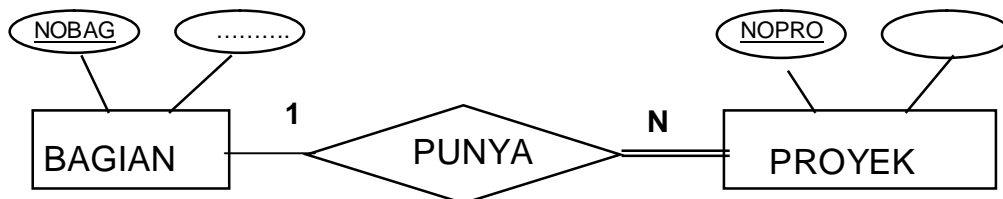
BAGIAN (NOBAG, ... , MANAGER)

7. Setiap Binary Relationship 1:1, dimana kedua Participation Constraintnya partial, maka selain kedua relasi perlu dibuat relasi baru yang berisi Primary Key gabungan dari Primary Key kedua tipe Entity yang berelasi.



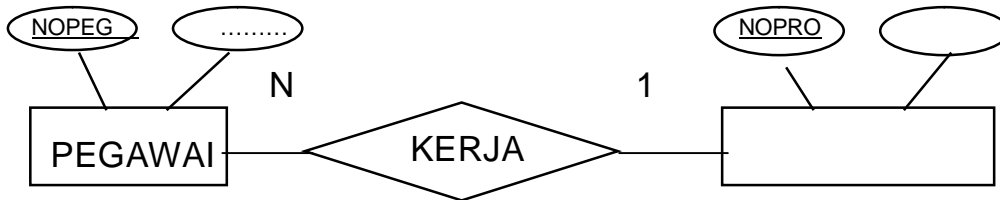
PEKERJAAN (NOPEG, NOPRO, ...)

8. Setiap Binary Relationship 1 : N, dimana tipe Entity yang bersisi N mempunyai Participation Constraint Total, maka Primary Key pada relasi yang bersisi 1 dijadikan Foreign Key pada relasi yang bersisi N.



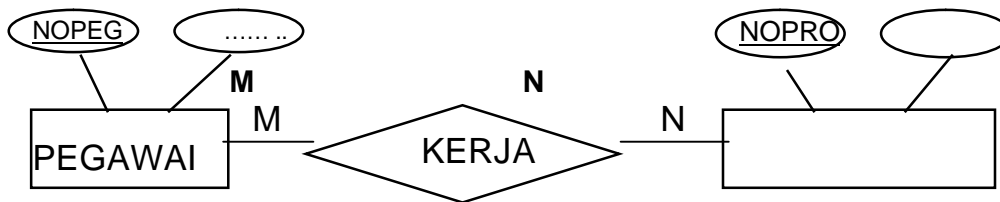
PROYEK (NOPRO, ... , NOBAG)

9. Setiap Binary Relationship 1 : N, dimana tipe Entity yang bersisi N mempunyai Participation Constraint partial, buatlah relasi baru dimana Primary Keynya merupakan gabungan dari Primary Key kedua tipe Entity yang berelasi.



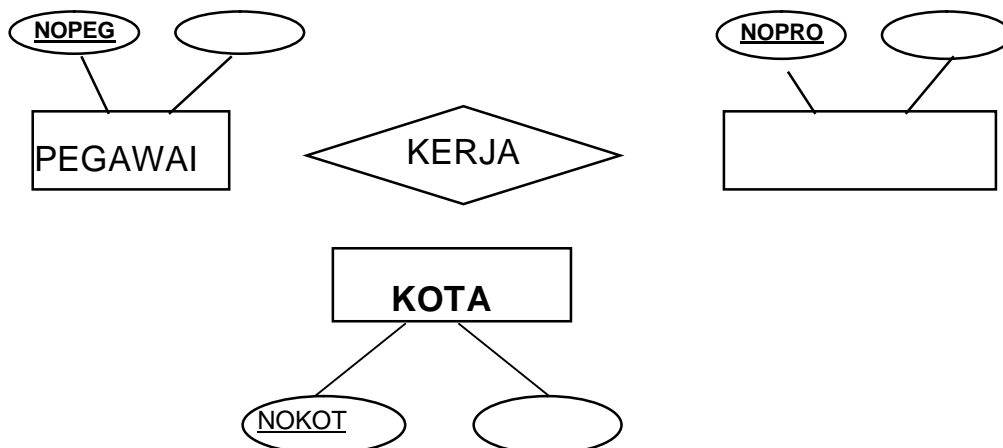
PEKERJAAN (NOPEG, NOPRO,)

10. Setiap Binary Relationship M:N, buatlah relasi baru dimana Primary Keynya merupakan gabungan dari Primary Key kedua tipe Entity yang berelasi.



