**LAPORAN RESMI**

**MODUL I**

**DDL dan DML**

**SISTEM MANAJEMEN BASIS DATA**



**NAMA : IMAM ARISHANDI IRFANTO**

**N.R.P : 220441100034**

**DOSEN : FITRI DAMAYANTI, S.Kom,M.Kom**

**ASISTEN : M. IQBAL FIRMANSYAH**

**TGL PRAKTIKUM : 22 MARET 2024**

**Disetujui : Maret 2023**

**Asisten**

**M. IQBAL FIRMANSYAH**

**21.04.411.00084**

**LABORATORIUM BISNIS INTELIJEN SISTEM**

**PRODI SISTEM INFORMASI**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS TRUNOJOYO MADURA**

# **BAB I PENDAHULUAN**

## **Latar Belakang**

Semakin berkembangnya teknologi saat ini maka kebutuhan penyimpanan data dalam skala yang besar sangat dibutuhkan terutama untuk perusahaan- perusahaan besar, perangkat data sangat berperan penting dalam perkembangan usahanya. Perangkat lunak yang digunakan untuk mengelola dan memanggil query basis data disebut dengan system manajemen basis data (database management system, DBMS) dalam system basis data dapat dipelajari dalam ilmu informasi. Dalam database tentunya kita mengenal DDL dan DML. Dalam bahasa pemrograman dua hal tersebut memiliki dungsi yang berbeda-beda. Sebelum mengetahui lebih dalam tentang DDL (Data Definition Language) dan DML (Data Manipulation Language), ada yang perlu anda ketahui sebelumnya, bahwa pembahasan ini masuk dalam teknologi database, dimana di dalamnya merupakan cara atau teknik pengelolaan data baik data teks gambar maupun data lainya. Pengelolaan database dilakukan dengan menuliskan kode perintah berupa Query SQL yang harus anda masukan sesuai dengan kebutuhan.Contohnya seperti DDL yang merupakan sekumpulan set perintah yang bertujuan untuk mendefinisikan atribut-atribut database, tabel, atribut kolom(field), maupun batasan-batasan terhadap suatu atribut dan relasi/hubungan antar tabel. Sedangkan DML merupakan kelompok perintah yang berfungsi untuk memanipulasi data dalam database.contohnya untuk memasukkan, pengambilan, pengubahan maupun penghapusan data.

## **Tujuan**

* Mampu memahami perintah-perintah untuk menjelaskan objek dari database dan mendefinisikan atribut-atribut database, table dan batasan-batasan terhadap suatu atribut serta hubungan antar table.
* Mampu memahami dan mengisi table dalam database dan memanipulasi data dalam database.

# **BAB II DASAR TEORI**

## **Data Definition Language**

DDL SQL memungkinkan dilakukannya spesifikasi tidak hanya pada himpunan relasi tetapi juga informasi untuk setiap relasi, yang meliputi :

* Skema setiap relasi
* Domain nilai setiap atribut relasi
* Konstrain integritas
* Himpunan indeks untuk setiap relasi
* Sekuriti dan autorisasi setiap relasi
* Struktur penyimpanan secara fisik untuk setiap relasi dalam disk.

DDL (*Data Definition Languange*) merupakan sekumpulan set perintah yang bertujuan untuk mendefinisikan atribut – atribut database, tabel, atribut kolom (field), maupun batasan – batasan terhadap suatu atribut dan relasi/hubungan antar tabel. Yang termasuk dalam kelompok perintah DDL adalah : CREATE, ALTER, dan DROP. CREATE merupakan perintah DDL yang digunakan untuk membuat database maupun tabel. Nama database maupun tabel tidak boleh mengandung spasi (space). Nama database tidak boleh sama antar database. ALTER merupakan perintah DDL yang digunakan untuk mengubah nama/struktur tabel. DROP merupakan perintah DDL yang digunakan untuk menghapus database ataupun tabel.



### **Database**

Database dapat dibayangkan sebagai sebuah lemari arsip, dimana arsip yang disimpan didalam lemari tersebut akan disusun dan dikelompokkan berdasarkan urutan tertentu seperti urutan abjad atau urutan kronologis. Hal ini dilakukan sebagai upaya untuk mempermudah pencarian arsip, karena arsip yang tersusun dengan rapi maka proses pencarian arsip dapat lebih cepat. Syntax untuk membuat database sebagai berikut :

|  |
| --- |
| CREATE DATABASE namadatabase; |

Sedangkan syntax tambahan untuk menampilkan daftar nama database yang terdapat dalam database server pada MySQL menggunakan perintah :

|  |
| --- |
| SHOW DATABASES; |

Sebelum kita membuat suatu tabel yang digunakan untuk menyimpan data, terlebih dahulu harus memilih/mengaktifkan salah satu database sebagai database aktif yang akan digunakan untuk menyimpan beberapa tabel yang akan kita buat. Untuk memilih/mengaktifkan salah satu database menggunakan syntax :

|  |
| --- |
| USE namadatabase; |

Contohnya kita akan mengaktifkan database db\_universitas : USE db\_universitas; Selain perintah - perintah di atas, terdapat perintah yang berfungsi untuk menghapus database maupun tabel. Perintah tersebut adalah sebagai berikut :

|  |
| --- |
| DROP DATABASE namadatabase; |

### **Tabel**

* Membuat Tabel

Dalam membuat sebuah tabel, nama tabel tidak boleh mengandung spasi (space). Ketika membuat tabel ada beberapa yang harus dideklarasikan dalam pembuatannya, yaitu antara lain meliputi :

