



BELAJAR SENDIRI

Pemrograman
Database Menggunakan



FoxPro 9.0

Panduan bagi programmer muda dan mahasiswa
dalam pengimplementasian konsep database

Free CD

- File setup aplikasi penjualan
- Bab-bab pilihan dari 100 judul buku komputer best seller Elex (PDF)



Menjadi mahir tanpa guru

Frieyadie, S.Kom



Daftar Isi

Dedikasi	iii
Prakata	v
Daftar Isi	vii
Bab 1 Permodelan Sistem Informasi	
1.1. Diagram Alir Data	3
1.2. Normalisasi	9
1.3. Entity Relationship Diagram	12
1.4. Logical Relationship Structure	16
Bab 2 Analisa Perancangan Sisten Basis Data.....	19
2.1. Normalisasi	20
2.2. Entity Relationship Diagram	27
2.3. Logical Relationship Structure	29
2.4. Rancangan Table	30
Bab 3 Perancangan DBMS Dengan Visual FoxPro.....	33
3.1. Membuat Project.....	33
3.2. Membuat Database	35
3.3. Membuat Table	39

Bab 4 Implementasi Perancangan Aplikasi dengan Visual FoxPro	49
4.1. Membuat Form Barang.....	49
4.2. Membuat Form Pelanggan	77
4.3. Membuat Form Transaksi Penjualan	91
4.4. Membuat Form Cetak Faktur	118
4.5. Membuat Form Cetak Kartu Pelangganr	124
4.6. Membuat Laporan Penjualan Bulanan Penjualan.....	130
4.7. Membuat Form Form About	137
Bab 5 Perancangan Report Dengan Visual FoxPro	139
5.1. Membuat Kartu Pelanggan	139
5.2. Membuat Faktur Penjualan	144
5.3. Membuat Faktur Penjualan Secara Wizard	155
5.4. Membuat Laporan Barang	162
5.5. Membuat Laporan Pelanggan	165
5.6. Membuat Laporan Penjualan	168
Bab 6 Perancangan Menu Utama	175
6.1. Membuat Sistem Menu	175
6.2. Membuat Form Menu	187
6.3. Membuat Form Utama	189
6.3. Mengkompilasi Project	191
Bab 7 Membuat Distribusi Aplikasi	195
7.1. Instalasi Install Shield Express Visual FoxPro Limited Edition Version 5.0	195
7.2. Memulai Pendistribusian Aplikasi.....	203
Daftar Pustaka	210



Permodelan Sistem Informasi

Sebelum mengembangkan dan menerapkan suatu sistem informasi alangkah baiknya merancang sistem informasi tersebut dengan menggunakan beberapa model perancangan sistem informasi..

Permodelan digunakan untuk mempermudah *system analyst* untuk melakukan perancangan dan pengembangan sistem informasi.

Ada beberapa hal mengapa permodelan digunakan, antara lain adalah :

1. Agar mudah dimengerti dan dipahami.
2. Mengidentifikasi kemungkinan kesalahan dan mencegah kerangkapan design yang dibutuhkan.
3. Memprediksi waktu pembuatan dan memprediksi biaya yang dibutuhkan.
4. Menganalisa jalannya sistem informasi.
5. Menjadi dasar pengembangan sistem informasi selanjutnya.

System Analyst harus mengetahui kebutuhan pengguna, seperti form yang digunakan, menu-menu yang digunakan, laporan yang dihasilkan, keamanan data dan informasi dan lain-lain, dengan dasar kebutuhan pengguna tersebut aplikasi dirancang, rancangan tersebut mulai dari yang sederhana sampai dengan rancangan yang lebih kompleks.

Untuk mengetahui kebutuhan-kebutuhan pengguna tersebut, beberapa hal yang dapat dilakukan oleh *system analyst*, antara lain :

1. Melakukan wawancara terhadap bagian-bagian yang terkait yang akan menggunakan atau berhubungan dengan sistem informasi tersebut.
2. Mengumpulkan data-data yang berkaitan dengan sistem informasi yang akan digunakan.

