**Nama : Anne Dwi Revina**

**Kelas : XI RPL 2**

**No. Absen : 03**

**SISTEM PERPUSTAKAAN SEKOLAH**

1. Tabel Buku

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID Buku | Judul Buku | Penulis | Kategori | Stok |
| 1 | Algoritma dan Pemrograman | Andi Wijaya | Teknologi | 5 |
| 2 | Dasar-dasar Database | Budi Santoso | Teknologi | 7 |
| 3 | Matematika Diskrit | Rina Sari | Matematika | 4 |
| 4 | Sejarah Dunia | John Smith | Sejarah | 3 |
| 5 | Pemrograman Web dengan PHP | Eko Prasetyo | Teknologi | 8 |
| 6 | Sistem Operasi | Dian Kurniawan | Teknologi | 6 |
| 7 | Jaringan Komputer | Ahmad Fauzi | Teknologi | 5 |
| 8 | Cerita Rakyat Nusantara | Lestari Dewi | Sastra | 9 |
| 9 | Bahasa Inggris untuk Pemula | Jane Doe | Bahasa | 10 |
| 10 | Biologi Dasar | Budi Rahman | Sains | 7 |
| 11 | Kimia Organik | Siti Aminah | Sains | 5 |
| 12 | Teknik Elektro | Ridwan Hakim | Teknik | 6 |
| 13 | Fisika Modern | Albert Einstein | Sains | 4 |
| 14 | Manajemen Waktu | Steven Covey | Pengembangan | 8 |
| 15 | Strategi Belajar Efektif | Tony Buzan | Pendidikan | 6 |

1. Tabel Siswa

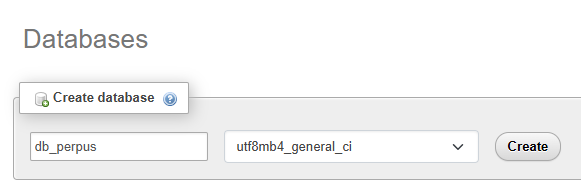
|  |  |  |
| --- | --- | --- |
| ID Siswa | Nama | Kelas |
| 1 | Andi Saputra | X-RPL |
| 2 | Budi Wijaya | X-TKJ |
| 3 | Citra Lestari | XI-RPL |
| 4 | Dewi Kurniawan | XI-TKJ |
| 5 | Eko Prasetyo | XII-RPL |
| 6 | Farhan Maulana | XII-TKJ |
| 7 | Gita Permata | X-RPL |
| 8 | Hadi Sucipto | X-TKJ |
| 9 | Intan Permadi | XI-RPL |
| 10 | Joko Santoso | XI-TKJ |
| 11 | Kartika Sari | XII-RPL |
| 12 | Lintang Putri | XII-TKJ |
| 13 | Muhammad Rizky | X-RPL |
| 14 | Novi Andriana | X-TKJ |
| 15 | Olivia Hernanda | XI-RPL |

1. Tabel Peminjaman

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID Peminjaman | ID Siswa | ID Buku | Tanggal Pinjam | Tanggal Kembali | Status |
| 1 | 11 | 2 | 2025-02-01 | 2025-02-08 | Dipinjam |
| 2 | 2 | 5 | 2025-01-28 | 2025-02-04 | Dikembalikan |
| 3 | 3 | 8 | 2025-02-02 | 2025-02-09 | Dipinjam |
| 4 | 4 | 10 | 2025-01-30 | 2025-02-06 | Dikembalikan |
| 5 | 5 | 3 | 2025-01-25 | 2025-02-01 | Dikembalikan |
| 6 | 15 | 7 | 2025-02-01 | 2025-02-08 | Dipinjam |
| 7 | 7 | 1 | 2025-01-29 | 2025-02-05 | Dikembalikan |
| 8 | 8 | 9 | 2025-02-03 | 2025-02-10 | Dipinjam |
| 9 | 13 | 4 | 2025-01-27 | 2025-02-03 | Dikembalikan |
| 10 | 10 | 11 | 2025-02-01 | 2025-02-08 | Dipinjam |

