

IF2240 - Basis Data

Tugas Besar

Milestone 3: *Integrity Constraints*



Disusun Oleh:

The Dragon Spirit of Lirili Larila

**Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung**

2025

IDENTITAS KELOMPOK

Nomor Kelompok : 4

Kode Kelompok : DRG

Nama Kelompok : The Dragon Spirit of Lirili Larila

Anggota Kelompok : 1. Muhammad Raihan Nazhim Oktana
2. Nathan Jovial Hartono
3. Muhammad Rayhan Farrukh
4. Abrar Abhirama Widyadhana
5. Aloisius Adrian Stevan Gunawan

Kelas : K1

Dosen Pengampu : Tricya Esterina Widagdo, S.T., M.Sc.

Problem Set : Patra

Nama Asisten : Daniel Mulia Putra Manurung

Tanggal Pengumpulan : 31 Mei 2025

1. FUNCTIONAL DEPENDENCIES\

Tabel	Audio (1)
Daftar Atribut	<ul style="list-style-type: none">• id_konten• durasi• kualitas
Daftar FD	id_konten → durasi, kualitas

Tabel	Creator (2)
Daftar Atribut	<ul style="list-style-type: none">• id_creator• nama• email• bidang_kreasi• deskripsi• tanggal_bergabung
Daftar FD	id_creator → nama, email, bidang_kreasi, deskripsi, tanggal_bergabung

Tabel	Gambar (3)
Daftar Atribut	<ul style="list-style-type: none">• id_gambar• id_konten• format• resolusi
Daftar FD	id_gambar → id_konten, format, resolusi

Tabel	Informasi_Hasil (4)
Daftar Atribut	<ul style="list-style-type: none">• id_gambar• id_spesial• status_pengerjaan• tanggal_penyelesaian• feedback
Daftar FD	id_gambar, id_spesial → status_pengerjaan, tanggal_penyelesaian, feedback

Tabel	Inspirasi (5)
Daftar Atribut	<ul style="list-style-type: none"> • id_publikasi • id_merchandise
Daftar FD	Tidak ada <i>non-key attribute</i> (trivial)

Tabel	Komentar (6)
Daftar Atribut	<ul style="list-style-type: none"> • id_komentar • id_supporter • id_konten • waktu • isi_komentar
Daftar FD	id_komentar → id_supporter, id_konten, waktu, isi_komentar

Tabel	Konten (7)
Daftar Atribut	<ul style="list-style-type: none"> • id_konten • nama_bidang • judul • deskripsi • tanggal_publikasi • jenis
Daftar FD	id_konten → nama_bidang, judul, deskripsi, tanggal_publikasi, jenis

Tabel	Manfaat (8)
Daftar Atribut	<ul style="list-style-type: none"> • id_tier • id_konten
Daftar FD	Tidak ada <i>non-key attribute</i> (trivial)

Tabel	Merchandise (9)
Daftar Atribut	<ul style="list-style-type: none"> • id_merchandise • nama • harga • stok

	<ul style="list-style-type: none"> • deskripsi
Daftar FD	id_merchandise → nama, harga, stok, deskripsi

Tabel	Publikasi (10)
Daftar Atribut	<ul style="list-style-type: none"> • id_publikasi • id_creator • id_konten
Daftar FD	id_publikasi → id_creator, id_konten

Tabel	Special_Content (11)
Daftar Atribut	<ul style="list-style-type: none"> • id_spesial • id_supporter • id_creator • judul • deskripsi • harga_dasar • tanggal_batas_revisi • batas_waktu_pengerjaan
Daftar FD	id_spesial → id_supporter, id_creator, judul, deskripsi, harga_dasar, tanggal_batas_revisi, batas_waktu_pengerjaan

Tabel	Struk_Langganan (12)
Daftar Atribut	<ul style="list-style-type: none"> • id_supporter • id_tier • tanggal_mulai • status • metode_pembayaran • jumlah • tanggal_pembayaran_terakhir
Daftar FD	id_supporter, id_tier, tanggal_mulai → status, metode_pembayaran, jumlah, tanggal_pembayaran_terakhir

Tabel	Struk_Pembelian (13)
Daftar Atribut	<ul style="list-style-type: none"> • id_supporter

	<ul style="list-style-type: none"> • id_merchandise • jumlah • tanggal_pembelian • total_harga • metode_pembayaran
Daftar FD	id_supporter, id_merchandise, tanggal_pembelian → jumlah, total_harga, metode_pembayaran

Tabel	Supporter (14)
Daftar Atribut	<ul style="list-style-type: none"> • id_supporter • nama • email • alamat • tanggal_bergabung
Daftar FD	id_supporter → nama, email, alamat, tanggal_bergabung

Tabel	Teks (15)
Daftar Atribut	<ul style="list-style-type: none"> • id_konten • jumlah_kata • format
Daftar FD	id_konten → jumlah_kata, format

Tabel	Tier (16)
Daftar Atribut	<ul style="list-style-type: none"> • id_tier • id_creator • nama_tier • deskripsi • harga
Daftar FD	id_tier → id_creator, nama_tier, deskripsi, harga

Tabel	Video (17)
Daftar Atribut	<ul style="list-style-type: none"> • id_konten • durasi • resolusi

Daftar FD	id_konten → durasi, resolusi
-----------	------------------------------

2. ***NORMAL FORM***

Tabel	Audio (1)
Bentuk Normal Form	BCNF (Boyce-Codd Normal Form)
Penjelasan	1NF: Terpenuhi (semua relasi adalah 1NF : atomik) 2NF: Terpenuhi karena Primary Key bukan composite key, sehingga tidak ada partial dependency 3NF: Terpenuhi karena semua atribut non-key bergantung langsung pada Primary Key. BCNF : id_creator adalah candidate key

Tabel	Creator (2)
Bentuk Normal Form	BCNF (Boyce-Codd Normal Form)
Penjelasan	1NF: Terpenuhi (semua relasi adalah 1NF : atomik). 2NF: Terpenuhi karena Primary Key (id_creator) bukan composite key, sehingga tidak ada partial dependency. 3NF: Terpenuhi karena semua atribut Non-key (nama, email, bidang_kreasi, deskripsi, tanggal_bergabung) bergantung langsung pada Primary Key (id_creator). BCNF: id_creator adalah candidate key.

Tabel	Gambar (3)
Bentuk Normal Form	BCNF (Boyce-Codd Normal Form)
Penjelasan	1NF: Terpenuhi (semua relasi adalah 1NF : atomik). 2NF: Terpenuhi karena Primary Key (id_gambar) bukan composite key, sehingga tidak ada partial dependency. 3NF: Terpenuhi karena semua atribut Non-key (id_konten, format, resolusi) bergantung langsung pada Primary Key (id_gambar). BCNF: id_gambar adalah candidate key.

Tabel	Informasi_Hasil (4)
Bentuk Normal Form	BCNF (Boyce-Codd Normal Form)
Penjelasan	<p>1NF: Terpenuhi (semua relasi adalah 1NF : atomik).</p> <p>2NF: Terpenuhi karena semua atribut Non-key (status_pengerjaan, tanggal_penyelesaian, feedback) bergantung penuh pada keseluruhan Primary Key (id_gambar, id_spesial).</p> <p>3NF: Terpenuhi karena semua atribut Non-key (status_pengerjaan, tanggal_penyelesaian, feedback) bergantung langsung pada Primary Key dan tidak ada dependensi transitif antar atribut Non-key.</p> <p>BCNF: (id_gambar, id_spesial) adalah candidate key.</p>

Tabel	Inspirasi (5)
Bentuk Normal Form	BCNF (Boyce-Codd Normal Form)
Penjelasan	<p>1NF: Terpenuhi (semua relasi adalah 1NF : atomik).</p> <p>2NF: Terpenuhi karena tidak ada atribut Non-key yang dapat bergantung sebagian pada Primary Key.</p> <p>3NF: Terpenuhi karena tidak ada atribut Non-key yang dapat memiliki dependensi transitif.</p> <p>BCNF: Semua atribut (id_publikasi, id_merchandise) adalah bagian dari candidate key.</p>

Tabel	Komentar (6)
Bentuk Normal Form	BCNF (Boyce-Codd Normal Form)
Penjelasan	<p>1NF: Terpenuhi (semua relasi adalah 1NF : atomik).</p> <p>2NF: Terpenuhi karena Primary Key (id_komentar) bukan composite key, sehingga tidak ada partial dependency.</p> <p>3NF: Terpenuhi karena semua atribut Non-key (id_supporter, id_konten, waktu, isi_komentar) bergantung langsung pada Primary Key (id_komentar).</p> <p>BCNF: id_komentar adalah candidate key.</p>

Tabel	Konten (7)
-------	------------

Bentuk Normal Form	BCNF (Boyce-Codd Normal Form)
Penjelasan	<p>1NF: Terpenuhi (semua relasi adalah 1NF : atomik).</p> <p>2NF: Terpenuhi karena Primary Key (id_konten) bukan composite key, sehingga tidak ada partial dependency.</p> <p>3NF: Terpenuhi karena semua atribut Non-key (nama_bidang, judul, deskripsi, tanggal_publicasi, jenis) bergantung langsung pada Primary Key (id_konten).</p> <p>BCNF: id_konten adalah candidate key.</p>

Tabel	Manfaat (8)
Bentuk Normal Form	BCNF (Boyce-Codd Normal Form)
Penjelasan	<p>1NF: Terpenuhi (semua relasi adalah 1NF : atomik).</p> <p>2NF: Terpenuhi karena tidak ada atribut Non-key yang dapat bergantung sebagian pada Primary Key.</p> <p>3NF: Terpenuhi karena tidak ada atribut Non-key yang dapat memiliki dependensi transitif.</p> <p>BCNF: Semua atribut (id_tier, id_konten) adalah bagian dari candidate key.</p>

Tabel	Merchandise (9)
Bentuk Normal Form	BCNF (Boyce-Codd Normal Form)
Penjelasan	<p>1NF: Terpenuhi (semua relasi adalah 1NF : atomik).</p> <p>2NF: Terpenuhi karena Primary Key (id_merchandise) bukan composite key, sehingga tidak ada partial dependency.</p> <p>3NF: Terpenuhi karena semua atribut Non-key (nama, harga, stok, deskripsi) bergantung langsung pada Primary Key (id_merchandise).</p> <p>BCNF: id_merchandise adalah candidate key.</p>

Tabel	Publikasi (10)
Bentuk Normal Form	BCNF (Boyce-Codd Normal Form)
Penjelasan	1NF: Terpenuhi (semua relasi adalah 1NF : atomik).