Biasanya *System Analyst* melakukan beberapa tahapan yang berkenaan dengan perancangan sistem, antara lain :

1. Analisa Sistem Berjalan.
Dalam analisa sistem berjalan ini, *system analyst* melakukan sistem yang telah berjalan disuatu perusahaan atau organisasi, dari sini dapat dilihat kelebihan dan kekurangan dari sistem yang ada.
2. Analisa Sistem Usulan atau Analisa Sistem yang dikembangkan dari sistem berjalan yang ada.
Jika sudah melakukan analisa sistem berjalan, *system analyst* biasanya memberikan usulan untuk memperbaiki bagian-bagian dari sistem berjalan yang kurang baik.

Tujuan dari perancangan sistem informasi adalah :

1. Untuk memenuhi kebutuhan dari pengguna sistem informasi agar supaya tujuan yang diinginkan oleh perusahaan atau organisasi tercapai.
2. Untuk memberikan gambaran yang jelas dari rancang bangun sistem informasi yang diinginkan oleh pengguna kepada pemrogram (*programmer*) dan ahli teknik (*engginer*) yang terlibat didalam pembuatan sistem informasi tersebut.

Sasaran utama dari perancangan sistem informasi adalah supaya pekerjaan menjadi efektif dan efisien dalam hal pengolahan data sampai dengan keluaran data.

Berikut ada beberapa permodelan sistem informasi yang dapat digunakan untuk merancang sistem informasi.

1.1. Diagram Alir Data

Diagram Alir Data (DAD) atau *Data Flow Diagram* (DFD), diperkenalkan pada akhir 1970 dan dipopulerkan oleh sebagai alat desain dan analisa sistem aplikasi oleh Gane dan Sarson atau Yoarn dan Coad pada tahun 1979. DAD menunjukkan setiap alur data pada entitas luar pada sebuah sistem, begitu pula pergerakan data dari proses ke proses satu ke proses lainnya pada media penyimpanan secara logika.

Diagram Alir Data adalah jaringan yang menggambarkan sebuah sistem, baik sistem manual, otomatis maupun campuran antara keduanya.

Diagram Arus Data digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem yang baru akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir serta disimpan.

1.1.1. Elemen Diagram Alir Data

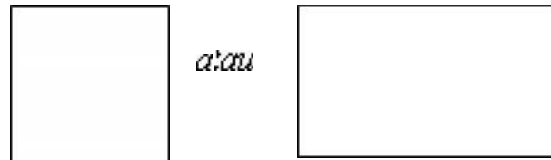
Diagram Alir Data mempunyai empat elemen dasar yang terdiri atas :

a. Entitas / Lingkungan Luar (*External Entity*)

Dikenal juga dengan Source atau sumber yang bisa berupa manusia, organisasi, departemen atau sistem lain. Source adalah masukan bersih kesistem yang dibuat modelnya tersebut. Terkadang source disebut Origin. Sink atau sasaran adalah keluaran bersih dari sistem yang dijadikan model tersebut. Sink juga disebut sebagai Destination. Sink juga bisa berupa orang, organisasi, departemen atau sistem lain.

Digambarkan dengan simbol Empat Persegi Panjang (model *Yourdon* dan *DeMarco*) atau Bujur Sangkar (model *Gane* dan *Sarson*). Menunjukkan entitas atau kesatuan yang berhubungan dengan sistem,

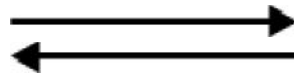
dapat berupa orang, organisasi atau sistem lainnya yang akan memberikan input atau menerima output dari sistem atau kedua-duanya.



Gambar 1.1. Lingkungan Luar

b. Arus Data (*Data Flow*).

Digambarkan dengan anak panah. Menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem dan mengalir di antara proses (*process*), simpanan data (*data store*) dan entitas (*external entity*).

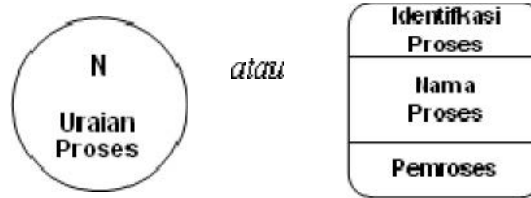


Gambar 1.2. Aliran Data

c. Proses (*Process*).

Proses mentransformasi masukan menjadi keluaran. Digambarkan dengan simbol Lingkaran (model *Yourdon* dan *DeMarco*). Simbol lingkaran juga disebut **Gelembung** atau **Transform**. Proses juga digambarkan dengan **Rounded Rectangle** atau persegi panjang yang mempunyai pojok-pojok yang bulat (model *Gane* dan *Sarson*). Proses menunjukkan kegiatan atau kerja yang dilakukan oleh orang, mesin atau

komputer dari hasil suatu data yang masuk kedalam proses untuk dihasilkan arus data yang akan keluar dari proses.