- Nama tabel,

- Nama Field (Kolom),

- Type data dari field dan

- Panjang data.

Adapun syntax yang digunakan untuk membuat tabel secara umum adalah sebagai berikut:

|  |
| --- |
| CREATE TABLE namatabel (Field1, TypeData1, Field2, TypeData2); |

Contoh berikut syntax untuk membuat Tabel mahasiswa: CREATE TABLE mahasiswa (nim CHAR (20), nama\_mhs CHAR (50), login CHAR(20), pass CHAR(20), umur INT, ipk real, PRIMARY KEY(nim));

* Menampilkan Tabel

Untuk menampilkan daftar nama tabel yang terdapat dalam database yang sedang aktif/digunakan menggunakan perintah :

|  |
| --- |
| SHOW TABLES; |

* Menampilkan deskripsi atribut table

Untuk menampilkan deskripsi atribut – atribut yang terdapat pada suatu tabel dengan menggunakan perintah:

|  |
| --- |
| DESC namaTabel; |

* Menghapus tabel

Untuk menghapus Tabel perintahnya sama dengan untuk menghapus database yaitu dengan menggunakan perintah DROP. Syntax yang digunakan adalah :

|  |
| --- |
| DROP TABLE namaTable; |

Tabel yang akan dihapus harus sesuai dengan nama tabel. Misal kita akan menghapus tabel mahasiswa, maka syntax nya adalah: DROP TABLE mahasiswa;

* Mendefinisikan Null/Not Null

Null ataupun Not Null merupakan pernyataan yang digunakan untuk membuat kolom yang kita buat boleh kosong (Null) atau tidak boleh kosong (Not Null). Ketika pada kolom tabel tidak diset, maka secara default akan bernilai Null (boleh kosong).

Untuk mendefinisikannya maka perintah yang digunakan adalah:

CREATE TABLE mahasiswa (nim CHAR (20) NOT NULL, nama\_mhs CHAR (50) NOT NULL,login CHAR(20) NOT NULL, pass CHAR(20) NOT NULL, umur INT, ipk real, PRIMARY KEY(nim));

* Mendefinisikan Nilai Default

Nilai default merupakan nilai yang diberikan secara otomatis oleh sistem untuk suatu kolom ketika terjadi penambahan baris baru, sementara nilai pada kolom tersebut tidak diisi oleh pengguna. Contohnya adalah :

CREATE TABLE mahasiswa (nim CHAR (20), nama\_mhs CHAR (50), login CHAR(20), pass CHAR(20), umur INT DEFAULT 0, ipk real, PRIMARY KEY(nim));

* Mendefinisikan PRIMARY KEY pada tabel

Suatu keharusan dalam suatu tabel adalah harus memiliki satu kolom yang dijadikan sebagai perwakilan dari tabel tersebut. Pembuatan perwakilan tabel ini berfungsi untuk melakukan hubungan/relasional dengan tabel lain. Bentuk perwakilan ini dalam database disebut sebagai PRIMARY KEY yang aturan pembuatannya adalah sebagai berikut:

* Satu tabel hanya diperbolehkan memiliki satu kolom kunci
* Nama kolom kunci tidak digunakan pada kolom lain dalam satu tabel
* Nama kolom kunci tidak boleh sama dengan kolom kunci yang ada pada tabel lain
* Bentuk kolom kunci harus diset NOT NULL

Terdapat tiga cara untuk mendefinisikan primary key. Berikut ini syntax yang digunakan:

*CREATE TABLE mahasiswa (nimCHAR (20), nama\_mhs CHAR (50), login CHAR(20), pass CHAR(20), umur INT, ipk real, PRIMARY KEY(nim));*

Atau :

*CREATE TABLE mahasiswa (nim CHAR (20) NOT NULL PRIMARY KEY, nama\_mhs CHAR (50), login CHAR(20), pass CHAR(20), umur INT, ipk real);*

Atau :

*ALTER TABLE mahasiswa ADD CONSTRAINT namaconstraint PRIMARY KEY(namakolom);*

* Mengahapus PRIMARY KEY pada tabel

Cara 1: Jika primary key dibuat menggunakan alter table: ALTER TABLE namatabel DROP CONSTRAINT namaconstraint; Cara 2: jika primary key dibuat melalui create table: ALTER TABLE namatabel DROP PRIMARY KEY;