	<p>2NF: Terpenuhi karena Primary Key (id_publikasi) bukan composite key, sehingga tidak ada partial dependency.</p> <p>3NF: Terpenuhi karena semua atribut Non-key (id_creator, id_konten) bergantung langsung pada Primary Key (id_publikasi).</p> <p>BCNF: id_publikasi adalah candidate key.</p>
--	---

Tabel	Special_Content (11)
Bentuk Normal Form	BCNF (Boyce-Codd Normal Form)
Penjelasan	<p>1NF: Terpenuhi (semua relasi adalah 1NF : atomik).</p> <p>2NF: Terpenuhi karena Primary Key (id_spesial) bukan composite key, sehingga tidak ada partial dependency.</p> <p>3NF: Terpenuhi karena semua atribut Non-key (id_supporter, id_creator, judul, deskripsi, harga_dasar, tanggal_batas_revisi, batas_waktu_pengerjaan) bergantung langsung pada Primary Key (id_spesial).</p> <p>BCNF: id_spesial adalah candidate key.</p>

Tabel	Struk_Langganan (12)
Bentuk Normal Form	BCNF (Boyce-Codd Normal Form)
Penjelasan	<p>1NF: Terpenuhi (semua relasi adalah 1NF : atomik).</p> <p>2NF: Terpenuhi karena semua atribut Non-key (status, metode_pembayaran, jumlah, tanggal_pembayaran_terakhir) bergantung penuh pada keseluruhan Primary Key (id_supporter, id_tier, tanggal_mulai).</p> <p>3NF: Terpenuhi karena semua atribut Non-key (status, metode_pembayaran, jumlah, tanggal_pembayaran_terakhir) bergantung langsung pada Primary Key dan tidak ada dependensi transitif antar atribut Non-key.</p> <p>BCNF: (id_supporter, id_tier, tanggal_mulai) adalah candidate key.</p>

Tabel	Struk_Pembelian (13)
Bentuk Normal Form	BCNF (Boyce-Codd Normal Form)

Penjelasan	<p>1NF: Terpenuhi (semua relasi adalah 1NF : atomik).</p> <p>2NF: Terpenuhi karena semua atribut Non-key (jumlah, total_harga, metode_pembayaran) bergantung penuh pada keseluruhan Primary Key (id_supporter, id_merchandise, tanggal_pembelian).</p> <p>3NF: Terpenuhi karena semua atribut Non-key (jumlah, total_harga, metode_pembayaran) bergantung langsung pada Primary Key dan tidak ada dependensi transitif antar atribut Non-key.</p> <p>BCNF: (id_supporter, id_merchandise, tanggal_pembelian) adalah candidate key.</p>
------------	--

Tabel	Supporter (14)
Bentuk Normal Form	BCNF (Boyce-Codd Normal Form)
Penjelasan	<p>1NF: Terpenuhi (semua relasi adalah 1NF : atomik).</p> <p>2NF: Terpenuhi karena Primary Key (id_supporter) bukan composite key, sehingga tidak ada partial dependency.</p> <p>3NF: Terpenuhi karena semua atribut Non-key (nama, email, alamat, tanggal_bergabung) bergantung langsung pada Primary Key (id_supporter).</p> <p>BCNF: id_supporter adalah candidate key.</p>

Tabel	Teks (15)
Bentuk Normal Form	BCNF (Boyce-Codd Normal Form)
Penjelasan	<p>1NF: Terpenuhi (semua relasi adalah 1NF : atomik).</p> <p>2NF: Terpenuhi karena Primary Key (id_konten) bukan composite key, sehingga tidak ada partial dependency.</p> <p>3NF: Terpenuhi karena semua atribut Non-key (jumlah_kata, format) bergantung langsung pada Primary Key (id_konten).</p> <p>BCNF: id_konten adalah candidate key.</p>

Tabel	Tier (16)
Bentuk Normal Form	BCNF (Boyce-Codd Normal Form)
Penjelasan	1NF: Terpenuhi (semua relasi adalah 1NF : atomik).

	2NF: Terpenuhi karena Primary Key (id_tier) bukan composite key, sehingga tidak ada partial dependency. 3NF: Terpenuhi karena semua atribut Non-key (id_creator, nama_tier, deskripsi, harga) bergantung langsung pada Primary Key (id_tier). BCNF: id_tier adalah candidate key.
--	---

Tabel	Video (17)
Bentuk Normal Form	BCNF (Boyce-Codd Normal Form)
Penjelasan	1NF: Terpenuhi (semua relasi adalah 1NF : atomik). 2NF: Terpenuhi karena Primary Key (id_konten) bukan composite key, sehingga tidak ada partial dependency. 3NF: Terpenuhi karena semua atribut Non-key (durasi, resolusi) bergantung langsung pada Primary Key (id_konten). BCNF: id_konten adalah candidate key.

3. IMPLEMENTASI *INTEGRITY CONSTRAINTS*

a. *Type Constraints*

- i. Pada tabel “Konten” atribut “jenis” bertipe JenisKonten. JenisKonten as ENUM ('Teks', 'Gambar', 'Audio', 'Video').
- ii. Semua tabel yang memiliki atribut primary key bertipe POSINTEGER. POSINTEGER adalah integer > 0.

```
CREATE TYPE JenisKonten AS ENUM ('Teks', 'Gambar',
'Audio', 'Video');
-> tidak bisa diimplementasikan di mariaDB karena tidak
ada fitur CREATE TYPE (hanya di postgres)

CREATE DOMAIN POSINTEGER AS INTEGER
CHECK (VALUE > 0);
-> tidak bisa diimplementasikan di mariaDB karena MySQL
tidak ada fitur CREATE DOMAIN (hanya di postgres)
```

b. *Attribute Constraints*

- i. Batasan nilai untuk atribut Harga pada tabel Merchandise dan Tier Menggunakan CHECK constraints pada tabel yang relevan untuk atribut yang dibatasi

```

ALTER TABLE Merchandise
ADD CONSTRAINT CK_Merchandise_Harga CHECK (harga <=
1500000);

ALTER TABLE Tier
ADD CONSTRAINT CK_Tier_Harga CHECK (harga <= 1000000);

```

- ii. Batasan panjang string maksimal 255 karakter untuk atribut “Nama” pada tabel “Creator” dan tabel “Supporter” dan Not Null.

```

ALTER TABLE Creator
MODIFY COLUMN nama VARCHAR(255) NOT NULL;

ALTER TABLE Supporter
MODIFY COLUMN nama VARCHAR(255) NOT NULL;

```

c. *Relation Constraints*

- i. Sebuah konten kreator hanya dapat memiliki maksimal 5 tier unik. Pembuatan trigger BEFORE INSERT sehingga sebelum sebuah tuple pada relasi Tier ditambahkan, akan dihitung id_creator yang terhubung pada tuple tersebut, apakah jumlah tuple Tier-nya melebihi jumlah batasan (5)

- ii. Tanggal batas revisi harus sebelum tanggal batas waktu pengerjaan pada relasi Special_Content

Penambahan CHECK constraint pada relasi Special_Content untuk memastikan constraint tersebut terpenuhi.

```

DELIMITER $$
CREATE TRIGGER check_max_5_tiers
BEFORE INSERT ON Tier
FOR EACH ROW
BEGIN
    DECLARE tier_count INT;
    SELECT COUNT(*) INTO tier_count
    FROM Tier
    WHERE id_creator = NEW.id_creator;
    IF tier_count >= 5 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Kreator hanya bisa memiliki
maksimal 5 tier.';
    END IF;
END $$

```

```

DELIMITER ;

ALTER TABLE Special_Content
ADD CONSTRAINT chk_special_content_revision_deadline
CHECK (tanggal_batas_revisi <= batas_waktu_pengerjaan);

```

d. *Database Constraints*

- i. Atribut jumlah pada Struk_Langganan harus sesuai atribut harga pada relasi Tier

Menambahkan trigger BEFORE INSERT pada tabel Struk_Langganan trigger ini akan memastikan bahwa sebelum ditambahkan data baru, akan dicek apakah atribut jumlah sesuai dengan atribut harga pada Tier yang bersesuaian

- ii. Pada relasi manfaat, id konten yang ada haruslah konten yang dibuat atau dimiliki oleh id kreator yang ada.

Menambahkan trigger BEFORE INSERT yang mengecek *foreign key-foreign key* yang ada di relasi Manfaat sesuai dengan attribut-attribut pada relasi Kreator dan relasi Konten.

```

DELIMITER $$
CREATE TRIGGER trg_check_jumlah_sesuai_harga_tier_insert
BEFORE INSERT ON Struk_Langganan
FOR EACH ROW
BEGIN
    DECLARE v_harga_tier INT;

    SELECT harga INTO v_harga_tier
    FROM Tier
    WHERE id_tier = NEW.id_tier;

    IF NEW.jumlah <> v_harga_tier THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Jumlah pembayaran pada struk
langganan tidak sesuai dengan harga tier.';
    END IF;
END $$
DELIMITER ;

DELIMITER $$
CREATE TRIGGER
trg_manfaat_check_content_creator_before_insert
BEFORE INSERT ON Manfaat

```

```

FOR EACH ROW
BEGIN
    DECLARE v_tier_creator_id INT;
    DECLARE v_content_is_owned_by_tier_creator INT
    DEFAULT 0;

    SELECT T.id_creator INTO v_tier_creator_id
    FROM Tier T
    WHERE T.id_tier = NEW.id_tier;

    IF v_tier_creator_id IS NOT NULL THEN
        SELECT 1 INTO v_content_is_owned_by_tier_creator
        FROM Publikasi P
        WHERE P.id_konten = NEW.id_konten AND
        P.id_creator = v_tier_creator_id
        LIMIT 1;
    END IF;

    IF v_content_is_owned_by_tier_creator = 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Konten yang dijadikan manfaat
        harus dibuat oleh kreator pemilik tier tersebut.';
    END IF;
END $$
DELIMITER ;

```

e. *Transition Constraints*

i. Alur status Commision

Menggunakan trigger BEFORE UPDATE yang melakukan validasi terhadap update pada relasi Informasi_Hasil sehingga setiap kali update, akan dicek apakah atribut status_pengerjaan berubah, dan jika berubah, apakah sesuai dengan aturan yang ada

ii. Alur status pada relasi Struk_Langgan

Menggunakan trigger BEFORE UPDATE juga pada relasi Struk_Langgan untuk memastikan bahwa status aktif hanya akan berubah menjadi “berakhir

```

DELIMITER $$
CREATE TRIGGER TRG_Informasi_Hasil_Status_Transition
BEFORE UPDATE ON Informasi_Hasil
FOR EACH ROW
BEGIN
    IF OLD.status_pengerjaan = 'menunggu_persetujuan' AND

```

```

NEW.status_pengerjaan NOT IN ('disetujui', 'ditolak')
THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Transisi dari
menunggu_persetujuan salah. Ke: disetujui/ditolak.';
    ELSEIF OLD.status_pengerjaan = 'disetujui' AND
NEW.status_pengerjaan <> 'dalam_pengerjaan' THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Transisi dari disetujui
salah. Ke: dalam_pengerjaan.';
    ELSEIF OLD.status_pengerjaan = 'dalam_pengerjaan' AND
NEW.status_pengerjaan <> 'selesai' THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Transisi dari
dalam_pengerjaan salah. Ke: selesai.';
    ELSEIF (OLD.status_pengerjaan = 'selesai' AND
NEW.status_pengerjaan <> OLD.status_pengerjaan)
OR
(OLD.status_pengerjaan = 'ditolak' AND
NEW.status_pengerjaan <> OLD.status_pengerjaan)
THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Status final
(selesai/ditolak), tak bisa diubah.';
    ELSEIF OLD.status_pengerjaan = 'menunggu_persetujuan'
AND
NEW.status_pengerjaan = 'dalam_pengerjaan' THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Lompatan status salah:
menunggu_persetujuan ke dalam_pengerjaan.';
    ELSEIF OLD.status_pengerjaan = 'menunggu_persetujuan'
AND
NEW.status_pengerjaan = 'selesai' THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Lompatan status salah:
menunggu_persetujuan ke selesai.';
    ELSEIF OLD.status_pengerjaan = 'disetujui' AND
NEW.status_pengerjaan = 'selesai' THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Lompatan status salah:
disetujui ke selesai.';
    END IF;
END $$
DELIMITER ;

