**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

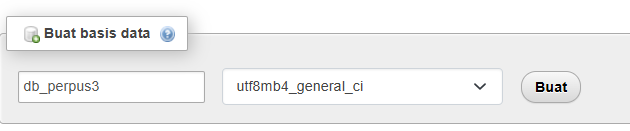
|  |  |  |
| --- | --- | --- |
| ID Siswa | Nama | Kelas |
| 1 | Andi Saputra | X-RPL |
| 2 | Budi Wijaya | X-TKJ |
| 3 | Citra Lestari | XI-RPL |
| 4 | Dewi Kurniawan | XI-TKJ |
| 5 | Eko Prasetyo | XII-RPL |
| 6 | Farhan Maulana | XII-TKJ |
| 7 | Gita Permata | X-RPL |
| 8 | Hadi Sucipto | X-TKJ |
| 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(25),

penulis VARCHAR(25),

kategori VARCHAR(10),

stok INT

);



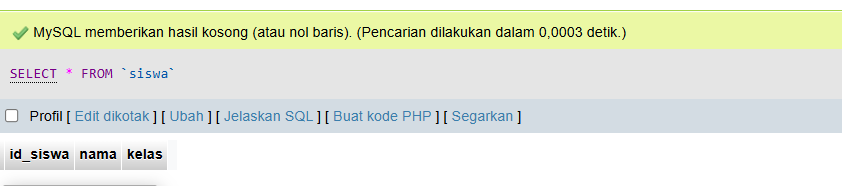
CREATE TABLE siswa (

id\_siswa INT PRIMARY KEY AUTO\_INCREMENT,

nama VARCHAR(25),

kelas VARCHAR(50)

);



CREATE TABLE peminjaman (

id\_peminjaman INT PRIMARY KEY,

id\_siswa INT,

id\_buku INT,

tanggal\_pinjam DATE,

tanggal\_kembali DATE,

status ENUM('Dipinjam', 'Dikembalikan')

);



1. Input 5 record di setiap table menggunakan query INSERT.

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

(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');



INSERT INTO siswa (nama\_siswa, kelas) VALUES

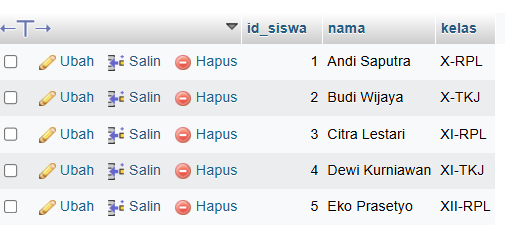
('Andi Saputra', 'X-RPL'),

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

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

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

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



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

Table 1

DELIMITER //

CREATE PROCEDURE InsertBuku()

BEGIN

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

('Sistem Operasi', 'Dian Kurniawan', 'Teknologi', 6),

('Jaringan Komputer', 'Ahmad Fauzi', 'Teknologi', 5),

('Cerita Rakyat Nusantara', 'Lestari Dewi', 'Sastra', 9),

('Bahasa Inggris untuk Pemula', 'Jane Doe', 'Bahasa', 10),

('Biologi Dasar', 'Budi Rahman', 'Sains', 7),

('Kimia Organik', 'Siti Aminah', 'Sains', 5),

('Teknik Elektro', 'Ridwan Hakim', 'Teknik', 6),

('Fisika Modern', 'Albert Einstein', 'Sains', 4),

('Manajemen Waktu', 'Steven Covey', 'Pengembangan', 8),

('Strategi Belajar Efektif', 'Tony Buzan', 'Pendidikan', 6);

END //

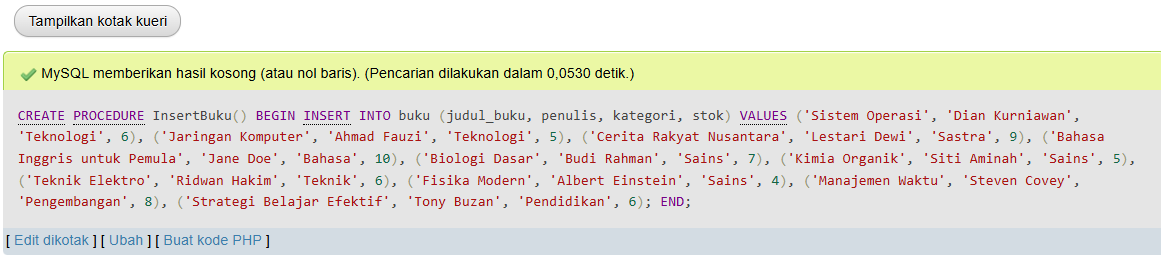


Table 2

DELIMITER $$

CREATE PROCEDURE INSERTpeminjaman()

