

$$\begin{array}{r} 20 \\ 24 \\ \hline 20 \\ 25 \end{array}$$


- Amaliah Faradibah,
S.Kom.,M.Kom.,MTA.,
MCF
- Ir. Dedy Atmajaya,
S.Kom.,M.Eng.,MTA.
- Andi Ulfah Tenripada,
S.Kom.,M.Kom.,MTA.
- Tim Asisten Laboratorium

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Kuasa, yang telah memberikan rahmat-Nya sehingga Modul Praktikum **Basis Data 2** untuk mahasiswa/i Program Studi Sistem Informasi Fakultas Ilmu Komputer Universitas Muslim Indonesia ini dapat diselesaikan dengan sebaik-baiknya.

Modul praktikum ini dibuat sebagai pedoman dalam melakukan kegiatan praktikum **Basis Data 2** yang merupakan kegiatan penunjang mata kuliah pada Program Studi Sistem Informasi. Modul praktikum ini diharapkan dapat membantu mahasiswa/i dalam mempersiapkan dan melaksanakan praktikum dengan lebih baik, terarah, dan terencana. Pada setiap topik telah ditetapkan capaian pembelajaran mata kuliah pelaksanaan praktikum dan semua kegiatan yang harus dilakukan oleh mahasiswa/i serta teori singkat untuk memperdalam pemahaman mahasiswa/i mengenai materi yang dibahas.

Penyusun menyakini bahwa dalam pembuatan Modul Praktikum **Basis Data 2** ini masih jauh dari sempurna. Oleh karena itu penyusun mengharapkan kritik dan saran yang membangun guna penyempurnaan modul praktikum ini dimasa yang akan datang.

Akhir kata, penyusun mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung.

Makassar, September 2024

Tim Penyusun

TATA TERTIB PELAKSANAAN PRAKTIKUM

Tata Tertib Pelaksanaan Praktikum pada Laboratorium Terpadu Fakultas Ilmu Komputer UMI adalah sebagai berikut:

1. Seluruh Pengguna laboratorium harus dalam keadaan sehat tidak menunjukkan gejala sakit (batuk, hidung tersumbat, dan suhu badan diatas 37°C).
2. Praktikan hanya diizinkan melaksanakan praktikum apabila :
 - a. Pria
 - Berpakaian rapi memakai kemeja putih polos;
 - Menggunakan celana kain berwarna hitam bukan dari bahan jeans/semi jeans;
 - Rambut rapi dan tidak panjang;
 - b. Wanita
 - Berpakaian rapi memakai kemeja tunik putih polos (tidak transparan)
 - Memakai Jilbab Segitiga Hitam (bukan pasmina) dan menutupi dada.
 - Menggunakan Rok Panjang berwarna hitam yang tidak terbelah dan tidak span serta bukan dari bahan jeans/semi jeans;
 - Memakai kaos kaki dengan tinggi minimal 10 cm di atas mata kaki;
3. Ketika memasuki dan selama berada dalam ruangan, praktikan diwajibkan :
 - Tenang, tertib, dan sopan;
 - Tidak mengganggu praktikan lain yang sedang melaksanakan praktikum;
 - Tidak diperbolehkan merokok, membawa makanan / minuman senjata tajam dan senjata api ke dalam ruangan praktikum;
 - Tidak diperbolehkan membawa *handphone* ke meja praktikum dan *handphone* dalam mode senyap;
 - Tidak diperbolehkan membawa media penyimpanan eksternal atau *flashdisk* ke meja praktikum tanpa seizin Dosen Pengampu atau Asisten;
4. Dilarang membawa, mengambil, serta memindahkan perangkat yang digunakan pada saat praktikum tanpa instruksi dari Dosen Pengampu atau Asisten.
5. Toleransi keterlambatan praktikan maksimal 5 menit.
6. Praktikan berada diarea laboratorium dengan mengikuti jadwal yang telah ditentukan oleh Kepala Laboratorium.

7. Penggunaan fasilitas Laboratorium menyesuaikan dengan kapasitas ruang Laboratorium.
8. Segala pelanggaran yang dilakukan oleh praktikan akan berakibat pada penutupan dan penghentian penggunaan seluruh fasilitas laboratorium dan ditindak sesuai dengan aturan yang berlaku.