```

f. *Referential Actions*

i. Penghapusan entry Tier

Menghapus sebuah entry pada tabel Tier akan menghapus entry pada tabel Struk_Langganan dan tabel Manfaat yang berhubungan dengan primary key entry pada Tier.

```
-- Untuk tabel Manfaat
ALTER TABLE Manfaat
DROP FOREIGN KEY Manfaat_ibfk_1;

ALTER TABLE Manfaat
ADD CONSTRAINT Manfaat_ibfk_1
FOREIGN KEY (id_tier) REFERENCES Tier(id_tier)
ON DELETE CASCADE;

-- Untuk tabel Struk_Langganan
ALTER TABLE Struk_Langganan
DROP FOREIGN KEY Struk_Langganan_ibfk_2;

ALTER TABLE Struk_Langganan
ADD CONSTRAINT Struk_Langganan_ibfk_2
FOREIGN KEY (id_tier) REFERENCES Tier(id_tier)
ON DELETE CASCADE;
```

ii. Penghapusan entry Merchandise

Menghapus sebuah entry pada tabel Merchandise akan menghapus entry pada tabel Struk_Pembelian dan tabel Inspirasi yang berhubungan dengan primary key entry pada Merchandise.

```
-- Untuk tabel Inspirasi
ALTER TABLE Inspirasi
DROP FOREIGN KEY Inspirasi_ibfk_2;

ALTER TABLE Inspirasi
ADD CONSTRAINT Inspirasi_ibfk_2
FOREIGN KEY (id_merchandise) REFERENCES
Merchandise(id_merchandise)
ON DELETE CASCADE;

-- Untuk tabel Struk_Pembelian
ALTER TABLE Struk_Pembelian
DROP FOREIGN KEY Struk_Pembelian_ibfk_2;

ALTER TABLE Struk_Pembelian
ADD CONSTRAINT Struk_Pembelian_ibfk_2
FOREIGN KEY (id_merchandise) REFERENCES
```

```
Merchandise(id_merchandise)
ON DELETE CASCADE;
```

4. FITUR TAMBAHAN

a. Otomatisasi Perhitungan Harga Merchandise

Tujuan	Mengotomatisasi perhitungan atribut total_harga pada Tabel Struk_Pembelian untuk mengatasi kesalahan input manual dengan membuat prosedur untuk kalkulasi ulang berdasarkan harga dan jumlah item, serta trigger yang mengisi dan memperbarui total harga saat penambahan data, pengurangan data, atau perubahan data. Dalam kasus ini, input data diasumsikan sudah valid, atau di-handle dengan nilai 0 pada beberapa kasus invalid.
Query Testing	<p>1) General Query Testing :</p> <pre>SELECT Supporter.nama AS Nama_Supporter , Merchandise.nama AS Nama_Merchandise , Struk_Pembelian.jumlah AS Jumlah , Merchandise.harga AS Harga_Satuan , Struk_Pembelian.total_harga AS Total_Harga FROM Supporter , Struk_Pembelian , Merchandise WHERE Supporter.id_supporter = Struk_Pembelian.id_supporter AND Struk_Pembelian.id_merchandise = Merchandise.id_merchandise ORDER BY Supporter.nama ASC , Struk_Pembelian.total_harga ASC , Merchandise.harga ASC , Struk_Pembelian.jumlah ASC , Merchandise.nama ASC;</pre> <hr/> <p>—</p> <p>2) Query Testing Trigger 1 :</p> <pre>INSERT INTO Supporter (id_supporter , nama) VALUES (120 , "Dummy Tester") ON DUPLICATE KEY UPDATE nama = "Dummy Tester";</pre> <hr/> <p>—</p> <pre>INSERT INTO Struk_Pembelian (id_supporter , id_merchandise , jumlah, tanggal_pembelian , total_harga , metode_pembayaran) VALUES (120 , 58 , 3 , CURDATE() , 0 , "QRIS");</pre> <hr/>

```

-
INSERT INTO Struk_Pembelian (
    id_supporter , id_merchandise , jumlah,
    tanggal_pembelian , total_harga ,
    metode_pembayaran
) VALUES (
    120 , 59 , 1 , CURDATE() , 0 , "Debit"
);

-
INSERT INTO Struk_Pembelian (
    id_supporter , id_merchandise , jumlah,
    tanggal_pembelian , total_harga ,
    metode_pembayaran
) VALUES (
    120 , 60 , -1 , CURDATE() , 0 , "Credit"
);

-
SELECT Supporter.nama AS Nama_Supporter ,
Merchandise.nama AS Nama_Merchandise ,
Struk_Pembelian.jumlah AS Jumlah ,
Merchandise.harga AS Harga_Satuan ,
Struk_Pembelian.total_harga AS Total_Harga
FROM Supporter , Struk_Pembelian , Merchandise
WHERE Supporter.id_supporter =
Struk_Pembelian.id_supporter AND
Struk_Pembelian.id_merchandise =
Merchandise.id_merchandise AND
Supporter.id_supporter = 120
ORDER BY Supporter.nama ASC ,
Struk_Pembelian.total_harga ASC ,
Merchandise.harga ASC , Struk_Pembelian.jumlah
ASC , Merchandise.nama ASC;

-
3) Query Testing Trigger 2 :
UPDATE Struk_Pembelian
SET jumlah = jumlah - 5
WHERE id_supporter = 120 AND id_merchandise =
58;

-
UPDATE Struk_Pembelian
SET jumlah = jumlah + 3
WHERE id_supporter = 120 AND id_merchandise =
59;

-
UPDATE Struk_Pembelian
SET jumlah = jumlah - 1

```

```
WHERE id_supporter = 120 AND id_merchandise = 60;
```

```
--  
SELECT Supporter.nama AS Nama_Supporter ,  
Merchandise.nama AS Nama_Merchandise ,  
Struk_Pembelian.jumlah AS Jumlah ,  
Merchandise.harga AS Harga_Satuan ,  
Struk_Pembelian.total_harga AS Total_Harga  
FROM Supporter , Struk_Pembelian , Merchandise  
WHERE Supporter.id_supporter =  
Struk_Pembelian.id_supporter AND  
Struk_Pembelian.id_merchandise =  
Merchandise.id_merchandise AND  
Supporter.id_supporter = 120  
ORDER BY Supporter.nama ASC ,  
Struk_Pembelian.total_harga ASC ,  
Merchandise.harga ASC , Struk_Pembelian.jumlah  
ASC , Merchandise.nama ASC;
```

```
--  
4) Query Testing Trigger 3 :
```

```
UPDATE Merchandise  
SET harga = harga + 50000  
WHERE id_merchandise = 58;
```

```
--  
UPDATE Merchandise  
SET harga = harga + 60000  
WHERE id_merchandise = 59;
```

```
--  
UPDATE Merchandise  
SET harga = harga + 70000  
WHERE id_merchandise = 60;
```

```
--  
UPDATE Struk_Pembelian  
SET jumlah = jumlah + 1  
WHERE id_supporter = 120 AND id_merchandise = 58;
```

```
--  
UPDATE Struk_Pembelian  
SET jumlah = jumlah - 4  
WHERE id_supporter = 120 AND id_merchandise = 59;
```

```
--  
UPDATE Struk_Pembelian  
SET jumlah = jumlah + 9
```

	<pre> WHERE id_supporter = 120 AND id_merchandise = 60; -- SELECT Supporter.nama AS Nama_Supporter , Merchandise.nama AS Nama_Merchandise , Struk_Pembelian.jumlah AS Jumlah , Merchandise.harga AS Harga_Satuan , Struk_Pembelian.total_harga AS Total_Harga FROM Supporter , Struk_Pembelian , Merchandise WHERE Supporter.id_supporter = Struk_Pembelian.id_supporter AND Struk_Pembelian.id_merchandise = Merchandise.id_merchandise AND Supporter.id_supporter = 120 ORDER BY Supporter.nama ASC , Struk_Pembelian.total_harga ASC , Merchandise.harga ASC , Struk_Pembelian.jumlah ASC , Merchandise.nama ASC; </pre>
Query Manipulasi	<pre> 1) Prosedur Hitung_Total_Harga : DELIMITER \$\$ CREATE PROCEDURE Hitung_Total_Harga (IN pid_supporter INT, IN pid_merch INT) BEGIN DECLARE harga INT; DECLARE jumlah INT; SELECT m.harga INTO harga FROM Merchandise m WHERE m.id_merchandise = pid_merch; SELECT s.jumlah INTO jumlah FROM Struk_Pembelian s WHERE s.id_supporter = pid_supporter AND s.id_merchandise = pid_merch; UPDATE Struk_Pembelian SET total_harga = harga * jumlah WHERE id_supporter = pid_supporter AND id_merchandise = pid_merch; END \$\$ DELIMITER ; -- 2) Trigger 1 (Before Insert Struk_Pembelian) : DELIMITER \$\$ CREATE TRIGGER SP_Before_Insert_Merchandise BEFORE INSERT ON Struk_Pembelian FOR EACH ROW BEGIN DECLARE cnt INT; </pre>

```

        IF (NEW.jumlah <= 0) THEN
            SET NEW.jumlah = 0;
            SET NEW.total_harga = 0;
        ELSE
            SELECT harga INTO cnt
            FROM Merchandise
            WHERE id_merchandise =
NEW.id_merchandise;
            SET NEW.total_harga = NEW.jumlah *
cnt;
        END IF;
    END $$
DELIMITER ;

```

—
3) Trigger 2 (Before Update Struk_Pembelian) :

```

DELIMITER $$
CREATE TRIGGER SP_Before_Update_Jumlah
BEFORE UPDATE ON Struk_Pembelian
FOR EACH ROW
BEGIN
    DECLARE cnt INT;
    IF (NEW.jumlah <= 0) THEN
        SET NEW.jumlah = 0;
        SET NEW.total_harga = 0;
    ELSE
        SELECT harga INTO cnt
        FROM Merchandise
        WHERE id_merchandise =
NEW.id_merchandise;
        SET NEW.total_harga = NEW.jumlah *
cnt;
    END IF;
END $$
DELIMITER ;

```

—
4) Trigger 3 (After Update Merchandise) :

```

DELIMITER $$
CREATE TRIGGER Merchandise_After_Update_Harga
AFTER UPDATE ON Merchandise
FOR EACH ROW
BEGIN
    DECLARE chk INT DEFAULT 0;
    DECLARE sid INT;
    DECLARE qty INT;
    DECLARE cur CURSOR FOR
        SELECT id_supporter , jumlah
        FROM Struk_Pembelian
        WHERE id_merchandise =
NEW.id_merchandise;

```

```

DECLARE CONTINUE HANDLER FOR NOT FOUND SET
chk = 1;
OPEN cur;
read_loop : LOOP
    FETCH cur INTO sid , qty;
    IF chk THEN
        LEAVE read_loop;
    END IF;
    UPDATE Struk_Pembelian
    SET total_harga = qty * NEW.harga
    WHERE id_supporter = sid AND
    id_merchandise = NEW.id_merchandise;
    END LOOP;
CLOSE cur;
END $$
DELIMITER ;

```

Tangkapan Layar Sebelum Manipulasi

```

MariaDB [DRG]> SELECT Supporter.nama AS Nama_Supporter , Merchandise.nama AS Nama_Merchandise , Struk_Pembelian.jumlah AS Jumlah , Merchandise.h
arga AS Harga_Satuan , Struk_Pembelian.total_harga AS Total_Harga
-> FROM Supporter , Struk_Pembelian , Merchandise
-> WHERE Supporter.id_supporter = Struk_Pembelian.id_supporter AND Struk_Pembelian.id_merchandise = Merchandise.id_merchandise
-> ORDER BY Supporter.nama ASC , Struk_Pembelian.total_harga ASC , Merchandise.harga ASC , Struk_Pembelian.jumlah ASC , Merchandise.nama ASC;

