

TUGAS PRESENTASI

Kelompok 3

101
fell in love with it.
village

LIBRATECH

LibraTech adalah sebuah platform digital yang dirancang untuk mempermudah akses informasi dan pengelolaan peminjaman buku di perpustakaan. Website ini menjadi solusi modern bagi perpustakaan untuk memberikan pelayanan yang efisien dan transparan kepada para anggota, baik mahasiswa, dosen, maupun masyarakat umum.

TABEL UTAMA

1. Tabel Anggota : Menyimpan data anggota perpustakaan.
2. Tabel Kategori : Menyimpan data kategori buku.
3. Tabel Buku : Menyimpan data buku di perpustakaan.
4. Tabel Peminjaman: Menyimpan data transaksi peminjaman buku.
5. Tabel Log Peminjaman : Menyimpan catatan aktivitas peminjaman.

RELASI ANTAR TABEL

1. Relasi antara tabel anggota dan peminjaman : Relasi one-to-many satu anggota dapat melakukan banyak peminjaman buku.
2. Relasi antara tabel kategori dan buku: Relasi one-to-many satu kategori dapat memiliki banyak buku.
3. Relasi antara tabel buku dan peminjaman: Relasi one-to-many satu buku dapat dipinjam dalam beberapa transaksi peminjaman yang berbeda.
4. Relasi antara tabel peminjaman dan logpeminjaman: Relasi one-to-many Satu transaksi peminjaman dapat memiliki beberapa log aktivitas terkait

CONSTRAIN, KEY, DAN INDEX

Primary Key:

- anggota: id_anggota
- kategori: id_kategori
- buku: id_buku
- peminjaman: id_peminjaman
- logpeminjaman: id_log

Foreign Key:

- peminjaman: id_anggota → anggota(id_anggota), id_buku → buku(id_buku)
- buku: id_kategori → kategori(id_kategori)
- logpeminjaman: id_peminjaman → peminjaman(id_peminjaman)

Indexing:

- anggota: nama
- buku: judul
- peminjaman: tanggal_pinjam

HAK AKSES PENGGUNA

1. Administrator: Akses penuh ke semua tabel (CREATE, READ, UPDATE, DELETE).
2. Petugas: Kelola data anggota, buku, dan peminjaman (READ, INSERT, UPDATE).
3. Anggota: Lihat daftar buku dan status peminjaman pribadi (READ).
4. Auditor: Periksa log aktivitas dan transaksi (READ).

PENAMBAHAN CURSOR, TRIGGER, DAN ERROR HANDLING

Error handling: Cek apakah buku sudah dipinjam

```
DECLARE buku_dipinjam INT;
SELECT COUNT(*) INTO buku_dipinjam
FROM Peminjaman
WHERE id_buku = p_id_buku AND tanggal_kembali IS NULL;

IF buku_dipinjam > 0 THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Buku ini sedang dipinjam dan
belum dikembalikan.';
ELSE
    -- Insert data peminjaman
    INSERT INTO Peminjaman (id_anggota, id_buku,
tanggal_pinjam, tanggal_kembali)
VALUES (p_id_anggota, p_id_buku, p_tanggal_pinjam,
p_tanggal_kembali);
END IF;
END;
//
DELIMITER ;
```

CREATE CURSOR

-- Procedure untuk menampilkan semua buku berdasarkan kategori tertentu

DELIMITER //

CREATE PROCEDURE LihatBukuPerKategori(IN p_id_kategori INT)
BEGIN

DECLARE selesai INT DEFAULT 0;

DECLARE buku_judul VARCHAR(100);

DECLARE buku_cursor CURSOR FOR

SELECT judul FROM Buku WHERE id_kategori = p_id_kategori;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET selesai = 1;

OPEN buku_cursor;

buku_loop: LOOP

FETCH buku_cursor INTO buku_judul;

IF selesai THEN

LEAVE buku_loop;

END IF;

SELECT buku_judul AS "Judul Buku";

END LOOP;

CLOSE buku_cursor;

END;

//

DELIMITER ;

PENAMBAHAN CURSOR, TRIGGER, DAN ERROR HANDLING

CREATE TRIGGER

-- Trigger untuk mencatat log setiap kali ada peminjaman baru

CREATE TABLE IF NOT EXISTS LogPeminjaman (

id_log INT AUTO_INCREMENT PRIMARY KEY,

id_peminjaman INT,

waktu_log TIMESTAMP DEFAULT CURRENT_TIMESTAMP

);

DELIMITER //

CREATE TRIGGER AfterInsertPeminjaman

AFTER INSERT ON Peminjaman

FOR EACH ROW

BEGIN

INSERT INTO LogPeminjaman (id_peminjaman)

VALUES (NEW.id_peminjaman);

END;

//

DELIMITER ;

-- Commit transaksi

COMMIT;

STORED VIEW

Stored Procedure untuk menambahkan data peminjaman

DELIMITER //

CREATE PROCEDURE TambahPeminjaman(

IN p_id_anggota INT,

IN p_id_buku INT,

IN p_tanggal_pinjam DATE,

IN p_tanggal_kembali DATE

)

BEGIN

TERIMA KASIH