BEGIN

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

(11, 2, '2025-02-01', '2025-02-08', 'Dipinjam'),

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

(3, 8, '2025-02-02', '2025-02-09', 'Dipinjam'),

(4, 10, '2025-01-30', '2025-02-06', 'Dikembalikan'),

(5, 3, '2025-01-25', '2025-02-01', 'Dikembalikan'),

(15, 7, '2025-02-01', '2025-02-08', 'Dipinjam'),

(7, 1, '2025-01-29', '2025-02-05', 'Dikembalikan'),

(8, 9, '2025-02-03', '2025-02-10', 'Dipinjam'),

(13, 4, '2025-01-27', '2025-02-03', 'Dikembalikan'),

(10, 11, '2025-02-01', '2025-02-08', 'Dipinjam');

END $$

DELIMITER ;

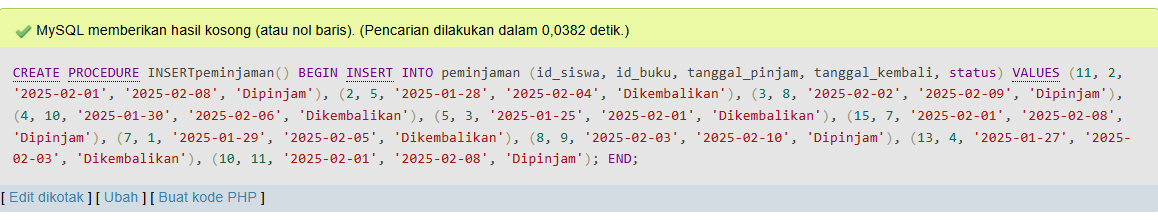


Table 3

DELIMITER $$

CREATE PROCEDURE INSERTsiswa()

BEGIN

INSERT INTO siswa (nama\_siswa, kelas) VALUES

('Farhan Maulana', 'XII-TKJ'),

('Gita Permata', 'X-RPL'),

('Hadi Sucipto', 'X-TKJ'),

('Intan Permadi', 'XI-RPL'),

('Joko Santoso', 'XI-TKJ'),

('Kartika Sari', 'XII-RPL'),

('Lintang Putri', 'XII-TKJ'),

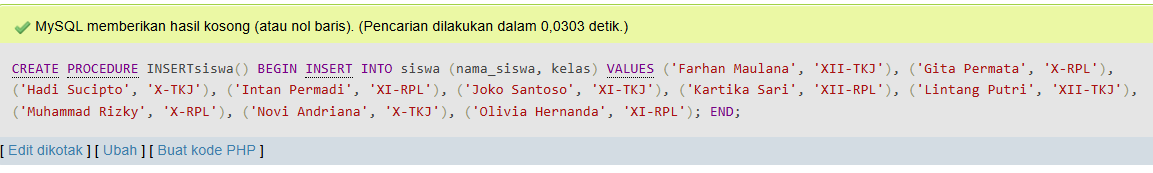
('Muhammad Rizky', 'X-RPL'),

('Novi Andriana', 'X-TKJ'),

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

END $$

DELIMITER ;



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

Table 1 (buku)

-Update

DELIMITER //

CREATE PROCEDURE UpdateBuku(

IN Id\_BukuBaru INT,

IN JudulBukuBaru VARCHAR(50),

IN PenulisBaru VARCHAR(20),

IN KategoriBaru VARCHAR(20),

IN StokBaru INT

)

BEGIN

UPDATE buku

SET judul\_buku = JudulBukuBaru,

penulis = PenulisBaru,

kategori = KategoriBaru,

stok = StokBaru

WHERE id\_buku = Id\_BukuBaru;

END //

-Delete

DELIMITER //

CREATE PROCEDURE DeleteBuku(

IN Id\_BukuBaru INT

)

BEGIN

DELETE FROM buku WHERE id\_buku = Id\_BukuBaru;