* Menambah kolom baru pada tabel

Pada saat kita membuat tabel terkadang kita ingin menambahkan kolom lagi pada tabel yang sudah kita buat. Dalam database, hal ini dapat dilakukan dengan menggunakan perintah sebagai berikut:

|  |
| --- |
| ALTER TABLE namatabel ADD fieldbaru typedata(lebar); |

Namatabel merupakan nama tabel yang akan ditambahkan kolomnya. Field baru merupakan nama kolom yang akan ditambahkan, typedata(lebar) merupakan type data dan lebar data yang akan ditambahkan. Misal kita akan menambahkan kolom telepon pada tabel mahasiswa setelah kolom umur:

ALTER TABLE mahasiswa ADD COLUMN telepon CHAR(15) AFTER umur;

* Mengubah tipe data atau lebar kolom pada tabel

Perintah yang digunakan adalah:

|  |
| --- |
| ALTER TABLE namatabel MODIFY COLUMN field |

Contoh :

ALTER TABLE mahasiswa MODIFY COLUMN CHANGE COLUMN telepon VARCHAR(12);

* Mengubah nama kolom (Field)

Perintah yang digunakan adalah:

|  |
| --- |
| ALTER TABLE namatabel namakolomlama namakolombaru typedatabaru(lebarbaru); |

Contoh :

ALTER TABLE mahasiswa CHANGE COLUMN telepon phone CHAR(25);

* Menghapus kolom pada tabel

Perintah yang digunakan:

|  |
| --- |
| ALTER TABLE namatabel DROP COLUMN namakolom; |

Contoh :

ALTER TABLE mahasiswa DROP COLUMN phone;

* Membuat dan menghapus index

Index berfungsi untuk mempercepat proses pencarian data dalam suatu tabel. Dengan adanya index pada suatu field tabel menyebabkan proses pencarian otomatis akan dilakukan terlebih dahulu ke dalam index, apabila ditemukan baru akan diambilkan data sesungguhnya dari tabel, apabila tidak ditemukan dalam index, sudah dapat dipastikan bahwa data tersebut tidak ada dalam tabel. Terdapat perintah untuk membuat dan menghapus index, tapi tidak ada perintah untuk merubah index. Perhatikan contoh berikut :

|  |
| --- |
| CREATE INDEX IDXNIM ON mahasiswa(nim); |

Atau :

|  |
| --- |
| ALTER TABLE mahasiswa ADD INDEX IDXNIM(nim); |

Sedangkan untuk menghapus :

|  |
| --- |
| DROP INDEX IDXNIM ON mahasiswa(nim); |

Atau :

|  |
| --- |
| ALTER TABLE mahasiswa DROP INDEX IDXNIM(nim); |

## **Data Manipulation Language**

Merupakan bentuk bahasa basis data yang berguna untuk melakukan manipulasi data dan pengambilan data pada suatu basis data. Manipulasi data dapat berupa :

1. penyisipan/penambahan data baru ke suatu basis data
2. penghapusan data dari suatu basis data
3. pengubahan data di suatu basis data

Pada level fisik, kita harus mendefinisikan algoritma yang memungkinkan pengaksesan yang efisien terhadap data. Pada level yang lebih tinggi, yang dipentingkan bukan hanya efisiensi akses, tetapi juga efisiensi interaksi manusia (pemakai) dengan sistem (kemudahan permintaan akses). Data Manipulation Language (DML) merupakan bahasa yang bertujuan memudahkan pemakai untuk mengakses data sebagaimana direpresentasikan oleh model data. Ada 2 jenis DML, yaitu :

* Prosedural, yang mensyaratkan agar pemakai menentukan, data apa yang diinginkan serta bagaimana cara mendapatkannya.
* Nonprosedural, yang membuat pemakai dapat menentukan data apa yang diinginkan tanpa menyebutkan bagaimana cara mendapatkannya.

Perintah yang termasuk dalam DML adalah: INSERT, DELETE, UPDATE, dan SELECT.

1. INSERT

Perintah INSERT bertujuan untuk menambahkan record data pada suatu tabel. Terdapat beberapa cara untuk menambahkan record, yaitu:

Cara 1: Menambahkan record dengan mengisi data pada setiap kolom:

|  |
| --- |
| INSERT INTO namatabel VALUES (nilai1, nilai2, nilai- |

Cara 2: Menambahkan baris dengan hanya mengisi pada kolom tertentu:

|  |
| --- |
| INSERT INTO namatabel VALUES (nilai1, nilai2, nilai- |

Ket : Jika data bertipe string, date, atau time (contoh : didi, basis data, 1984- 03-18) maka pemberian nilainya diapit menggunakan tanda petik tunggal ('Didi') atau petik ganda (“Basis Data”). Jika data bertipe numerik (29, 4) maka pemberian nilainya tidak diapit tanda petik tunggal maupun ganda.

