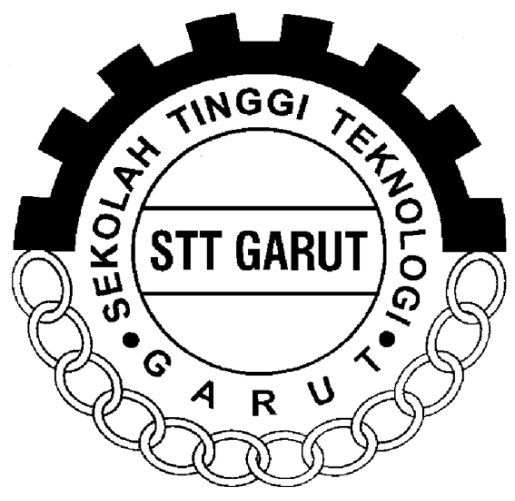
**LAPORAN PRAKTIKUM BASIS DATA**

**MODUL 1- 8**

*Ditujukan untuk memenuhi tugas Praktikum Basis Data*



Disusun Oleh :

Asep Saepul Milah ( 1406022 )

Raden Nazar L ( 1406096 )

Sifa Fauziah ( 1406122 )

Toni Nugraha ( 1406127 )

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TINGGI TEKNOLOGI GARUT**

**2015/2016**

**MODUL I**

**KONSEP DASAR BASIS DATA**

* 1. **Tujuan**

Tujuan praktikum basis data “ Konsep Dasar Basis Data” adalah agar :

* Mengetahui konsep dasar perancangan basis data
* Memahami hirarki data dan implementasi data
  1. **Dasar Teori**

1. Konsep Data, Informasi dan Basis Data

Data merupakan fakta mengenai suatu objek seperti manusia, benda, peristiwa, konsep, keadaan dan sebagainya yang dapat dicatat dan mempunyai arti secara implisit. Basis data merupakan kumpulan informasi bermanfaat yang diorganisasikan ke dalam suatu aturan yang khusus. Informasi ini adalah data yang telah diorganisasikan ke dalam bentuk yang sesuai dengan kebutuhan seseorang.

1. Hirarki Data

Data diorganisasikan kedalam bentuk elemen data (*field*), rekaman (*record*), dan berkas (*file*). Definisi dari ketiganya adalah sebagai berikut :

* Elemen data adalah satuan data terkecil yang tidak dapat dipecah lagi menjadi unit lain yang bermakna. Misalnya data siswa terdiri dari NIS, Nama, Alamat, Telepon atau Jenis Kelamin.
* Rekaman merupakan gabungan sejumlah elemen data yang Saling terkait. Istilah lain dari rekaman adalah baris atau tabel.
* Berkas adalah himpunan seluruh rekaman yang bertipe sama.

1. Sistem Basis Data

Gabungan antara basis data dan perangkat lunak SMBD (Sistem Manajemen Basis Data) termasuk di dalamnya program aplikasi yang dibuat dan bekerja dalam satu sistem disebut dengan Sistem Basis Data.

1. Databases Managment Sistem

DBMS dapat diartikan sebagai program komputer yang digunakan untuk memasukkan, mengubah, menghapus, memodifikasi dan memperoleh data/informasi dengan praktis dan efisien.

1. Model Data

Model data dapat dikelompokkan berdasarkan konsep pembuatan deskripsi struktur basis data, yaitu:

* Model data konsepsual (*high level*) menyajikan konsep tentang bagaiman *user* memandang atau memperlakukan data. Dalam model ini dikenalkan tiga konsep penyaj data yaitu:
* *Entity* (entitas) merupakan penyajian obyek, kejadian atau konsep dunia nyata yang keberadaannya seeara eksplisit didefinisikan dan disimpan dalam basis data, contohnya Mahasiswa, Matakuliah, Dosen, Nilai dan lain sebagainya.
* *Atribute* (atribut) adalah keterangan-keterangan yang menjelaskan karakteristik dari Suatu entitas seperti NIM, Nama, Fakultas, Jurusan untuk entitas Mahasiswa.
* *Relationship* (hubungan) merupakan hubungan atau interaksi antara satu entitas dengan yang lainnya, misalnya entitas pelanggan berhubungan dengan entitas barang yang dibclinya.
* Model data fisikal (*low level*) merupakan konsep bagaimana deskripsi detail data disimpan ke dalam komputer dengan menyajikan informasi tentang format rekaman, urutan rekaman, dan jalur pengaksesan data yang dapat membuat pemcarian rekaman data lebih elisien.
* Model data implementasi (representational) merupakan konsep deskripsi data disimpan dalam komputer dengan menyembunyikan sebagian detail deskripsi data sehingga para *user* mendapat gambaran global bagaimana data disimpan dalam komputer. Model ini merupakan konsep model data yang digunakan Oleh model hirarki, jaringan dan relasional.

1. Metode perancangan Basis Data

Metode perancangan basis data adalah kumpulan teknik terorganisasi untuk pembuatan rancangan basis data. Teknik terorganisasi ini merupakan kumpulan tahap-tahapan yang memiliki aturan-aturan terurut. Teknik yang digunakan pada perancangan basis data dibagi menjadi dua, yaitu:

1. Perancangan basis data tingkat logik.
2. Perancangan basis data tingkat fisik.

Perancangan basis data secara logik dimulai dengan penciptaan model konseptual dari organisasi dan seluruhnya tak bergantung rincian implementasi seperti perangkat lunak DBMS, program aplikasi, bahasa pemrograman, platform perangkat keras, dan pertimbangan fisik lainnya. Model konsep ini kemudian dipetakan menjadi model data secara logik yang telah dipengaruhi model data target basis data seperti model relasional. Dalam perancangan basis data secara logik, kita dapat melakukannya dengan:

* Menerapkan Normalisasi terhadap struktur tabel yang telah diketahui.
* Langsung membuat model *Entity-Relationship* (ER).

Perancangan basis data secara fisik adalah proses memproduksi deskripsi implementasi basis data pada penyimpanan sekunder, mendeskripsikan struktur-struktur penyimpanan dan metode-metode pengaksesan dalam meningkatkan efektifitas pengaksesan. pada tahap ini, perancangan fisik telah ditujukan untuk sistem DBMS tertentu.

Langkah – langkah perancangan basis data :

Berikut adalah perancangan basis data relasional :

* Dimulai dari perancangan basis data logik untuk basis data relasional pada tahap I sampai dengan tahap 3.
* Perancangan dan implementasi basis data fisik untuk basis data relasional pada tahap 4 sampai dengan tahap 7.

1. Tahap 1

Membangun rancangan data konseptual lokal berdasarkan pandangan pemakai.Yaitu ngidentifikasikan himpunan entitas - himpunan entitas. Mengidentifikasikan keterhubungan-keterhubungan (*relationship*), mengidentifikasikan dan mengasosiasikan atribut-atribut pada entitas atau keterhubungan, menentukan domain atribut, menentukan atribut-atribut *candidate key* dan *primary key*, melakukan spesialisasi/generalisasi, menggambarkan diagram ER, melakukan *review model* data konsep dengan pemakai.

1. Tahap 2

Membangun dan validasi model data logik lokal. Yaitu memetakan model data konsep ke model data logik, melakukan turunan relasi-relasi dari model data logik, validasi model menggunakan normalisasi, validasi model berdasarkan transaksi transaksi pemakai, menggambarkan ER nya, mendefinisikan kontsrain-konstrain (batasan-batasan) integritas, melakukan *review model* data logik dengan pemakai.

1. Tahap 3

Membangun dan validasi model data logik global. Yaitu menggabungkan model data logik lokal menjadi model global, validasi model data logik global, periksa untuk pertumbuhan masa datang, menggambarkan diagram ER akhir, melakukan *review model* logik global dengan pemakai.

1. Tahap 4

Menerjemahkan model data logik global untuk DBMS target. Yaitu merancang relasi-relasi basis untuk DBMS target, merancang aturan-aturan integritas untuk DBMS target.

1. Tahap 5

Merancang dan implementasi representasi fisik. Yaitu menganalisa transaksi- transaksi, memilih organisasi *file*, memilih indeks-indeks sekunder, mempertimbangkan penambahan redudansi (ambigu) yang terkendali, estimasikan ruang disk yang diperlukan.

1. Tahap 6

Merancang dan mengimplementasikan mekanisme pengamanan. Yaitu merancang view-view pemakai, merancang aturan-aturan pengaksesan.

1. Tahap 7

Memonitor dan menyesuaikan sistem yang sedang beroperasi.

* 1. **Praktikum**

1. Jelaskan yang dimaksud dengan basis data? Sebutkan manfaatnya!
2. Sebutkan model data yang anda ketahui dan jelaskan dengan contoh!
3. Apakah yang dimaksud dengan perancangan basis data relasional? berikan contohnya dan jelaskan!

Jawaban :

1. Basis data merupakan kumpulan informasi bermanfaat yang diorganisasikan ke dalam suatu aturan yang khusus. Informasi ini adalah data yang telah diorganisasikan ke dalam bentuk yang sesuai dengan kebutuhan seseorang.

Manfaat :

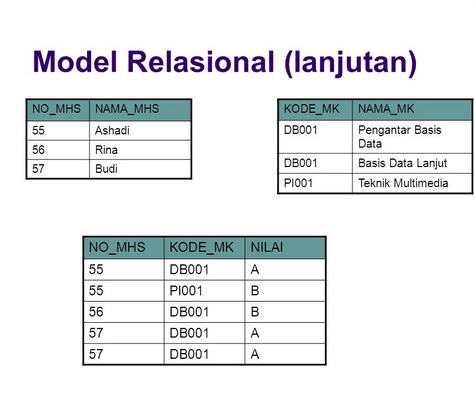
* Kecepatan dan Kemudahan (Speed)
* Kebersamaan Pemakaian (Sharability)
* Pemusatan Kontrol Data
* Efisiensi Ruang Penyimpanan (Space)
* Keakuratan (Accuracy)
* Ketersediaan (Availability)
* Kelengkapan (Completeness)
* Keamanan (Security)
* Kemudahan dalam Pembuatan Program Aplikasi Baru
* User View

1. Model data konsepsual , Model data implementasi, Model data fisikal

Contoh :

1. Basis data relational adalah sekumpulan tabel-tabel yang memiliki hubungan relasi secara matematika dan logika.

Contoh :



**MODUL II**

**RELASI TABEL**

* 1. **Tujuan**

Tujuan praktikum basis data “ Relasi Table” adalah agar :

* Untuk mengetahui hubungan antara data dalam basis data
* Untuk mengetahui perancangan ERD
* Untuk mengetahui kardinalitas antar entitas
  1. **Dasar Teori**

1. Model Relasi Entitas

Merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data yang berdasarkan suatu persepsi bahwa di dunia nyata terdiri dari objek-objek dasar yang memilikl hubungan atau relasi dari objek-objek tersebut. Entitas dalam ER diagram dibedakan menjadi 2 yaitu :

* *Strong entity* (entitas kuat)

Entitas yang mandiri, yang keberadaannya tidak bergantung pada keberadaan entitas yang Iainnya. Instansiasi entitas kuat selalu memiliki karakteristik yang unik disebut *identifier* (sebuah atribut tunggal atau gabungan atribut atribut yang secara unik dapat digunakan untuk membedakannya dari entitas kuat yang lain).

* *Weak entity* (entitas lemah)

Entitas yang keberadaannya sangat bergantung pada keberadaan entitas yang lainnya. Entitas lemah tidak memiliki arti apa-apa dan tidak dikehendaki kehadirannya dalam diagram ER tanpa kehadiran entitas di mana mereka bergantung.

Mengapa jika terdapat set entitas yang sama muncul beberapa kali dalam satu set ER-Diagram ini harus dihindari?

* Untuk menghindari *redundancy*(data yang ambigu)
* Menghemat penyimpanan (*storage*) data
* Mengurangi efektifitas dan kecepatan akses
* Untuk menghindari terjadinya asinkronisasi data pada saat di *update*

1. Derajat Relationship

Derajat Relationship menjelaskan jumlah entitas yang terlibat dalam suatu relationship.