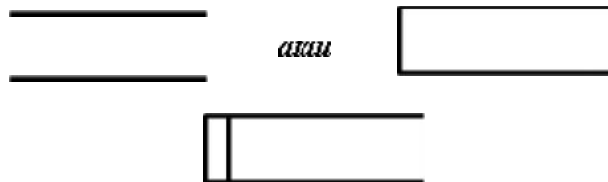
Gambar 1.3. Proses

Keterangan :

- **Identifikasi Proses (N)**, berupa angka yang menunjukkan nomor acuan dari proses.
- **Nama Proses (Uraian Proses)**, menunjukkan apa yang sedang dikerjakan oleh proses tersebut.
- **Pemroses**, menunjukkan siapa atau dimana suatu proses dilakukan

d. Simpanan Data (Data Store)

Digambarkan dengan sepasang garis horizontal. Menunjukkan suatu tempat penyimpanan data yang dapat berupa suatu file di sistem komputer, arsip atau catatan manual, tabel acuan manual dan lain-lain.



Gambar 1.4. Simpanan Data

1.1.2. Tingkatan Diagram Alir Data

Tingkatan-tingkatan yang dimiliki oleh Diagram Alir Data sebanyak 3(tiga) tingkatan. Berikut penjelasan tingkatan – tingkatan yang dimiliki oleh Diagram Alir Data (*DAD*)

1. Diagram Konteks.

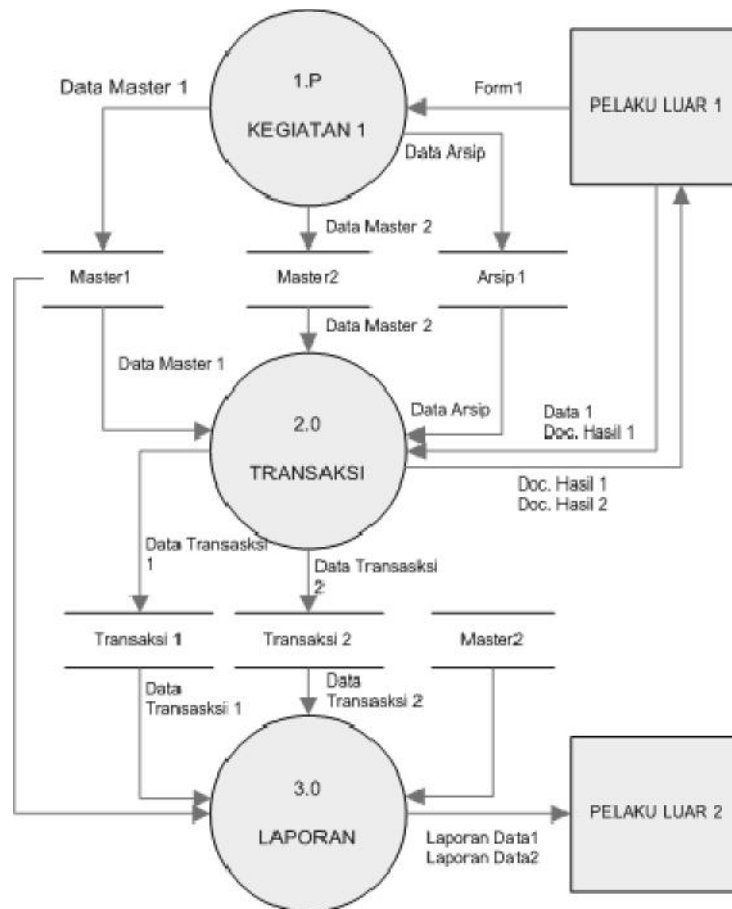
Diagram konteks merupakan diagram yang paling atas yang terdiri dari satu proses dan menggambarkan ruang lingkup sistem.



Gambar 1.5. Diagram Konteks

2. Diagram Nol (*Overview*).

Diagram Nol merupakan tingkatan diagram setelah diagram konteks, yang menggambarkan proses utama dari Diagram Arus Data.