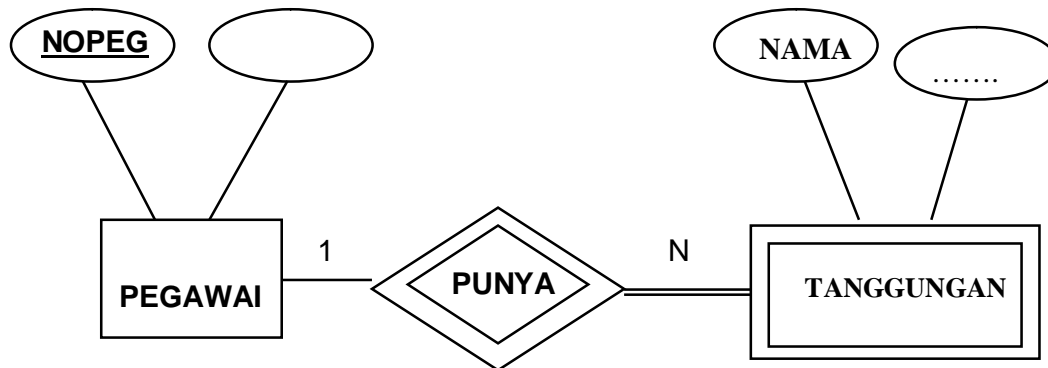
PEKERJAAN (NOPEG, NOPRO,)

11. Setiap Ternary Relationship, buatlah relasi baru dimana Primary Keynya merupakan gabungan dari Primary Key ketiga tipe Entity yang berelasi.



PEKERJAAN (NOPEG, NOPRO, NOKOT)

12. Setiap tipe Weak Entity, dibuat suatu relasi yang memuat semua atributnya dimana Primary Keynya adalah gabungan dari Partial Key dan Primary Key dari relasi induknya (identifying owner).



TANGGUNGAN (NOPEG, NAMA,)

Hasil Transformasi dari Diagram ER ke Database Relasional :

Skema Database

PEGAWAI	(<u>NOPEG</u> , NAPEG, ALM1, KDPOS, TGLLAH, UMUR, SUPERVISOR-ID, NOBAG)
BAGIAN	(<u>NOBAG</u> , NABAG, LOKASI, MANAGER)
PROYEK	(<u>NOPRO</u> , NAPRO, NOBAG)
LOKPR	(<u>NOPRO</u> , LOKAPR)
PEKERJAAN	(<u>NOPEG</u> , <u>NOPRO</u> , JAM)
TANGGUNGAN	(<u>NOPEG</u> , NAMA, JNKELT, HUBUNGAN)

3. NORMALISASI

Normalisasi adalah suatu teknik untuk mengorganisasi data ke dalam tabel-tabel untuk memenuhi kebutuhan pemakai di dalam suatu organisasi.