* Unary Degree (Derajat satu) hanya satu entitas yang terlibat



* *Binary Degree* (Derajat Dua) menghubungkan dua entitas



* T*ernary Degree* (Derajat Tiga) menghubungkan tiga entitas



1. Cardinality Ratio Constraint

Cardinality Ratio Constraint menjelaskan batasan jumlah relasi suatu entitas dengan entitas lainnya. Jenis rasio kardinalitas :

1. *One to one*(1 : 1)



1. One to many ( 1 : m )



1. Many to many ( M : M )



1. Atribut

Atribut adalah properti deskriptif yang dimiliki oleh setiap himpunan entitas. Jenis-jenis atribut :

* Atribut *key* digunakan untuk mengidentifikasi suatu entity secara unik
* Atribut tunggal memilikl nilai tunggal
* Atribut multivalue memiliki sekelompok nilai untuk setiap instant entity Atribut composite dapat didekomposisi menjadi beberapa atribut Iain
* Atribut derivative dihasilkan dari atribut yang lain

1. Key

*Key* adalah sejumlah atribut yang mengidentifikasi record/baris dalam sebuah relation secara *unique*(tidak redudan). Beberapa jenis *key*yaitu:

* Super Key

Satu atribut atau kumpulan atribut yang secara unik mengidentifikasi sebuah record di dalam relasi atau himpunan dari satu atau lebih entitas yang dapat digunakan untuk mengidentifikasi secara unik sebuah entitas dalam entitas set.

* Candidate Key

Atribut-atribut yang menjadi determinan yang dapat dijadikan identitas record pada sebuah relation bisa terdapat satu atau lebih candidate key

* Primary Key

Candidate key yang menjadi identitas record karena dapat mengidentifikasi *record* secan unik

* Altenate Key

Kandidate key yang tidak dijadikan *primary key*

* Composite Key

*Key* yang terdiri dari 2 atribut atau lebih. Atribut-atribut tersebut bila berdiri sendiri tidak menjadi identitas *record*, tetapi bila dirangkaikan menjadi satu kesatuan akan dapat mengidentifikasi secara unik.

* Foreign Key

*Non key* atribut pada sebuah relasi yang juga menjadi *key* (*primary*) atribut di relasi lainnya. *Foreign key* biasanya digunakan sebagai penghubung antara *record-record* dan kedua relasi tersebut.

Contoh :

|  |  |  |  |
| --- | --- | --- | --- |
| **S#** | **SName** | **Kode** | **P#** |
| S1 | Riska | 1002 | 2648 |
| S2 | Sandi | 1001 | 2649 |
| S3 | Santi | 1003 | 2641 |

***Keterangan :***

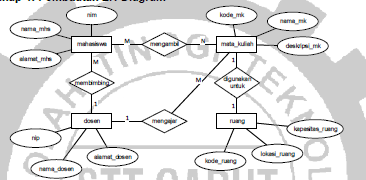
* Super Key : SName, Kode
* Candidate Key : S#, SName
* Primary Key : S#
* Alternative Key : SName
* Foreign Key : Kode

1. Contoh Pembuatan ERD

* Tahap 1 : Penentuan Entitas
* mahasiswa: menyimpan semua informasi pribadi mengenai semua mahasiswa
* dosen: menyimpan semua informasi pribadi mengenai semua dosen
* mata kuliah: menyimpan semua informasi mengenai semua mata kuliah yang ditawarkan
* ruang: menyimpan semua informasi mengenai ruang kelas yang digunakan
* Tahap 2: Penentuan Atribut
* mahasiswa:
* nim: nomor induk mahasiswa (*integer*) *Primary Key*
* nama\_mhs: nama lengkap mahasiswa (*string*)
* alamat\_mhs: alamat lengkap mahasiswa (*string*)
* dosen :
* nip: nomor induk pegawai (*integer*) *Primary Key*
* nama dosen: nama lengkap dosen (*string*)
* alamat dosen: alamat lengkap dosen (*string*)
* mata kuliah :
* kode\_mk: kode untuk mata kuliah (*integer*) PK
* nama\_mk: nama lengkap mata kuliah (*string*)
* deskripsi\_mk: deskripsi singkat mengenai mata kuliah (*string*)
* ruang:
* kode\_ruang: kode untuk ruang kelas (*string*)
* *Primary Key* lokasi\_ruang: deskripsi singkat mengenai lokasi ruang kelas (*string*)
* kapasitas ruang: banyaknya mahasiswa yang dapat ditampung (*integer*)
* Tahap 3: Penentuan *Relationship*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *mahasiswa* | *dosen* | *Mata kuliah* | *ruang* |
| *Mahasiswa* |  | *N : 1* | *M : 1* |  |
| *Dosen* |  |  | *1 : n* |  |
| *Mata kuliah* |  |  |  | *1:1* |
| *ruang* |  |  |  |  |

* Tahap 4 : Pembuatan ERD



* 1. **Praktikum**

Buatlah ER diagram dari kasus tersebut :

Suatu perusahaan *software* dlminta membuatkan basis data yang akan menangani data-data inventaris sebuah toko kecil. Karena tokonya kecil, maka ada beberapa gudang yang khusus untuk menyimpan stock produk. Data-data yang akan ditanganinya adalah: data produk yang ditawarkan toko, data pemasok produk, data transaksi pembelian produk dari pemasok (nota pembelian), dan data gudang tempat penyimpanan produk. Satu produk yang sama bisa disimpan di beberapa gudang yang berbeda, dan tentu saja tiap gudang menyimpan berbagai macam produk. Di *database*harus ada data mengenai sisa stock yang ada di masing-masing gudang untuk semua produk.

* Tahap 1 : Penentuan Entitas

**produk** : menyimpan semua informasi mengenai semua produk yang ditawarkan

**pemasok**: menyimpan semua informasi mengenai semua pemasok

**nota\_pembelian**: menyimpan semua informasi mengenai semua transaksi pembelian produk dari pemasok

**gudang**: menyimpan semua informasi mengenai gudang untuk penyimpanan produk

* Tahap 2 : Penentuan Entitas

**produk**:

kode\_produk : kode unik untuk tiap macam produk (string) PK

nama\_produk : nama lengkap untuk produk (string)

harga\_jual : harga jual produk di toko (integer)

**pemasok** :

kode\_pemasok: kode unik untuk tiap pemasok (string) PK

nama\_pemasok: nama lengkap untuk pemasok (string)

alamat\_pemasok: alamat lengkap untuk pemasok (string)

**nota pembelian** :

no\_nota: kode untuk mata kuliah (integer) PK

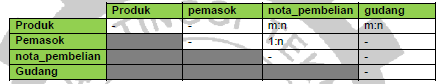
tanggal: tanggal transaksi dilakukan (date)

**gudang:**

kode\_gudang: kode untuk ruang kelas (string) PK

alamat\_gudang: alamat lengkap untuk gudang (string)

* Tahap 3: Penentuan *Relationship*

Hubungan :

* produk disimpan di gudang:

Tabel utama: produk, gudang

Tabel kedua: stok\_produk

Relationship : Many-to-many (m:n)

Attribute penghubung : kode\_produk,kode\_gudang (FK kode\_produk, kode\_gudang di stok produk)

* produk tercatat di nota\_pembelian:

Tabel utama: produk, nota\_pembelian

Tabel kedua : rincian\_nota\_pembelian

Relationship: Many-to-many (m:n)

Attribute penghubung: kode\_produk,no\_nota (FK kode\_produk, no\_nota, di rincian\_nota\_pembelian)

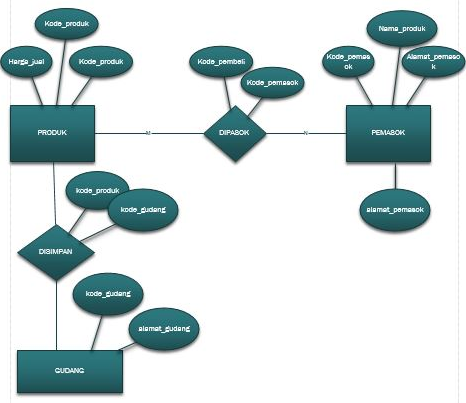
* pemasok tercatat di nota\_pembelian:
* Tabel utama: pemasok

Tabel kedua: nota\_pembelian

Relationship : One-to-many (1:m)

Attribute penghubung: kode\_pemasok (FK kode\_pemasok di nota\_pembelian)12x

* ER Diagram



**MODUL III**

**NORMALISASI**

* 1. **Tujuan**

Tujuan praktikum basis data “ Normalisasi” adalah agar :

* Untuk mengetahui cara mengurangi redudansi data
* Untuk memastikan depedensi data ( data berada pada tabel yang tepat )
  1. **Dasar Teori**

1. Normalisasi

Normalisasi adalah suatu proses memperbaiki / membangun dengan model data relasional, dan secara umum lebih tepat dikoneksikan dengan model data logika. Proses normalisasi adalah proses pengelompokan data elemen menjadi tabel tabel yang menunjukkan entitasdan relasinya.

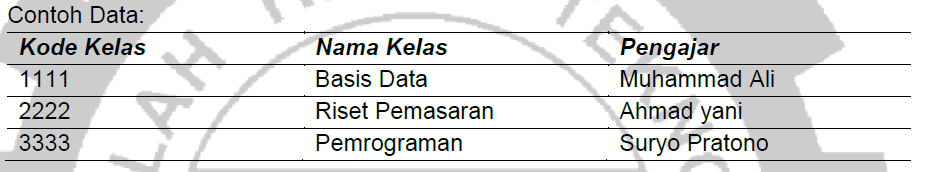
1. Tahapan Normalisasi

* Bentuk Normal Kesatu (1NF / First Normal Form)

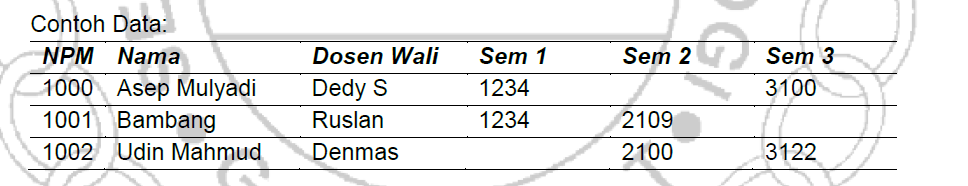
Bentuk Bentuk Normal Kesatu mempunyai ciri yaitu setiap data dibentuk dalam file flat, data dibentuk dalam satu record demi satu record dan nilai dari field berupa *"atomic value"*. Tidak ada set atribut yang berulang ulang atau atribut bernilai ganda (*multiple value*). Tiap field hanya satu pengertian, bukan merupakan kumpulan data yang mempunyai arti mendua. Hanya satu arti saja dan juga bukanlah pecahan kata kata sehingga artinya lain.

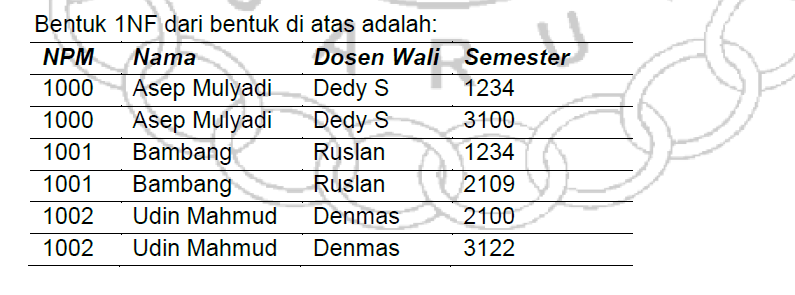
Contoh : Kelas (Kode Kelas, Nama Kelas, Pengajar)

Ini merupakan bentuk 1NF karena tidak ada yang berganda dan tiap atribut satu pengetian yang tunggal

Mahasiswa (NPM, Nama, Dosen Wali, Semester l, Semester2 Semester3)

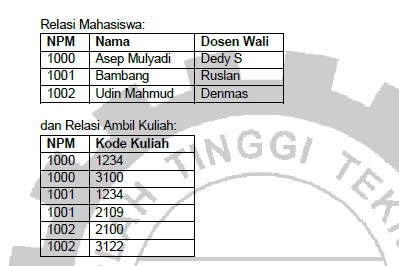
Mahasiswa yang punya NPM, Nama, Dosen Wali mengikuti 3 mata kuliah. Di sini ada perulangan semester sebanyak 3 kali. Bentuk seperti ini bukanlah 1NF.