END //

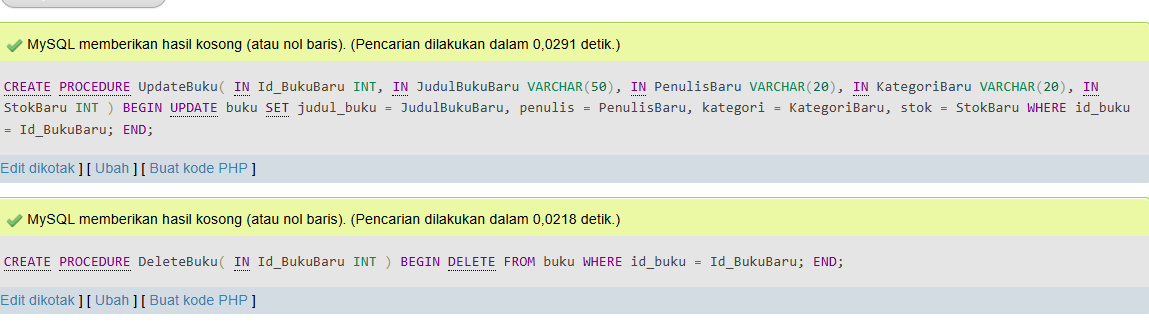


Table 2 (peminjaman)

-Update

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 VARCHAR(15) -- ENUM diganti ke VARCHAR karena tidak bisa di parameter prosedur

)

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

DELIMITER $$

CREATE PROCEDURE DeletePeminjaman(

IN p\_id\_peminjaman INT

)

BEGIN

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

END $$

DELIMITER ;

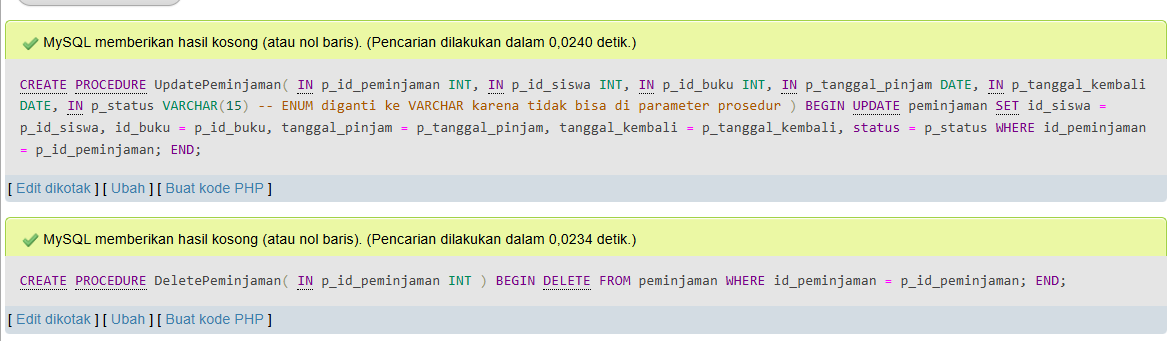


Table 3(siswa)

DELIMITER $$

CREATE PROCEDURE UpdateSiswa(

IN p\_id\_siswa INT,

IN p\_nama\_siswa VARCHAR(100),

IN p\_kelas VARCHAR(20)

)

BEGIN

UPDATE siswa

SET nama\_siswa = p\_nama\_siswa,

kelas = p\_kelas

WHERE id\_siswa = p\_id\_siswa;

END $$

DELIMITER ;

DELIMITER $$

CREATE PROCEDURE DeleteSiswa(

IN p\_id\_siswa INT

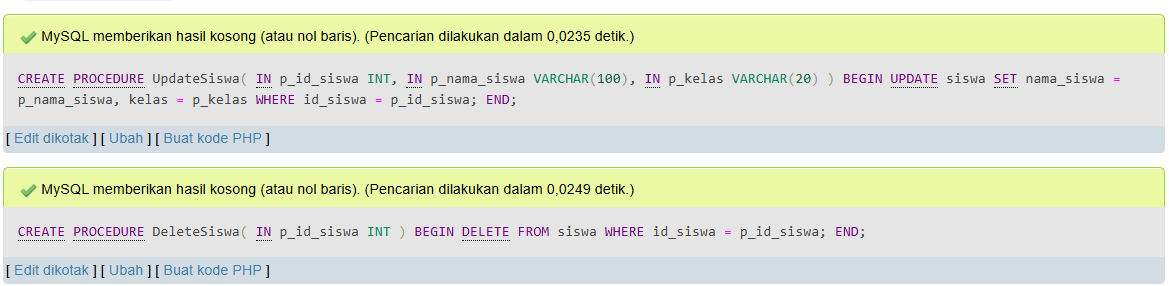
)

