

## **PRAKTIKUM BASIS DATA**

Studi Kasus:

Aplikasi Pencatatan Data Inventaris Alat Sekolah



Dipersiapkan Oleh :  
**Kelas / Kelompok: C/Silent**

Darryl Azzuri - 223040138  
Muhammad Raihan Nugraha - 243040072  
Ilham Syaugi - 243040074  
Akbar Putra Pamungkas – 243040093

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS PASUNDAN**  
**BANDUNG**  
**2026**

## DAFTAR ISI

<b>DAFTAR ISI.....</b>	<b>1</b>
<b>BAB I PENDAHULUAN.....</b>	<b>2</b>
1.1 Latar Belakang.....	2
1.2 Kakas Yang Digunakan.....	3
1.3 Jobdesk Tim.....	4
<b>BAB II PEMBAHASAN.....</b>	<b>5</b>
2.1 Langkah Pengeraan.....	5
2.2 Schema Database.....	5
2.3 Pengeraan dan Pembahasan.....	8
2.3.1 Pembuatan & Pengisian Tabel.....	8
2.3.2 Pembuatan View.....	13
2.3.3. Pembuatan Procedure.....	28
2.3.4 Pembuatan Trigger.....	30
<b>BAB III PENUTUP.....</b>	<b>33</b>
3.1 Kendala Yang Dihadapi.....	33
3.2 Solusi Kendala.....	33
3.3 Kesimpulan.....	33
<b>LAMPIRAN.....</b>	<b>34</b>

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Pengelolaan data sarana dan prasarana pendidikan merupakan salah satu aktivitas vital dalam manajemen operasional sekolah, khususnya pada pengelolaan inventaris alat sekolah. Proses ini berkaitan langsung dengan kesiapan fasilitas belajar mengajar, pemeliharaan aset, serta ketepatan informasi yang dibutuhkan oleh pihak sekolah dan yayasan dalam perencanaan anggaran maupun pengambilan keputusan. Apabila pengelolaan data inventaris tidak dilakukan secara baik dan terstruktur, maka dapat menimbulkan berbagai permasalahan seperti hilangnya aset tanpa jejak, ketidaksesuaian jumlah fisik dengan catatan pembukuan, serta kesulitan dalam memantau kondisi kelayakan alat sekolah.

Pada praktiknya, masih banyak sekolah yang melakukan pencatatan data inventaris alat sekolah secara manual menggunakan buku induk atau semi-terkomputerisasi menggunakan aplikasi lembar kerja sederhana (spreadsheet). Data jenis barang, lokasi penempatan (ruang kelas/lab), data penanggung jawab, hingga riwayat kondisi barang (baik/rusak) sering kali dicatat dalam berkas yang terpisah tanpa sistem basis data yang terintegrasi. Kondisi ini menyebabkan proses penelusuran aset menjadi lambat, risiko duplikasi atau kehilangan data cukup tinggi, serta terjadinya kesalahan dalam pelaporan kekayaan aset sekolah, yang pada akhirnya menghambat efisiensi tata usaha sekolah.

Seiring dengan perkembangan teknologi informasi, pemanfaatan sistem basis data relasional menjadi solusi yang tepat untuk mengatasi permasalahan tersebut. Dengan adanya sistem yang terkomputerisasi, seluruh data inventaris dapat disimpan secara terpusat dan saling terhubung melalui relasi antar tabel. Selain itu, penggunaan fitur basis data seperti view untuk penyajian laporan aset per ruangan, store procedure untuk pengelolaan mutasi barang, serta trigger untuk pembaruan status kondisi barang secara otomatis dapat meningkatkan keakuratan data dan mempercepat proses audit inventaris.

Berdasarkan permasalahan tersebut, pada tugas besar ini dirancang sebuah Aplikasi Pencatatan Data Inventaris Alat Sekolah yang bertujuan untuk mengelola data alat sekolah, lokasi penempatan, kategori barang, serta staf penanggung jawab secara terintegrasi menggunakan basis data. Sistem ini diharapkan dapat meningkatkan efektivitas dan akuntabilitas pengelolaan aset sekolah, mempermudah pemantauan kondisi barang secara real-time, serta menyajikan informasi laporan inventaris yang valid sebagai dasar evaluasi

sarana prasarana. Selain itu, aplikasi ini juga menjadi bentuk penerapan konsep basis data pada studi kasus dunia nyata di lingkungan pendidikan.