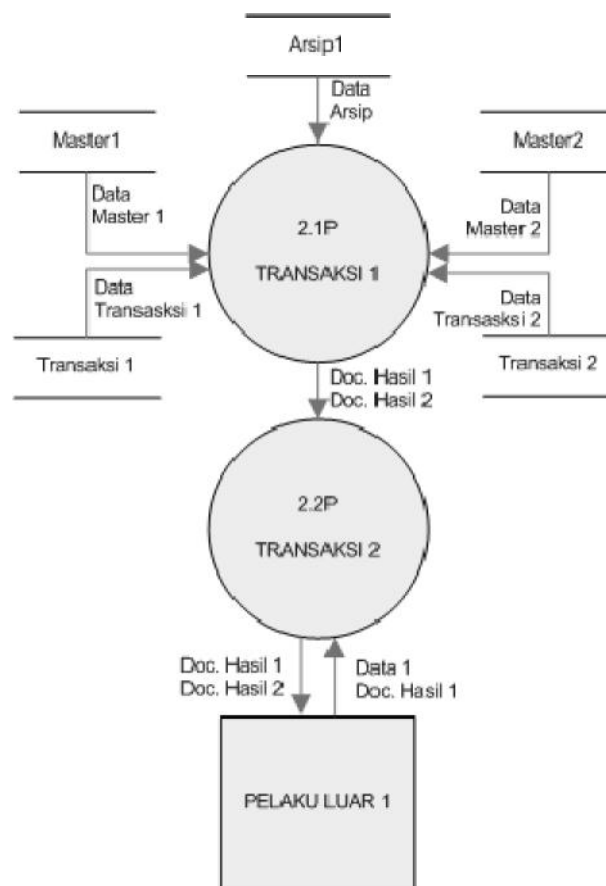
Gambar 1.6. Diagram Konteks

Keterangan :

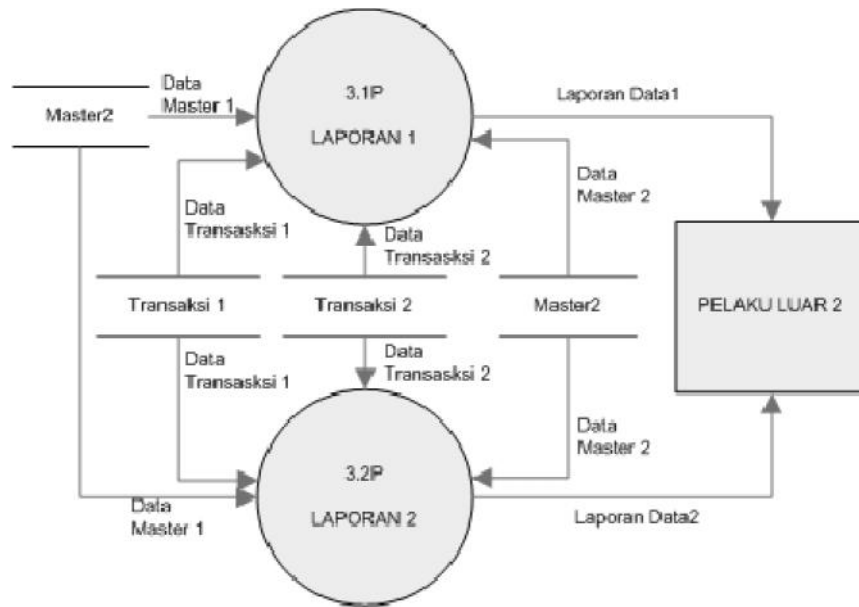
P pada proses Kegiatan 1 adalah Primitif yang artinya bahwa tidak mempunyai proses lagi didalamnya.

3. Diagram Detail

Diagram Detail merupakan tingkatan diagram setelah diagram Nol, yang menggambarkan subproses dari masing-masing proses utama dari Diagram Nol



Gambar 1.7. Diagram Detail Proses 2.0



Gambar 1.8. Diagram Detail Proses 3.0

1.2. Normalisasi

Normalisasi bertujuan untuk mengurangi ketidaknormalan desain tabel yang redundansi atau tabel yang mempunyai struktur atau nilai ganda.