1. DELETE

Perintah DELETE digunakan untuk menghapus satu baris, baris dengan kondisi tertentu maupun seluruh baris. Syntax yang digunakan:

|  |
| --- |
| DELETE FROM namatabel WHERE [kondisi]; |

Perintah dalam tanda [] bersifat pilihan/opsional untuk menghapus suatu baris dengan kondisi tertentu yang dipersyaratkan. Contoh perintah untuk menghapus suatu baris dalam tabel dengan kondisi persyaratan tertentu :

DELETE FROM mahasiswa WHERE nim 13120070;

1. UPDATE

Perintah UPDATE digunakan untuk mengubah isi data pada satu atau beberapa kolom pada suatu tabel. Syntax yang digunakan secara umum adalah sebagai berikut:

|  |
| --- |
| UPDATE namatabel SET field1=nilai1, field2=nilai2 [WHERE kondisi]; |

Perintah dalam tanda [] bersifat pilihan/opsional untuk mengubah suatu baris dengan kondisi tertentu yang dipersyaratkan.

1. SELECT

Perintah SELECT digunakan untuk menampilkan isi dari suatu tabel yang dapat dihubungkan dengan beberapa tabel lainnya. Menampilkan data semua kolom dengan menggunakan asterisk (\*):

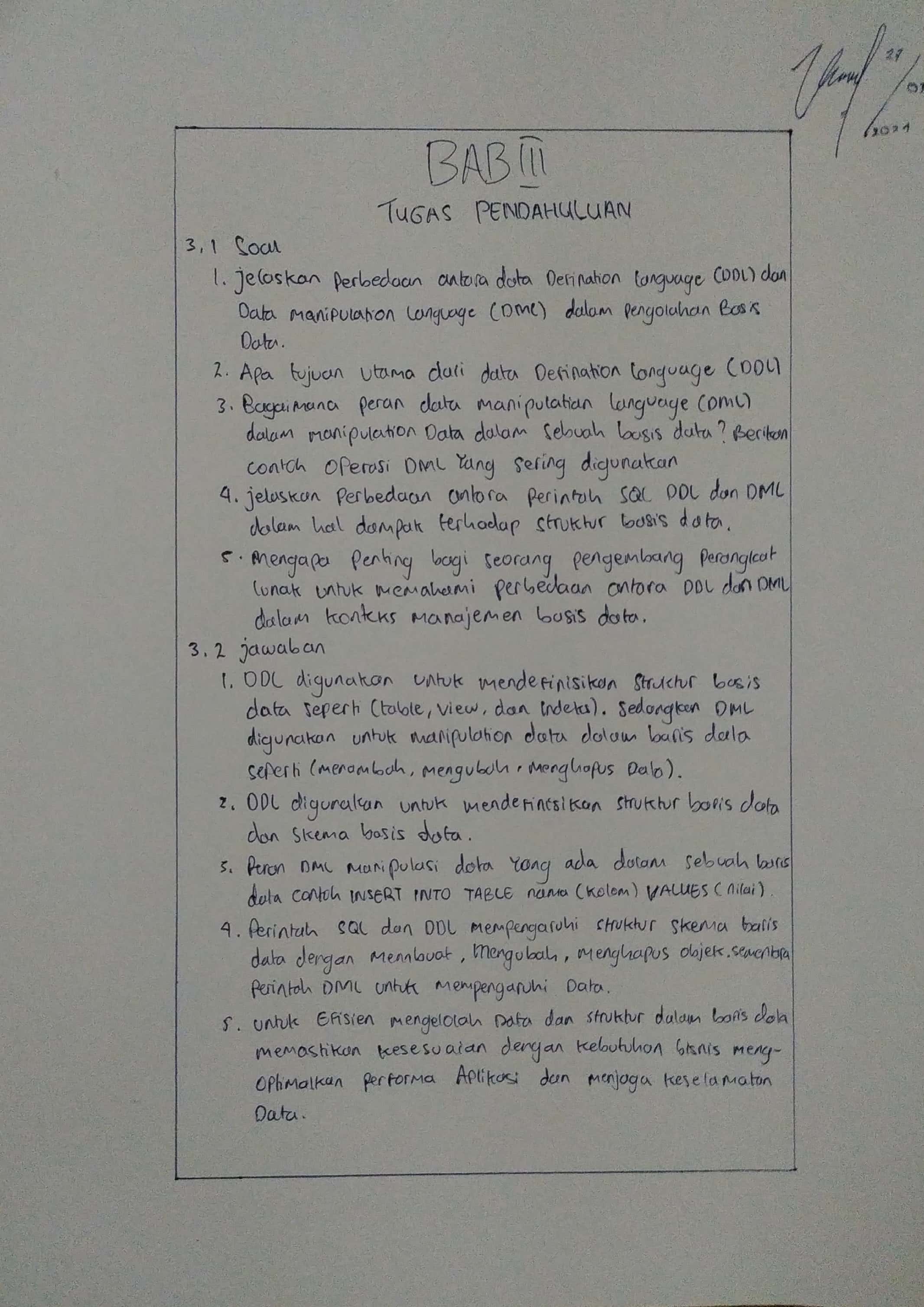
|  |
| --- |
| SELECT \* FROM namatabel; |