## 1.2 Kakas Yang Digunakan

Pengembangan Aplikasi Pencatatan Data Inventaris Alat Sekolah ini melibatkan penggunaan sejumlah perangkat lunak pendukung yang disesuaikan dengan kebutuhan teknis. Pemanfaatan alat bantu (tools) tersebut mencakup seluruh rangkaian kerja, mulai dari fase perancangan skema basis data, eksekusi kode program, hingga dokumentasi laporan akhir. Hal ini bertujuan agar luaran yang dihasilkan terstruktur dengan baik dan mampu menjawab permasalahan yang ada. Rincian mengenai kakas beserta kegunaannya disajikan dalam tabel berikut.

No	Nama Kakas	Fungsi
1	SQL Server Management Studio 20	Mengimplementasikan rancangan basis data pada SQL Server. Fungsinya mencakup pembuatan struktur tabel dan relasi, pengembangan logika sistem (view, procedure, trigger), hingga tahap pengujian script SQL.
2	Google Docs	Media penyusunan laporan tugas besar secara sistematis.
3	Visual Studio Code	Penulisan dan penyimpanan kode SQL.
4	Git & Github	Manajemen kode antar anggota supaya perubahan kode tercatat dengan rapi dan sistematis.

## 1.3 Jobdesk Tim

Pada pelaksanaan tugas besar Aplikasi Pencatatan Data Inventaris Alat Sekolah, distribusi tugas dilakukan untuk memperjelas peran setiap anggota tim dalam proses perancangan maupun pembangunan sistem. Hal ini bertujuan agar koordinasi tim berjalan efektif dan hasil akhir yang diperoleh sesuai dengan kebutuhan. Tabel berikut menampilkan rincian pembagian tugas yang diemban oleh masing-masing anggota.

No	Nama	Tugas
1	Darryl Azzuri	<b>Database designer:</b> Menyusun desain konseptual basis data dalam bentuk ERD, memastikan struktur dan relasi data terdefinisi dengan jelas sebagai fondasi sistem, memperbaiki penggeraan

		masing-masing anggota, dan membuat laporan akhir.
2	Ilham Syaugi	<b>Data collector:</b> Menyusun data sampel dan melakukan proses populasi data ( <i>data entry</i> ) ke dalam basis data
3	Muhammad Raihan Nugraha	<b>Programmer:</b> Mengimplementasikan struktur tabel dan pembuatan <i>view</i> untuk penyajian data.
4	Akbar Putra Pamungkas	<b>Programmer:</b> Membuat <i>stored procedure</i> untuk transaksi dan <i>trigger</i> untuk respons data otomatis