* Bentuk Normal Kedua (2NF)

Bentuk Normal kedua mempunyai syarat yaitu bentuk data telah memenuhi kriteria bentuk Normal Kesatu. Atribut bukan kunci haruslah bergantung secara fungsi pada kunci utama, sehingga untuk membentuk Normal Kedua haruslah sudah ditentukan kunci-kunci field. Dari contoh relasi mahasiswa pada bentuk Normal Kesatu, terlihat bahwa kunci utama adalah NPM. Nama Mahasiswa dan Dosen Wali bergantung pada NPM, Tetapi Kode Semester bukanlah fungsi dari Mahasiswa maka file siswa dipecah menjadi 2 relasi yaitu :

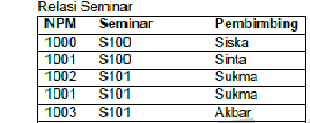


* Bentuk Normal Ketiga (3NF)

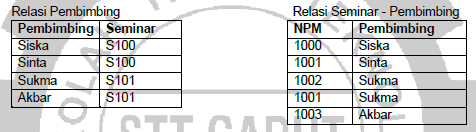
Untuk menjadi bentuk Normal Ketiga maka relasi haruslah dalam bentuk Normal Kedua dan semua atribut bukan primer tidak punya hubungan yang transitif.

* *Boyce-Codd Normal Form* (BNCF)

*Boyce-Codd Normal Form* mempunyai paksaan yang lebih kuat dari bentuk Normal ketiga. Untuk menjadi BNCF, relasi harus dalam bentuk Normal Kesatu dan setiap atribut dipaksa bergantung pada fungsi pada atribut super key.



Bentuk Relasi Seminar adalah bentuk Normal Ketiga, tetapi tidak BCNF karena Kode Seminar masih bergantung fungsi pada Pembimbing, jika setiap Pembimbing dapat mengajar hanya satu seminar. Seminar bergantung pada satu atribut bukan super key seperti yang disayaratakan oleh BCNF. Maka relasi Seminar harus dipecah menjadi dua yaitu :



1. Penerapan Bentuk Normalisasi

Proses perancangan basis data dapat dimulai dari dokumen dasar yang dipakai dalarn sistem sesungguhnya. Kadang-kadang basis data dibentuk dari sistem nyata yang mempunyai bentuk masih belum menggambarkan entitas-entitas secara baik. Sebagai contoh basis data yang dibangun dari daftar faktur pembelian sebagai berikut:

* Langkah Pertama

Bentuklah menjadi tabel Un-Normalized, dengan mencantumkan semua field data yang ada.

* Langkah Kedua

Ubahlah menjadi bentuk Normal Kesatu dengan memisahkan data padaJieId-fieId yang tepat dan bernilai atomik, juga seluruh record harus lengkap datanya.

* Langkah Ketiga

Pembentukan Normal Kedua dengan mencari field kunei yang dapat dipakai sebagai patokan dalam pencarian dan yang mempunyai sifat yang unik.

* Langkah Keempat

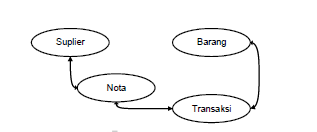
Bentuk normal ketiga mempunyai syarat setiap tabel tidak mempunyai field yang bergantung transitif, harus bergantung penuh pada kunci utama.

* Langkah Kelima

Pengujian di sini untuk memastikan kebenaran isi tabel dan hubungan antara tabel tersebut. Ujian bahwa setiap tabel haruslah punya hubungan dengan tabel yang lainnya

* Langkah Keenam - Relasi Antar tabel

Gambarkan hubungan relasi antar file yang ada sebagai berikut :

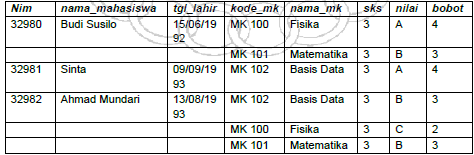


* Langkah Ketujuh

Permasalahan di atas hanya terbatas pada satu dokumen Faktur pembelian barang, padahal pada kenyataannya tentu faktur tersebut mempunyai dokumen pelengkap misalnya nota penjualan barang, laporan stok barang, laporan penjualan, laporan pembelian dan masih banyak lagi laporan dan dokumen data enty lainya.

* 1. **Praktikum**

Diberikan tabel Mahasiswa di bawah ini .

****

Dari tabel diatas, buatlah normalisasi sampai bentuk normal ke tiga !

* Tabel normal pertama ( 1 NF )

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Nim** | **Nama\_mahasiswa** | **Tgl\_lahir** | **Kode\_mk** | **Nama\_mk** | **sks** | **nilai** | **bobot** |
| **32980** | **Budi susilo** | **15/06/1992** | **Mk 100** | **fisika** | **3** | **A** | **4** |
| **32980** | **Budi susilo** | **09/09/1992** | **Mk 101** | **matematika** | **3** | **B** | **3** |
| **32981** | **Sinta** | **09/09/1993** | **Mk 102** | **Basis data** | **3** | **A** | **4** |
| **32982** | **Ahmad mundari** | **13/08/1993** | **Mk 102** | **Basis Data** | **3** | **B** | **3** |
| **32982** | **Ahmad mundari** | **13/08/1993** | **Mk 100** | **fisika** | **3** | **C** | **2** |
| **32982** | **Ahmad mundari** | **13/08/1993** | **Mk 101** | **matematika** | **3** | **B** | **3** |

* Tabel normalisasi ke dua ( 2 NF )
* Tabel Mahasiswa

|  |  |  |
| --- | --- | --- |
| **Nim** | **Nama\_mhs** | **Tgl\_lahir** |
| **32980** | **Budi susilo** | **15/06/1993** |
| **32981** | **Sinta** | **13/09/1993** |
| **32982** | **Ahmad mundari** | **13/08/1993** |

* Tabel kuliah

|  |  |  |
| --- | --- | --- |
| **Kode\_mk** | **Nama\_mk** | **Sks** |
| **Mk 100** | **Fisika** | **3** |
| **Mk 101** | **Matematika** | **3** |
| **Mk 102** | **Basis data** | **3** |

* Tabel Nilai

|  |  |  |  |
| --- | --- | --- | --- |
| **Nim** | **Kd\_mk** | **nilai** | **bobot** |
| **32980** | **Mk 100** | **A** | **4** |
| **32980** | **Mk 101** | **B** | **3** |
| **32981** | **Mk 102** | **A** | **4** |
| **32982** | **Mk 102** | **B** | **3** |
| **32982** | **Mk 100** | **C** | **2** |
| **32982** | **Mk 101** | **B** | **3** |

* Tabel Normalisasi ke tig ( 3 NF )
* Tabel Mahasiswa

|  |  |  |
| --- | --- | --- |
| **Nim** | **Nama\_mhs** | **Tgl\_lahir** |
| **32980** | **Budi susilo** | **15/06/1993** |
| **32981** | **Sinta** | **13/09/1993** |
| **32982** | **Ahmad mundari** | **13/08/1993** |

* Tabel kuliah

|  |  |  |
| --- | --- | --- |
| **Kode\_mk** | **Nama\_mk** | **Sks** |
| **Mk 100** | **Fisika** | **3** |
| **Mk 101** | **Matematika** | **3** |
| **Mk 102** | **Basis data** | **3** |

* Tabel Nilai

|  |  |  |  |
| --- | --- | --- | --- |
| **Nim** | **Kd\_mk** | **nilai** | **bobot** |
| **32980** | **Mk 100** | **A** | **4** |
| **32980** | **Mk 101** | **B** | **3** |
| **32981** | **Mk 102** | **A** | **4** |
| **32982** | **Mk 102** | **B** | **3** |
| **32982** | **Mk 100** | **C** | **2** |
| **32982** | **Mk 101** | **B** | **3** |

* Tabel transaksi nilai bobot

|  |  |
| --- | --- |
| **Nilai** | **Bobot** |
| **A** | **4** |
| **B** | **3** |
| **C** | **2** |
| **D** | **1** |
| **E** | **21** |

**MODUL IV**

**IMPLEMENTASI BASIS DATA**

**( TRANSFORMASI MODUL BATA )**

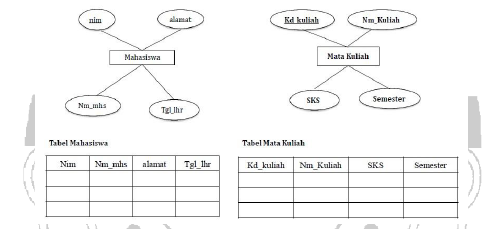
* 1. **Tujuan**

Tujuan dari praktikum basis data “ implementasi basis data “ adalah agar :

* Untuk mengetahui perubahan dari suatu bentuk ke bentuk yang lain
* Untuk mengetahui transformasi ERD
  1. **Dasar Teori**

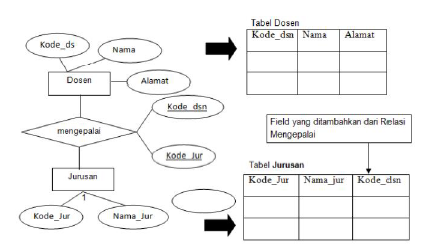
Transformasi merupakan perubahan dari suatu bentuk ke bentuk yang lain. Transformasi ERD ini merepresentasikan ERD kedalam tabel dalam basis data, dan atribut yang terdapat pada masing-masing entitas akan dinyatakan sebagai field atau kolom dari tabel yang sesuai.

1. Setiap entitas akan diimplementasikan dalam bentuk sebuah tabel. Contohnya sebagai berikut:



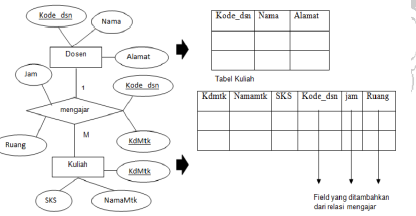
1. Transformasi 1 – 1

Relasi dengan derajat relasi satu-ke-satu, yang menghubungkan 2 buah himpunan entitas akan direpresentasikan dalam bentuk penambahan / penyertaan atribut-atribut relasi ke tabel yang mewakili salah satu dari kedua himpunan entitas. Contoh: Dosen sebagai Ketua Jurusan.



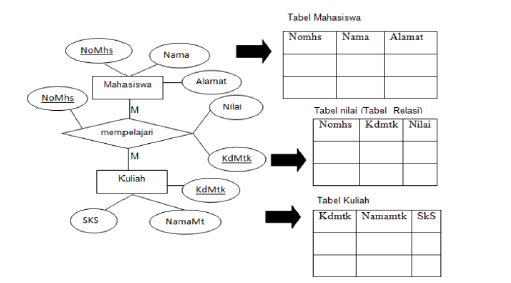
1. Transformasi Relasi 1 – M

Relasi dengan derajat satu ke banyak, yang menghubungkan 2 buah himpunan Entitas, akan direpresentasikan dalam bentuk pemberian atribut kunci dari himpunan entitas pertama (yang berderajat 1) ke tabel yang mewakili himpunan entitas kedua (yang berderajat M)



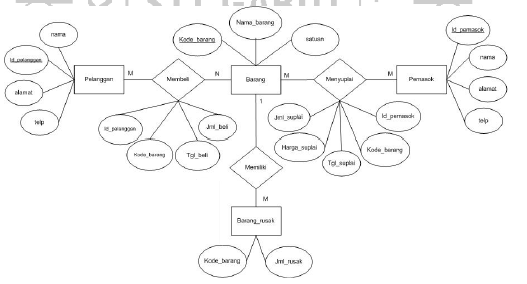
1. Transformasi M – M

Relasi dengan derajat banyak ke banyak, yang menghubungkan 2 buah himpunan entitas akan diwujudkan dalam bentuk tabel khusus, yang memiliki field (atau foreign key) yang berasal dari kunci-kunci dari himpunan entitas yang dihubungkannya.



* 1. **Praktikum**

Lakukan transformasi model data pada ERD berikut ini!