```

Nama_Supporter	Nama_Merchandise	Jumlah	Harga_Satuan	Total_Harga
Asmadi Rahimah	Cumque Keychain	2	210000	420000
Banara Wastuti	Accusamus Mug	5	280000	1400000
Bella Prasetyo	Nesciunt Sticker Pack	1	500000	500000
Bella Prasetyo	Fugiat T-Shirt	4	300000	1200000
Cager Siregar	Reiciendis Keychain	5	460000	2300000
Cakrawala Laksita	Ex Sticker Pack	2	370000	740000
Cakrawala Laksita	Quos Sticker Pack	3	270000	810000
Candra Tampubolon	Vel T-Shirt	4	200000	800000
Candra Tampubolon	Fuga Sticker Pack	2	400000	800000
Candra Tampubolon	Cum Poster	4	220000	880000
Taswin Yolanda	Vitae Sticker Pack	2	500000	1000000
Tgk. Dewi Halimah	Fugiat T-Shirt	1	110000	110000
Tgk. Dewi Halimah	Cumque Keychain	3	210000	630000
Tgk. Gaman Majmudin, M.M.	Sinilique Keychain	2	420000	840000
Yoga Manullang	Qui Poster	1	50000	50000
Yoga Manullang	Minima Poster	1	490000	490000
Yoga Manullang	Magnam Sticker Pack	3	230000	690000
Yoga Manullang	Vitae Sticker Pack	2	500000	1000000
Zelda Mandasari	Ducimus Mug	5	280000	1400000
Zizi Sudiati	Minus Mug	3	290000	870000

100 rows in set (0.001 sec)

Tangkapan Layar Proses Manipulasi

1) Prosedur Hitung_Total_Harga :

```

MariaDB [DRG]> DELIMITER $$
MariaDB [DRG]> CREATE PROCEDURE Hitung_Total_Harga (
-> IN pid_supporter INT,
-> IN pid_merch INT
-> )
-> BEGIN
-> DECLARE harga INT;
-> DECLARE jumlah INT;
-> SELECT m.harga INTO harga
-> FROM Merchandise m
-> WHERE m.id_merchandise = pid_merch;
-> SELECT s.jumlah INTO jumlah
-> FROM Struk_Pembelian s
-> WHERE s.id_supporter = pid_supporter AND s.id_merchandise = pid_merch;
-> UPDATE Struk_Pembelian
-> SET total_harga = harga * jumlah
-> WHERE id_supporter = pid_supporter AND id_merchandise = pid_merch;
-> END $$
Query OK, 0 rows affected (0.011 sec)

```

2) Trigger 1 (Before Insert Struk_Pembelian) :

```

MariaDB [DRG]> DELIMITER $$
MariaDB [DRG]> CREATE TRIGGER SP_Before_Insert_Merchandise
-> BEFORE INSERT ON Struk_Pembelian
-> FOR EACH ROW
-> BEGIN
->     DECLARE cnt INT;
->     IF (NEW.jumlah <= 0) THEN
->         SET NEW.jumlah = 0;
->         SET NEW.total_harga = 0;
->     ELSE
->         SELECT harga INTO cnt
->         FROM Merchandise
->         WHERE id_merchandise = NEW.id_merchandise;
->         SET NEW.total_harga = NEW.jumlah * cnt;
->     END IF;
-> END $$
Query OK, 0 rows affected (0.019 sec)

```

3) Trigger 2 (Before Update Struk_Pembelian) :

```

MariaDB [DRG]> DELIMITER $$
MariaDB [DRG]> CREATE TRIGGER SP_Before_Update_Jumlah
-> BEFORE UPDATE ON Struk_Pembelian
-> FOR EACH ROW
-> BEGIN
->     DECLARE cnt INT;
->     IF (NEW.jumlah <= 0) THEN
->         SET NEW.jumlah = 0;
->         SET NEW.total_harga = 0;
->     ELSE
->         SELECT harga INTO cnt
->         FROM Merchandise
->         WHERE id_merchandise = NEW.id_merchandise;
->         SET NEW.total_harga = NEW.jumlah * cnt;
->     END IF;
-> END $$
Query OK, 0 rows affected (0.030 sec)

```

4) Trigger 3 (After Update Merchandise) :

```

MariaDB [DRG]> DELIMITER $$
MariaDB [DRG]> CREATE TRIGGER Merchandise_After_Update_Harga
-> AFTER UPDATE ON Merchandise
-> FOR EACH ROW
-> BEGIN
->     DECLARE chk INT DEFAULT 0;
->     DECLARE sid INT;
->     DECLARE qty INT;
->     DECLARE cur CURSOR FOR
->         SELECT id_supporter , jumlah
->         FROM Struk_Pembelian
->         WHERE id_merchandise = NEW.id_merchandise;
->     DECLARE CONTINUE HANDLER FOR NOT FOUND SET chk = 1;
->     OPEN cur;
->     read_loop : LOOP
->         FETCH cur INTO sid , qty;
->         IF chk THEN
->             LEAVE read_loop;
->         END IF;
->         UPDATE Struk_Pembelian
->         SET total_harga = qty * NEW.harga
->         WHERE id_supporter = sid AND id_merchandise = NEW.id_merchandise;
->     END LOOP;
->     CLOSE cur;
-> END $$
Query OK, 0 rows affected (0.017 sec)

```

Tangkapan Layar Setelah Manipulasi

1) General Query Testing :

```

MariaDB [DRG]> SELECT Supporter.nama AS Nama_Supporter , Merchandise.nama AS Nama_Merchandise , Struk_Pembelian.jumlah AS Jumlah , Merchandise.harga AS Harga_Satuan , Struk_Pembelian.total_harga AS Total_Harga
-> FROM Supporter , Struk_Pembelian , Merchandise WHERE Supporter.id_supporter = Struk_Pembelian.id_supporter AND Struk_Pembelian.id_merchandise = Merchandise.id_merchandise
-> ORDER BY Supporter.nama ASC , Struk_Pembelian.total_harga ASC , Merchandise.harga ASC , Struk_Pembelian.jumlah ASC , Merchandise.nama ASC;

```

Nama_Supporter	Nama_Merchandise	Jumlah	Harga_Satuan	Total_Harga
Asmadi Rahimah	Cumque Keychain	2	260000	520000
Banara Wastuti	Accusamus Mug	5	280000	1400000
Bella Prasetyo	Nesciunt Sticker Pack	1	500000	500000
Bella Prasetyo	Fugiat T-Shirt	4	300000	1200000
Cager Siregar	Reiciendis Keychain	5	460000	2300000
Cakrawala Laksita	Ex Sticker Pack	2	370000	740000
Cakrawala Laksita	Quos Sticker Pack	3	270000	810000
Candra Tampubolon	Vel T-Shirt	4	200000	800000
Candra Tampubolon	Fuga Sticker Pack	2	400000	800000
Candra Tampubolon	Cum Poster	4	220000	880000

Taswir Yolanda	Vitae Sticker Pack	2	560000	1120000
Tgk. Dewi Halimah	Fugiat T-Shirt	1	110000	110000
Tgk. Dewi Halimah	Cumque Keychain	3	260000	780000
Tgk. Gaman Najmudin, M.M.	Similique Keychain	2	420000	840000
Voga Manullang	Qui Poster	1	50000	50000
Voga Manullang	Minima Poster	1	490000	490000
Voga Manullang	Magnam Sticker Pack	3	300000	900000
Voga Manullang	Vitae Sticker Pack	2	560000	1120000
Zelda Mandasari	Ducimus Mug	5	280000	1400000
Zizi Sudiati	Minus Mug	3	290000	870000

103 rows in set (0.001 sec)

2) Test Case Trigger 1 :


```

MariaDB [DRG]> INSERT INTO Supporter (id_supporter, nama)
-> VALUES (120, "Dummy Tester");
-> ON DUPLICATE KEY UPDATE nama = "Dummy Tester";
Query OK, 1 row affected (0.004 sec)

MariaDB [DRG]> INSERT INTO Struk_Pembelian (
-> id_supporter, id_merchandise, jumlah,
-> tanggal_pembelian, total_harga,
-> metode_pembayaran
-> ) VALUES (
-> 120, 58, 3, CURDATE(), 0, "QRIS"
-> );
Query OK, 1 row affected (0.005 sec)

MariaDB [DRG]> INSERT INTO Struk_Pembelian (
-> id_supporter, id_merchandise, jumlah,
-> tanggal_pembelian, total_harga,
-> metode_pembayaran
-> ) VALUES (
-> 120, 59, 1, CURDATE(), 0, "Debit"
-> );
Query OK, 1 row affected (0.005 sec)

MariaDB [DRG]> INSERT INTO Struk_Pembelian (
-> id_supporter, id_merchandise, jumlah,
-> tanggal_pembelian, total_harga,
-> metode_pembayaran
-> ) VALUES (
-> 120, 60, -1, CURDATE(), 0, "Credit"
-> );
Query OK, 1 row affected (0.003 sec)

MariaDB [DRG]> SELECT Supporter.nama AS Nama_Supporter, Merchandise.nama AS Nama_Merchandise, Struk_Pembelian.jumlah AS Jumlah, Merchandise.harga AS Harga_Satuan, Struk_Pembelian.total_harga AS Total_Harga
-> FROM Supporter, Struk_Pembelian, Merchandise WHERE Supporter.id_supporter = Struk_Pembelian.id_supporter AND Struk_Pembelian.id_merchandise = Merchandise.id_merchandise AND Supporter.id_supporter = 120
-> ORDER BY Supporter.nama ASC, Struk_Pembelian.total_harga ASC, Merchandise.harga ASC, Struk_Pembelian.jumlah ASC, Merchandise.nama ASC;
+-----+-----+-----+-----+-----+
| Nama_Supporter | Nama_Merchandise | Jumlah | Harga_Satuan | Total_Harga |
+-----+-----+-----+-----+-----+
| Dummy Tester  | Magmam Sticker Pack | 0      | 220000       | 0           |
| Dummy Tester  | Vitae Sticker Pack | 1      | 500000       | 500000      |
| Dummy Tester  | Cumque Keychain    | 3      | 210000       | 630000      |
+-----+-----+-----+-----+-----+
3 rows in set (0.001 sec)

```

3) Test Case Trigger 2 :

```

MariaDB [DRG]> UPDATE Struk_Pembelian
-> SET jumlah = jumlah - 5
-> WHERE id_supporter = 120 AND id_merchandise = 58;
Query OK, 1 row affected (0.005 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [DRG]> UPDATE Struk_Pembelian
-> SET jumlah = jumlah + 3
-> WHERE id_supporter = 120 AND id_merchandise = 59;
Query OK, 1 row affected (0.005 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [DRG]> UPDATE Struk_Pembelian
-> SET jumlah = jumlah - 1
-> WHERE id_supporter = 120 AND id_merchandise = 60;
Query OK, 0 rows affected (0.001 sec)
Rows matched: 1 Changed: 0 Warnings: 0

MariaDB [DRG]> SELECT Supporter.nama AS Nama_Supporter, Merchandise.nama AS Nama_Merchandise, Struk_Pembelian.jumlah AS Jumlah, Merchandise.harga AS Harga_Satuan, Struk_Pembelian.total_harga AS Total_Harga
-> FROM Supporter, Struk_Pembelian, Merchandise WHERE Supporter.id_supporter = Struk_Pembelian.id_supporter AND Struk_Pembelian.id_merchandise = Merchandise.id_merchandise AND Supporter.id_supporter = 120
-> ORDER BY Supporter.nama ASC, Struk_Pembelian.total_harga ASC, Merchandise.harga ASC, Struk_Pembelian.jumlah ASC, Merchandise.nama ASC;
+-----+-----+-----+-----+-----+
| Nama_Supporter | Nama_Merchandise | Jumlah | Harga_Satuan | Total_Harga |
+-----+-----+-----+-----+-----+
| Dummy Tester  | Cumque Keychain    | 0      | 210000       | 0           |
| Dummy Tester  | Magmam Sticker Pack | 0      | 230000       | 0           |
| Dummy Tester  | Vitae Sticker Pack | 4      | 500000       | 2000000     |
+-----+-----+-----+-----+-----+
3 rows in set (0.001 sec)