**TUGAS**

1. Buatlah database dengan nama db\_perpus.



1. Buatlah table buku, siswa, dan peminjaman.

CREATE TABLE buku (

id\_buku INT PRIMARY KEY AUTO\_INCREMENT,

judul\_buku VARCHAR(255),

penulis VARCHAR(255),

kategori VARCHAR(100),

stok INT

);

CREATE TABLE siswa (

id\_siswa INT PRIMARY KEY AUTO\_INCREMENT,

nama VARCHAR(255),

kelas VARCHAR(50)

);

CREATE TABLE peminjaman (

id\_peminjaman INT PRIMARY KEY AUTO\_INCREMENT,

id\_siswa INT,

id\_buku INT,

tanggal\_pinjam DATE,

tanggal\_kembali DATE,

status ENUM('Dipinjam', 'Dikembalikan'),

FOREIGN KEY (id\_siswa) REFERENCES siswa(id\_siswa),

FOREIGN KEY (id\_buku) REFERENCES buku(id\_buku)

);

1. Input 5 record di setiap table menggunakan query INSERT, UPDATE, DELETE.

INSERT INTO buku (judul\_buku, penulis, kategori, stok) VALUES

('Algoritma dan Pemrograman', 'Andi Wijaya', 'Teknologi', 5),

('Dasar-dasar Database', 'Budi Santoso', 'Teknologi', 7),

('Matematika Diskrit', 'Rina Sari', 'Matematika', 4),

('Sejarah Dunia', 'John Smith', 'Sejarah', 3),

('Pemrograman Web dengan PHP', 'Eko Prasetyo', 'Teknologi', 8);

INSERT INTO siswa (nama, kelas) VALUES

('Andi Saputra', 'X-RPL'),

('Budi Wijaya', 'X-TKJ'),

('Citra Lestari', 'XI-RPL'),

('Dewi Kurniawan', 'XI-TKJ'),

('Eko Prasetyo', 'XII-RPL');

INSERT INTO peminjaman (id\_siswa, id\_buku, tanggal\_pinjam, tanggal\_kembali, status) VALUES

(1, 2, '2025-02-01', '2025-02-08', 'Dipinjam'),

(2, 5, '2025-01-28', '2025-02-04', 'Dikembalikan'),

(3, 3, '2025-02-02', '2025-02-09', 'Dipinjam'),

(4, 4, '2025-01-30', '2025-02-06', 'Dikembalikan'),

(5, 1, '2025-01-25', '2025-02-01', 'Dikembalikan');

1. Input 10 record di setiap table menggunakan stored procedure INSERT.

Insert Buku :

DELIMITER //

CREATE PROCEDURE InsertBuku(

IN judul\_buku varchar(225),

IN penulis varchar(225),

IN kategori varchar(100),

IN stok int

)

BEGIN

INSERT INTO buku (judul\_buku, penulis, kategori, stok) VALUES

(judul\_buku, penulis, kategori, stok);

END //

DELIMITER ;

Insert Siswa :

DELIMITER //

CREATE PROCEDURE InsertSiswa2(

IN nama varchar(50),

IN kelas varchar(50)

)

BEGIN

INSERT INTO siswa (nama, kelas) VALUES

(nama, kelas);

END //

DELIMITER ;

Insert Peminjaman :

DELIMITER //

CREATE PROCEDURE InsertPeminjaman2(

IN Id\_siswa int,

IN Id\_buku int,

IN Tanggal\_pinjam date,

IN Tanggal\_kembali date,

IN Status ENUM('Dipinjam','Dikembalikan')

)

BEGIN

INSERT INTO peminjaman (id\_siswa, id\_buku, tanggal\_pinjam, tanggal\_kembali, status) VALUES

(Id\_siswa, Id\_buku, Tanggal\_pinjam, Tanggal\_kembali, Status);