* Transformasi model data
* Tabel pelanggan

|  |  |  |  |
| --- | --- | --- | --- |
| Id\_pelanggan | nama | alamat | Tlpn |
|  |  |  |  |

* Tabel barang

|  |  |  |
| --- | --- | --- |
| Kode\_barang | Nama\_barang | Satuan |
|  |  |  |

* Tabel pemasok

|  |  |  |  |
| --- | --- | --- | --- |
| Id\_pemasok | nama | alamat | tlp |
|  |  |  |  |

* Tabel barang rusak

|  |  |
| --- | --- |
| Kd\_barang | Jml \_rusak |
|  |  |

* Tabel transaksi
* Tabel pembelian

|  |  |  |  |
| --- | --- | --- | --- |
| Id\_pelanggan | Kode\_barang | Tgl\_beli | Jml\_beli |
|  |  |  |  |

* Tabel penyuplai

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id\_pemasok | Kd\_barang | Haraga\_suplai | Tgl\_suplai | Jml\_suplai |
|  |  |  |  |  |

**MODUL V**

**DDL ( DATA DEFINITION LANGUAGE )**

* 1. **Tujuan**

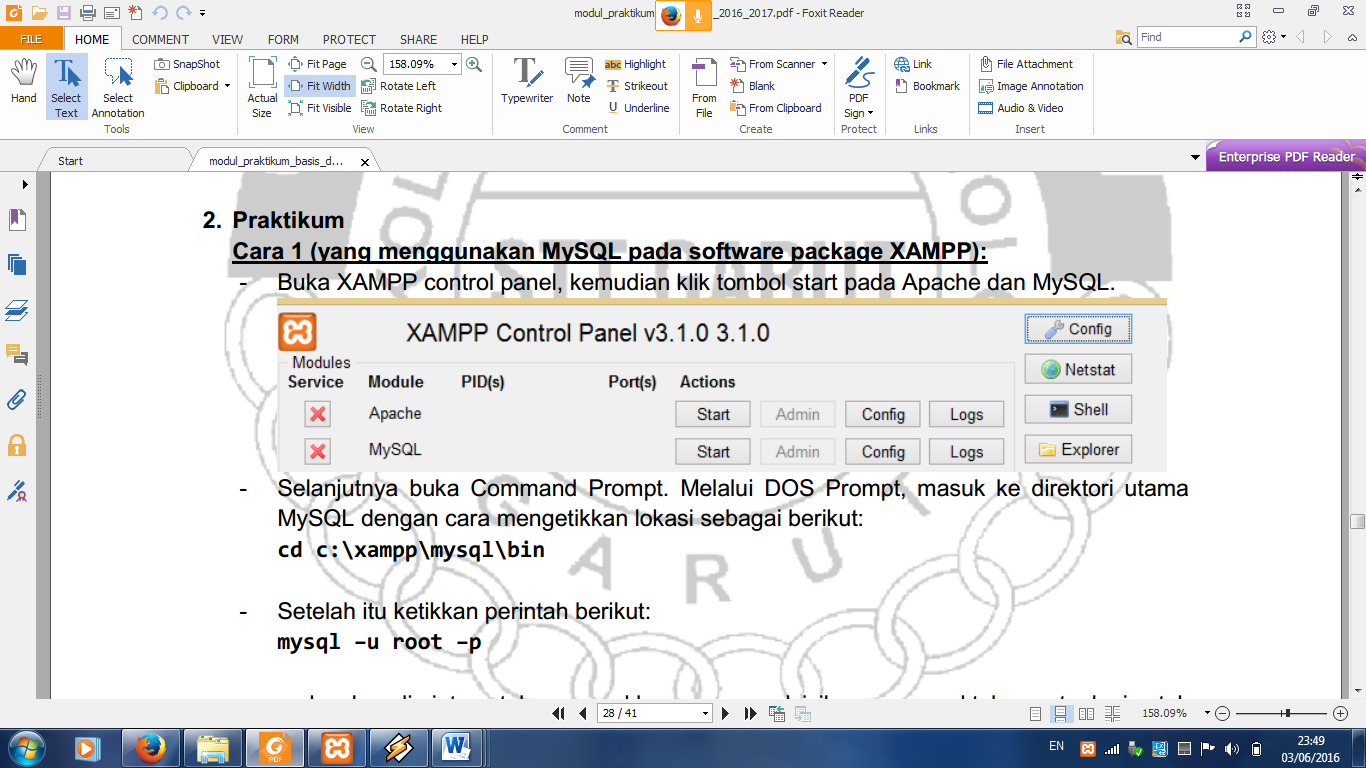
Tujuan dari praktikum basis data “ DDL ( Data Definition Language ) adalah agar :

* Untuk mengetahui pendefinisian database dan pendefinisian table
* Untuk mengetahui pembuatan database dan pembuatan table
  1. **Dasar Teori**

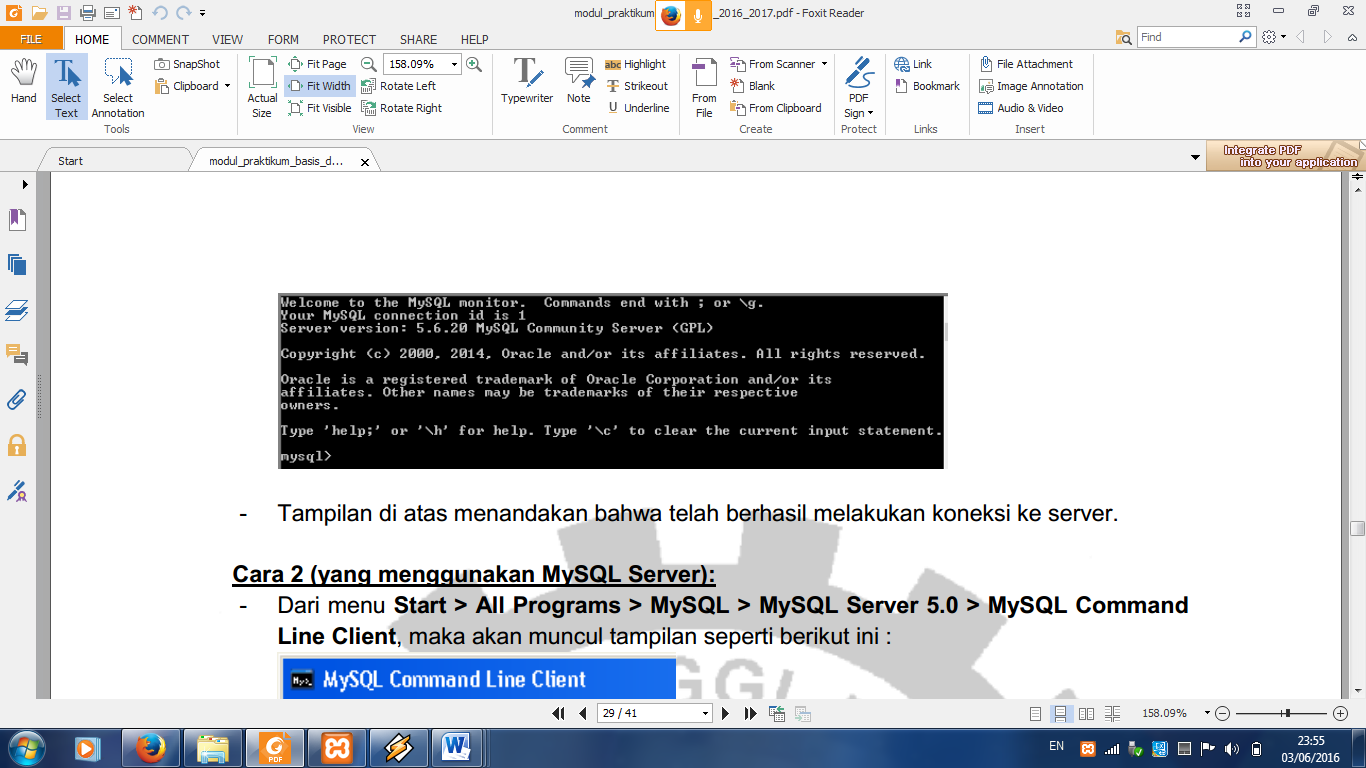
1. DDL ( Data Definition Language )

Kelompok perintah ini bisa digunakan untuk melakukan pendefinisian database dan pendefinisian tabel. Sehingga dengan menggunakan perintah-perintah ini, kita bias memerintahkan untuk membuat database, membuat tabel, mengubah strukturnya, menghapus tabel, membuat index tabel dan lain-lain yang berhubungan dengan pendefinisian database dan tabel. Data Definision Language(DDL) merupakan sebuah bahasa SQL pendifinisi data, perintah-perintah yang digunakan antara lain:

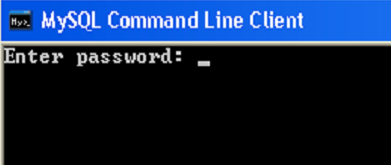
* CREATE DATABASE : Digunakan untuk membuat database beru
* USE DATABASE : Digunakan untuk masuk dalam database
* CREATE TABLE : Digunakan untuk membuat tabel
* ALTER TABLE : Digunakan untuk mengubah struktur tabel
* RENAME TABLE : Digunakan untuk merubah nama table
* DROP DATBASE : Digunakan untuk menghapus database
* DROP TABLE : Digunakan untuk menghapus tabel
  1. **Praktikum**
* Cara 1 (yang menggunakan MySQL pada software package XAMPP):
* Buka XAMPP control panel, kemudian klik tombol start pada Apache dan MySQL



* Selanjutnya buka Command Prompt. Melalui DOS Prompt, masuk ke direktori utama MySQL dengan cara mengetikkan lokasi sebagai berikut: **cd c: \xampp\mysql\bin**
* Setelah itu ketikkan perintah berikut: **mysql –u root –p**
* maka akan diminta untuk memasukkan passoword, isikan password,tekan enter lagi untuk masuk.
* Selanjutnya akan ada respon dari server seperti gambar berikut :



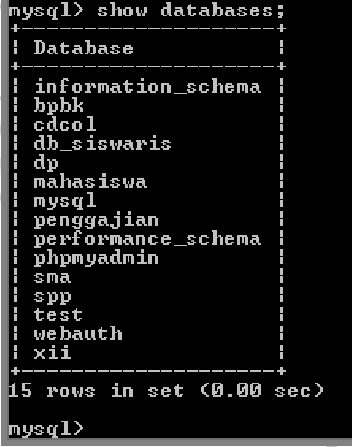
* Tampilan di atas menandakan bahwa telah berhasil melakukan koneksi ke server.
* Menggunakan MySQL Server:
* Cara 2 (yang menggunakan MySQL Server):
* Dari menu **Start > All Programs > MySQL > MySQL Server 5.0 > MySQL Command** **Line Client**, maka akan muncul tampilan seperti berikut ini :



* Masukkan password yang telah ditentukan pada saat instalasi, kemudian tekan enter.

**Membuat Data Basse baru:**

Untuk mengetahui atau melihat database yang sudah ada, bisa digunakan perintah **show** **database;**

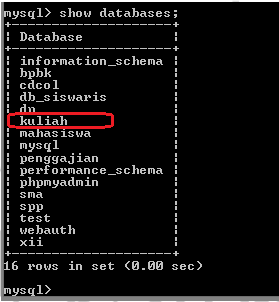


Dalam pembuatan database perlu perhatikan penulisan nama database tidak boleh menggunakan spasi dan karakter non standar. Bentuk penulisan perintah untuk membuat database baru adalah dengan

menggunakan perintah **create databasenama\_database**;



Untuk memastikan bahwa database yang kita buat sudah jadi, perintahkan **show** **databases**, sehingga semua database yang ada di server dan bisa diakses oleh user akan ditampilkan.



**Membuat Table Baru**

Memulai membuat tabel di dalam database Mahasiswa, maka kita harus mengaktifkan terlebih dulu database mahasiswa dengan menggunakan perintah **use nama\_database.**



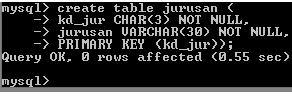
Setelah database aktif, kita baru bisa memulai untuk membuat tabel yang kita perlukan. Pada database kuliah, akan disimpan data-data tentang mahasiswa, jurusan, dan mata kuliah.Pertama kali, kita akan membuat tabel jurusan yang fungsinya untuk menyimpan data tentang jurusan dengan perintah.

**Create nama\_table (**

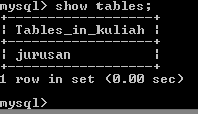
**Kolom\_1 type(panjang),**

**Kolom\_2 type(panjang),**

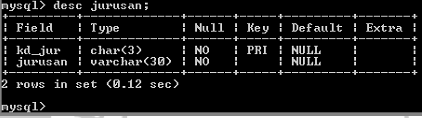
**......** **Kolom\_n type(panjang) );**



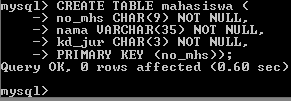
Untuk membuktikan bahwa kita berhasil, maka tabel yang ada database bisa kita tampilkan dengan menggunakan perintah **show tables**



Pada hasil perintah diatas ditampilkan, bahwa tabel yang sudah ada di database kuliah adalah jurusan. Untuk menampilkan struktur dari tabel jurusan dengan menggunakan perintahkan**desc nama\_tabel**



selanjutnya mmembuat tabel mahasiswa.



Selanjutnya silahkan buat tabel propinsi yang memiliki 2 buah *filed* (kolom) yaitu kd\_propinsi dan propinsi.