SANKSI-SANKSI

Sanksi terhadap pelanggaran **TATA TERTIB**:

Dosen Pengampu dan Asisten laboratorium berhak menjatuhkan sanksi, sesuai dengan aturan yang berlaku di Laboratorium Terpadu Fakultas Ilmu Komputer UMI apabila :

1. Praktikan merusak peralatan praktikum (*Personal Computer*) secara sengaja, maka praktikan bertanggung jawab untuk mengganti kerusakan tersebut.
2. Praktikan tidak mematuhi dan mentaati aturan praktikum maka tidak diperkenankan mengikuti praktikum.

Pelanggaran point lainnya dikenakan sanksi teguran, dikeluarkan/dicoret namanya dalam kegiatan praktikum (mengulang mata kuliah sesuai dengan semester berjalan) sampai sanksi akademik.



Kepala Laboratorium Terpadu,

Ir. Abdul Rachman Manga', S.Kom., M.T., MTA., MCF

DAFTAR ISI

KATA PENGANTAR	2
TATA TERTIB PELAKSANAAN PRAKTIKUM.....	3
DAFTAR ISI.....	5
MODUL 1 – DDL Tingkat Lanjut.....	6
MODUL 2 – DML TINGKAT LANJUT	11
MODUL 3 – TRIGGER.....	15
MODUL 4 – SQL ROUTINES.....	21

MODUL 1 – DDL Tingkat Lanjut

1. Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa dapat menggunakan SQL tingkat lanjut untuk DDL (Data Definition Language)

2. Sub Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa mampu mengimplementasikan penggunaan SQL Constraint:

- PRIMARY KEY
- FOREIGN KEY
- UNIQUE KEY

3. Teori Dasar

a. Primary Key

```
CREATE TABLE nama_table (
    ...,
    PRIMARY KEY(field1, field2, ...)
);
```

b. Foreign Key

```
CREATE TABLE nama_table (
    ...,
    CONSTRAINT nama_ForeignKey
    FOREIGN KEY (nama_field)
    REFERENCES nama_tabel_referensi (nama_field) EVENT ACTION
);
```

Event:

- ON DELETE
- ON UPDATE

Action

- RESTRICT
- NO ACTION
- CASCADE
- SET NULL

Ketika Event dikosongkan, otomatis action yang digunakan adalah RESTRICT.

c. Unique Constraint

```
CREATE TABLE nama_table (
    ...,
    UNIQUE(field1, field2, ...)
);
```