Menampilkan data untuk field/kolom tertentu:

|  |
| --- |
| SELECT field1, field2, field-n FROM namatabel; |

Menampilkan data dengan kondisi tertentu menggunakan klausa WHERE:

|  |
| --- |
| SELECT \* FROM namatabel WHERE kondisi; |



# **BAB IV IMPLEMENTASI**

## **Soal**

1. Universitas trunojoyo ingin mengembangkan sebuah database untuk mencatat berbagai transaksi dan keungan yang terjadi,berikut adalah detai administrasi,keungan,dan laporan yang perlu dicatat:

Administraasi:

1. Pencatatan mahasiswa:catat setiap data mahasiswa yang ada di Universitas Trunojoyo
2. Pencatatan dosen: catat setiap data dosen yang ada di Universitas Trunojoyo
3. Pencattan mahasiswa berhenti kuliah dan lulus: catat setiap mahasiswa yang berhenti kuliah dan lulus
4. Pendataan mata kuliah :catat setiap pengambilan mata kuliah oleh mahsiswa.

Kuangan

1. Pembayaran UKT: Masukkan setiap transaksi Pembayaran UKT mahasiswa
2. Pembayaran gaji dosen: catat setiap pembayaran gaji kepada dosen
3. Pembelian peralatan dan bahan ajar:catat setiap pembelian peraltan dan bahan ajar untuk keperluan pengajaran
4. Pembayaran tagihan Listrik,air,dll:catat setiap pembayaran tagihan ruti Universitas Trunojoyo

Laporan

1. Laporan daftar mahasiswa: Laporan jumlah mahasiswa di universitas trunojoyo
2. Laporan daftar dosen: : Laporan jumlah dosen di universitas trunojoyo
3. Laporan keungan: : Laporan keungan di universitas trunojoyo
4. Isi data pada table masing-masing 10 data

## **Query**

1. Membuat Database

CREATE DATABASE db\_university;

DROP DATABASE db\_university;

use db\_university;

CREATE TABLE tb\_mahasiswa (

nim INT NOT NULL PRIMARY KEY,

nama\_mahasiswa VARCHAR (50) NOT NULL,

prodi VARCHAR (20) NOT NULL,

tanggal\_masuk DATE NOT NULL

);

CREATE TABLE tb\_dosen (

nip INT NOT NULL PRIMARY KEY,

nama\_dosen VARCHAR (50) NOT NULL,

bidang\_keahlian VARCHAR (20) NOT NULL,

kode\_matakuliah VARCHAR (10) NOT NULL,

FOREIGN KEY (kode\_matakuliah) REFERENCES tb\_matakuliah(kode\_mata\_kuliah)

);  
CREATE TABLE tb\_status\_mahasiswa (

id INT NOT NULL Primary Key,

nim INT NOT NULL,

status\_mahasiswa ENUM('Aktif', 'Berhenti', 'Lulus') NOT NULL,

tanggal DATE NOT NULL,

FOREIGN KEY (nim) REFERENCES tb\_mahasiswa(nim)

);

CREATE TABLE tb\_matakuliah (

kode\_mata\_kuliah VARCHAR (10) NOT NULL PRIMARY KEY,

nama\_matakuliah VARCHAR (50) NOT NULL,

sks\_matakuliah INT NOT NULL

);

CREATE TABLE tb\_pengambilan\_matakuliah (

id INT NOT NULL PRIMARY KEY,

nim INT NOT NULL,

kode\_mata\_kuliah VARCHAR (10) NOT NULL,

tahun\_ajaran VARCHAR (10) NOT NULL,

semester INT (5) NOT NULL,

FOREIGN KEY (nim) REFERENCES tb\_mahasiswa(nim),

FOREIGN KEY (kode\_mata\_kuliah) REFERENCES tb\_matakuliah(kode\_mata\_kuliah)

);

CREATE TABLE tb\_Pembayaran\_ukt (

id int NOT NULL PRIMARY KEY,

nim INT NOT NULL,

jumlah\_pembayaran INT NOT NULL,

tanggal\_pembayaran DATE NOT NULL,

FOREIGN KEY (nim) REFERENCES tb\_mahasiswa(nim)

);

CREATE TABLE tb\_pembayaran\_gaji\_dosen (

id int NOT NULL PRIMARY KEY,

nip INT NOT NULL,

jumlah\_pembayaran INT NOT NULL,

tanggal\_pembayaran DATE NOT NULL,

FOREIGN KEY (nip) REFERENCES tb\_dosen(nip)