**Merubah dan Memodifikasi Table**

Terkadang ketika kita, sedang membuat tabel, ternyata ada nama field yang kurang, atau nama\_fieldnya susah dipahami, atau bisa juga type data serta ukurannya tidak/ kurang sesuai. Sekarang coba kita perhatikan tabel yang sudah kita buat sebelumnya. Pada table mahasiswa terdapat field nama dengan type varchar ukurannya 35, akan dirubah ukurannya menjadi 30, caranya dengan menggunakan perintah **alter table** **nama\_table modify nama\_kolom type(panjang)**;



**Merubah NamaTabel**

Untuk merubah nama tabel gunakan perintah **alter table nama\_table rename** **nama\_table\_baru;**



**Menghapus Tabel dan Database**

Terkadang kita sudah membuat suatu tabel atau database, tapi ternyata tabel tersebut sebetulnya tidak diperlukan. Mungkin karena pembatasan masalah, ataupun mungkin karena diketahui setelahnya bahwa tabel tersebut tidaklah diperlukan, dan hanya memboroskan tempat penyimpanan saja, atau dengan tabel tersebut menjadikan tidak efektif dan efisien. Atau bisa saja tabel tersebut diciptakan hanya untuk keperluan sementara saja(temporari), maka kita harus atau perlu untuk menghapus tabel tersebut. Begitu juga dengan database yang sudah kita buat ,ternyata database tersebut sudah kita  
perlukan lagi. Perintah yang digunakan untuk menghapus adalah **DROP**.

* Menghapus Tabel **drop table nama\_table** contoh: **drop table mahasiswa;**
* Menghapus Database **drop database nama\_database** contoh: **drop database kuliah;**

**MODUL VI**

**DML ( DATA MANIPULATION LANGUAGE )**

* 1. **Tujuan**

Tujuan peraktikum basis data “ DML “ adalah agar :

* Untuk mengetahui cara memanipulasi data
  1. **Dasat Teori**
* **DML (Data ManipulationLanguage)**

DML adalah kelompok perintah yang berfungsi untuk memanipulasi data dalam basis data, misalnya untuk pengambilan, penyisipan, pengubahan, dan penghapusan data. Perintah yang masuk kategori DML adalah :

 SELECT : Perintah ini digunakan untuk menampilkan data

 INSERT : Perintah yang digunakan untuk memasukan data

 UPDATE : Perintah yang digunakan untuk memperbarui isi data

 DELETE : Perintah yang digunakan untuk melakukan penghapusan data

 LOAD DATA : Perintah yang dgunakan untuk mengambil data dari file berformat “.SQL”

* 1. **Praktikum**
* **Memasukkan Data ke dalam Tabel**

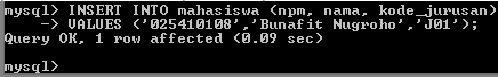
Memasukan data merupakan langkah yang dilakukan untuk dapat mendokumentasikan data-data mahasiswa, istilah ini juga sering disebut dengan ungkapan Input data, Merekam data, atau Insert. Perintah Insert digunakan untuk menyisipkan data baru ke dalam sebuah tabel database. Data yang akan dimasukan dalam tabel mahasiswa seperti pada tabel berikut :



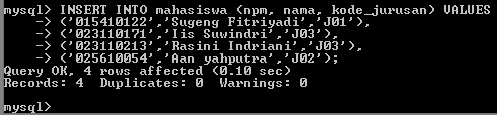
Ada dua macam perintah Insert, yaitu yang digunakan untuk menyisipkan data satu persatu dan yang menyisipkan banyak data sekaligus. Untuk sintaknya adalah sebagai berikut :

**insert into nama\_tabel (kolom1, kolom2, …, kolom\_n) values ('nilai1' , 'nilai2', …, 'nilai\_n')**

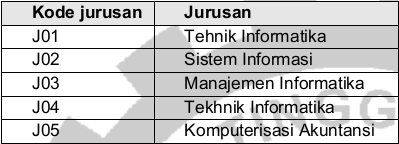
Untuk menyisipkan data mahasiswa tersebut dapat dilakukan dengan melakukan penambahan data satu-persatu seperti berikut :



Atau dengan menambahkannya secara sekaligus



Sekarang masukanlah data-data berikut pada tabel jurusan, yang merupakan bentuk relasi yang dipanggil pada tabel mahasiswa.

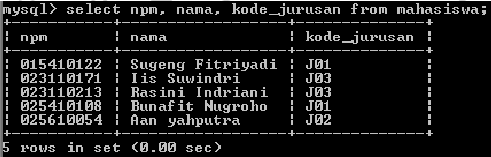


Selanjutnya masukanlah data-data berikut pada tabel propinsi, yang merupakan bentuk relasi yang dipanggil pada tabel mahasiswa.

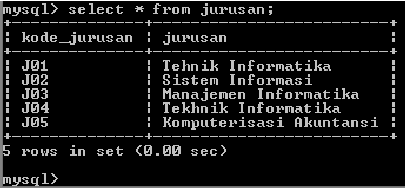


* **Menampilkan Data**

Perintah SELECT pada MySQL digunkana untuk menampilkan atau memunculkan isi data pada satu tabel atau lebih. Sekarang cobalah untuk melihat isi data dari ketiga tabel di atas, perintah yang digunakan untuk melihat isi data pada tabel tersebut adalah dengan menggunakan perintah **select kolom1, kolom2, ..., kolom\_n from nama\_tabel;**



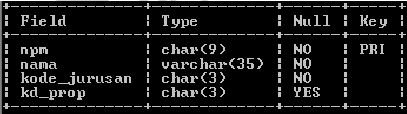
Atau dengan menggunakan perintah **select \* from nama\_tabel;**



* **Mengubah isi data yang sudah ada pada Tabel**

Perintah Update digunakan untuk memodifikasi atau memperbaharui nilai-nilai record pada sebuah tabel. Sintaksnya adalah sebagai berikut : **UPDATE nama\_tabel SET field1 = nilai1, field2 = nilai2 , ..., filed\_n = nilai\_n WHERE kondisi;**

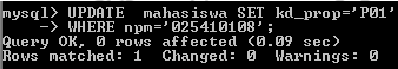
Sebelum mengubah isi tabel, modifikasilah tabel mahasiswa dengan menambahkan kolom kd\_prop, sehingga struktur tabel mahasiswa menjadi seperti berikut.



Untuk melakukan perubahan stuktur tabel pada tabel mahasiswa, maka anda dapat menggunakan perintah **alter table mahasiswa add kd\_prop CHAR(3);**



Selanjutnya, masukan data propinsi pada tabel mahasiswa dengan cara mengupdate (merubah) data pada tabel mahasiwa dikarenakan kolom kd\_prop masih kosong karena sebelumnya pada proses INSERT tidak beserta dengan data pada kolom kd\_prop, untuk melakukan UPDATE perintah yang digunakan adalah:











Hati-hati juga dalam menjalankan perintah update ini, jangan sampai lupa untuk menuliskan kondisi data dalam tabel yang akan diperbaiki, karena jika lupa data akan berubah semuanya.

* **Menghapus isi data pada Tabel**

Perintah Delete digunakan untuk menghapus record pada suatu tabel. Sintaks dari perintah Delete adalah sebagai berikut :

**delete from nama\_tabel where kriteria**

Hati-hati juga dalam memberikan perintah delete ini, karena jika lupa memberikan pernyataan kondisi, perintah ini akan menyebabkan terhapusnya seluruh isi tabel. Sebagai contoh kita hapus dari tabel mahasiswa, mahasiwa dengan npm=’015410122’, maka perintahnya:



Selanjutnya coba lihat dengan menggunakan perintah select, apakah data mahasiswa dengan npm 015410122 sudah terhapus.

**MODUL VII**

**FUNGSI SQL LANJUTAN**

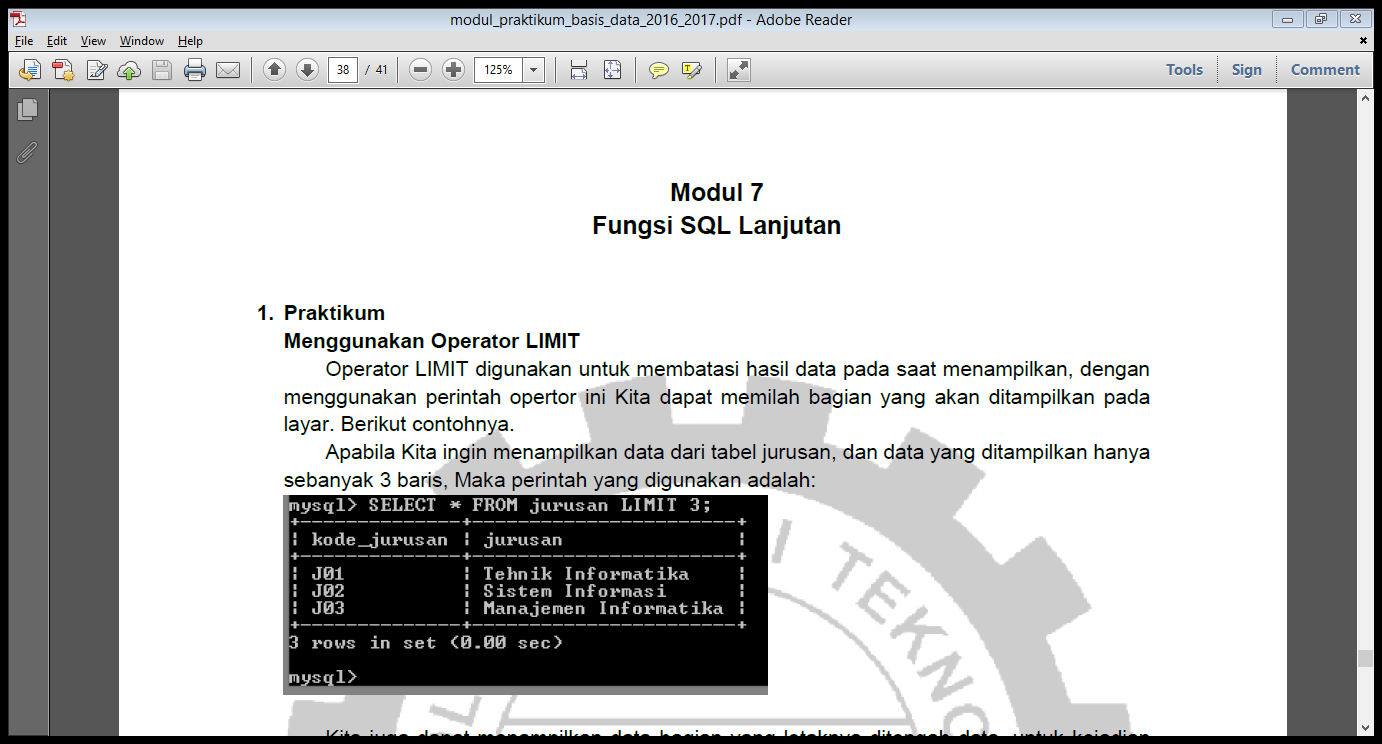
* 1. **Tujuan**

Tujuan praktikum basis data “ fungsi SQL “ adalah agar :