1.2.1. Jenis Kunci Atribut (Kunci Field)

Kunci Atribut atau kunci field yang dapat mewakili suatu record data.

a. Kunci Kandidat (Candidate Key)

Kunci Kandidat adalah suatu kunci yang mengidentifikasi secara unik dari suatu kejadian spesifik dari entity. Jika suatu kunci kandidat berisi lebih dari satu atribut, disebut dengan *composite key* atau kunci gabungan. Setiap kunci kandidat berpeluang menjadi kunci utama.

b. Kunci Utama (Primary Key)

Kunci Utama adalah suatu kunci yang mengidentifikasi secara unik dari suatu kejadian spesifik dan mewakili setiap kejadian dari suatu entity. Kunci utama bersifat unik dan nilai dari kunci utama tidak sama antara nilai yang satu dengan nilai yang lainnya.

Akan tetapi kunci gabungan dapat juga dijadikan Primary Key.

c. Kunci Alternatif (Alternate Key)

Kunci Alternatif adalah suatu kunci kandidat yang tidak digunakan sebagai kunci utama. Kunci alternatif ini biasa digunakan untuk pencarian suatu data atau sebagai kunci pengurutan suatu data.

d. Kunci Tamu (Foreign Key)

Kunci Tamu adalah suatu kunci yang ditempatkan pada tabel lain yang berguna untuk hubungan ketabel induk.

1.2.2. Bentuk Normalisasi

a. Normalisasi Pertama (1NF)

Bentuk normalisasi ke satu ini setiap atribut hanya memiliki satu pengertian saja dan tidak ada atribut yang bernilai ganda.

b. Normalisasi Kedua (2NF)

Bentuk normalisasi ke dua ini haruslah ditentukan kunci field yang bersifat unik yang dapat mewakili atribut lain yang menjadi anggotanya dan setiap atribut bukan kunci haruslah bergantung fungsi pada kunci utama (*primary key*).

c. Normalisasi Ketiga (3NF)

Sebuah relasi variabel (tabel) berada dalam bentuk 3NF bila dan hanya bila ia berada dalam 2NF dan setiap atribut bukan kunci adalah bergantung secara tidak transitif pada kunci utama yaitu setiap atribut bukan kunci haruslah bergantung pada kunci utama secara menyeluruh.

Jika pada normalisasi ke dua apa bila seluruh atributnya sudah tergantung penuh pada kunci utamanya maka bisa dikatakan normalisasi.

d. Normalisasi Boyce/Codd (BCNF)

Normalisasi BCNF ini merupakan sebuah relasi variabel atau tabel yang berada dalam BCNF apabila satu-satunya *determinan* adalah kunci kandidat. Apabila tidak terdapat BCNF, maka BCNF merupakan normalisasi ketiga (3NF) itu sendiri.

1.3. Entity Relationship Diagram

Permodelan *Entity Relationship Diagram* (ER-D) banyak digunakan dalam merancang tabel dan database serta keterhubungannya atau relasinya. Berikut simbol-simbol yang digunakan didalam merancang dengan menggunakan ER-D.

a. Entitas

Entitas adalah suatu data yang dapat disimpan dan berguna bagi badan atau perusahaan, dengan katalain suatu objek yang dapat dibedakan dengan objek lainnya. Entitas digambarkan dengan kotak persegi panjang.



Gambar 1.9. Entitas

Terdapat juga Entitas Lemah (*Weak Entity*) yaitu suatu entitas sangat bergantung dengan entitas biasa, dengan kata lain adalah entitas lemah tidak akan ada apabila tidak ada entitas biasa. Sebagai contoh, jika terdapat entitas pembelian apabila tidak terdapat entitas permintaan pembelian. Entitas lemah digambarkan dengan kotak persegi panjang dengan garis ganda.



Gambar 1.10. Entitas Lemah

b. Atribut

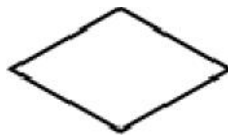
Atribut menunjukan karkarakteristik dari tiap-tiap entitas. Atribut digambarkan dengan bentuk oval.



Gambar 1.11. Atribut

c. Relasi

Relasi menunjukkan hubungan yang terjadi antar entitas. Relasi digambarkan dengan bentuk belah ketupat.



Gambar 1.12. Relasi

d. Line Connector

Line Connector digambarkan dengan bentuk garis tunggal



Gambar 1.13. Line Connector

e. Atribut Utama

Atribut digambarkan dengan bentuk oval, dengan keterangan diberi garis bawah absolut.