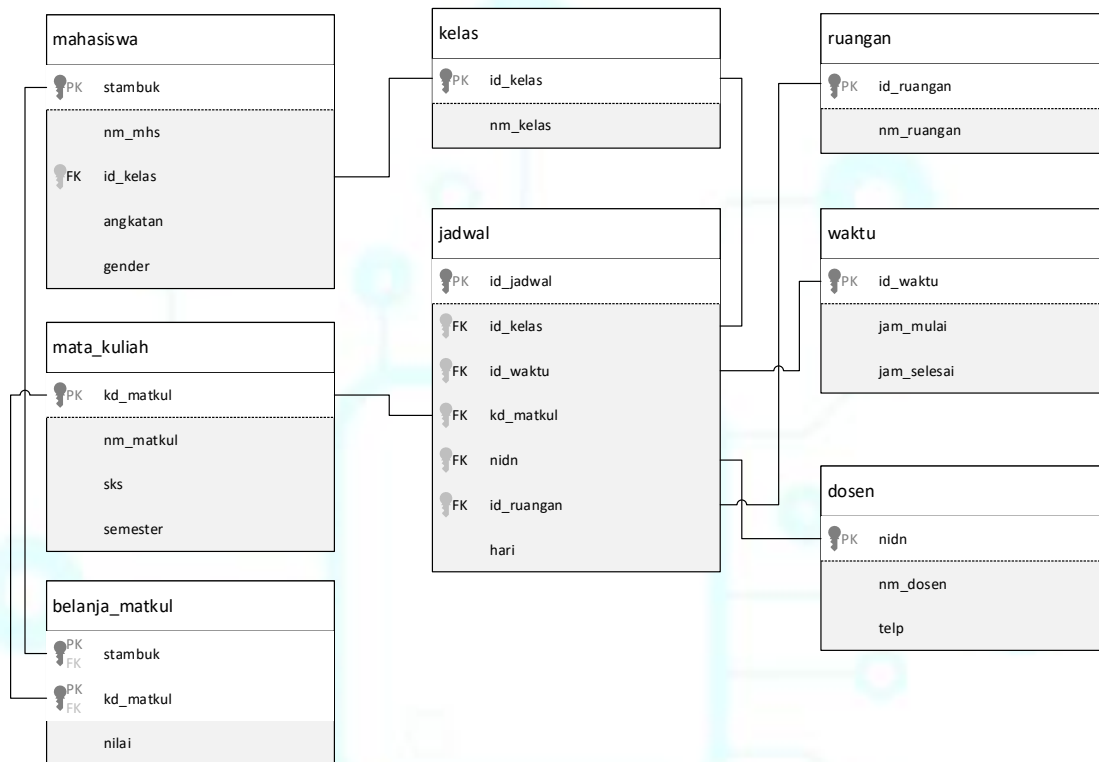
4. Kegiatan Praktikum

a. Instrument

- 1) Perangkat Komputer
- 2) Sistem Operasi Windows / Linux / OSX
- 3) MySQL Server
- 4) MySQL Workbench

b. Prosedur

- 1) Baca dan pahami semua tahapan praktikum dengan cermat.
- 2) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- 3) Rapihkan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- 4) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain.
- 5) Pastikan diri anda tidak menyentuh sumber listrik.

c. Studi Kasus

- 1) Buat database dengan nama database, **db_akademik_stambuk**. Stambuk disesuaikan dengan stambuk masing-masing mahasiswa.
- 2) Berdasarkan rancangan skema database di atas, implementasikan pembuatan tabel-tabelnya pada database yang dibuat pada langkah sebelumnya, lengkap dengan *constraint foreign key* dan *unique* pada masing-masing tabel tersebut.
- 3) Set *UNIQUE constraint* untuk field:
 - nm_kelas pada tabel kelas
 - telp pada tabel dosen
- 4) Set untuk setiap *foreign key* memiliki *event action* ON DELETE RESTRICT dan ON UPDATE CASCADE

5) Isi data dari masing-masing tabel sesuai dengan data di bawah ini:

a) Tabel kelas

id_kelas	nm_kelas
1	A1
2	B1
3	C1

b) Tabel ruangan

id_ruangan	nm_ruangan
1	U-3.01
2	U-3.02
3	U-3.03

c) Tabel waktu

id_waktu	jam_mulai	jam_selesai
1	07:00:00	09:40:00
2	09:40:00	12:10:00
3	13:00:00	15:30:00
4	15:40:00	18:10:00

d) Tabel dosen

nidn	nm_dosen	telp
0031056905	Dr. Ir. Hj. Harlinda L., MM.,M.Kom.,MTA.	08111111111
0428077401	Dr. Ir. Dolly Indra, S.Kom.,M.MSi.,MTA.	08222222222
0901019302	Dewi Widyawati, S.Kom., M.Kom.	08333333333

e) Tabel mata_kuliah

kd_matkul	nm_matkul	sks	semester
1303KKA713	Pemrograman Mobile	3	5
1303PPA105	Algoritma dan Pemrograman 1	3	1
1313KKB309	Basis Data II	3	3

f) Tabel mahasiswa

stambuk	nm_mhs	id_kelas	angkatan	gender
13020190096	A. ANUGRAH AQSA	3	2019	Laki-laki
13020220009	ANDIKA PRATAMA	1	2022	Laki-laki
13020220014	RAIHAN APRIANSYAH	1	2022	Laki-laki
13120210009	RISMA	2	2021	Perempuan
13120210011	NURHALISA	2	2021	Perempuan

g) Tabel jadwal

	id_jadwal	id_kelas	id_waktu	kd_matkul	nidn	id_ruangan	hari
▶	1	3	4	1303KKA713	0428077401	3	Senin
	2	1	1	1303PPA105	0031056905	1	Senin
	3	2	2	1313KKB309	0901019302	1	Selasa

h) Tabel belanja_matkul

	stambuk	kd_matkul	nilai
▶	13020190096	1303KKA713	87.5
	13020220009	1303PPA105	66
	13020220014	1303PPA105	91.2
	13120210009	1313KKB309	89.8
	13120210011	1313KKB309	92.5

LEMBAR EVALUASI PRAKTIKUM

Evaluasi Praktikum 1 :

No	Indikator	Skor Penilaian				
		Sangat Kurang (E) ≤ 40	Kurang (D) 41-55	Cukup (C) 55-65	Baik (B) 66-85	Sangat Baik (A) ≥ 86
1.						
2.						
3.						
4.						