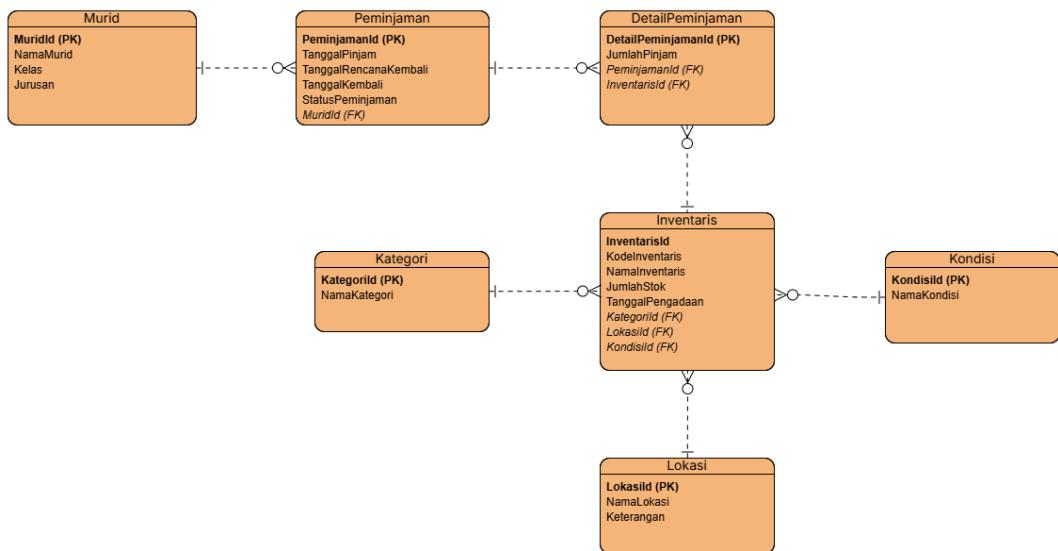
## BAB II

# PEMBAHASAN

### 2.1 Langkah Pengerjaan

1. Menentukan tugas untuk masing-masing anggota
2. Membuat repository untuk manajemen kode antar anggota
3. Membuat skema basis data (Database Designer)
4. Memasukkan 30 sampel data ke dalam setiap tabel (Data Collector)
5. Membuat 15 view (5 agregasi, 5 join, 5 bebas) (Programmer)
6. Membuat 3 procedure (Programmer)
7. Membuat 2 trigger (Programmer)
8. Menguji semua view, procedure, dan trigger yang sudah dibuat
9. Membuat laporan akhir

### 2.2 Schema Database



#### Penjelasan Schema Database:

##### 1. Murid

**Fungsi:** Menyimpan data identitas murid sebagai peminjam inventaris.

**Atribut:** MuridId, NamaMurid, Kelas, Jurusan

**Relasi:** Murid → Peminjaman

**Tipe Relasi:** One-to-Many (1:N)

**Penjelasan:** Satu murid dapat melakukan beberapa transaksi peminjaman inventaris.

**Foreign Key:** Tidak memiliki foreign key

## 2. Peminjaman

**Fungsi:** Mencatat transaksi peminjaman inventaris oleh murid.

**Atribut:** PeminjamanId, TanggalPinjam, TanggalRencanaKembali, TanggalKembali, StatusPeminjaman, MuridId

**Relasi:** Peminjaman → Murid, Peminjaman → DetailPeminjaman

**Tipe Relasi:** Many-to-One (N:1) ke Murid, One-to-Many (1:N) ke DetailPeminjaman

**Penjelasan:** Setiap peminjaman dilakukan oleh satu murid dan dapat terdiri dari beberapa barang.

**Foreign Key:** MuridId

## 3. DetailPeminjaman

**Fungsi:** Menyimpan rincian barang yang dipinjam dalam satu transaksi.

**Atribut:** DetailPeminjamanId, JumlahPinjam, PeminjamanId, InventarisId

**Relasi:** DetailPeminjaman → Peminjaman, DetailPeminjaman → Inventaris

**Tipe Relasi:** Many-to-One (N:1)

**Penjelasan:** Tabel penghubung antara peminjaman dan inventaris.

**Foreign Key:** PeminjamanId, InventarisId

## 4. Inventaris

**Fungsi:** Menyimpan data barang inventaris sekolah.

**Atribut:** InventarisId, KodeInventaris, NamaInventaris, JumlahStok, TanggalPengadaan, KategoriId, LokasiId, KondisiId

**Relasi:** Inventaris → DetailPeminjaman, Inventaris → KategoriInventaris, Inventaris → LokasiPenyimpanan, Inventaris → KondisiInventaris