BEGIN

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

END $$

DELIMITER ;



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

Table 1 (buku)

DELIMITER $$

CREATE PROCEDURE GetAllBuku()

BEGIN

SELECT \* FROM buku;

END $$

DELIMITER ;

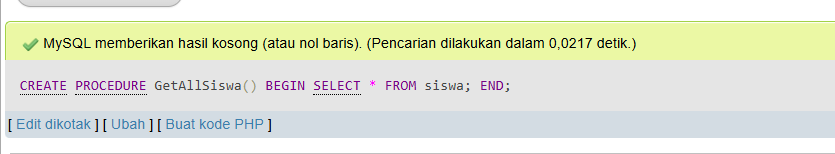


Table 2 (peminjaman)

DELIMITER $$

CREATE PROCEDURE GetAllPeminjaman()

BEGIN

SELECT \* FROM peminjaman;

END $$

DELIMITER ;

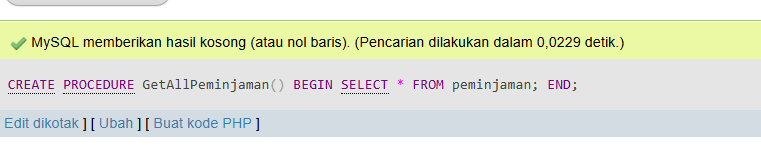


Table 3 (siswa)

DELIMITER $$

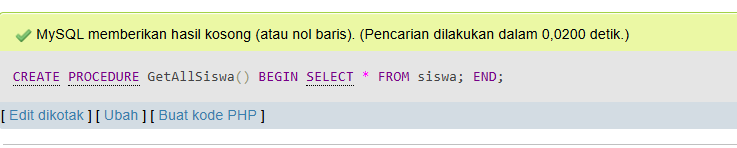
CREATE PROCEDURE GetAllSiswa()

BEGIN

SELECT \* FROM siswa;

END $$

DELIMITER ;



1. Stok buku pada saat dipinjam berkurang secara otamatis.

DELIMITER $$

CREATE TRIGGER KurangiStokBuku

AFTER INSERT ON peminjaman

FOR EACH ROW

BEGIN

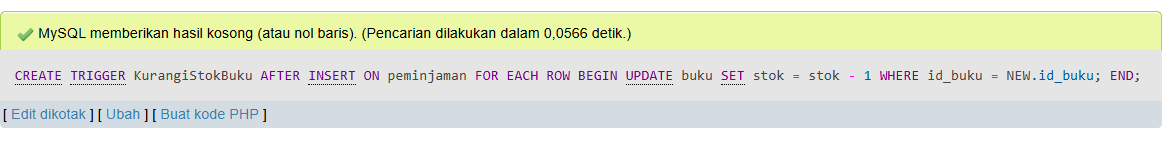
UPDATE buku

SET stok = stok - 1

WHERE id\_buku = NEW.id\_buku;

END $$

DELIMITER ;



1. Stok buku pada saat dikembalikan bertambah secara otomatis.

DELIMITER $$

CREATE TRIGGER TambahStokBuku

AFTER UPDATE ON peminjaman

FOR EACH ROW

BEGIN

IF NEW.status = 'Dikembalikan' AND OLD.status != 'Dikembalikan' THEN

UPDATE buku

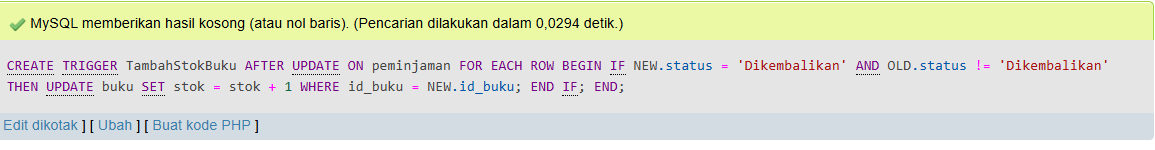
SET stok = stok + 1

WHERE id\_buku = NEW.id\_buku;

END IF;

END $$

DELIMITER ;



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

DELIMITER $$

CREATE PROCEDURE KembalikanBuku(

IN p\_id\_peminjaman INT

)