);

CREATE TABLE tb\_pembelian\_peralatan (

id INT NOT NULL PRIMARY KEY,

deskripsi VARCHAR (20) NOT NULL,

jumlah INT NOT NULL,

tanggal\_pembelian DATE NOT NULL,

total INT NOT NULL

);

CREATE TABLE tb\_pembayaran\_tagihan\_rutin (

id INT NOT NULL PRIMARY KEY,

deskripsi\_tagihan VARCHAR (50) NOT NULL,

jumlah\_pembayaran INT NOT NULL,

tanggal\_pembayaran DATE NOT NULL

);

CREATE TABLE tb\_transaksi\_keuangan (

id INT NOT NULL AUTO\_INCREMENT PRIMARY KEY,

pembayaran\_ukt\_id INT,

pembayaran\_gaji\_dosen\_id INT,

pembayaran\_pembelian\_id INT,

pembayaran\_tagihan\_rutin\_id INT,

FOREIGN KEY (pembayaran\_ukt\_id) REFERENCES tb\_pembayaran\_ukt(id),

FOREIGN KEY (pembayaran\_gaji\_dosen\_id) REFERENCES tb\_pembayaran\_gaji\_dosen(id),

FOREIGN KEY (pembayaran\_pembelian\_id) REFERENCES tb\_pembelian\_peralatan(id),

FOREIGN KEY (pembayaran\_tagihan\_rutin\_id) REFERENCES tb\_pembayaran\_tagihan\_rutin(id)

);

1. Mengisi Tabel

INSERT INTO tb\_mahasiswa (nim, nama\_mahasiswa, prodi, tanggal\_masuk)

VALUES

(220410001, 'Angga Jalasena Saputra', 'Sistem Informasi', '2024-03-10'),

(220410108, 'Della Ayu Vernandah', 'Sistem Informasi', '2024-03-25'),

(220410082, 'Amelia Zalfa Sahira', 'Sistem Informasi', '2024-03-17'),

(220410034, 'Imam Arishandi Irfanto', 'Sistem Informasi', '2024-03-09'),

(220410154, 'Dwi Nurjannah', 'Sistem Informasi', '2024-03-02'),

(220410079, 'Viko Alfian P', 'Sistem Informasi', '2024-03-14'),

(220410022, 'Tesa Putri Hutagaol', 'Sistem Informasi', '2024-03-29'),

(220410014, 'Bintang Wahyu Ariyono', 'Sistem Informasi', '2024-03-22'),

(220410085, 'Khoirotun Nisak', 'Sistem Informasi', '2024-03-15'),

(220410119, 'Romauli Napitupulu', 'Sistem Informasi', '2024-03-24'),

(220410159, 'M. Irfan Faruq Zain', 'Sistem Informasi', '2024-03-27'),

(220410009, 'Naela Nahdiah', 'Sistem Informasi', '2024-03-16'),

(220410107, 'Oxa Indi Vixzio', 'Sistem Informasi', '2024-03-19');

INSERT INTO tb\_dosen (nip, nama\_dosen, bidang\_keahlian, kode\_matakuliah)

VALUES

(101, 'Budi Satoto', 'Sistem Informasi','SPK'),

(102, 'Yeni', 'Sistem Informasi', 'DM'),

(103, 'Fitri', 'Sistem Informasi', 'PEMBER'),

(104, 'Zain', 'Sistem Informasi', 'SMBD'),

(105, 'Hera', 'Sistem Informasi', 'PSDP'),

(106, 'Novi', 'Sistem Informasi', 'MPIT'),

(107, 'Budi Susilo', 'Sistem Informasi', 'IPPL'),

(108, 'Rika', 'Sistem Informasi', 'SPK'),

(109, 'Doni', 'Sistem Informasi', 'PEMBER'),

(110, 'Syarif', 'Sistem Informasi', 'MPIT');

INSERT INTO tb\_status\_mahasiswa (id,nim,status\_mahasiswa,tanggal)

VALUES

(1,220410001,'Aktif','2024-03-27'),

(2,220410108,'Aktif','2024-03-27'),

(3,220410082,'Aktif','2024-03-27'),

(4,220410034,'Lulus','2024-03-27'),

(5,220410154,'Aktif','2024-03-27'),

(6,220410079,'Berhenti','2024-03-27'),

(7,220410022,'Aktif','2024-03-27'),

(8,220410014,'Aktif','2024-03-27'),

(9,220410085,'Aktif','2024-03-27'),

(10,220410119,'Aktif','2024-03-27'),

(11,220410159,'Aktif','2024-03-27'),

(12,220410009,'Aktif','2024-03-27'),

(13,220410107,'Aktif','2024-03-27');

INSERT INTO tb\_matakuliah (kode\_mata\_kuliah, nama\_matakuliah, sks\_matakuliah) VALUES