**Tipe Relasi:** Many-to-One (N:1), One-to-Many (1:N)

**Penjelasan:** Setiap barang inventaris memiliki kategori, lokasi, dan kondisi tertentu.

**Foreign Key:** KategoriId, LokasiId, KondisiId

## 5. KategoriInventaris

**Fungsi:** Mengelompokkan inventaris berdasarkan kategorinya.

**Atribut:** KategoriId, NamaKategori

**Relasi:** KategoriInventaris → Inventaris

**Tipe Relasi:** One-to-Many (1:N)

**Penjelasan:** Satu kategori dapat memiliki banyak inventaris.

**Foreign Key:** Tidak memiliki foreign key

## 6. LokasiPenyimpanan

**Fungsi:** Menyimpan informasi lokasi penyimpanan inventaris.

**Atribut:** LokasiId, NamaLokasi, Keterangan

**Relasi:** LokasiPenyimpanan → Inventaris

**Tipe Relasi:** One-to-Many (1:N)

**Penjelasan:** Setiap lokasi dapat menyimpan banyak inventaris.

**Foreign Key:** Tidak memiliki foreign key

## 7. KondisiInventaris

**Fungsi:** Menyimpan status kondisi fisik inventaris.

**Atribut:** KondisiId, NamaKondisi

**Relasi:** KondisiInventaris → Inventaris

**Tipe Relasi:** One-to-Many (1:N)

**Penjelasan:** Menunjukkan kondisi barang untuk keperluan monitoring dan perawatan.

**Foreign Key:** Tidak memiliki foreign key

## 2.3 Pengerjaan dan Pembahasan

### 2.3.1 Pembuatan & Pengisian Tabel

#### 1. Tabel Murid

```
-- Menyimpan data murid yang dapat melakukan peminjaman
CREATE TABLE Murid (
    MuridId INT IDENTITY(1,1) PRIMARY KEY,
    NamaMurid VARCHAR(100) NOT NULL,
    Kelas VARCHAR(10) NOT NULL,
    Jurusan VARCHAR(10) NOT NULL
);
```

```
-- Data Murid
INSERT INTO Murid (NamaMurid, Kelas, Jurusan) VALUES
('Ilham Saugi', 'XII', 'IPA'),
('Muhamad Nur Salam', 'XII', 'IPS'),
('Afsal Prima Maula', 'X', 'IPA'),
('Billy Wicaksono', 'XI', 'IPS'),
('Muhamad Phatur Rahman', 'XI', 'IPA'),
('Yanto Santoso', 'XII', 'IPS'),
('Riski Pratama', 'XII', 'IPA'),
('Muhammad Raehan', 'X', 'IPS'),
('Akbar Putra Pamungkas', 'XI', 'IPA'),
('Siti Maimunah', 'XI', 'IPS'),
('Sabrina Azizah', 'XII', 'IPA'),
('Ali Fauzi', 'XII', 'IPS'),
('Muhammad Thoriq', 'X', 'IPA'),
('Rifky Pratama', 'XI', 'IPS'),
('Khoirun Rozan', 'XI', 'IPA'),
('Reza Putra', 'XII', 'IPS'),
('Raffi Ahmad', 'XII', 'IPA'),
('Bahlil Ilahi', 'X', 'IPS'),
('Rafli Rahman', 'XI', 'IPA'),
('Rafii Khoirun', 'XI', 'IPS'),
('Arneu Samsul', 'XII', 'IPA'),
('Vicka Aulia', 'XII', 'IPS'),
('Nazala Qisti', 'X', 'IPA'),
('Areva Munira', 'XI', 'IPS'),
('Aditya Frahaz', 'XI', 'IPA'),
('Daniel Aditya', 'XII', 'IPS'),
('Denis Figo', 'XII', 'IPA'),
('Prima Maula', 'X', 'IPS'),
('Pratama Arhan', 'XI', 'IPA'),
('Farel Rahmat', 'XI', 'IPS');
```

Data Collector melakukan proses pengisian data pada tabel **Murid**. Data yang dimasukkan berjumlah tiga puluh data murid dengan informasi utama berupa MuridId, NamaMurid, Kelas, dan Jurusan.