```

4) Test Case Trigger 3 :

```

MariaDB [DRG]> UPDATE Merchandise
-> SET harga = harga + 50000
-> WHERE id_merchandise = 58;
Query OK, 1 row affected (0.006 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [DRG]> UPDATE Merchandise
-> SET harga = harga + 60000
-> WHERE id_merchandise = 59;
Query OK, 1 row affected (0.003 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [DRG]> UPDATE Merchandise
-> SET harga = harga + 70000
-> WHERE id_merchandise = 60;
Query OK, 1 row affected (0.005 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [DRG]> UPDATE Struk_Pembelian
-> SET jumlah = jumlah + 1
-> WHERE id_supporter = 120 AND id_merchandise = 58;
Query OK, 1 row affected (0.003 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [DRG]> UPDATE Struk_Pembelian
-> SET jumlah = jumlah - 4
-> WHERE id_supporter = 120 AND id_merchandise = 59;
Query OK, 1 row affected (0.005 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [DRG]> UPDATE Struk_Pembelian
-> SET jumlah = jumlah + 9
-> WHERE id_supporter = 120 AND id_merchandise = 60;
Query OK, 1 row affected (0.005 sec)
Rows matched: 1 Changed: 1 Warnings: 0

```

```

MariaDB [DRG]> SELECT Supporter.nama AS Nama_Supporter , Merchandise.nama AS Nama_Merchandise , Struk_Pembelian.jumlah AS Jumlah , Merchandise.harga
a AS Harga_Satuan , Struk_Pembelian.total_harga AS Total_Harga
-> FROM Supporter , Struk_Pembelian , Merchandise WHERE Supporter.id_supporter = Struk_Pembelian.id_supporter AND Struk_Pembelian.id_merchandise
= Merchandise.id_merchandise AND Supporter.id_supporter = 120
-> ORDER BY Supporter.nama ASC , Struk_Pembelian.total_harga ASC , Merchandise.harga ASC , Struk_Pembelian.jumlah ASC , Merchandise.nama ASC;
+-----+-----+-----+-----+-----+
| Nama_Supporter | Nama_Merchandise | Jumlah | Harga_Satuan | Total_Harga |
+-----+-----+-----+-----+-----+
| Dummy Tester  | Vitae Sticker Pack | 0      | 560000       | 0           |
| Dummy Tester  | Cumque Keychain   | 1      | 260000       | 260000      |
| Dummy Tester  | Magnam Sticker Pack | 9      | 300000       | 2700000     |
+-----+-----+-----+-----+-----+
3 rows in set (0.001 sec)

```

b. Data Kreator Populer

Tujuan	Menyediakan data 10 kreator paling populer untuk monitoring pada dashboard admin melalui sebuah view bernama KreatorPopuler. View ini akan menampilkan nama kreator, bidang kreasi, jumlah pendukung aktif, jumlah total komentar, dan skor popularitas yang dihitung berdasarkan kombinasi jumlah pendukung dan interaksi konten
Query Testing	SELECT * FROM KreatorPopuler;
Query Manipulasi	<pre> DELIMITER \$\$ CREATE VIEW KreatorPopuler AS WITH KreatorPendukungAktif AS (SELECT T.id_creator, COUNT(DISTINCT SL.id_supporter) AS jumlah_pendukung_aktif FROM Tier T JOIN Struk_Langganan SL ON T.id_tier = SL.id_tier WHERE SL.status = 'Aktif' GROUP BY T.id_creator), KreatorJumlahKomentar AS (SELECT P.id_creator, COUNT(K.id_komentar) AS jumlah_total_komentar FROM Publikasi P JOIN Komentar K ON P.id_konten = K.id_konten GROUP BY P.id_creator) SELECT C.nama AS nama_kreator, C.bidang_kreasi, COALESCE(KPA.jumlah_pendukung_aktif, 0) AS jumlah_pendukung_aktif, COALESCE(KJK.jumlah_total_komentar, 0) AS </pre>

```

jumlah_komentar,
    (COALESCE(KPA.jumlah_pendukung_aktif, 0) *
 3 + COALESCE(KJK.jumlah_total_komentar, 0)) AS
skor_popularitas
FROM Creator C
LEFT JOIN KreatorPendukungAktif KPA ON
C.id_creator = KPA.id_creator
LEFT JOIN KreatorJumlahKomentar KJK ON
C.id_creator = KJK.id_creator
ORDER BY skor_popularitas DESC
LIMIT 10;

$$

DELIMITER ;

```

Tangkapan Layar Sebelum Manipulasi

```

MariaDB [drg]> show tables;
+-----+
| Tables_in_drg |
+-----+
| audio          |
| creator        |
| gambar         |
| informasi_hasil |
| inspirasi      |
| komentar       |
| konten         |
| manfaat        |
| merchandise    |
| publikasi      |
| special_content |
| struk_langganan |
| struk_pembelian |
| supporter      |
| teks           |
| tier            |
| video          |
+-----+
17 rows in set (0.002 sec)

MariaDB [drg]>

```

```

MariaDB [drg]> SELECT * FROM KreatorPopuler;
ERROR 1146 (42S02): Table 'drg.kreatorpopuler' doesn't exist
MariaDB [drg]>

```

Tangkapan Layar Proses Manipulasi

```

MariaDB [DRG]> DELIMITER $$
MariaDB [DRG]>
MariaDB [DRG]> CREATE VIEW KreatorPopuler AS
-> WITH KreatorPendukungAktif AS (
->   SELECT T.id_creator,
->     COUNT(DISTINCT SL.id_supporter) AS jumlah_pendukung_aktif
->   FROM Tier T
->   JOIN Struk_Langganan SL ON T.id_tier = SL.id_tier
->   WHERE SL.status = 'Aktif'
->   GROUP BY T.id_creator
-> ),
-> KreatorJumlahKomentar AS (
->   SELECT P.id_creator,
->     COUNT(K.id_komentar) AS jumlah_total_komentar
->   FROM Publikasi P
->   JOIN Komentar K ON P.id_konten = K.id_konten
->   GROUP BY P.id_creator
-> )
-> SELECT
->   C.nama AS nama_kreator,
->   C.bidang_kreasi,
->   COALESCE(KPA.jumlah_pendukung_aktif, 0) AS jumlah_pendukung_aktif
-> ,
->   COALESCE(KJK.jumlah_total_komentar, 0) AS jumlah_komentar,
->   (COALESCE(KPA.jumlah_pendukung_aktif, 0) * 3 + COALESCE(KJK.jumlah_total_komentar, 0)) AS skor_popularitas
-> FROM Creator C
-> LEFT JOIN KreatorPendukungAktif KPA ON C.id_creator = KPA.id_creator
-> LEFT JOIN KreatorJumlahKomentar KJK ON C.id_creator = KJK.id_creator
-> ORDER BY skor_popularitas DESC
-> LIMIT 10;
-> $$
Query OK, 0 rows affected (0.015 sec)

MariaDB [DRG]>
MariaDB [DRG]> DELIMITER ;

```

Tangkapan Layar Setelah Manipulasi

```

MariaDB [drg]> SELECT * FROM kreatorpopuler
-> ;

```

nama_kreator	bidang_kreasi	jumlah_pendukung_aktif	jumlah_komentar	skor_popularitas
Gina Figrantoro	Podcast Komedi	3	0	9
Sutan Jail Marpaung	Animasi Pendek	3	0	9
Yulia Agustina	Review Teknologi	2	0	6
Queen Waluyo, M.M.	Seni Digital	2	0	6
Tgk. Gina Usamah	Review Teknologi	1	0	3
Dr. Marwata Narpati, S.Pd	Animasi Pendek	1	0	3
drg. Banawi Yuniar, S.IP	Musik Akustik	1	0	3
Salwa Santoso	Tutorial Masak	1	0	3
Hesti Sihombing	Seni Digital	1	0	3
Irwan Widodo	Animasi Pendek	1	0	3

```

10 rows in set (0.100 sec)

MariaDB [drg]>

```

c. Aktivitas Pendukung

Tujuan	Membangun sistem untuk mencatat berbagai aktivitas pembantu untuk analisis engagement, yang diimplementasikan dengan pembuatan tabel
--------	--

	<p>AktivitasPendukung beserta kolom-kolom yang ditentukan, dan tiga trigger yang otomatis mencatat aktivitas saat pendukung berlangganan tier, memberikan komentar, atau membeli merchandise.</p>
<p>Query Testing</p>	<p>1) Lihat semua aktivitas pendukung</p> <pre>SELECT * FROM Aktivitas Pendukung ORDER BY waktu_aktivitas DESC;</pre> <p>2) Lihat semua aktivitas pendukung</p> <pre>-- Semua langganan SELECT * FROM AktivitasPendukung WHERE jenis_aktivitas = 'Berlangganan'; -- Semua komentar SELECT * FROM AktivitasPendukung WHERE jenis_aktivitas = 'Komentar'; -- Semua pembelian SELECT * FROM AktivitasPendukung WHERE jenis_aktivitas = 'Pembelian';</pre>
<p>Query Manipulasi</p>	<p>1) Buat Tabel AktivitasPendukung</p> <pre>CREATE TABLE AktivitasPendukung (id_aktivitas INT AUTO_INCREMENT PRIMARY KEY, id_supporter INT NOT NULL, jenis_aktivitas VARCHAR(50) NOT NULL, waktu_aktivitas TIMESTAMP DEFAULT CURRENT_TIMESTAMP, keterangan TEXT, FOREIGN KEY (id_supporter) REFERENCES Supporter(id_supporter));</pre> <p>2) Trigger: Saat Pendukung Berlangganan Tier</p> <pre>DELIMITER \$\$ CREATE TRIGGER trg_log_langganan AFTER INSERT ON Struk_Langganan FOR EACH ROW BEGIN INSERT INTO AktivitasPendukung (id_supporter, jenis_aktivitas, keterangan) VALUES (NEW.id_supporter,</pre>

```

        'Berlangganan',
        CONCAT('Berlangganan tier ID: ',
NEW.id_tier, ', jumlah: ', NEW.jumlah)
    );
END $$

```

```

DELIMITER ;

```

3) Trigger: Saat Pendukung Memberi Komentar
DELIMITER \$\$

```

CREATE TRIGGER trg_log_komentar
AFTER INSERT ON Komentar
FOR EACH ROW
BEGIN
    INSERT INTO AktivitasPendukung
(id_supporter, jenis_aktivitas,
waktu_aktivitas, keterangan)
VALUES (
    NEW.id_supporter,
    'Komentar',
    NEW.waktu,
    CONCAT('Komentar di konten ID: ',
NEW.id_konten)
);
END $$

```

```

DELIMITER ;

```

4) Trigger: Saat Pendukung Membeli Merchandise
DELIMITER \$\$

```

CREATE TRIGGER trg_log_pembelian
AFTER INSERT ON Struk_Pembelian
FOR EACH ROW
BEGIN
    INSERT INTO AktivitasPendukung
(id_supporter, jenis_aktivitas, keterangan)
VALUES (
    NEW.id_supporter,
    'Pembelian',
    CONCAT('Membeli merchandise ID: ',
NEW.id_merchandise, ', jumlah: ', NEW.jumlah,
', total: ', NEW.total_harga)
);
END $$