('SPK','Sistem Pendukung Keputusan',3),

('DM','Data Mining',3),

('PEMBER','Pemrograman Bergerak',4),

('SMBD','Sistem Manajemen Basis Data',4),

('PSDP','Perencanaan Sumber Daya Perusahaan',2),

('MPIT','Manajemen Proyek IT',3),

('IPPL','Implementasi Dan Pengujuan Perangkat Lunak',3);

INSERT INTO tb\_pengambilan\_matakuliah (id,nim,kode\_mata\_kuliah,tahun\_ajaran,semester)

VALUE

(1,220410001,'SPK','2024/2025',4),

(2,220410001,'MPIT','2024/2025',4),

(3,220410001,'IPPL','2024/2025',4),

(4,220410034,'PSDP','2024/2025',4),

(5,220410034,'SMBD','2024/2025',4),

(6,220410034,'PEMBER','2024/2025',4),

(7,220410034,'SPK','2024/2025',4),

(8,220410022,'SPK','2024/2025',4),

(9,220410022,'PEMBER','2024/2025',4),

(10,220410022,'MPIT','2024/2025',4),

(11,220410159,'SMBD','2024/2025',4),

(12,220410159,'SPK','2024/2025',4),

(13,220410159,'IPPL','2024/2025',4),

(14,220410014,'SMBD','2024/2025',4),

(15,220410014,'PEMBER','2024/2025',4),

(16,220410014,'IPPL','2024/2025',4),

(17,220410014,'SPK','2024/2025',4);

INSERT INTO tb\_pembayaran\_ukt (id,nim,tanggal\_pembayaran, jumlah\_pembayaran)

VALUES

(1,'220410001', '2024-03-25', 5000000),

(2,'220410108', '2024-03-24', 4500000),

(3,'220410082', '2024-03-23', 4800000),

(4,'220410034', '2024-03-22', 5200000),

(5,'220410154', '2024-03-21', 4700000),

(6,'220410079', '2024-03-20', 4900000),

(7,'220410022', '2024-03-19', 5100000),

(8,'220410014', '2024-03-18', 4600000),

(9,'220410085', '2024-03-17', 5300000),

(10,'220410119', '2024-03-16', 4800000),

(11,'220410159', '2024-03-16', 3500000),

(12,'220410009', '2024-03-16', 5000000),

(13,'220410107', '2024-03-16', 4500000);

INSERT INTO tb\_pembayaran\_gaji\_dosen (id, nip, tanggal\_pembayaran, jumlah\_pembayaran)

VALUES

(1, 101, '2024-03-25', 10000000),

(2, 102, '2024-03-24', 9500000),

(3, 103, '2024-03-23', 9800000),

(4, 104, '2024-03-22', 10500000),

(5, 105, '2024-03-17', 10300000),

(6, 106, '2024-03-21', 9700000),

(7, 107, '2024-03-20', 9900000),

(8, 108, '2024-03-19', 10100000),

(9, 109, '2024-03-18', 9600000),

(0, 110, '2024-03-16', 9800000);

INSERT INTO tb\_pembelian\_peralatan (id, deskripsi, tanggal\_pembelian, jumlah, total)

VALUES

('123', 'Proyektor' ,'2024-03-25', 5, 250000),

('124', 'Layar Proyektor' ,'2024-03-24', 3, 180000),

('125', 'Laptop','2024-03-23', 2, 120000),

('126', 'PC','2024-03-22', 4, 240000),

('127', 'Kipas Angin','2024-03-21', 6, 300000),

('128', 'AC','2024-03-20', 7, 350000),

('129', 'Karpet','2024-03-19', 1, 600000),

('130', 'Kursi' ,'2024-03-18', 8, 400000),

('131', 'ATK','2024-03-17', 3, 180000),

('132', 'Sapu','2024-03-16', 5, 250000);

INSERT INTO tb\_pembayaran\_tagihan\_rutin (id, deskripsi\_tagihan, jumlah\_pembayaran, tanggal\_pembayaran)

VALUES

(001, 'Tagihan Listrik', 500000, '2024-02-25'),

(002, 'Tagihan Air', 450000, '2024-02-25'),

(003, 'Tagihan Internet', 480000, '2024-02-25'),

(004, 'Tagihan Telepon', 520000, '2024-02-25'),

(005, 'Tagihan TV Kabel', 470000, '2024-02-25'),

(006, 'Tagihan Asuransi', 490000, '2024-02-25'),

(007, 'Tagihan Pajak', 510000, '2024-02-25'),

(008, 'Tagihan Pemeliharaan', 460000, '2024-02-25'),

(009, 'Tagihan Parkir', 530000, '2024-02-25'),

(010, 'Tagihan Keanggotaan', 480000, '2024-02-25');

INSERT INTO tb\_transaksi\_keuangan(pembayaran\_ukt\_id,pembayaran\_gaji\_dosen\_id,pembayaran\_pembelian\_id,pembayaran\_tagihan\_rutin\_id)

