

1. Memahami Proses dan Mengidentifikasi DDL, DQL, dan DML

Proses dari DFD untuk Pendaftaran:

- Formulir input untuk pendaftaran siswa.
- Cek apakah siswa sudah terdaftar.
- Simpan data siswa jika belum terdaftar, jika sudah maka tampilkan data yang ada.
- Update status pembayaran siswa setelah interaksi dengan bank.

Proses untuk Tes Seleksi:

- Siswa melakukan tes.
- Jawaban disimpan ke tabel hasil tes.
- Update hasil tes sesuai dengan nilai yang diperoleh siswa.

Proses untuk Pembayaran:

- Siswa dapat mengecek status pembayaran.
- Kwitansi otomatis dihasilkan berdasarkan data pembayaran.

2. DDL (Data Definition Language)

DDL digunakan untuk membuat tabel dalam basis data berdasarkan proses yang dijelaskan. Tabel-tabel yang diperlukan antara lain:

- Tabel: `siswa` (students)
 - Kolom: `id_siswa`, `nama`, `tanggal_lahir`, `no_ktp`, `no_pendaftaran`, `status_pembayaran`.
- Tabel: `hasil_tes` (test_results)
 - Kolom: `id_tes`, `id_siswa`, `nilai`, `hasil`.
- Tabel: `pembayaran` (payments)
 - Kolom: `id_pembayaran`, `id_siswa`, `jumlah`, `tanggal`, `status`.

Contoh DDL untuk membuat tabel:

```
CREATE TABLE siswa (  
    id_siswa INT PRIMARY KEY AUTO_INCREMENT,  
    nama VARCHAR(100),  
    tanggal_lahir DATE,  
    no_ktp VARCHAR(20) UNIQUE,  
    no_pendaftaran VARCHAR(50),
```

```
status_pembayaran ENUM('Belum Bayar', 'Sudah Bayar')
);
```

-- Tabel hasil tes

```
CREATE TABLE hasil_tes (
    id_tes INT PRIMARY KEY AUTO_INCREMENT,
    id_siswa INT,
    nilai DECIMAL(5,2),
    hasil VARCHAR(50),
    FOREIGN KEY (id_siswa) REFERENCES siswa(id_siswa)
);
```

-- Tabel pembayaran

```
CREATE TABLE pembayaran (
    id_pembayaran INT PRIMARY KEY AUTO_INCREMENT,
    id_siswa INT,
    jumlah DECIMAL(10,2),
    tanggal DATE,
    status ENUM('Pending', 'Selesai'),
    FOREIGN KEY (id_siswa) REFERENCES siswa(id_siswa)
);
---
```

3. DML (Data Manipulation Language)

DML digunakan untuk memasukkan, memperbarui, atau menghapus data dalam tabel.

Contoh DML:

- Menambahkan Data Siswa Baru:

```
INSERT INTO siswa (nama, tanggal_lahir, no_ktp, no_pendaftaran, status_pembayaran)
VALUES ('Siti Nurcica', '2001-12-15', '1234567890123456', 'REG20241001', 'Belum Bayar');
```

- Memperbarui Status Pembayaran:

```
UPDATE siswa
SET status_pembayaran = 'Sudah Bayar'
```

```
WHERE no_pendaftaran = 'REG20241001';
```

- Memasukkan Hasil Tes:

```
INSERT INTO hasil_tes (id_siswa, nilai, hasil)  
VALUES (1, 85.5, 'Lulus');
```

4. DQL (Data Query Language)

DQL digunakan untuk mengambil data dari tabel menggunakan perintah `SELECT`.

Contoh DQL:

- Menampilkan Data Siswa Berdasarkan Nomor Pendaftaran:

```
SELECT * FROM siswa  
WHERE no_pendaftaran = 'REG20241001';
```

- Menampilkan Hasil Tes Berdasarkan ID Siswa:

```
SELECT * FROM hasil_tes  
WHERE id_siswa = 1;
```

5. Store Procedure untuk DML dan DQL

Untuk mempermudah eksekusi perintah DML dan DQL, kita dapat membuat store procedure. Berikut contoh store procedure untuk memperbarui status pembayaran:

```
DELIMITER //
```

```
CREATE PROCEDURE updateStatusPembayaran(IN p_no_pendaftaran VARCHAR(50), IN p_status  
ENUM('Belum Bayar', 'Sudah Bayar'))
```

```
BEGIN
```

```
    UPDATE siswa
```

```
    SET status_pembayaran = p_status
```

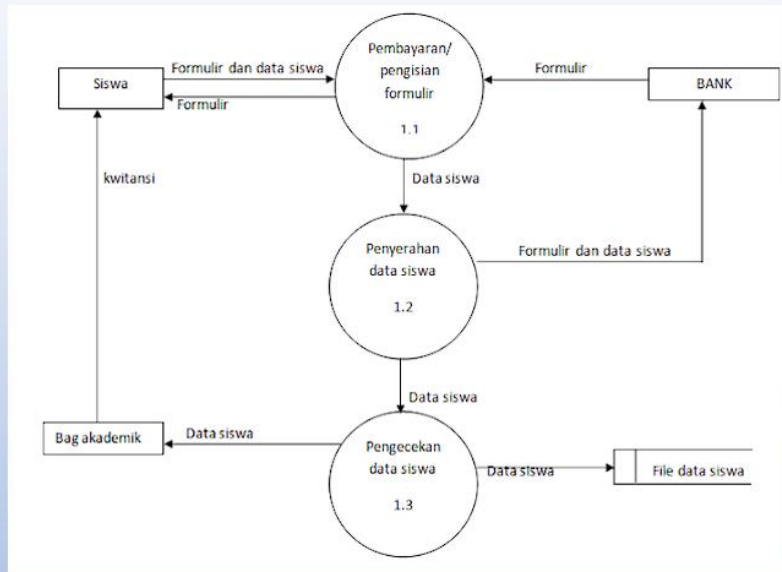
```
    WHERE no_pendaftaran = p_no_pendaftaran;
```

```
END //
```

```
DELIMITER ;
```