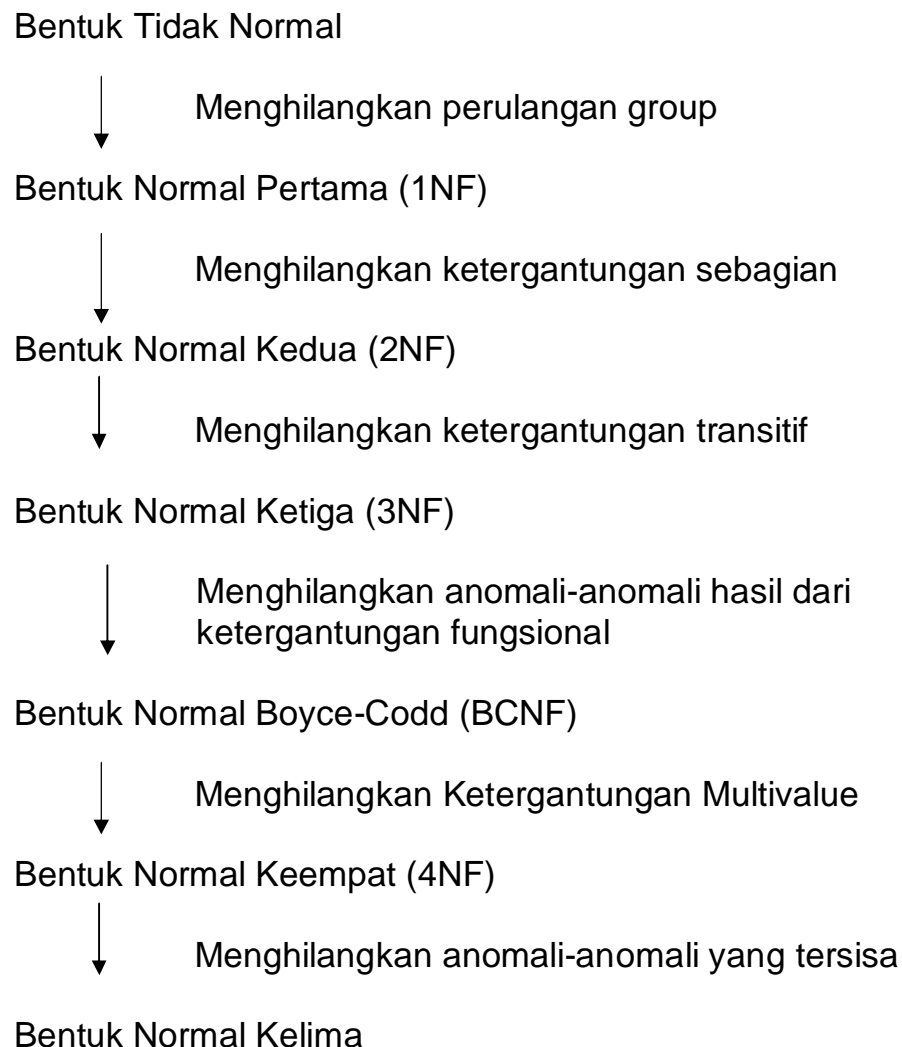
Tujuan dari Normalisasi

- § Untuk menghilangkan kerangkapan data
- § Untuk mengurangi kompleksitas
- § Untuk mempermudah pemodifikasian data

Proses Normalisasi

- § Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat.
- § Apabila tabel yang diuji belum memenuhi persyaratan tertentu, maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal.

Tahapan Normalisasi



Ketergantungan Fungsional

Definisi :

Atribut Y pada relasi R dikatakan tergantung fungsional pada atribut X ($R.X \rightarrow R.Y$), jika dan hanya jika setiap nilai X pada relasi R mempunyai tepat satu nilai Y pada R.

Misal, terdapat skema database Pemasok-barang :
Pemasok (No-pem, Na-pem)

Tabel PEMASOK-BARANG

<u>No-pem</u>	Na-pem
P01	Baharu
P02	Sinar
P03	Harapan

Ketergantungan fungsional dari tabel PEMASOK-BARANG adalah :
No-pem \rightarrow Na-pem

Ketergantungan Fungsional Penuh

Definisi :

Atribut Y pada relasi R dikatakan tergantung fungsional penuh pada atribut X pada relasi R, jika Y tidak tergantung pada subset dari X (bila X adalah key gabungan)

Contoh :

KIRIM-BARANG(No-pem, Na-pem, No-bar, Jumlah)

<u>No-pem</u>	Na-pem	<u>No-bar</u>	Jumlah
P01	Baharu	B01	1000
P01	Baharu	B02	1500
P01	Baharu	B03	2000
P02	Sinar	B03	1000
P03	Harapan	B02	2000

Ketergantungan fungsional :

No-pem --> Na-pem

No-bar, No-pem --> Jumlah (Tergantung penuh terhadap key-nya)

Ketergantungan Transitif

Definisi :