END //

CALL InsertBuku('Jaringan Komputer', 'Ahmad Fauzi', 'Teknologi', 5);

CALL InsertSiswa('Olivia Hernanda', 'XI-RPL');

CALL InsertPeminjaman(6, 7, '2025-02-05', '2025-02-12', 'Dipinjam');

1. Buatlah stored procedure UPDATE, DELETE di setiap table.

Update tabel buku :

DELIMITER //

CREATE PROCEDURE UpdateBuku(

IN p\_id\_buku INT,

IN p\_judul\_buku VARCHAR(255),

IN p\_penulis VARCHAR(255),

IN p\_kategori VARCHAR(100),

IN p\_stok INT

)

BEGIN

UPDATE buku

SET judul\_buku = p\_judul\_buku,

penulis = p\_penulis,

kategori = p\_kategori,

stok = p\_stok

WHERE id\_buku = p\_id\_buku;

END //

DELIMITER ;

Delete tabel buku :

DELIMITER //

CREATE PROCEDURE DeleteBuku(

IN p\_id\_buku INT

)

BEGIN

DELETE FROM buku WHERE id\_buku = p\_id\_buku;

END //

DELIMITER ;

Update tabel siswa :

DELIMITER //

CREATE PROCEDURE UpdateSiswa(

IN p\_id\_siswa INT,

IN p\_nama VARCHAR(255),

IN p\_kelas VARCHAR(50)

)

BEGIN

UPDATE siswa

SET nama = p\_nama,

kelas = p\_kelas

WHERE id\_siswa = p\_id\_siswa;

END //

DELIMITER ;

Delete tabel siswa :

DELIMITER //

CREATE PROCEDURE DeleteSiswa(

IN p\_id\_siswa INT

)

BEGIN

DELETE FROM siswa WHERE id\_siswa = p\_id\_siswa;

END //

DELIMITER ;

Update tabel peminjaman :

DELIMITER //

CREATE PROCEDURE UpdatePeminjaman(

IN p\_id\_peminjaman INT,

IN p\_id\_siswa INT,

IN p\_id\_buku INT,

IN p\_tanggal\_pinjam DATE,

IN p\_tanggal\_kembali DATE,

IN p\_status ENUM('Dipinjam','Dikembalikan')

)

BEGIN

UPDATE peminjaman

SET id\_siswa = p\_id\_siswa,

id\_buku = p\_id\_buku,

tanggal\_pinjam = p\_tanggal\_pinjam,

tanggal\_kembali = p\_tanggal\_kembali,

status = p\_status

WHERE id\_peminjaman = p\_id\_peminjaman;

END //

DELIMITER ;

Delete tabel peminjaman :

DELIMITER //

CREATE PROCEDURE DeletePeminjaman(

IN p\_id\_peminjaman INT

)

BEGIN

DELETE FROM peminjaman WHERE id\_peminjaman = p\_id\_peminjaman;

END //

DELIMITER ;

1. Buatlah stored procedure untuk menampilkan seluruh record di setiap table.

Table buku :

DELIMITER //

CREATE PROCEDURE GetAllBuku()

BEGIN

SELECT \* FROM buku;

END //

DELIMITER ;

Tabel siswa :

DELIMITER //

CREATE PROCEDURE GetAllSiswa()

BEGIN

SELECT \* FROM siswa;

END //

DELIMITER ;

Tabel peminjaman :

DELIMITER //

CREATE PROCEDURE GetAllPeminjaman()

BEGIN

SELECT \* FROM peminjaman;

END //

DELIMITER ;

CALL GetAllBuku();

CALL GetAllSiswa();

CALL GetAllPeminjaman();

1. Stok buku pada saat dipinjam berkurang secara otamatis.

DELIMITER //

CREATE PROCEDURE PinjamBuku(

IN p\_id\_siswa INT,

IN p\_id\_buku INT,

IN p\_tanggal\_pinjam DATE

)