Catatan Asisten :

Dosen : _____

Asisten 1 : _____

Asisten 2 : _____

TINGKAT KEBERHASILAN

MODUL 2 – DML TINGKAT LANJUT

1. Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa dapat menggunakan DML tingkat lanjut.

2. Sub Capaian Pembelajaran Mata Kuliah (CPMK)

- Mahasiswa mampu menggunakan keyword GROUP BY, diikuti dengan penguunaan fungsi agregat.
- Mahasiswa mampu menggunakan klausa HAVING untuk pengkondisian hasil fungsi agregat.
- Mahasiswa mampu mengimplementasikan penggunaan SQL JOIN:
 - INNER JOIN
 - LEFT JOIN
 - RIGHT JOIN
 - FULL OUTER JOIN

3. Teori Dasar

a. GROUP BY

Digunakan bersamaan dengan fungsi agregat (count, min, max, sum, avg) untuk mengelompokkan *result-set/record* yang dihasilkan.

Syntax:

```
select kolom1, kolom2, ...
from nama_tabel
where kondisi
group by nama_kolom
order by kolom1, kolom2, ...;
```

b. SQL HAVING Clause

Keyword WHERE tidak bisa digunakan jika kondisi yang diberikan menggunakan fungsi agregat. Maka untuk mengatasinya, klausa/keyword HAVING harus digunakan. Keyword HAVING digunakan jika terdapat fungsi agregat pada kondisi yg diberikan.

Syntax:

```
select kolom1, kolom2, ...
from nama_tabel
where kondisi
group by nama_kolom
having kondisi
order by kolom1, kolom2, ...;
```

c. INNER JOIN

Keyword INNER JOIN digunakan untuk memilih record yang memiliki nilai yang sama pada kedua tabel.

Syntax:

```
select kolom1, kolom2, ...
from tabel1
inner join tabel2 on tabel1.nama_kolom = tabel2.nama_kolom;
```

d. LEFT JOIN

LEFT JOIN mengembalikan nilai dari semua record pada Tabel Kiri (tabel1), dan record yang memiliki nilai yang sama dari kedua tabel.

Syntax:

```
select kolom1, kolom2, ...
from tabel1
left join tabel2 on tabel1.nama_kolom = tabel2.nama_kolom;
```

e. RIGHT JOIN

RIGHT JOIN mengembalikan nilai dari semua record pada Tabel Kanan (tabel2), dan record yang memiliki nilai yang sama dari kedua tabel.

Syntax:

```
select kolom1, kolom2, ...
from tabel1
right join tabel2 on tabel1.nama_kolom = tabel2.nama_kolom;
```

f. FULL OUTER JOIN

Kombinasi dari LEFT JOIN dan RIGHT JOIN, mengembalikan semua record pada kedua tabel.

Syntax:

```
select kolom1, kolom2, ...
from tabel1
left join tabel2 on tabel1.nama_kolom = tabel2.nama_kolom union
select kolom1, kolom2, ...
from tabel1
right join tabel2 on tabel1.nama_kolom = tabel2.nama_kolom;
```

4. Kegiatan Praktikum**a. Instrument**

- 1) Perangkat Komputer
- 2) Sistem Operasi Windows / Linux / OSX
- 3) MySQL Server
- 4) MySQL Workbench

b. Prosedur

- 1) Baca dan pahami semua tahapan praktikum dengan cermat.
- 2) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- 3) Rapihkan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- 4) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain.
- 5) Pastikan diri anda tidak menyentuh sumber listrik.

c. Studi Kasus

Berdasarkan tabel-tabel yang telah dibuat pada praktikum sebelumnya, tuliskan query untuk:

- 1) Tampilkan nidn, nama, dan telp dosen yang mengajar mata kuliah Pemrograman Mobile.
- 2) Tampilkan jumlah mahasiswa berdasarkan angkatan.