```

```

DELIMITER ;

```

Tangkapan Layar Sebelum Manipulasi

```

Database changed
MariaDB [DRG]> SHOW TABLES;
+-----+
| Tables_in_DRG |
+-----+
| Audio          |
| Creator        |
| Gambar         |
| Informasi_Hasil |
| Inspirasi      |
| Komentar       |
| Konten         |
| Manfaat        |
| Merchandise    |
| Publikasi      |
| Special_Content |
| Struk_Langganan |
| Struk_Pembelian |
| Supporter      |
| Teks           |
| Tier           |
| Video          |
+-----+
17 rows in set (0.000 sec)

```

Tangkapan Layar Proses Manipulasi

```

MariaDB [DRG]> CREATE TABLE AktivitasPendukung (
->   id_aktivitas INT AUTO_INCREMENT PRIMARY KEY,
->   id_supporter INT NOT NULL,
->   jenis_aktivitas VARCHAR(50) NOT NULL,
->   waktu_aktivitas TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
->   keterangan TEXT,
->   FOREIGN KEY (id_supporter) REFERENCES Supporter(id_supporter)
-> );
Query OK, 0 rows affected (0.060 sec)

MariaDB [DRG]> DELIMITER $$
MariaDB [DRG]>
MariaDB [DRG]> CREATE TRIGGER trg_log_langganan
-> AFTER INSERT ON Struk_Langganan
-> FOR EACH ROW
-> BEGIN
->   INSERT INTO AktivitasPendukung (id_supporter, jenis_aktivitas, keterangan)
->   VALUES (
->     NEW.id_supporter,
->     'Berlangganan',
->     CONCAT('Berlangganan tier ID: ', NEW.id_tier, ', jumlah: ', NEW.jumlah)
->   );
-> END $$
Query OK, 0 rows affected (0.025 sec)

```

```

MariaDB [DRG]>
MariaDB [DRG]> DELIMITER ;
MariaDB [DRG]> DELIMITER $$
MariaDB [DRG]>
MariaDB [DRG]> CREATE TRIGGER trg_log_komentar
-> AFTER INSERT ON Komentar
-> FOR EACH ROW
-> BEGIN
->     INSERT INTO AktivitasPendukung (id_supporter, jenis_aktivitas, waktu_aktivitas, keterangan)
->     VALUES (
->         NEW.id_supporter,
->         'Komentar',
->         NEW.waktu,
->         CONCAT('Komentar di konten ID: ', NEW.id_konten)
->     );
-> END $$
Query OK, 0 rows affected (0.023 sec)

MariaDB [DRG]>
MariaDB [DRG]> DELIMITER ;
MariaDB [DRG]> DELIMITER $$
MariaDB [DRG]>
MariaDB [DRG]> CREATE TRIGGER trg_log_pembelian
-> AFTER INSERT ON Struk_Pembelian
-> FOR EACH ROW
-> BEGIN
->     INSERT INTO AktivitasPendukung (id_supporter, jenis_aktivitas, keterangan)
->     VALUES (
->         NEW.id_supporter,
->         'Pembelian',
->         CONCAT('Membeli merchandise ID: ', NEW.id_merchandise, ', jumlah: ', NEW.jumlah, ', total: ', NEW.total_harga)
->     );
-> END $$
Query OK, 0 rows affected (0.021 sec)

MariaDB [DRG]>
MariaDB [DRG]> DELIMITER ;

MariaDB [DRG]> INSERT INTO Struk_Langganan
-> (id_supporter, id_tier, tanggal_mulai, status, metode_pembayaran, jumlah, tanggal_pembayaran_terakhir)
-> VALUES
-> (1, 2, CURDATE(), 'aktif', 'transfer', 50000, CURDATE());
Query OK, 1 row affected (0.008 sec)

MariaDB [DRG]> INSERT INTO Komentar
-> (id_komentar, id_supporter, id_konten, isi_komentar)
-> VALUES
-> (101010, 1, 5, 'Komentar pengujian trigger');
Query OK, 1 row affected (0.004 sec)

MariaDB [DRG]> INSERT INTO Struk_Pembelian
-> (id_supporter, id_merchandise, jumlah, tanggal_pembelian, total_harga, metode_pembayaran)
-> VALUES
-> (1, 3, 2, CURDATE(), 150000, 'e-wallet');
Query OK, 1 row affected (0.004 sec)

```

Tangkapan Layar Setelah Manipulasi


```
MariaDB [DRG]> SHOW TABLES;
```

Tables_in_DRG				
AktivitasPendukung	Audio	Creator	Gambar	Informasi_Hasil
Inspirasi	Komentar	Konten	Manfaat	Merchandise
Publikasi	Special_Content	Struk_Langganan	Struk_Pembelian	Supporter
Teks	Tier	Video		

18 rows in set (0.002 sec)

```
MariaDB [DRG]> SELECT * FROM AktivitasPendukung WHERE id_supporter = 1;
```

id_aktivitas	id_supporter	jenis_aktivitas	waktu_aktivitas	keterangan
1	1	Berlangganan	2025-05-31 14:28:53	Berlangganan tier ID: 2, jumlah: 50000
2	1	Komentar	2025-05-31 14:29:26	Komentar di konten ID: 5
3	1	Pembelian	2025-05-31 14:29:29	Membeli merchandise ID: 3, jumlah: 2, total: 150000

3 rows in set (0.002 sec)

```
MariaDB [DRG]> SELECT * FROM AktivitasPendukung ORDER BY waktu_aktivitas DESC;
```

id_aktivitas	id_supporter	jenis_aktivitas	waktu_aktivitas	keterangan
3	1	Pembelian	2025-05-31 14:29:29	Membeli merchandise ID: 3, jumlah: 2, total: 150000
2	1	Komentar	2025-05-31 14:29:26	Komentar di konten ID: 5
1	1	Berlangganan	2025-05-31 14:28:53	Berlangganan tier ID: 2, jumlah: 50000

3 rows in set (0.001 sec)

```
MariaDB [DRG]> SELECT * FROM AktivitasPendukung WHERE jenis_aktivitas = 'Berlangganan';
```

id_aktivitas	id_supporter	jenis_aktivitas	waktu_aktivitas	keterangan
1	1	Berlangganan	2025-05-31 14:28:53	Berlangganan tier ID: 2, jumlah: 50000

1 row in set (0.000 sec)

```
MariaDB [DRG]> SELECT * FROM AktivitasPendukung WHERE jenis_aktivitas = 'Komentar';
```

id_aktivitas	id_supporter	jenis_aktivitas	waktu_aktivitas	keterangan
2	1	Komentar	2025-05-31 14:29:26	Komentar di konten ID: 5

1 row in set (0.000 sec)

```
MariaDB [DRG]> SELECT * FROM AktivitasPendukung WHERE jenis_aktivitas = 'Pembelian';
```

id_aktivitas	id_supporter	jenis_aktivitas	waktu_aktivitas	keterangan
3	1	Pembelian	2025-05-31 14:29:29	Membeli merchandise ID: 3, jumlah: 2, total: 150000

1 row in set (0.000 sec)

d. Validasi Email

Tujuan	Memastikan semua pengguna (Kreator dan Pendukung) memiliki alamat email yang valid sesuai format standar "<...>@<...>.com" (di mana <...> tidak mengandung '@') sebelum data disimpan, dengan mengimplementasikan dua trigger (satu untuk Kreator, satu untuk Pendukung) yang melakukan validasi otomatis saat data email dimasukkan atau diperbarui.
Query Testing	<p>1. Test insert email supporter valid</p> <pre>INSERT INTO Supporter (id_supporter, nama, email, alamat, tanggal_bergabung) VALUES (2001, 'Budi Valid', 'budi.valid@example.com', 'Jl. Valid No. 1', CURDATE());</pre> <pre>SELECT id_supporter, email FROM Supporter WHERE id_supporter = 2001;</pre> <p>2. Test insert email supporter tidak valid</p> <pre>INSERT INTO Supporter (id_supporter, nama, email, alamat, tanggal_bergabung) VALUES (2002, 'Cici Invalid', 'cici@invalid', 'Jl. Invalid No. 2', CURDATE());</pre> <pre>INSERT INTO Supporter (id_supporter, nama, email, alamat, tanggal_bergabung) VALUES (2003, 'Dedi Invalid', 'dedi@@example.com', 'Jl. Invalid No. 3', CURDATE());</pre> <pre>INSERT INTO Supporter (id_supporter, nama, email, alamat, tanggal_bergabung) VALUES (2004, 'Euis Invalid', 'euis@lokal@example.com', 'Jl. Invalid No. 4', CURDATE());</pre> <pre>SELECT id_supporter, email FROM Supporter WHERE id_supporter IN (2002, 2003, 2004);</pre> <p>3. Test update email supporter jadi tidak valid</p> <pre>UPDATE Supporter SET email = 'budi.salah.com' WHERE id_supporter = 2001;</pre> <pre>UPDATE Supporter SET email = 'budi@salah@domain.com' WHERE id_supporter = 2001;</pre> <pre>SELECT id_supporter, email FROM Supporter WHERE id_supporter = 2001;</pre> <p>4. Test update email supporter valid</p> <pre>UPDATE Supporter SET email = 'budi.baru.valid@domainkeren.com' WHERE id_supporter = 2001;</pre> <pre>SELECT id_supporter, email FROM Supporter WHERE id_supporter = 2001;</pre>

	<p>5. Test insert email creator valid</p> <pre>INSERT INTO Creator (id_creator, nama, email, bidang_kreasi, deskripsi, tanggal_bergabung) VALUES (3001, 'Ani Kreatif', 'ani.kreatif@patra.com', 'Seni Digital', 'Deskripsi Ani', CURDATE());</pre> <pre>SELECT id_creator, email FROM Creator WHERE id_creator = 3001;</pre> <p>6. Test insert email creator tidak valid</p> <pre>INSERT INTO Creator (id_creator, nama, email, bidang_kreasi, deskripsi, tanggal_bergabung) VALUES (3002, 'Banu Gagal', 'banu@gagalcom', 'Musik', 'Deskripsi Banu', CURDATE()); INSERT INTO Creator (id_creator, nama, email, bidang_kreasi, deskripsi, tanggal_bergabung) VALUES (3003, 'Citra Error', 'citra.error@domain@salah.com', 'Game Dev', 'Deskripsi Citra', CURDATE());</pre> <pre>SELECT id_creator, email FROM Creator WHERE id_creator IN (3002, 3003);</pre> <p>7. Test update email creator jadi tidak valid</p> <pre>UPDATE Creator SET email = 'ani@.com' WHERE id_creator = 3001; UPDATE Creator SET email = 'ani.kreatif@patra.net' WHERE id_creator = 3001;</pre> <pre>SELECT id_creator, email FROM Creator WHERE id_creator = 3001;</pre> <p>8. Test update email creator valid</p> <pre>UPDATE Creator SET email = 'ani.kreatif.baru@patracool.com' WHERE id_creator = 3001;</pre> <pre>SELECT id_creator, email FROM Creator WHERE id_creator = 3001;</pre>
Query Manipulasi	<p>1) Trigger 1 (Before Insert email creator) :</p> <pre>DELIMITER //</pre> <pre>CREATE TRIGGER trg_validasi_email_creator_insert BEFORE INSERT ON Creator FOR EACH ROW BEGIN IF NOT (NEW.email LIKE '_%@%.%' AND NEW.email LIKE '%.com' AND LOCATE('@', NEW.email, LOCATE('@', NEW.email) + 1) = 0 AND SUBSTRING_INDEX(SUBSTRING_INDEX(NEW.email, '@', 1), '@', -1) NOT LIKE '%@%') THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Format email kreator tidak valid.'; END IF; END //</pre>

```
DELIMITER ;
```