Store procedure ini bisa dipanggil dengan:

```
CALL updateStatusPembayaran('REG20241001', 'Sudah Bayar');
```



DFD Level 2 : 1.0 Pendaftaran

Apa saja DML, DQL yang dapat dibuat dari proses berikut?

- Uraikan proses
- Akan dibutuhkan banyak perintah SQL
- Buatkan kedalam store procedure

Uraian Proses

1. Pembayaran/Pengisian Formulir (1.1):

- Siswa mengisi formulir pendaftaran dan melakukan pembayaran.
- Data siswa, seperti nama, tanggal lahir, nomor KTP, dan nomor pendaftaran, akan dimasukkan ke dalam sistem.
- Kwitansi pembayaran dihasilkan.

2. Penyerahan Data Siswa (1.2):

- Data yang sudah dimasukkan akan diserahkan ke pihak akademik.
- Data siswa harus dicatat dengan baik dan benar dalam sistem.

3. Pengecekan Data Siswa (1.3):

- Pihak akademik melakukan pengecekan data siswa untuk memastikan bahwa semua data yang diperlukan sudah lengkap dan benar.
- Hasil pengecekan bisa diambil untuk keperluan administrasi lebih lanjut.

DML dan DQL yang Dapat Dihasilkan

DML

- Insert Data Siswa: Menambahkan data siswa baru ke dalam tabel `siswa`.

- Update Status Pembayaran: Memperbarui status pembayaran siswa setelah pembayaran dilakukan.
- Insert Kwitansi: Menyimpan data kwitansi ke dalam tabel `kwitansi` (jika ada tabel tersebut).

DQL

- Select Data Siswa: Mengambil data siswa berdasarkan nomor pendaftaran untuk memastikan data siswa sudah terdaftar.
- Select Kwitansi: Mengambil data kwitansi untuk verifikasi pembayaran.
- Select Data untuk Pengecekan: Mengambil data siswa yang akan diperiksa oleh pihak akademik.

Contoh SQL

1. DML Contoh

- Menambahkan Data Siswa Baru

```
INSERT INTO siswa (nama, tanggal_lahir, no_ktp, no_pendaftaran, status_pembayaran)
VALUES ('John Doe', '2002-01-15', '1234567890123456', 'REG20241002', 'Belum Bayar');
```

- **Memperbarui Status Pembayaran**

```
UPDATE siswa
SET status_pembayaran = 'Sudah Bayar'
WHERE no_pendaftaran = 'REG20241002';
```

- **Menyimpan Kwitansi Pembayaran**

```
INSERT INTO kwitansi (id_siswa, jumlah, tanggal)
VALUES (1, 1000000, NOW());
```

2. DQL Contoh

- Menampilkan Data Siswa Berdasarkan Nomor Pendaftaran

```
SELECT * FROM siswa
WHERE no_pendaftaran = 'REG20241002';
```

- **Menampilkan Kwitansi Berdasarkan ID Siswa**

```
SELECT * FROM kwitansi
WHERE id_siswa = 1;
```

Store Procedure

DELIMITER //

CREATE PROCEDURE ProsesPendaftaran(

IN p_nama VARCHAR(100),

IN p_tanggal_lahir DATE,

IN p_no_ktp VARCHAR(20),

IN p_no_pendaftaran VARCHAR(50),

IN p_jumlah DECIMAL(10,2)

)

BEGIN

DECLARE p_id_siswa INT;

-- Insert Data Siswa

INSERT INTO siswa (nama, tanggal_lahir, no_ktp, no_pendaftaran, status_pembayaran)

VALUES (p_nama, p_tanggal_lahir, p_no_ktp, p_no_pendaftaran, 'Belum Bayar');

SET p_id_siswa = LAST_INSERT_ID();

-- Insert Kwitansi

INSERT INTO kwitansi (id_siswa, jumlah, tanggal)

VALUES (p_id_siswa, p_jumlah, NOW());

-- Update Status Pembayaran

UPDATE siswa

SET status_pembayaran = 'Sudah Bayar'

WHERE id_siswa = p_id_siswa;

END //

DELIMITER ;

...

Memanggil Store Procedure

Untuk memanggil store procedure tersebut, gunakan perintah berikut:

```
CALL ProsesPendaftaran('John Doe', '2002-01-15', '1234567890123456', 'REG20241002', 1000000);
```

Dengan prosedur dan query yang telah diuraikan di atas, Anda dapat dengan mudah mengelola data siswa dan proses pembayaran. Jika Anda memerlukan bantuan lebih lanjut atau ada pertanyaan, silakan beri tahu saya!