- 3) Tampilkan nama mata kuliah, nama mahasiswa dan stambuk mahasiswa yang nilai mata kuliahnya di bawah 70.
- 4) Tampilkan nama matakuliah dan nama dosen yang mengajar pada hari senin pukul 07:00.
- 5) Tampilkan nama ruangan yang kosong pada hari senin pukul 07:00.
- 6) Tampilkan nama ruangan yang sama sekali belum ditempati untuk perkuliahan.
- 7) Tampilkan stambuk dan nama mahasiswa yang memiliki nilai tertinggi pada matakuliah Basis Data II.
- 8) Tampilkan data jadwal perkuliahan secara lengkap terdiri dari field:
 - hari
 - jam_mulai
 - jam_selesai
 - nm_matakuliah
 - nm_kelas
 - nm_ruangan
 - nm_dosen

LEMBAR EVALUASI PRAKTIKUM

Evaluasi Praktikum 2 :

No	Indikator	Skor Penilaian				
		Sangat Kurang (E) ≤ 40	Kurang (D) 41-55	Cukup (C) 55-65	Baik (B) 66-85	Sangat Baik (A) ≥ 86
1.						
2.						
3.						
4.						

Catatan Asisten :

Dosen : _____

Asisten 1 : _____

Asisten 2 : _____

TINGKAT KEBERHASILAN

MODUL 3 – TRIGGER

1. Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa dapat membuat Trigger pada sebuah tabel

2. Sub Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa mampu mengimplementasikan penggunaan event trigger meliputi:

- BEFORE dan AFTER INSERT
- BEFORE dan AFTER UPDATE
- BEFORE dan AFTER DELETE

3. Teori Dasar

Trigger adalah sekumpulan kode program yang terdiri dari statemen prosedural dan deklarasi yang tersimpan dalam katalog dan diaktifkan oleh server database pada event tertentu.

Event yang dapat memicu, sbb :

- Penambahan data baru (INSERT)
- Update data (UPDATE)
- Penghapusan (DELETE)

Dalam trigger kita bisa menggunakan :

- OLD.nama kolom → mengembalikan nama kolom lama sebelum terjadi perubahan / penghapusan.
- NEW.nama kolom → mengembalikan nama kolom baru setelah terjadi pemasukan/perubahan.
- Pada proses INSERT, hanya NEW row saja yang tersedia.
- Pada proses UPDATE, tersedia NEW dan OLD row.
- Pada proses DELETE, hanya OLD row saja yang tersedia.

Syntax:

```
DELIMITER ##
CREATE TRIGGER nama_trigger
BEFORE/AFTER nama_event ON nama_tabel
FOR EACH ROW
BEGIN
    -- deklarasi variabel (jika diperlukan)
    DECLARE [nama_variabel] [tipe data];

    -- SQL SYNTAX/STATEMENT(S);
END##
delimiter ;
```