2) Trigger 2 (Before Update email creator) :

```
DELIMITER //
```

```
CREATE TRIGGER trg_validasi_email_creator_update
BEFORE UPDATE ON Creator
FOR EACH ROW
BEGIN
    IF NEW.email <> OLD.email THEN
        IF NOT (NEW.email LIKE '_%@_._%' AND
NEW.email LIKE '%.com' AND LOCATE('@', NEW.email,
LOCATE('@', NEW.email) + 1) = 0 AND
SUBSTRING_INDEX(SUBSTRING_INDEX(NEW.email, '@', 1),
 '@', -1) NOT LIKE '%@%') THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Format email kreator
tidak valid.';
        END IF;
    END IF;
END //
```

```
DELIMITER ;
```

3) Trigger 3 (Before Insert email supporter) :

```
DELIMITER //
```

```
CREATE TRIGGER trg_validasi_email_supporter_insert
BEFORE INSERT ON Supporter
FOR EACH ROW
BEGIN
    IF NOT (NEW.email LIKE '_%@_._%' AND NEW.email
LIKE '%.com' AND LOCATE('@', NEW.email, LOCATE('@',
NEW.email) + 1) = 0 AND
SUBSTRING_INDEX(SUBSTRING_INDEX(NEW.email, '@', 1),
 '@', -1) NOT LIKE '%@%') THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Format email pendukung
tidak valid.';
    END IF;
END //
```

```
DELIMITER ;
```

4) Trigger 4 (Before Update email supporter) :

```
DELIMITER //
```

```
CREATE TRIGGER trg_validasi_email_supporter_update
BEFORE UPDATE ON Supporter
FOR EACH ROW
BEGIN
    IF NEW.email <> OLD.email THEN
        IF NOT (NEW.email LIKE '_%@_._%' AND
NEW.email LIKE '%.com' AND LOCATE('@', NEW.email,
```

```

LOCATE('@', NEW.email) + 1) = 0 AND
SUBSTRING_INDEX(SUBSTRING_INDEX(NEW.email, '@', 1),
'@', -1) NOT LIKE '%@%') THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Format email
pendukung tidak valid.';
END IF;
END IF;
END //

DELIMITER ;

```

Tangkapan Layar Sebelum Manipulasi

1. Test insert email supporter valid

```

MariaDB [drg]> INSERT INTO Supporter (id_supporter, nama, email, alamat, tanggal_bergabung) VALUES (2001, 'Budi Valid', 'budi.valid@example.com', 'Jl. Valid No. 1', CURDATE());
Query OK, 1 row affected (0.025 sec)

MariaDB [drg]> SELECT id_supporter, email FROM Supporter WHERE id_supporter = 2001;
+-----+-----+
| id_supporter | email                |
+-----+-----+
| 2001         | budi.valid@example.com |
+-----+-----+
1 row in set (0.001 sec)

```

2. Test insert email supporter tidak valid

```

MariaDB [drg]> INSERT INTO Supporter (id_supporter, nama, email, alamat, tanggal_bergabung) VALUES (2002, 'Cici Invalid', 'cici@invalid', 'Jl. Invalid No. 2', CURDATE());
Query OK, 1 row affected (0.027 sec)

MariaDB [drg]> INSERT INTO Supporter (id_supporter, nama, email, alamat, tanggal_bergabung) VALUES (2003, 'Dedi Invalid', 'dedi@example.com', 'Jl. Invalid No. 3', CURDATE());
Query OK, 1 row affected (0.026 sec)

MariaDB [drg]> INSERT INTO Supporter (id_supporter, nama, email, alamat, tanggal_bergabung) VALUES (2004, 'Euis Invalid', 'euis@lokal@example.com', 'Jl. Invalid No. 4', CURDATE());
Query OK, 1 row affected (0.025 sec)

MariaDB [drg]> SELECT id_supporter, email FROM Supporter WHERE id_supporter IN (2002, 2003, 2004);
+-----+-----+
| id_supporter | email                |
+-----+-----+
| 2002         | cici@invalid         |
| 2003         | ded@example.com      |
| 2004         | euis@lokal@example.com |
+-----+-----+
3 rows in set (0.001 sec)

```

3. Test update email supporter tidak valid

```

MariaDB [drg]> UPDATE Supporter SET email = 'budi.salah.com' WHERE id_supporter = 2001;
Query OK, 1 row affected (0.018 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [drg]> SELECT id_supporter, email FROM Supporter WHERE id_supporter = 2001;
+-----+-----+
| id_supporter | email                |
+-----+-----+
| 2001         | budi.salah.com      |
+-----+-----+
1 row in set (0.001 sec)

MariaDB [drg]> UPDATE Supporter SET email = 'budi@salah@domain.com' WHERE id_supporter = 2001;
Query OK, 1 row affected (0.024 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [drg]> SELECT id_supporter, email FROM Supporter WHERE id_supporter = 2001;
+-----+-----+
| id_supporter | email                |
+-----+-----+
| 2001         | budi@salah@domain.com |
+-----+-----+
1 row in set (0.001 sec)

```

4. Test update email supporter valid

```
MariaDB [drg]> UPDATE Supporter SET email = 'budi.baru.valid@domainkeren.com' WHERE id_supporter = 2001;
Query OK, 1 row affected (0.027 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
MariaDB [drg]> SELECT id_supporter, email FROM Supporter WHERE id_supporter = 2001;
+-----+-----+
| id_supporter | email |
+-----+-----+
|          2001 | budi.baru.valid@domainkeren.com |
+-----+-----+
1 row in set (0.001 sec)
```

5. Test insert email creator valid

```
MariaDB [drg]> INSERT INTO Creator (id_creator, nama, email, bidang_kreasi, deskripsi, tanggal_bergabung) VALUES (3001, 'Ani Kreatif', 'ani.kreatif@patra.com', 'Seni Digital', 'Deskripsi Ani', CURDATE());
Query OK, 1 row affected (0.025 sec)

MariaDB [drg]> SELECT id_creator, email FROM Creator WHERE id_creator = 3001;
+-----+-----+
| id_creator | email |
+-----+-----+
|          3001 | ani.kreatif@patra.com |
+-----+-----+
1 row in set (0.001 sec)
```

6. Test insert email creator tidak valid

```
MariaDB [drg]> INSERT INTO Creator (id_creator, nama, email, bidang_kreasi, deskripsi, tanggal_bergabung) VALUES (3002, 'Banu Gagel', 'banu@gagalcom', 'Musik', 'Deskripsi Banu', CURDATE());
Query OK, 1 row affected (0.026 sec)

MariaDB [drg]> INSERT INTO Creator (id_creator, nama, email, bidang_kreasi, deskripsi, tanggal_bergabung) VALUES (3003, 'Citra Error', 'citra.error@domain@salah.com', 'Game Dev', 'Deskripsi Citra', CURDATE());
Query OK, 1 row affected (0.026 sec)

MariaDB [drg]> SELECT id_creator, email FROM Creator WHERE id_creator IN (3002, 3003);
+-----+-----+
| id_creator | email |
+-----+-----+
|          3002 | banu@gagalcom |
|          3003 | citra.error@domain@salah.com |
+-----+-----+
2 rows in set (0.003 sec)
```

7. Test update email creator tidak valid

```
MariaDB [drg]> UPDATE Creator SET email = 'ani@.com' WHERE id_creator = 3001;
Query OK, 1 row affected (0.025 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [drg]> SELECT id_creator, email FROM Creator WHERE id_creator = 3001;
+-----+-----+
| id_creator | email |
+-----+-----+
|          3001 | ani@.com |
+-----+-----+
1 row in set (0.001 sec)

MariaDB [drg]> UPDATE Creator SET email = 'ani.kreatif@patra.net' WHERE id_creator = 3001;
Query OK, 1 row affected (0.026 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [drg]> SELECT id_creator, email FROM Creator WHERE id_creator = 3001;
+-----+-----+
| id_creator | email |
+-----+-----+
|          3001 | ani.kreatif@patra.net |
+-----+-----+
1 row in set (0.000 sec)
```

8. Test update email creator valid

```
MariaDB [drg]> UPDATE Creator SET email = 'ani.kreatif.baru@patracool.com' WHERE id_creator = 3001;
Query OK, 1 row affected (0.003 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [drg]> SELECT id_creator, email FROM Creator WHERE id_creator = 3001;
+-----+-----+
| id_creator | email |
+-----+-----+
|          3001 | ani.kreatif.baru@patracool.com |
+-----+-----+
1 row in set (0.001 sec)
```

Tangkapan Layar Proses Manipulasi

```

MariaDB [DRG]> DELIMITER //
MariaDB [DRG]> CREATE TRIGGER trg_validasi_email_creator_insert
-- BEFORE INSERT ON Creator
-> FOR EACH ROW
-> BEGIN
-> IF NOT (NEW_email LIKE '%_@_%.' AND NEW_email LIKE '%.com' AND LOCATE('@', NEW_email, LOCATE('@', NEW_email) + 1) = 0 AND SUBSTRING_INDEX(SUBSTRING_INDEX(NEW_email, '@', 1), '@', -1) NOT LIKE '%n%') THEN
-> SIGNAL SQLSTATE '45000'
-> SET MESSAGE_TEXT = 'Format email kreator tidak valid.';
-> END IF;
-> END //
Query OK, 0 rows affected (0.035 sec)

MariaDB [DRG]> DELIMITER ;
MariaDB [DRG]> DELIMITER //
MariaDB [DRG]> CREATE TRIGGER trg_validasi_email_creator_update
-- BEFORE UPDATE ON Creator
-> FOR EACH ROW
-> BEGIN
-> IF NEW_email <> OLD_email THEN
-> IF NOT (NEW_email LIKE '%_@_%.' AND NEW_email LIKE '%.com' AND LOCATE('@', NEW_email, LOCATE('@', NEW_email) + 1) = 0 AND SUBSTRING_INDEX(SUBSTRING_INDEX(NEW_email, '@', 1), '@', -1) NOT LIKE '%n%') THEN
-> SIGNAL SQLSTATE '45000'
-> SET MESSAGE_TEXT = 'Format email kreator tidak valid.';
-> END IF;
-> END IF;
-> END //
Query OK, 0 rows affected (0.058 sec)

MariaDB [DRG]> DELIMITER ;
MariaDB [DRG]> DELIMITER //
MariaDB [DRG]> CREATE TRIGGER trg_validasi_email_supporter_insert
-- BEFORE INSERT ON Supporter
-> FOR EACH ROW
-> BEGIN
-> IF NOT (NEW_email LIKE '%_@_%.' AND NEW_email LIKE '%.com' AND LOCATE('@', NEW_email, LOCATE('@', NEW_email) + 1) = 0 AND SUBSTRING_INDEX(SUBSTRING_INDEX(NEW_email, '@', 1), '@', -1) NOT LIKE '%n%') THEN
-> SIGNAL SQLSTATE '45000'
-> SET MESSAGE_TEXT = 'Format email pendukung tidak valid.';
-> END IF;
-> END //
Query OK, 0 rows affected (0.033 sec)

MariaDB [DRG]> DELIMITER ;
MariaDB [DRG]> DELIMITER //
MariaDB [DRG]> CREATE TRIGGER trg_validasi_email_supporter_update
-- BEFORE UPDATE ON Supporter
-> FOR EACH ROW
-> BEGIN
-> IF NEW_email <> OLD_email THEN
-> IF NOT (NEW_email LIKE '%_@_%.' AND NEW_email LIKE '%.com' AND LOCATE('@', NEW_email, LOCATE('@', NEW_email) + 1) = 0 AND SUBSTRING_INDEX(SUBSTRING_INDEX(NEW_email, '@', 1), '@', -1) NOT LIKE '%n%') THEN
-> SIGNAL SQLSTATE '45000'
-> SET MESSAGE_TEXT = 'Format email pendukung tidak valid.';
-> END IF;
-> END IF;
-> END //
Query OK, 0 rows affected (0.054 sec)

MariaDB [DRG]> DELIMITER ;
MariaDB [DRG]>