Atribut Z pada relasi R dikatakan tergantung transitif pada atribut X , jika atribut Y tergantung pada atribut X pada relasi R dan atribut Z tergantung pada atribut Y pada relasi R. $\rightarrow X \rightarrow Y, \rightarrow Y \rightarrow Z, \text{ maka } X \rightarrow Z$

Contoh :

<u>No-pem</u>	Kode-kota	Kota	<u>No-bar</u>	Jumlah
P01	1	Jakarta	B01	1000
P01	1	Jakarta	B02	1500
P01	1	Jakarta	B03	2000
P02	3	Bandung	B03	1000
P03	2	Surabaya	B02	2000

Ketergantungan fungsional :

No-pem → Kode-kota
 Kode-kota → Kota , maka
 No-pem → Kota

Bentuk Normal Kesatu (1NF)

Suatu relasi dikatakan sudah memenuhi Bentuk Normal Kesatu bila setiap data bersifat atomik yaitu setiap irisan baris dan kolom hanya mempunyai satu nilai data

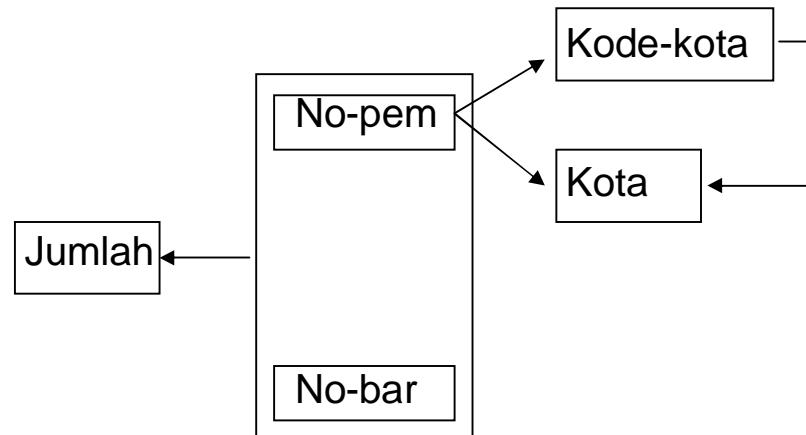
Tabel KIRIM-1 (Unnormal)

No-pem	Kode-kota	Kota	No-bar	Jumlah
P01	1	Jakarta	B01	1000
			B02	1500
			B03	2000
P02	3	Bandung	B03	1000
P03	2	Surabaya	B02	2000

Tabel KIRIM-2 (1NF)

<u>No-pem</u>	Kode-kota	Kota	<u>No-bar</u>	Jumlah
P01	1	Jakarta	B01	1000
P01	1	Jakarta	B02	1500
P01	1	Jakarta	B03	2000
P02	3	Bandung	B03	1000
P03	2	Surabaya	B02	2000

Diagram Ketergantungan Fungsional



Bentuk Normal Kedua (2NF)

Suatu relasi dikatakan sudah memenuhi Bentuk Normal Kedua bila relasi tersebut sudah memenuhi bentuk Normal kesatu, dan atribut yang bukan key sudah tergantung penuh terhadap keynya.

Tabel PEMASOK-1 (2NF)

<u>No-pem</u>	Kode-kota	Kota
P01	1	Jakarta
P02	3	Bandung
P03	2	Surabaya

Bentuk Normal Ketiga (3NF)

Suatu relasi dikatakan sudah memenuhi Bentuk Normal ketiga bila relasi tersebut sudah memenuhi bentuk Normal kedua dan atribut yang bukan key tidak tergantung transitif terhadap keynya.

Tabel KIRIM-3 (3NF)

<u>No-pem</u>	<u>No-bar</u>	Jumlah
P01	B01	1000
P01	B02	1500
P01	B03	2000
P02	B03	1000
P03	B02	2000

Tabel PEMASOK-2 (3NF)

<u>No-pem</u>	Kode-kota
P01	1
P02	3
P03	2

Tabel PEMASOK-3 (3NF)

<u>Kode-kota</u>	Kota
1	Jakarta
2	Surabaya
3	Bandung

Normalisasi pada database perkuliahan

Asumsi :