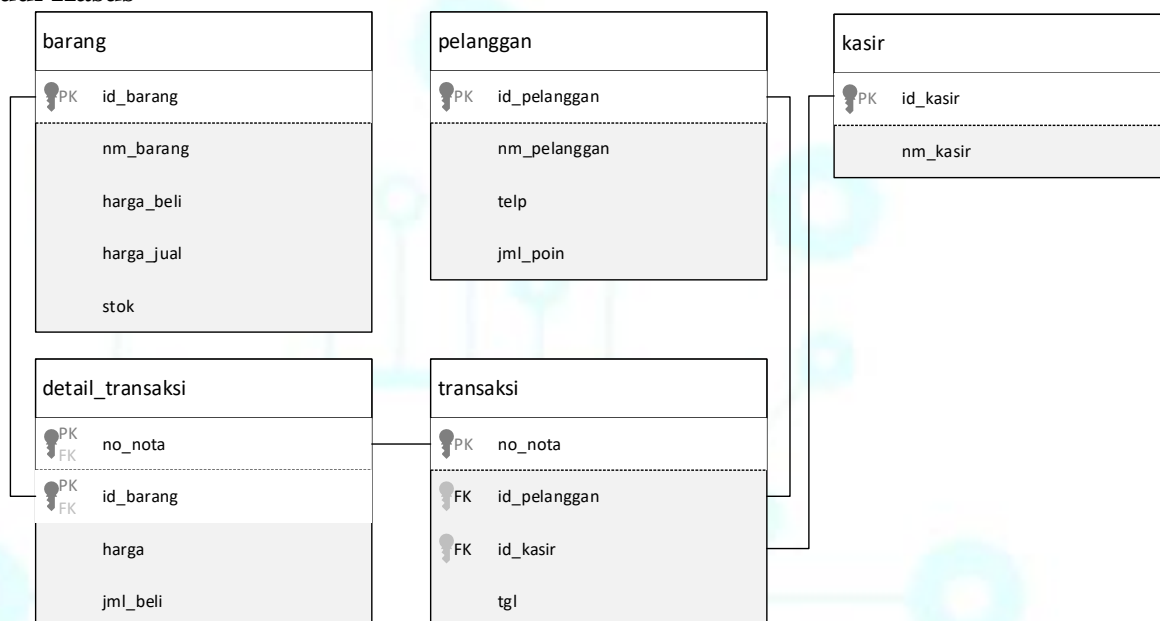
4. Kegiatan Praktikum

a. Instrument

- 1) Perangkat Komputer
- 2) Sistem Operasi Windows / Linux / OSX
- 3) MySQL Server
- 4) MySQL Workbench

b. Prosedur

- 1) Baca dan pahami semua tahapan praktikum dengan cermat.
- 2) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- 3) Rapihkan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- 4) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain.
- 5) Pastikan diri anda tidak menyentuh sumber listrik.

c. Studi Kasus

- 1) Implementasi dari rancangan skema *database* di atas, dapat [diakses pada halaman ini](#).
- 2) Lakukan langkah berikut untuk melakukan restore database:
 - a) Buka MySQL Workbench
 - b) Create database db_penjualan
 - c) Klik menu **Server** > pilih **Data Import**
 - d) Klik **Import from Self-Contained File** > pilih file db_penjualan.sql yang telah di-download pada tahap sebelumnya
 - e) Pada dropdown **Default Target Schema**, pilih db_penjualan
 - f) Klik **Start Import**
- 3) Isi data sesuai pada tabel di bawah:
 - Tabel barang

id_barang	nm_barang	harga_beli	harga_jual	stok
1	Gula	2000	2500	100
2	Beras	5000	5700	100
3	Terigu	2500	3000	100
4	Mie Instan	2000	2550	100
5	Teh	1000	1500	100
6	Kopi	1200	1700	100
7	Susu	3000	3600	100

- Tabel kasir

id_kasir	nm_kasir
1	Andi
2	Hasan

- Tabel pelanggan

id_pelanggan	nm_pelanggan	telp	jml_poin
1	Budi	11111	0
2	Ilham	22222	0
3	Fatur	33333	0
4	Nasrullah	44444	0

- 4) Buat trigger dengan nama `kurangi_stok`, **setelah** proses **insert** dilakukan pada tabel detail_transaksi, maka stok pada tabel barang akan berkurang sesuai dengan jumlah barang yang dibeli (field jml_beli).
- 5) Buat trigger dengan nama `update_stok`, **setelah** proses **update** dilakukan pada tabel detail_transaksi, maka stok pada tabel barang akan disesuaikan sesuai dengan nilai update pada field jml_beli.
- 6) Buat trigger dengan nama `tambah_stok`, **setelah** proses **delete** dilakukan pada tabel detail_transaksi, maka stok pada tabel barang akan bertambah sesuai dengan nilai field jml_beli pada row yang dihapus.
- 7) Buat trigger dengan nama `update_harga`, **sebelum** proses **insert** dilakukan pada tabel detail_transaksi, maka nilai dari field harga pada row yang akan diinsert (new.harga) sesuai dengan nilai dari field harga_beli pada tabel barang.
- 8) Buat trigger dengan nama `update_poin`, **setelah** proses **insert** selesai dilakukan pada tabel transaksi, jml_poin pada tabel pelanggan akan bertambah 100 poin. Jadi setiap transaksi yang dilakukan, akan menambah nilai poin pada pelanggan sebesar 100 poin.
- 9) Eksekusi proses insert berikut:

```
INSERT INTO transaksi (no_nota, id_pelanggan, id_kasir, tgl) VALUES ('N0001', 1, 1, '2022-12-04');
INSERT INTO detail_transaksi (no_nota, id_barang, jml_beli) VALUES ('N0001', 1, 3);
INSERT INTO detail_transaksi (no_nota, id_barang, jml_beli) VALUES ('N0001', 2, 2);
INSERT INTO detail_transaksi (no_nota, id_barang, jml_beli) VALUES ('N0001', 3, 1);
```

Cek isi data dari tabel detail_transaksi, tabel barang, dan tabel pelanggan datanya harus sesuai dengan tabel di bawah ini:

- Tabel detail_transaksi

no_nota	id_barang	harga	jml_beli
N0001	1	2500	3
N0001	2	5700	2
N0001	3	3000	1

- Tabel barang

id_barang	nm_barang	harga_beli	harga_jual	stok
1	Gula	2000	2500	97
2	Beras	5000	5700	98
3	Terigu	2500	3000	99
4	Mie Instan	2000	2550	100
5	Teh	1000	1500	100
6	Kopi	1200	1700	100
7	Susu	3000	3600	100

- Tabel Pelanggan

id_pelanggan	nm_pelanggan	telp	jml_poin
1	Budi	11111	100
2	Ilham	22222	0
3	Fatur	33333	0
4	Nasrullah	44444	0

10) Eksekusi proses update di bawah ini:

```
UPDATE detail_transaksi SET jml_beli = 2 WHERE no_nota = 'N0001' AND id_barang = 1;
```

```
UPDATE detail_transaksi SET jml_beli = 5 WHERE no_nota = 'N0001' AND id_barang = 2;
```

Cek isi data dari tabel barang datanya harus sesuai dengan tabel di bawah ini:

id_barang	nm_barang	harga_beli	harga_jual	stok
1	Gula	2000	2500	98
2	Beras	5000	5700	95
3	Terigu	2500	3000	99
4	Mie Instan	2000	2550	100
5	Teh	1000	1500	100
6	Kopi	1200	1700	100
7	Susu	3000	3600	100

11) Eksekusi proses delete di bawah ini:

```
DELETE FROM detail_transaksi WHERE no_nota = 'N0001' AND id_barang = 3
```

Cek isi data dari tabel barang datanya harus sesuai dengan tabel di bawah ini:

id_barang	nm_barang	harga_beli	harga_jual	stok
1	Gula	2000	2500	98
2	Beras	5000	5700	95
3	Terigu	2500	3000	100
4	Mie Instan	2000	2550	100
5	Teh	1000	1500	100
6	Kopi	1200	1700	100
7	Susu	3000	3600	100

LEMBAR EVALUASI PRAKTIKUM

Evaluasi Praktikum 3 :

No	Indikator	Skor Penilaian				
		Sangat Kurang (E) ≤ 40	Kurang (D) 41-55	Cukup (C) 55-65	Baik (B) 66-85	Sangat Baik (A) ≥ 86
1.						
2.						
3.						
4.						

Catatan Asisten :

Dosen : _____

Asisten 1 : _____

Asisten 2 : _____

TINGKAT KEBERHASILAN

MODUL 4 – SQL ROUTINES

1. Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa dapat mengimplementasikan pembuatan SQL Routines.

2. Sub Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa mampu mengimplementasikan PEMBUATAN SQL Routines meliputi:

- Stored Procedure
- Stored Function

3. Teori Dasar

a. Stored Procedure

- Stored procedure adalah kumpulan dari statemen SQL yang digunakan secara bersama-sama.
- Stored procedure mengizinkan penggunaan statemen SQL lebih dari sekedar statemen SQL biasa untuk meretrieve dan mengupdate database.
- Stored procedure mendukung penggunaan variable, kondisi, flow control dan cursor.
- Stored procedure dapat terdiri dari paling tidak satu statemen sampai dengan ratusan/ribuan baris.
- Dapat memiliki parameter input dan parameter output

Syntax:

```
CREATE PROCEDURE nama_procedure(
    IN/OUT parameter1 tipe_data,
    IN/OUT parameter2 tipe_data, ...)
BEGIN
    SqlStatement1;
    SqlStatement2;
    ...;
END
```

b. Stored Function

- Dapat memiliki parameter input, tetapi tidak memiliki parameter output
- Stored function tersebut adalah parameter outputnya
- Setelah berhasil dibuat maka dapat dipanggil seperti memanggil fungsi-fungsi standar
- Stored fuction harus memiliki statement RETURN

Syntax:

```
CREATE FUNCTION nama_function
    (parameter1 tipe_data, parameter2 tipe_data, ...)
RETURNS tipe_data
BEGIN
    SqlStatement1;
    SqlStatement2;
    ...;
    RETURN nilai;
END
```