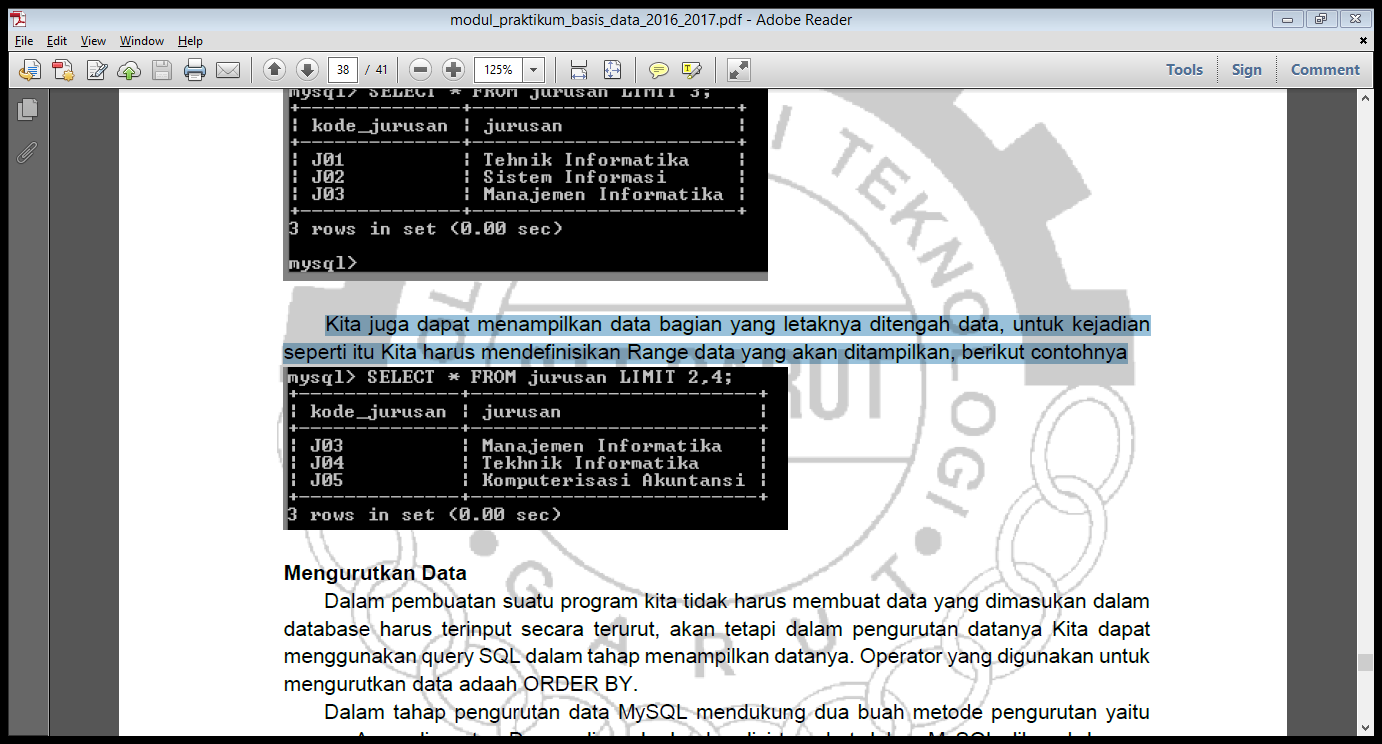
* Untuk semakin mengetahui perancangan basis data dan perancangan tabel
* Untuk semakin mengetahui pendefinisian databes dan pendefinisian table
  1. **praktikum**
* Menggunakan Operator LIMIT

Operator LIMIT digunakan untuk membatasi hasil data pada saat menampilkan, dengan menggunakan perintah opertor ini Kita dapat memilah bagian yang akan ditampilkan pada layar. Berikut contohnya.

Apabila Kita ingin menampilkan data dari tabel jurusan, dan data yang ditampilkan hanya sebanyak 3 baris, Maka perintah yang digunakan adalah:

****

Kita juga dapat menampilkan data bagian yang letaknya ditengah data, untuk kejadian seperti itu Kita harus mendefinisikan Range data yang akan ditampilkan, berikut contohnya.

****

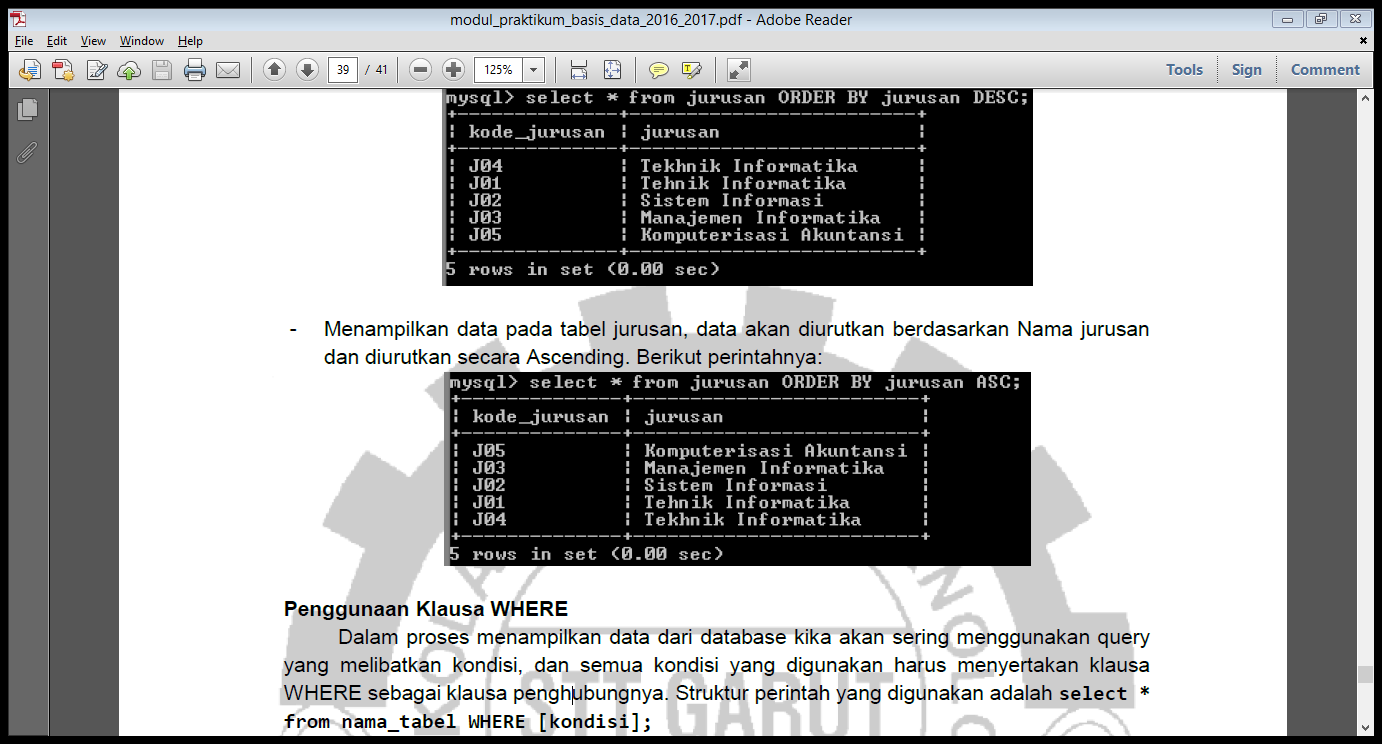
* Mengurutkan Data

Dalam pembuatan suatu program kita tidak harus membuat data yang dimasukan dalam database harus terinput secara terurut, akan tetapi dalam pengurutan datanya Kita dapat menggunakan query SQL dalam tahap menampilkan datanya. Operator yang digunakan untuk mengurutkan data adaah ORDER BY. Dalam tahap pengurutan data MySQL mendukung dua buah metode pengurutan yaitu secara Ascending atau Descending, kedua kondisi tersebut dalam MySQL dikenal dengan operotor ASC (Ascending) dan DESC (Descending). Brikut adalah contoh penggunaanya: Menampilkan data pada tabel Jurusan, data diurutkan berdasarkan pada kode Jurusan yang diurutkan secara Descending, Perintah yang digunakan adalah :



Menampilkan data pada tabel jurusan, data akan diurutkan berdasarkan

Nama jurusan dan diurutkan secara Ascending. Berikut perintahnya:



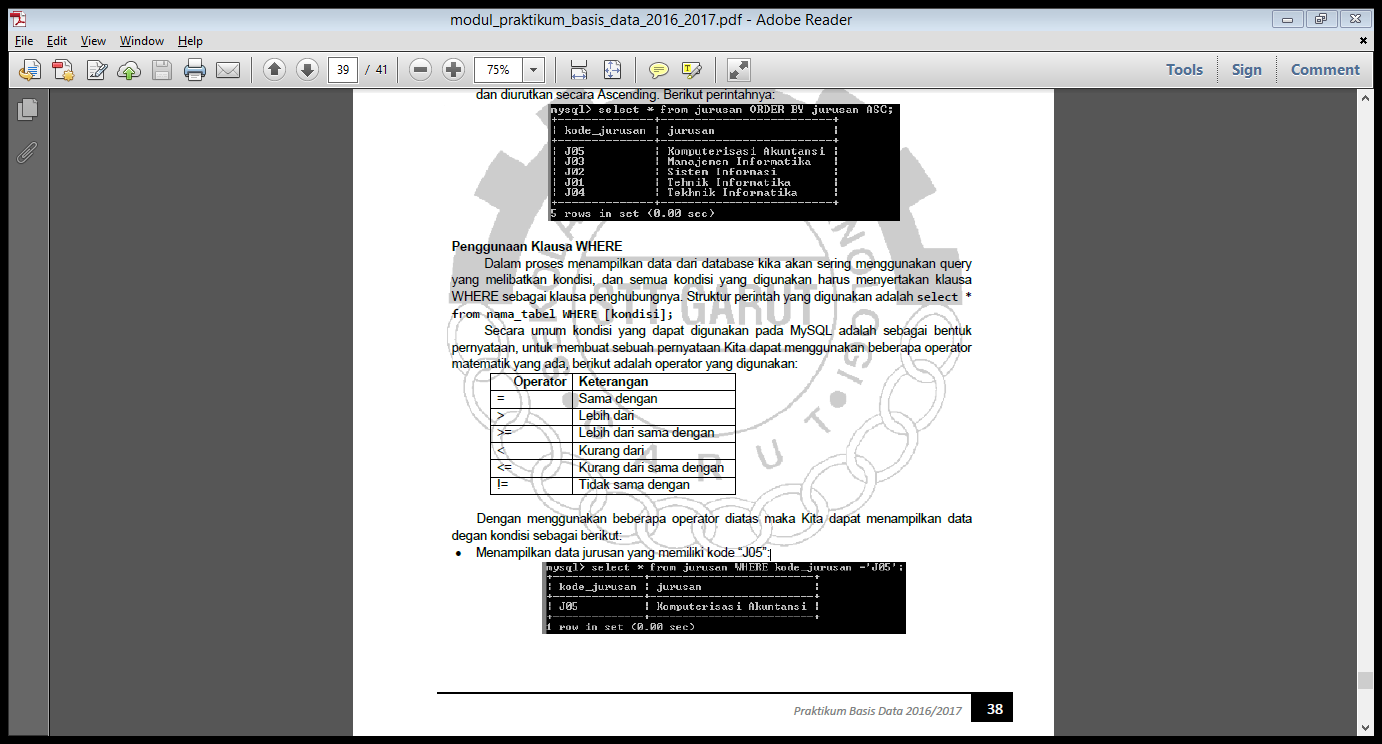
* Penggunaan Klausa WHERE

Dalam proses menampilkan data dari database kika akan sering menggunakan query yang melibatkan kondisi, dan semua kondisi yang digunakan harus menyertakan klausa WHERE sebagai klausa penghubungnya. Struktur perintah yang digunakan adalah select \* from nama\_tabel WHERE [kondisi]; Secara umum kondisi yang dapat digunakan pada MySQL adalah sebagai bentuk pernyataan, untuk membuat sebuah pernyataan Kita dapat menggunakan beberapa operator matematik yang ada, berikut adalah operator yang digunakan:

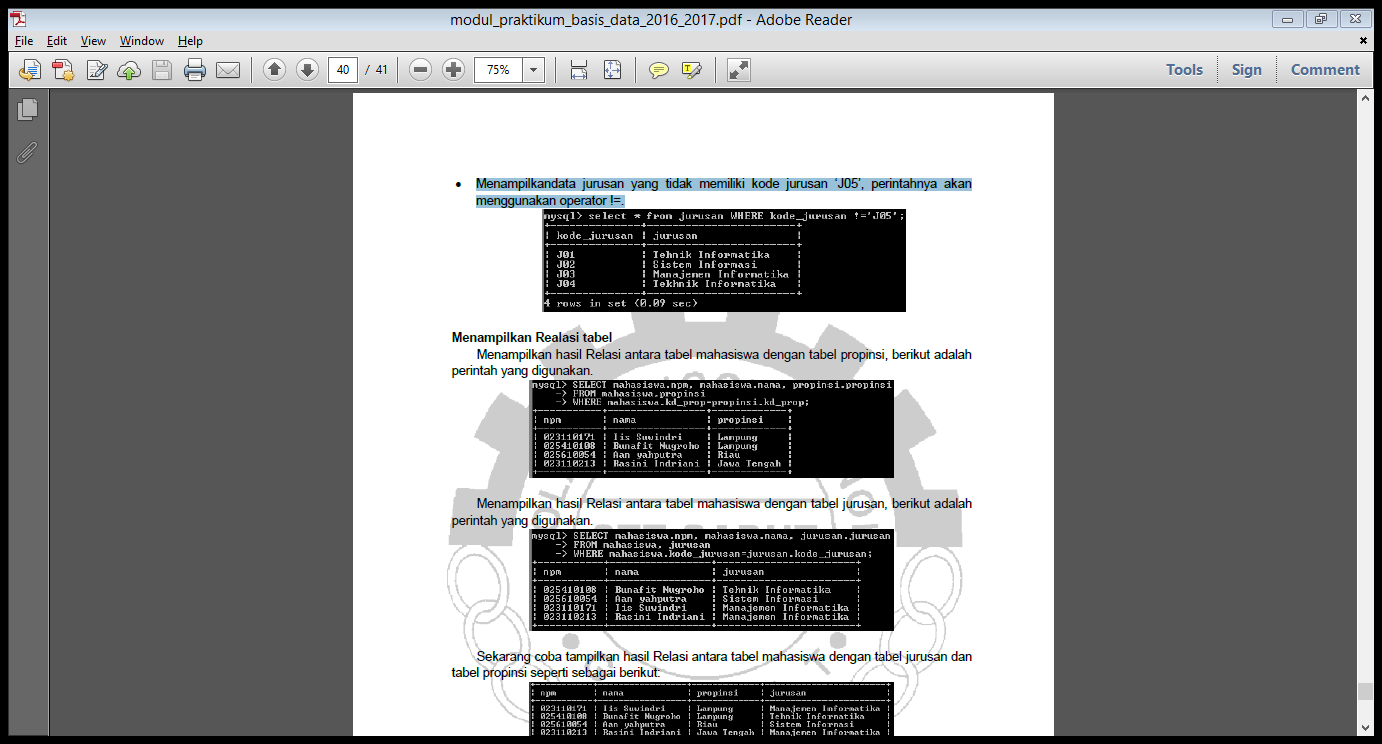
|  |  |
| --- | --- |
| **Operator** | **Keterangan** |
| = | Sama dengan |
| > | Lebih dari |
| >= | Lebih dari sama dengan |
| < | Kurang dari |
| <= | Kurang dari sama dengan |
| != | Tidak sama dengan |

Dengan menggunakan beberapa operator diatas maka Kita dapat menampilkan data degan kondisi sebagai berikut:

* Menampilkan data jurusan yang memiliki kode “J05”:

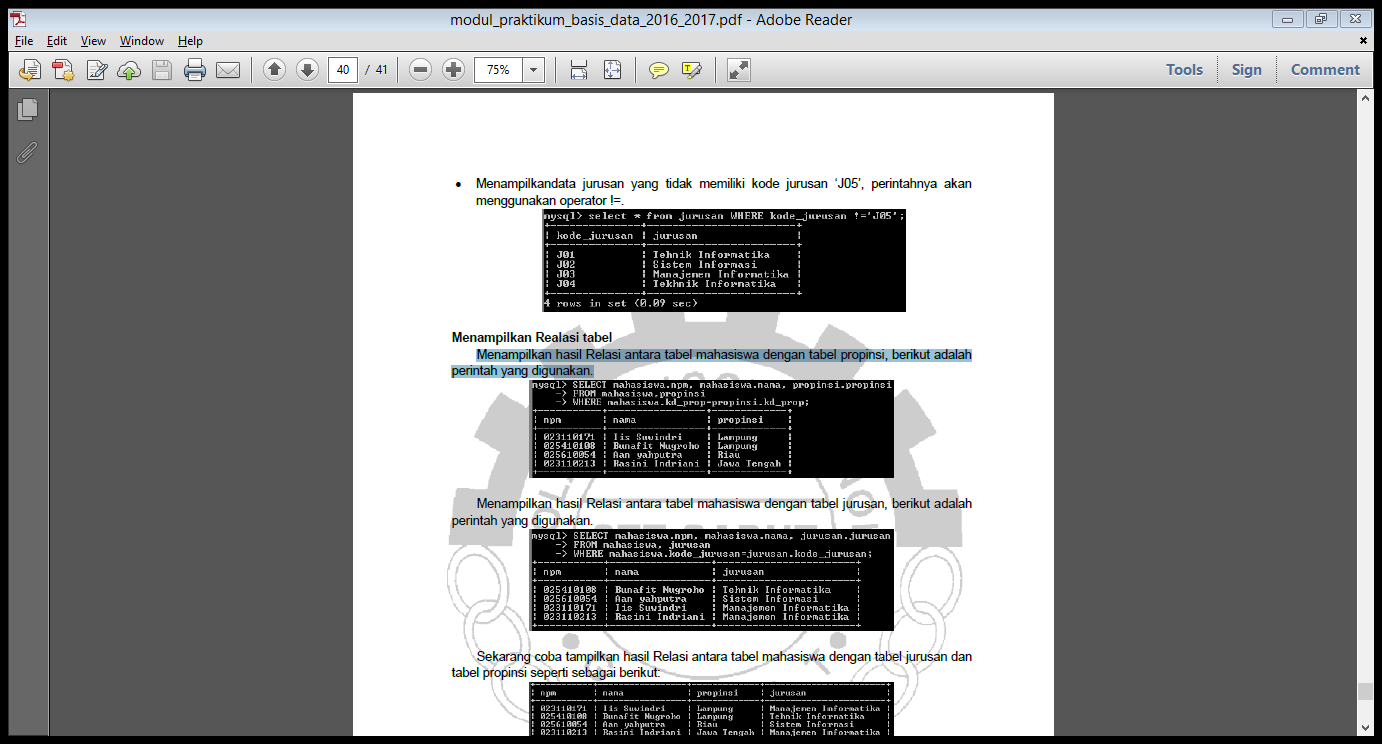


Menampilkandata jurusan yang tidak memiliki kode jurusan ‘J05’, perintahnya akan menggunakan operator !=.

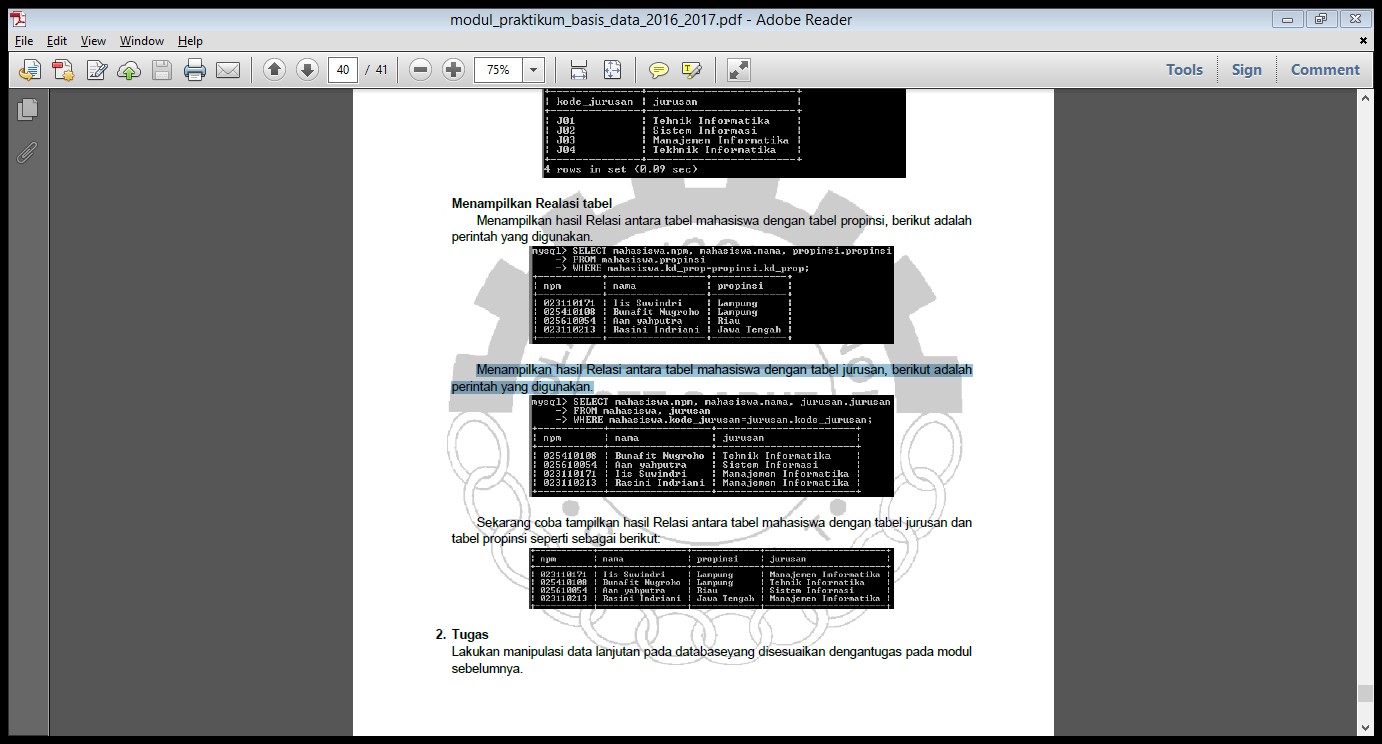


* Menampilkan Realasi table

Menampilkan hasil Relasi antara tabel mahasiswa dengan tabel propinsi, berikut adalah perintah yang digunakan.



Menampilkan hasil Relasi antara tabel mahasiswa dengan tabel jurusan, berikut adalah perintah yang digunakan.



Sekarang coba tampilkan hasil Relasi antara tabel mahasiswa dengan tabel jurusan dan tabel propinsi seperti sebagai berikut:



**BAB VIII**

**PROYEK AKHIR MATA KULIAH BASIS DATA**

**MENGGUNAKAN MODEL ER**

* 1. **Penentuan Entitas**
* Pelanggan
* Fhoto
* Petugas
  1. **Matrik Relasi**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Pelanggan | Photo | petugas |
| Pelanggan |  | mencetak |  |
| Photo | Kirim |  | Mencetak |
| petugas |  | Dicetak |  |

* 1. **Erd Percetakan Foto**

Pelanggan

Photo

Order

Petugas

Dicetakk

* 1. **Kunci Utama**
* Pelangga = Id\_Pelanggan
* Petugas = id\_Petugas
* Photo = Kode\_photo
  1. **Atribut**
* Pelanggan : Id\_Pelanggan,Nama,Alamat
* Photo :Id\_Photo,Jenis Photo,Size
* Petugas:Id\_Petugas,Nama
  1. **Transformasi Erd kedalam Table Basis Data**

Pelanggan

Photo

Order

Petugas

Dicetakk

* **Tabel Basis Data**
* Transformasi

Tabel Pelanggan

|  |  |  |
| --- | --- | --- |
| Id\_Pelanggan | Nama | Alamat |
|  |  |  |

Tabel Fhoto

|  |  |  |  |
| --- | --- | --- | --- |
| Id\_Foto | Jenis Foto | Size | Harga |
|  |  |  |  |

Tabel Petugas

|  |  |
| --- | --- |
| Id\_Petugas | Nama |
|  |  |

* Tabel Transaksi

Tabel Order

|  |  |  |  |
| --- | --- | --- | --- |
| Id\_pelanggan | Id\_fhoto | Jumlah | Total |
|  |  |  |  |

Tabel pencetakan

|  |  |  |
| --- | --- | --- |
| Id\_Fhoto | Id\_Petugas | Jumlah\_cetak |
|  |  |  |

* 1. **Normalisasi**

Tabel Nolmalisasi Unormal :

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Id\_pelanggan | Nama | Alamat | Id\_Photo | Size | Tanggal | Jumlah | Harga | Total | Id\_petugas | Petugas |
| 100001 | Abi | Bayongbong | 1001 | 3R | 10/05/2016 | 1 | 5000 | 5000 | 1011 | Gumgum |
|  |  |  | 1002 | 1R |  | 5 | 3000 | 15000 |  |  |
|  |  |  | 1003 | 4r |  | 4 | 7500 | 30000 |  |  |
| 100004 | Aji | Samarang | 1001 | 3R | 11/05/2016 | 1 | 5000 | 5000 | 1012 | Asep |
|  |  |  | 1002 | 1R |  | 5 | 3000 | 15000 |  |  |
| 100003 | Ahmad | Tarogong | 1001 | 3R | 12/05/2016 | 1 | 5000 | 5000 | 1011 | Gumgum |
|  |  |  | 1002 | 1R |  | 5 | 3000 | 15000 | 1012 | Asep |

Table Normal Pertama ( 1 NF )