## 2. Tabel Peminjaman

```
CREATE TABLE Peminjaman (
    PeminjamanId INT IDENTITY(1,1) PRIMARY KEY,
    TanggalPinjam DATE NOT NULL,
    TanggalRencanaKembali DATE NOT NULL,
    TanggalKembali DATE NULL,
    MuridId INT NOT NULL,
    StatusPeminjaman VARCHAR(30) NOT NULL,
    CONSTRAINT FK_Peminjaman_Murid
        FOREIGN KEY (MuridId) REFERENCES Murid(MuridId)
);
```

```
-- Data Peminjaman
INSERT INTO Peminjaman (TanggalPinjam, TanggalRencanaKembali,
TanggalKembali, MuridId, StatusPeminjaman) VALUES
('2024-09-15', '2024-09-29', '2024-09-29', 5, 'Selesai'),
('2024-09-20', '2024-10-04', '2024-10-04', 12, 'Selesai'),
('2024-10-05', '2024-10-19', '2024-10-21', 8, 'Terlambat'),
('2024-10-12', '2024-10-26', '2024-10-26', 23, 'Selesai'),
('2024-10-18', '2024-11-01', '2024-11-01', 3, 'Selesai'),
('2024-11-02', '2024-11-16', '2024-11-16', 17, 'Selesai'),
('2024-11-08', '2024-11-22', '2024-11-25', 29, 'Terlambat'),
('2024-11-15', '2024-11-29', '2024-11-29', 6, 'Selesai'),
('2024-11-22', '2024-12-06', '2024-12-06', 14, 'Selesai'),
('2024-11-28', '2024-12-12', '2024-12-12', 21, 'Selesai'),
('2024-12-03', '2024-12-17', '2024-12-17', 11, 'Selesai'),
('2024-12-09', '2024-12-23', '2024-12-26', 25, 'Terlambat'),
('2024-12-15', '2024-12-29', NULL, 4, 'Dipinjam'),
('2024-12-18', '2025-01-01', NULL, 19, 'Dipinjam'),
('2024-12-22', '2025-01-05', NULL, 7, 'Dipinjam'),
('2024-12-26', '2025-01-09', NULL, 16, 'Dipinjam'),
('2024-12-29', '2025-01-12', NULL, 28, 'Dipinjam'),
('2025-01-03', '2025-01-17', NULL, 2, 'Dipinjam'),
('2025-01-07', '2025-01-21', NULL, 13, 'Dipinjam'),
('2025-01-10', '2025-01-24', NULL, 22, 'Dipinjam'),
('2025-01-14', '2025-01-28', NULL, 9, 'Dipinjam'),
('2025-01-17', '2025-01-31', NULL, 30, 'Dipinjam'),
('2025-01-21', '2025-02-04', NULL, 1, 'Dipinjam'),
('2025-01-24', '2025-02-07', NULL, 15, 'Dipinjam'),
('2025-01-28', '2025-02-11', NULL, 26, 'Dipinjam'),
('2025-02-01', '2025-02-15', NULL, 10, 'Dipinjam'),
('2025-02-05', '2025-02-19', NULL, 18, 'Dipinjam'),
('2025-02-09', '2025-02-23', NULL, 24, 'Dipinjam'),
('2025-02-12', '2025-02-26', NULL, 20, 'Dipinjam'),
('2025-02-16', '2025-03-02', NULL, 27, 'Dipinjam');
```

Data Collector melakukan proses pengisian data pada tabel **Peminjaman**. Data yang dimasukkan berjumlah tiga puluh data peminjaman dengan informasi utama berupa MuridId, TanggalPinjam, TanggalRencanaKembali, TanggalKembali, dan StatusPeminjaman.

### 3. Tabel DetailPeminjaman

```
CREATE TABLE DetailPeminjaman (
    DetailPeminjamanId INT IDENTITY(1,1) PRIMARY KEY,
    PeminjamanId INT NOT NULL,
    InventarisId INT NOT NULL,
    JumlahPinjam INT NOT NULL,
    CONSTRAINT FK_DetailPeminjaman_Peminjaman
    | FOREIGN KEY (PeminjamanId) REFERENCES Peminjaman(PeminjamanId),
    CONSTRAINT FK_DetailPeminjaman_Inventaris
    | FOREIGN KEY (InventarisId) REFERENCES Inventaris(InventarisId)
);

-- Data Detail Peminjaman
INSERT INTO DetailPeminjaman (PeminjamanId, InventarisId, JumlahPinjam) VALUES
(1, 3, 2),(2, 15, 1),(3, 7, 1),(4, 22, 1),(5, 11, 3),
(6, 5, 2),(7, 18, 1),(8, 9, 1),(9, 26, 1),(10, 14, 2),
(11, 6, 4),(12, 28, 1),(13, 19, 5),(14, 8, 1),(15, 23, 1),
(16, 12, 2),(17, 19, 1),(18, 6, 1),(19, 17, 2),(20, 4, 3),
(21, 25, 1),(22, 13, 1),(23, 27, 1),(24, 12, 2),(25, 20, 1),
(26, 10, 1),(27, 16, 1),(28, 24, 1),(29, 21, 2),(24, 27, 1);
```

Data Collector melakukan proses pengisian data pada tabel DetailPeminjaman. Data yang dimasukkan berjumlah tiga puluh data detail peminjaman dengan informasi utama berupa PeminjamanId, InventarisId, dan JumlahPinjam.