BEGIN

-- Insert record peminjaman dengan status 'Dipinjam' dan tanggal\_kembali NULL

INSERT INTO peminjaman (id\_siswa, id\_buku, tanggal\_pinjam, tanggal\_kembali, status)

VALUES (p\_id\_siswa, p\_id\_buku, p\_tanggal\_pinjam, NULL, 'Dipinjam');

-- Kurangi stok buku

UPDATE buku SET stok = stok - 1 WHERE id\_buku = p\_id\_buku;

END //

DELIMITER ;

CALL PinjamBuku(2, 3, '2025-02-15');

1. Stok buku pada saat dikembalikan bertambah secara otomatis.

DELIMITER //

CREATE PROCEDURE KembalikanBuku(

IN p\_id\_peminjaman INT,

IN p\_id\_buku INT

)

BEGIN

-- Update status peminjaman dan set tanggal\_kembali dengan CURRENT\_DATE

UPDATE peminjaman

SET status = 'Dikembalikan', tanggal\_kembali = CURRENT\_DATE

WHERE id\_peminjaman = p\_id\_peminjaman;

-- Tambah stok buku

UPDATE buku SET stok = stok + 1 WHERE id\_buku = p\_id\_buku;

END //

DELIMITER ;

CALL KembalikanBuku(7, 3);

1. Buatlah stored procedure untuk mengembalikan buku dan gunakan tanggal pengembalian sesuai dengan tanggal saat mengembalikan (CURRENT DATE).

DELIMITER //

CREATE PROCEDURE ReturnBook(

IN p\_id\_peminjaman INT,

IN p\_id\_buku INT

)

BEGIN

-- Update status peminjaman menjadi 'Dikembalikan' dan set tanggal\_kembali dengan CURRENT\_DATE()

UPDATE peminjaman

SET status = 'Dikembalikan',

tanggal\_kembali = CURRENT\_DATE()

WHERE id\_peminjaman = p\_id\_peminjaman;

-- Tambahkan stok buku karena buku sudah dikembalikan

UPDATE buku

SET stok = stok + 1

WHERE id\_buku = p\_id\_buku;

END //

DELIMITER ;

CALL ReturnBook(1, 2);

1. Buatlah stored procedure untuk menampilkan daftar siswa yang pernah meminjam buku.

DELIMITER //

CREATE PROCEDURE GetSiswaPernahPinjam()

BEGIN

SELECT DISTINCT s.id\_siswa, s.nama, s.kelas

FROM siswa s

JOIN peminjaman p ON s.id\_siswa = p.id\_siswa;

END //

DELIMITER ;

CALL GetSiswaPernahPinjam();

1. Buatlah stored procedure untuk menampilkan semua siswa, termasuk yang tidak pernah meminjam buku.

DELIMITER //

CREATE PROCEDURE GetSemuaSiswa()

BEGIN

SELECT s.id\_siswa, s.nama, s.kelas,

CASE

WHEN p.id\_peminjaman IS NULL THEN 'Belum Pernah Meminjam'

ELSE 'Pernah Meminjam'

END AS status\_peminjaman

FROM siswa s

LEFT JOIN peminjaman p ON s.id\_siswa = p.id\_siswa;

END //

DELIMITER ;

CALL GetSemuaSiswa();

1. Buatlah stored procedure untuk menampilkan semua buku, termasuk yang belum pernah dipinjam.

DELIMITER //

CREATE PROCEDURE GetSemuaBuku()

BEGIN

SELECT b.id\_buku, b.judul\_buku, b.penulis, b.kategori, b.stok,

CASE

WHEN p.id\_peminjaman IS NULL THEN 'Belum Pernah Dipinjam'

ELSE 'Pernah Dipinjam'

END AS status\_peminjaman

FROM buku b

LEFT JOIN peminjaman p ON b.id\_buku = p.id\_buku;

END //

DELIMITER ;

CALL GetSemuaBuku();

**~ PUSH File SQL ke gitHub dengan nama repository Database-Pepustakaan-Sekolah ~**