Gambar 1.14. Atribut Utama

f. Atribut Pilihan

Atribut digambarkan dengan bentuk oval dengan keterangan diberi garis bawah putus-putus



Gambar 1.15. Atribut Pilihan

Kardinalitas merupakan tingkat hubungan yang terjadi antar entitas didalam sebuah sistem. Terdapat 3 (tiga) tingkat hubungan yang terjadi, yaitu :

1. Hubungan Satu pada Satu (One To One atau 1:1)

Tingkat hubungan dinyatakan satu pada satu, jika satu kejadian pada entitas pertama hanya mempunyai satu hubungan dengan suatu kejadian pada entitas ke dua. Demikian juga sebaliknya, satu kejadian pada entitas kedua hanya bisa mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama.

Contoh : Setiap Satu Mobil hanya mempunyai Satu STNK



Gambar 1.16. Hubungan One To One

2. Hubungan Satu pada Banyak (One To Many atau 1:M)

Tingkat hubungan satu pada Banyak (1 : M) adalah sama dengan banyak pada satu (M : 1), tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua. Sebaliknya satu kejadian pada entitas yang kedua, hanya bisa mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama.

Contoh : Satu Bus dinaki oleh banyak Penumpang



Gambar 1.17. Hubungan One To Many

3. Hubungan Banyak pada Banyak (Many To Many atau M:N)

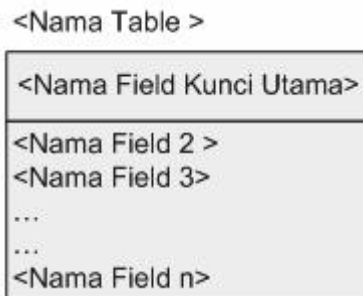
Tingkat hubungan banyak pada banyak (M : N) terjadi jika, tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya, baik dilihat dari sisi entitas yang pertama maupun dilihat dari sisi entitas yang kedua



Gambar 1.18. Hubungan Many To Many

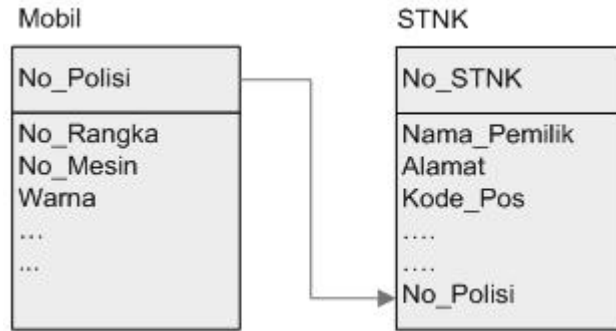
1.4. Logical Relationship Structure

LRS merupakan hasil dari pemodelan ER beserta atributnya. Pada LRS bisa terlihat hubungan-hubungan setiap entitas. LRS bisa dilihat relational model secara logic atau level external dan konsep sebelum dibentuk tabel secara level internal atau fisik, yang dari *field* dari atribut entitas. Penggambaran LRS seperti gambar dibawah ini.



Gambar 1.19. Logical Relationship Structure

Sebagai contoh bisa dilihat pada gambar 1.16, menunjukkan terdapat 2(dua) buah Entitas. Pada Entitas Mobil sebagai Primary Key adalah No_Polisi dan Entitas STNK sebagai Primary Key adalah No_STNK. Pada Saat bersamaan No_Polisi menjadi Foreign Key pada Entitas STNK



Gambar 1.20. LRS Entitas Mobil dan STNK

Dalam pembuatan LRS terdapat 3 (tiga) hal yang dapat mempengaruhi, yaitu :

1. Jika tingkat hubungan (*cardinality*) satu pada satu (*one-to-one*), maka digabungkan dengan entitas yang lebih kuat (*strong entity*), atau digabungkan dengan entitas yang memiliki atribut yang lebih sedikit.
2. Jika tingkat hubungan (*cardinality*) satu pada banyak (*one-to-many*), maka hubungan relasi atau gabungkan dengan entitas yang tingkat hubungannya banyak
3. Jika tingkat hubungan (*cardinality*) banyak pada banyak (*many-to-many*), maka hubungan relasi tidak akan digabungkan dengan entitas manapun, melainkan menjadi sebuah LRS.

lembar ini sengaja dikosongkan