#### 4. Tabel Inventaris

```

CREATE TABLE Inventaris (
    InventarisId INT IDENTITY(1,1) PRIMARY KEY,
    KodeInventaris VARCHAR(30) NOT NULL UNIQUE,
    NamaInventaris VARCHAR(100) NOT NULL,
    KategoriId INT NOT NULL,
    LokasiId INT NOT NULL,
    KondisiId INT NOT NULL,
    TanggalPengadaan DATE NOT NULL,
    JumlahStok INT NOT NULL,
    StatusInventaris VARCHAR(30) NOT NULL,

    CONSTRAINT FK_Inventaris_Kategori
        FOREIGN KEY (KategoriId)
        REFERENCES KategoriInventaris(KategoriId),

    CONSTRAINT FK_Inventaris_Lokasi
        FOREIGN KEY (LokasiId)
        REFERENCES LokasiPenyimpanan(LokasiId),

    CONSTRAINT FK_Inventaris_Kondisi
        FOREIGN KEY (KondisiId)
        REFERENCES KondisiInventaris(KondisiId)
);

-- Data Inventaris
INSERT INTO Inventaris (KodeInventaris, NamaInventaris, KategoriId, LokasiId, KondisiId, TanggalPengadaan, JumlahStok, StatusInventaris) VALUES
-- Peralatan Olahraga (KategoriId = 1) - Lokasi: Ruang Olahraga (LokasiId = 3)
('P0001', 'Bola Basket Spalding', 1, 3, 1, '2024-09-15', 8, 'Tersedia'),
('P0002', 'Matras Yoga', 1, 3, 1, '2023-10-10', 10, 'Tersedia'),
('P0003', 'Bola Voli Molten', 1, 3, 1, '2023-09-01', 12, 'Tersedia'),
('P0004', 'Net Badminton', 1, 3, 2, '2024-03-20', 5, 'Tersedia'),
('P0005', 'Raket Tenis Meja', 1, 3, 1, '2024-07-10', 15, 'Tersedia'),

-- Peralatan Musik (KategoriId = 2) - Lokasi: Ruang Musik (LokasiId = 5)
('PM001', 'Gitar Akustik Yamaha', 2, 5, 1, '2024-08-12', 4, 'Tersedia'),
('PM002', 'Keyboard Roland', 2, 5, 1, '2023-08-20', 2, 'Tersedia'),
('PM003', 'Drum Set Pearl', 2, 5, 2, '2024-02-15', 1, 'Tersedia'),
('PM004', 'Microphone Shure', 2, 5, 1, '2024-05-18', 6, 'Tersedia'),

-- Peralatan Laboratorium (KategoriId = 3) - Lokasi: Lab Kimia (LokasiId = 4)
('PL001', 'Mikroskop Cahaya', 3, 4, 1, '2022-12-05', 15, 'Tersedia'),
('PL002', 'Tabung Reaksi', 3, 4, 2, '2022-11-20', 100, 'Tersedia'),
('PL003', 'Bunsen Burner', 3, 4, 1, '2023-05-15', 10, 'Tersedia'),
('PL004', 'Gelas Ukur 500ml', 3, 4, 1, '2023-08-22', 25, 'Tersedia'),
('PL005', 'Pipet Tetes', 3, 4, 2, '2024-01-10', 50, 'Tersedia'),

-- Peralatan Seni Rupa (KategoriId = 4) - Lokasi: Ruang Lukis (LokasiId = 1)
('PSR001', 'Kanvas Lukis A3', 4, 1, 1, '2023-01-10', 50, 'Tersedia'),
('PSR002', 'Cat Akrilik Set', 4, 1, 1, '2024-04-05', 20, 'Tersedia'),
('PSR003', 'Kuas Lukis Set', 4, 1, 2, '2024-06-18', 30, 'Tersedia'),
('PSR004', 'Easel Kayu', 4, 1, 1, '2023-11-25', 10, 'Tersedia'),

-- Peralatan UKS (KategoriId = 5) - Lokasi: Ruang UKS (LokasiId = 6)
('PU001', 'Stetoskop', 5, 6, 1, '2022-10-15', 5, 'Tersedia'),
('PU002', 'Timbangan Badan', 5, 6, 2, '2022-09-30', 2, 'Tersedia'),
('PU003', 'Tensimeter Digital', 5, 6, 1, '2024-03-12', 3, 'Tersedia'),
('PU004', 'Tempat Tidur Pasien', 5, 6, 2, '2023-07-20', 4, 'Tersedia'),