4. Kegiatan Praktikum

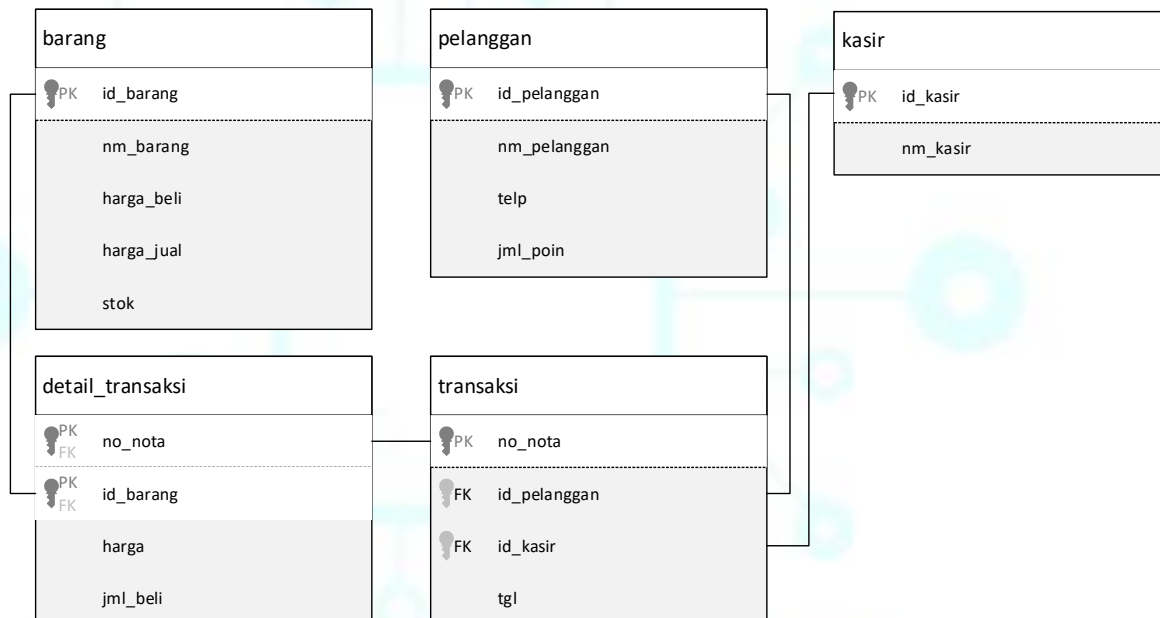
a. Instrument

- 1) Perangkat Komputer
- 2) Sistem Operasi Windows / Linux / OSX
- 3) MySQL Server
- 4) MySQL Workbench

b. Prosedur

- 1) Baca dan pahami semua tahapan praktikum dengan cermat.
- 2) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- 3) Rapihkan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- 4) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain.
- 5) Pastikan diri anda tidak menyentuh sumber listrik.

c. Studi Kasus



- 1) Lanjutkan database yang telah dibuat sebelumnya pada modul 3.
- 2) Buat stored procedure untuk melakukan insert data-data transaksi dengan nama ``insert_data_transaksi()``, data diinsert pada tabel **transaksi** dan **detail_transaksi**. Parameter yang digunakan pada procedure tersebut adalah sbb:
 - noNota CHAR(5)
 - idPelanggan INT
 - idKasir INT
 - tglTransaksi DATE
 - idBarang INT
 - jmlBeli INT
- 3) Buat stored function dengan nama ``hitungTotal()`` untuk menghitung total bayar pada setiap transaksi, parameter dari function tersebut adalah **no_nota**.

- 4) Buat stored function dengan nama ``getJmlTransaksi()`` untuk menghitung jumlah transaksi yang telah dilayani per kasirnya. Parameter yang diterima adalah `id_kasir`.
- 5) Buat stored procedure dengan nama ``tampilDataTransaksi()``, untuk menampilkan data transaksi secara detail dengan menerima parameter berupa `no_nota`. Data yang ditampilkan terdiri dari field:
 - `no_nota`
 - `nm_pelanggan`
 - `nm_kasir`
 - `tgl`
 - `nama_barang`
 - `harga`
 - `sub_total`
 - `grand_total` (boleh menggunakan function `hitungTotal(noNota)` yang dibuat sebelumnya)

LEMBAR EVALUASI PRAKTIKUM

Evaluasi Praktikum 4 :

No	Indikator	Skor Penilaian				
		Sangat Kurang (E) ≤ 40	Kurang (D) 41-55	Cukup (C) 55-65	Baik (B) 66-85	Sangat Baik (A) ≥ 86
1.						
2.						
3.						
4.						

Catatan Asisten :

Dosen : _____

Asisten 1 : _____

Asisten 2 : _____

TINGKAT KEBERHASILAN