BEGIN

-- Update status peminjaman menjadi 'Dikembalikan' dengan tanggal kembali = CURRENT\_DATE

UPDATE peminjaman

SET status = 'Dikembalikan',

tanggal\_kembali = CURRENT\_DATE

WHERE id\_peminjaman = p\_id\_peminjaman;

-- Tambah stok buku kembali

UPDATE buku

SET stok = stok + 1

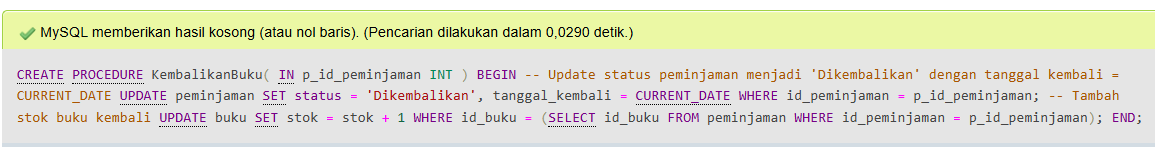
WHERE id\_buku = (SELECT id\_buku FROM peminjaman WHERE id\_peminjaman = p\_id\_peminjaman);

END $$

DELIMITER ;

Manggilnya

CALL KembalikanBuku(1);



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

DELIMITER $$

CREATE PROCEDURE DaftarSiswaPeminjam()

BEGIN

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

FROM siswa s

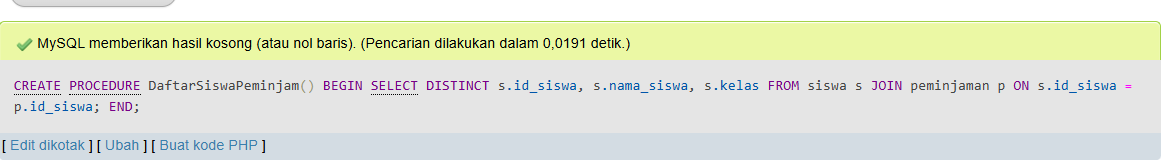
JOIN peminjaman p ON s.id\_siswa = p.id\_siswa;

END $$

DELIMITER ;

Cara manggilnya

CALL DaftarSiswaPeminjam();



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

DELIMITER $$

CREATE PROCEDURE DaftarSeluruhSiswa()

BEGIN

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

IFNULL(COUNT(p.id\_peminjaman), 0) AS total\_peminjaman

FROM siswa s

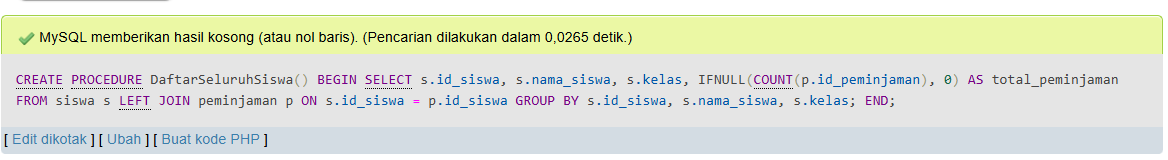
LEFT JOIN peminjaman p ON s.id\_siswa = p.id\_siswa

GROUP BY s.id\_siswa, s.nama\_siswa, s.kelas;

END $$

DELIMITER ;

Manggilnya : CALL DaftarSeluruhSiswa();



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

DELIMITER $$

CREATE PROCEDURE DaftarSeluruhBuku()

BEGIN

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

IFNULL(COUNT(p.id\_peminjaman), 0) AS total\_dipinjam

FROM buku b

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

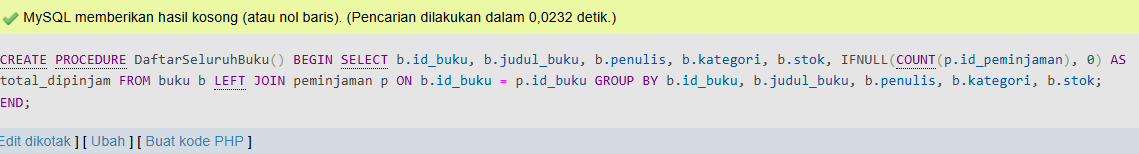
GROUP BY b.id\_buku, b.judul\_buku, b.penulis, b.kategori, b.stok;

END $$

DELIMITER ;

Manggilnya:

CALL DaftarSeluruhBuku();



**SUMYAR**

**XI RPL 1**

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