- § Seorang mahasiswa dapat mengambil beberapa mata kuliah
- § Satu mata kuliah dapat diambil oleh lebih dari satu mahasiswa
- § Satu mata kuliah hanya diajarkan oleh satu dosen
- § Satu dosen dapat mengajar beberapa mata kuliah
- § Seorang mahasiswa pada mata kuliah tertentu hanya mempunyai satu nilai

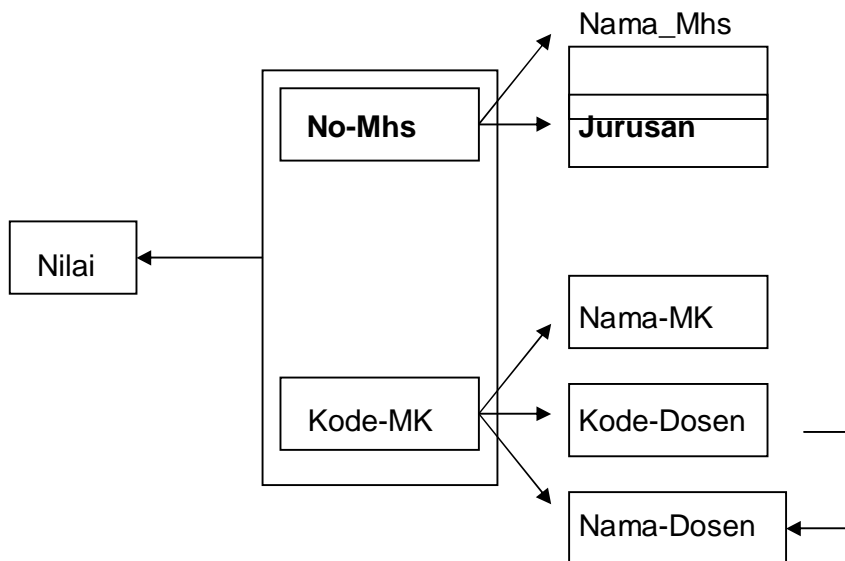
Tabel MAHASISWA-1 (Unnormal)

No-Mhs	Nama - Mhs	Jurusan	Kode-MK	Nama-MK	Kode-Dosen	Nama-Dosen	Nilai
2683	Welli	MI	MI350	Manajamen DB	B104	Ati	A
			MI465	Analnsis Prc. Sistem	B317	Dita	B
5432	Bakri	Ak.	MI350	Manajemen DB	B104	Ati	C
			AKN201	Akuntansi Keuangan	D310	Lia	B
			MKT300	Dasar Pemasaran	B212	Lola	A

Tabel MAHASISWA-2 (1NF)

No-Mhs	Nama-Mhs	Jurusan	Kode-MK	Nama-MK	Kode-Dosen	Nama-Dosen	Nilai
2683	Welli	MI	MI350	Manajamen DB	B104	Ati	A
2683	Welli	MI	MI465	Analnsis Prc. Sistem	B317	Dita	B
5432	Bakri	Ak.	MI350	Manajemen DB	B104	Ati	C
5432	Bakri	Ak.	AKN201	Akuntansi Keuangan	D310	Lia	B
5432	Bakri	Ak.	MKT300	Dasar Pemasaran	B212	Lola	A

Diagram Ketergantungan Fungsional



Tabel KULIAH (2NF)

	Nama-MK	Kode-Dosen	
MI350	Manajamen DB	B104	Ati
MI465	Analsis Prc. Sistem	B317	Dita
AKN201	Akuntansi Keuangan	D310	Lia
MKT300	Dasar Pemasaran	B212	Lola

Tabel MAHASISWA-3 (3NF)

	Nama-Mhs	Jurusan
2683	Welli	MI
5432	Bakri	Ak.

Tabel NILAI (3NF)

		Nilai
2683	MI350	A
2683	MI465	B
5432	MI350	C
5432	AKN201	B
5432	MKT300	A

Tabel MATAKULIAH (3NF)

<u>Kode-MK</u>	Nama-MK	Kode-Dosen
MI350	Manajamen DB	B104
MI465	Analsis Prc. Sistem	B317
AKN201	Akuntansi Keuangan	D310
MKT300	DasarPemasaran	B212

Tabel DOSEN (3NF)

<u>Kode- Dosen</u>	Nama-Dosen
B104	Ati
B317	Dita
B310	Lia
B212	Lola