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Id\_pelanggan | Nama | Alamat | Id\_Photo | Size | Tanggal | Jumlah | Harga | Total | Id\_petugas | Nama |
| 100001 | Abi | Bayongbong | 1001 | 3R | 10/05/2016 | 1 | 5000 | 5000 | 1011 | Gumgum |
| 100001 | Abi | Bayongbong | 1002 | 1R | 10/05/2016 | 5 | 5000 | 15000 | 1011 | Gumgum |
| 100001 | Abi | Bayongbong | 1003 | 4r | 10/05/2016 | 4 | 5000 | 30000 | 1011 | Gumgum |
| 100004 | Aji | Samarang | 1001 | 3R | 11/05/2016 | 1 | 5000 | 5000 | 1012 | Asep |
| 100004 | Aji | Samarang | 1002 | 1R | 11/05/2016 | 5 | 5000 | 15000 | 1012 | Asep |
| 100003 | Ahmad | Tarogong | 1001 | 3R | 12/05/2016 | 1 | 5000 | 5000 | 1011 | Gumgum |
| 100003 | Ahmad | Tarogong | 1002 | 1R | 12/05/2016 | 5 | 5000 | 15000 | 1012 | Asep |

Table normalisasi ke dua ( 2 NF )

Tabel Pelanggan

|  |  |  |
| --- | --- | --- |
| Id\_pelanggan | Nama | Alamat |
| 100001 | Abi | Bayongbong |
| 100004 | Aji | Samarang |
| 100003 | Ahmad | Tarogong |

Tabel Photo

|  |  |  |
| --- | --- | --- |
| Id\_Photo | Size | Harga |
| 1001 | 3R | 5000 |
| 1002 | 1R | 5000 |
| 1003 | 4r | 5000 |

Tabel Petugas

|  |  |
| --- | --- |
| Id\_petugas | Nama |
| 1011 | Gumgum |
| 1012 | Asep |

Tabel Transaksi

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Id\_pelanggan | Id\_Photo | Id\_petugas | Tanggal | Jumlah | Harga | Total |
| 100001 | 1001 | 1011 | 10/05/2016 | 1 | 5000 | 5000 |
| 100001 | 1002 | 1011 | 10/05/2016 | 5 | 5000 | 15000 |
| 100001 | 1003 | 1011 | 10/05/2016 | 4 | 5000 | 30000 |
| 100004 | 1001 | 1012 | 11/05/2016 | 1 | 5000 | 5000 |
| 100004 | 1002 | 1012 | 11/05/2016 | 5 | 5000 | 15000 |
| 100003 | 1001 | 1011 | 12/05/2016 | 1 | 5000 | 5000 |
| 100003 | 1002 | 1012 | 12/05/2016 | 5 | 5000 | 15000 |

Tabel normalisasi ke tiga ( 3 NF )

Tabel Pelanggan

|  |  |  |
| --- | --- | --- |
| Id\_pelanggan | Nama | Alamat |
| 100001 | Abi | Bayongbong |
| 100004 | Aji | Samarang |
| 100003 | Ahmad | Tarogong |

Tabel Photo

|  |  |  |
| --- | --- | --- |
| Id\_Photo | Size | Harga |
| 1001 | 3R | 5000 |
| 1002 | 1R | 5000 |
| 1003 | 4r | 5000 |

Tabel Petugas

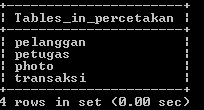
|  |  |
| --- | --- |
| Id\_petugas | Nama |
| 1011 | Gumgum |
| 1012 | Asep |

Tabel Transaksi

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Id\_pelanggan | Id\_Photo | Id\_petugas | Tanggal | Jumlah | Harga | Total |
| 100001 | 1001 | 1011 | 10/05/2016 | 1 | 5000 | 5000 |
| 100001 | 1002 | 1011 | 10/05/2016 | 5 | 5000 | 15000 |
| 100001 | 1003 | 1011 | 10/05/2016 | 4 | 5000 | 30000 |
| 100004 | 1001 | 1012 | 11/05/2016 | 1 | 5000 | 5000 |
| 100004 | 1002 | 1012 | 11/05/2016 | 5 | 5000 | 15000 |
| 100003 | 1001 | 1011 | 12/05/2016 | 1 | 5000 | 5000 |
| 100003 | 1002 | 1012 | 12/05/2016 | 5 | 5000 | 15000 |

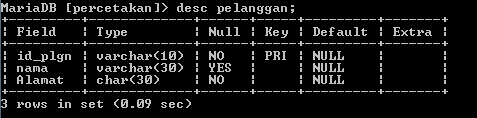
* 1. **Data Definition Language**

1. Tampilan table pada Database percetakan foto



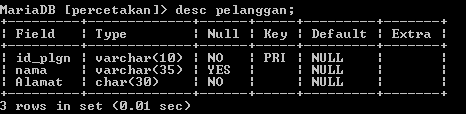
1. Merubah atau memodifikasi table

Sebelum dimodifikasi :



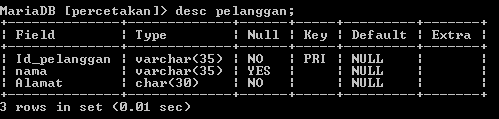
Seletah di masukan code : alter table pelanggan modify id\_plgn varchar(35);

Tampilan table menjadi :

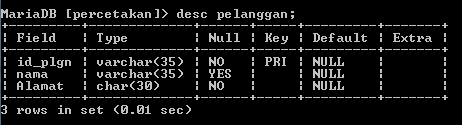


1. Merubah Nama kolom field

Sebelum :

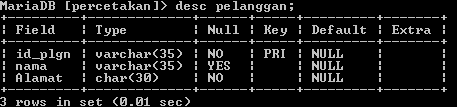


Setelah di tambahkan code : alter table pelanggan change Id\_pelanggan id\_plgn varchar(35);

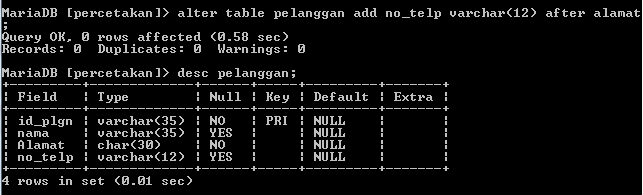


1. Menambahkan Kolom

Tampilan Sebelum dimasukan coding menambahkan kolom :

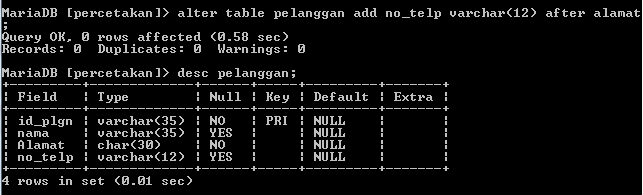


Setelah di masukan coding : alter table pelanggan add no\_telp varchar(12) after alamat;

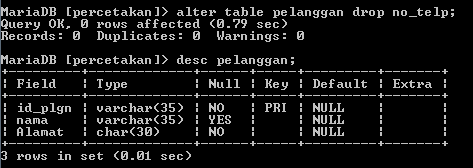


1. Menghapus Kolom (Field)

Sebelum di masukan coding :

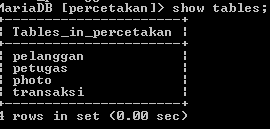


Setelah di masukan coding : alter table pelanggan drop no\_telp.

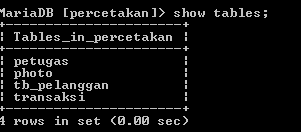


1. Merubah Nama Table

Tampilan sebelum nama table dirubah :



Sesudah dimasukan code : alter table pelanggan rename tb\_pelanggan;



* 1. **Data Manipulation Language**

1. menginputkan data ke table

meninputkan data berupa table ke databases dengan menggunakan code insert into (kolom1,kolom2,...kolomN)values(‘nilai1’,’nilai2’....,,nilaiN’);

data yang akan di inputkan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id\_plgn | nama | Alamat | Id\_plgn | Nama | Alamat |
| 1 | nazar | Bayongbong | 11 | dina | Bayongbong |
| 2 | asep | Garut | 12 | toni | Bayongbong |
| 3 | cucu | Garut | 13 | firman | Bayongbong |
| 4 | caca | Garut | 14 | dikdik | Bayongbong |
| 5 | cici | Garut | 15 | syipa | Tarogong |
| 6 | dede | Tarogong | 16 | abi | Tarogong |
| 7 | dedot | Tarogong | 17 | abu | Tarogong |
| 8 | Agung | Garut | 18 | tina | Tarogong |
| 9 | rifqi | Cilawu | 19 | dedi | Ranca Bolang |
| 10 | anisa | Cilawu | 20 | barik | Ranca Bolang |

Tampilan di di mysqlnya :



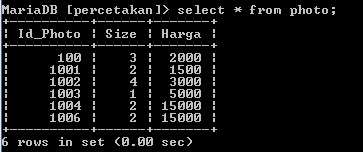
1. Menampilkan Data

Code untuk menampilkan data adalah select \* from nama\_tabel atau select kolom1,kolom2,....kolomN from nama\_table Disini saya memanggil dari tabel pelanggan,photo,dan petugas

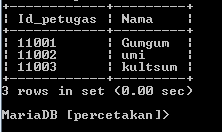
* Tabel pelanggan



* Tabel photo

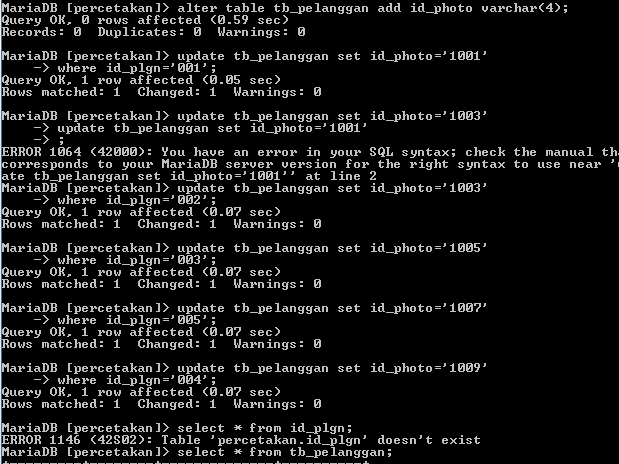


* Tabel petugas

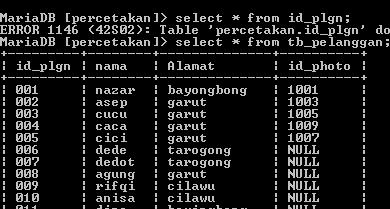


1. Merubah isi data pada table

Contoh codenya :



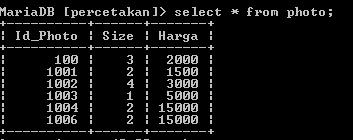
Setelah code dimasukan otomatis update menjadi tabel



Data null itu berarti data kosong;

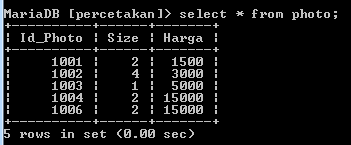
1. menghapus table

Tampilan sebelumnya



isi codenya ada delete from photo where Id\_photo=’100’ ;

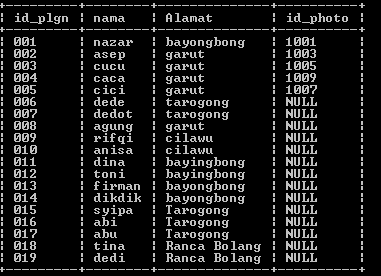
tampilan data yang npm id\_photo di hapus



* 1. **Fungsi SQL**
* Menggunakan Operator Limit

IsI Codenya : select \* from tb\_pelanggan limit 19;

Tampilan programnya :



* Mengurutkan Data

1. mengurut data ascending

Codenya : select \* from tb\_pelanggan order by tb\_pelanggan

asc;

Tampilan pada cmdnya



1. mengurut data descending

Codenya : select \* from tb\_pelanggan order by alamat desc;

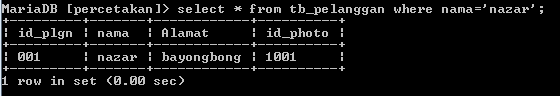
Tampilan pada cmdnya



* Penggunaan Clausa Where

Contoh Code : select \* from jurusan where nama=’nazar’;

Tampilan program



Contoh Code 2: select \* from jurusan where alamat!=’nazar’;

Tampilan program



* Menampilkan Relasi Tabel

Menampilkan Relasi antara tabel pelanggan dengan tabel photo

Contoh codingnya :

Select \* tb\_pelanggan.id\_plgn,photo.id\_photo

* From tb\_pelanggan,photo
* Where pelanggan.id\_photo=photo.id\_photo;