VALUES

(1,1,123,001),

(2,2,124,002),

(3,3,125,003),

(4,4,126,004),

(5,5,127,005),

(6,6,128,006),

(7,7,129,007),

(8,8,130,008),

(9,9,131,009),

(10,0,132,010);

1. Menampilkan Tabel

select count(\*) AS Jumlah\_Mahasiswa from tb\_mahasiswa;

select count(\*) AS Jumlah\_Dosenfrom tb\_dosen;

select \* FROM tb\_transaksi\_keuangan;

select \* from tb\_dosen;

select \* from tb\_pengambilan\_matakuliah;

select \* from tb\_Pembayaran\_ukt;

select \* from tb\_pembayaran\_gaji\_dosen;

select \* from tb\_pembelian\_peralatan;

select \* from tb\_pembayaran\_tagihan\_rutin;

## **Hasil**

1. Laporan Keuangan



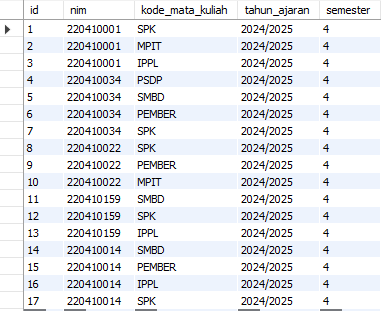
1. Table Mahasiswa



1. Table Dosen



1. Tabel Pendaftaran Matakuliah



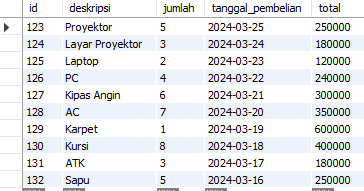
1. Tabel Pembayaran UKT



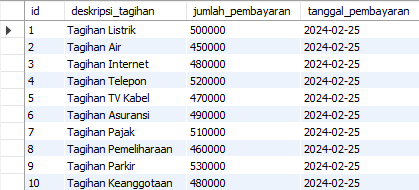
1. Tabel pembayaran Gaji Dosen



1. Table Pembelian Peralatan



1. Tabel Pembayaran Tagihan Rutin



## **Penjelasan**

Pada tugas kali ini saya membuat database university sesuai dengan query yang saya buat, yah tentu pertama-tama saya membuat query untuk membuat database university, lalu saya membuat query untuk membuat tabel sesuai yang ada di tugas, dengan itu saya juga mengisi tabel yang sudah saya buat dengan query yang sudah ada di atas, mengisi tabel mahasiswa, dosen, dan lain-lain. Setelah itu saya membuat beberapa query lagi untuk menampilkan hasil dari tabel atau menampilkan isi dari tabel, dan juga saya membuat file ERD untuk menampilkan hasil database yang sudah saya buat, dan melihat antara relasi-relasi antar tabel yang sudah saya atur, semua itu saya lakukan menggunakan software yang sama yaitu SQL Workbench. Outputnya di sini universitas sudah bisa mengontrol keoangan yang masuk dan keluar, dengan adanya tabel-tabel baru yang sudah di buat.

# **BAB V PENUTUP**

## **Analisa**

DDL (Data Definition Language) adalah salah satu dari Querry yang digunakan untuk membuat dan mendefinisikan struktur dan schema dari suatu database. DDL berisi perintah-perintah untuk membuat tabel, mengubah struktur tabel, menambahkan atau menghapus indeks dan constraint, serta membatasi dan menentukan hubungan antar tabel. DDL digunakan pada tahap desain dan pengembangan database, sehingga sangat penting untuk menghasilkan database yang baik dan efisien.

Sementara itu, DML (Data Manipulation Language) juga merupakan salah satu dari Querry yang digunakan untuk memanipulasi dan mengambil data dari tabel-tabel yang sudah didefinisikan oleh DDL. DML berisi perintah-perintah untuk menambah, menghapus, dan memperbarui data yang tersimpan dalam tabel. Selain itu, DML juga digunakan untuk mengambil data dari tabel untuk dianalisis dan diproses lebih lanjut. DML umumnya digunakan pada tahap operasional database, seperti saat melakukan transaksi atau mengambil data untuk keperluan laporan.

## **Kesimpulan**

DDL (Data Definition Languange) merupakan sekumpulan set perintah yang bertujuan untuk membuat struktur dan schema database, mendefinisikan atribut – atribut database, tabel, atribut kolom (field), maupun batasan – batasan terhadap suatu atribut dan relasi/hubungan antar tabel. Sedangkan DML merupakan bentuk bahasa basis data yang berguna untuk pengambilan data pada suatu basis data dan melakukan manipulasi data seperti penambahan data baru ke suatu basis data, menghapus data dari suatu basis data, dan pengubahan data di suatu basis data. Kedua bentuk bahasa basis data tersebut sangat penting dalam menjaga keutuhan dan efisiensi basis data.