```

Tangkapan Layar Setelah Manipulasi

1. Test insert email supporter valid

```

MariaDB [DRG]> INSERT INTO Supporter (id_supporter, nama, email, alamat, tanggal_bergabung) VALUES (2001, 'Budi Valid', 'budi.valid@example.com', 'Jl. Valid No. 1', CURDATE());
Query OK, 1 row affected (0.046 sec)

MariaDB [DRG]> SELECT id_supporter, email FROM Supporter WHERE id_supporter = 2001
+-----+-----+
| id_supporter | email |
+-----+-----+
| 2001 | budi.valid@example.com |
+-----+-----+
1 row in set (0.001 sec)

```

2. Test insert email supporter tidak valid

```

MariaDB [DRG]> INSERT INTO Supporter (id_supporter, nama, email, alamat, tanggal_bergabung) VALUES (2002, 'Cici Invalid', 'cici@invalid', 'Jl. Invalid No. 2', CURDATE());
ERROR 1644 (45000): Format email pendukung tidak valid.
MariaDB [DRG]> INSERT INTO Supporter (id_supporter, nama, email, alamat, tanggal_bergabung) VALUES (2003, 'Dedi Invalid', 'dedi@example.com', 'Jl. Invalid No. 3', CURDATE());
ERROR 1644 (45000): Format email pendukung tidak valid.
MariaDB [DRG]> INSERT INTO Supporter (id_supporter, nama, email, alamat, tanggal_bergabung) VALUES (2004, 'Euis Invalid', 'euis@tosal@example.com', 'Jl. Invalid No. 4', CURDATE()); -- 0 di bagian lokal
ERROR 1644 (45000): Format email pendukung tidak valid.
MariaDB [DRG]> SELECT id_supporter, email FROM Supporter WHERE id_supporter IN (2002, 2003, 2004);
Empty set (0.002 sec)

```

3. Test update email supporter tidak valid

```

MariaDB [DRG]> UPDATE Supporter SET email = 'budi.salah.com' WHERE id_supporter = 2001;
ERROR 1644 (45000): Format email pendukung tidak valid.
MariaDB [DRG]> UPDATE Supporter SET email = 'budi@salah@domain.com' WHERE id_supporter = 2001;
ERROR 1644 (45000): Format email pendukung tidak valid.
MariaDB [DRG]> SELECT id_supporter, email FROM Supporter WHERE id_supporter = 2001;
+-----+-----+
| id_supporter | email |
+-----+-----+
| 2001 | budi.valid@example.com |
+-----+-----+
1 row in set (0.001 sec)

```

4. Test update email supporter valid

```

MariaDB [DRG]> UPDATE Supporter SET email = 'budi.baru.valid@domainkeren.com' WHERE id_supporter = 2001;
Query OK, 1 row affected (0.026 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [DRG]> SELECT id_supporter, email FROM Supporter WHERE id_supporter = 2001;
+-----+-----+
| id_supporter | email |
+-----+-----+
| 2001 | budi.baru.valid@domainkeren.com |
+-----+-----+
1 row in set (0.001 sec)

```

5. Test insert email creator valid

```

MariaDB [DRG]> INSERT INTO Creator (id_creator, nama, email, bidang_kreatif, deskripsi, tanggal_bergabung) VALUES (3001, 'Ani Kreatif', 'ani.kreatif@patra.com', 'Seni Digital', 'Deskripsi Ani', CURDATE());
Query OK, 1 row affected (0.024 sec)

MariaDB [DRG]> SELECT id_creator, email FROM Creator WHERE id_creator = 3001;
+-----+-----+
| id_creator | email |
+-----+-----+
| 3001 | ani.kreatif@patra.com |
+-----+-----+
1 row in set (0.001 sec)

```

6. Test insert email creator tidak valid

```
MariaDB [DRG]> INSERT INTO Creator (id_creator, nama, email, bidang_kreatif, deskripsi, tanggal_berpindah) VALUES (3002, 'Bani Gagal', 'banugagal.com', 'Musik', 'Deskripsi Bani', CURDATE()); -- tidak ada . sebelum com
ERROR 1644 (45000): Format email kreator tidak valid.
MariaDB [DRG]> INSERT INTO Creator (id_creator, nama, email, bidang_kreatif, deskripsi, tanggal_berpindah) VALUES (3003, 'Citra Error', 'citra.error@domainpatra.com', 'Game Dev', 'Deskripsi Citra', CURDATE()); -- # di bagian domain kedua
ERROR 1644 (45000): Format email kreator tidak valid.
MariaDB [DRG]> SELECT id_creator, email FROM Creator WHERE id_creator IN (3002, 3003);
Empty set (0.001 sec)
```

7. Test update email creator tidak valid

```
MariaDB [DRG]> UPDATE Creator SET email = 'ani@.com' WHERE id_creator = 3001; -- tidak ada domain
ERROR 1644 (45000): Format email kreator tidak valid.
MariaDB [DRG]> UPDATE Creator SET email = 'ani.kreatif@patra.net' WHERE id_creator = 3001; -- bukan .com
ERROR 1644 (45000): Format email kreator tidak valid.
MariaDB [DRG]> SELECT id_creator, email FROM Creator WHERE id_creator = 3001;
+-----+-----+
| id_creator | email |
+-----+-----+
| 3001 | ani.kreatif@patra.com |
+-----+-----+
1 row in set (0.001 sec)
```

8. Test update email creator valid

```
MariaDB [DRG]> UPDATE Creator SET email = 'ani.kreatif.baru@patracool.com' WHERE id_creator
Query OK, 1 row affected (0.028 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [DRG]> SELECT id_creator, email FROM Creator WHERE id_creator = 3001;
+-----+-----+
| id_creator | email |
+-----+-----+
| 3001 | ani.kreatif.baru@patracool.com |
+-----+-----+
1 row in set (0.000 sec)
```

5. IMPLEMENTASI ACCESS BUSINESS RULE

a. Validasi Jenis Konten

Jenis konten yang masuk harus memiliki tipe yang sesuai dengan yang diminta, sehingga perlu ditambahkan kode dibawah ini, agar bisa menyaring data yang masuk, agar tidak terjadi ketidaksesuaian dari batasan jenis konten yang diinginkan.

```
ALTER TABLE Konten
ADD CONSTRAINT chk_validasi_jenis_konten
CHECK (jenis IN ('Teks', 'Gambar', 'Audio', 'Video'));
```

b. Batasan Harga Merchandise dan Tier

Pada tabel Merchandise dan Tier, diberikan batasan terhadap harga yang ada, sehingga cukup ditambahkan syarat nilai integer, agar harga pada tabel Merchandise tidak 0 dan tidak lebih dari 1500000, dan juga ditambahkan syarat pada tabel tier agar harga tidak 0 dan tidak melebihi 1000000.


```
ALTER TABLE Merchandise
ADD CONSTRAINT chk_batasan_harga_merchandise
CHECK (harga > 0 AND harga <= 1500000);
```

```
ALTER TABLE Tier
ADD CONSTRAINT chk_batasan_harga_tier
CHECK (harga > 0 AND harga <= 1000000);
```

c. Maksimal Tier per Kreator

Perlu dibatasi untuk banyak tier tiap kreator, sehingga setiap kali mencoba untuk menambahkan tier dengan id kreator tertentu, akan selalu dicek pada tabel tier, untuk dicek banyak tier yang sudah ada dan memiliki id_creator sekarang. Apabila jumlah sudah lebih dari atau sama dengan 5, maka akan dihentikan. Apabila belum sampai 5, maka insert akan dijalankan seperti biasanya.

```
DELIMITER //

CREATE TRIGGER trg_cek_maksimal_tier_per_kreator
BEFORE INSERT ON Tier
FOR EACH ROW
BEGIN
    DECLARE jumlah_tier_kreator INT;
    SELECT COUNT(*) INTO jumlah_tier_kreator
    FROM Tier
    WHERE id_creator = NEW.id_creator;

    IF jumlah_tier_kreator >= 5 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Business Rule Violation: Kreator
telah mencapai batas maksimal 5 tier.';
    END IF;
END //

DELIMITER ;
```

d. Alur Status Commission

Setiap kali ingin mengubah isi string status_pengerjaan pada tabel informasi_hasil, diperlukan sebuah trigger untuk menyaring perubahan,

agar selalu sesuai yang diinginkan yaitu dari 'menunggu_persetujuan' ke 'disetujui' atau 'ditolak', dari 'disetujui' ke 'dalam_pengerjaan', dan dari 'dalam_pengerjaan' ke 'selesai'.

```
DELIMITER //
```

```
CREATE TRIGGER trg_alur_status_pengerjaan_hasil
BEFORE UPDATE ON Informasi_Hasil
FOR EACH ROW
BEGIN
    DECLARE old_status_pengerjaan VARCHAR(75);
    DECLARE new_status_pengerjaan VARCHAR(75);

    SET old_status_pengerjaan = OLD.status_pengerjaan;
    SET new_status_pengerjaan = NEW.status_pengerjaan;

    IF old_status_pengerjaan IS NULL OR old_status_pengerjaan =
new_status_pengerjaan THEN
        LEAVE BEGIN;
    END IF;

    CASE old_status_pengerjaan
        WHEN 'menunggu_persetujuan' THEN
            IF NOT (new_status_pengerjaan = 'disetujui' OR
new_status_pengerjaan = 'ditolak') THEN
                SIGNAL SQLSTATE '45000'
                SET MESSAGE_TEXT = 'Business Rule Violation: Dari
status pengerjaan "menunggu_persetujuan", status hanya bisa
berubah menjadi "disetujui" atau "ditolak".';
            END IF;
        WHEN 'disetujui' THEN
            IF NOT (new_status_pengerjaan = 'dalam_pengerjaan')
THEN
                SIGNAL SQLSTATE '45000'
                SET MESSAGE_TEXT = 'Business Rule Violation: Dari
status pengerjaan "disetujui", status hanya bisa berubah menjadi
"dalam_pengerjaan".';
            END IF;
        WHEN 'dalam_pengerjaan' THEN
            IF NOT (new_status_pengerjaan = 'selesai') THEN
                SIGNAL SQLSTATE '45000'
                SET MESSAGE_TEXT = 'Business Rule Violation: Dari
status pengerjaan "dalam_pengerjaan", status hanya bisa berubah
menjadi "selesai".';
            END IF;
        WHEN 'ditolak' THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Business Rule Violation: Status
pengerjaan "ditolak" tidak dapat diubah.';
        WHEN 'selesai' THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Business Rule Violation: Status
pengerjaan "selesai" tidak dapat diubah.';
        ELSE
    
```

```
        IF old_status_pengerjaan NOT IN
('menunggu_persetujuan', 'disetujui', 'ditolak',
'dalam_pengerjaan', 'selesai') THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = CONCAT('Business Rule
Violation: Status pengerjaan lama "', old_status_pengerjaan, '"
tidak dikenal dalam alur yang ditetapkan.');
```

END IF;

END CASE;

END //

DELIMITER ;