```

Data Collector melakukan proses pengisian data pada tabel **Inventaris**. Data yang dimasukkan berjumlah tiga puluh data inventaris dengan informasi utama berupa KodeInventaris, NamaInventaris, JumlahStok, TanggalPengadaan, KategoriId, LokasiId, dan KondisiId.

## 5. Tabel KategoriInventaris

```
CREATE TABLE KategoriInventaris (
    KategoriId INT IDENTITY(1,1) PRIMARY KEY,
    NamaKategori VARCHAR(50) NOT NULL
);

-- Data Kategori Inventaris
INSERT INTO KategoriInventaris (NamaKategori) VALUES
('Peralatan Olahraga'),
('Peralatan Musik'),
('Peralatan Laboratorium'),
('Peralatan Seni Rupa'),
('Peralatan UKS'),
('Peralatan Masak'),
('Peralatan Elektronik');
```

Data Collector melakukan proses pengisian data pada tabel **KategoriInventaris**. Data yang dimasukkan berjumlah tiga puluh data kategori inventaris dengan informasi utama berupa NamaKategori.

## 6. Tabel LokasiPenyimpanan

```
CREATE TABLE LokasiPenyimpanan (
    LokasiId INT IDENTITY(1,1) PRIMARY KEY,
    NamaLokasi VARCHAR(50) NOT NULL,
    Keterangan VARCHAR(100)
);

-- Data Lokasi Penyimpanan
INSERT INTO LokasiPenyimpanan (NamaLokasi, Keterangan) VALUES
('Ruang Lukis', 'Lantai 1 Gedung A'),
('Lab Komputer', 'Lantai 2 Gedung A'),
('Ruang Olahraga', 'Samping Lapangan'),
('Lab Kimia', 'Lantai 1 Tengah'),
('Ruang Musik', 'Lantai 2 Gedung A'),
('Ruang UKS', 'Lantai 1 Gedung A'),
('Kantin', 'Lantai 2 Gedung B');
```

Data Collector melakukan proses pengisian data pada tabel **LokasiPenyimpanan**. Data yang dimasukkan berjumlah tiga puluh data lokasi penyimpanan dengan informasi utama berupa NamaLokasi dan Keterangan.

## 7. Tabel KondisiInventaris

```
CREATE TABLE KondisiInventaris (
    KondisiId INT IDENTITY(1,1) PRIMARY KEY,
    NamaKondisi VARCHAR(25) NOT NULL
);

-- Data Kondisi Inventaris
INSERT INTO KondisiInventaris (NamaKondisi) VALUES
('Baik'),
('Rusak Ringan'),
('Rusak Berat');
```

Data Collector melakukan proses pengisian data pada tabel **KondisiInventaris**. Data yang dimasukkan berjumlah tiga puluh data kondisi inventaris dengan informasi utama berupa NamaKondisi.

### 2.3.2 Pembuatan View

View merupakan representasi virtual dari hasil query yang dirancang untuk mempercepat proses pengambilan data laporan. Pada sistem ini, terdapat 15 view yang dikembangkan untuk memvisualisasikan data inventaris dan operasional sekolah secara efisien. Kategori view yang dibuat mencakup penggabungan relasi antar-tabel (join) dan rekапitulasi jumlah (agregasi).

#### 1. View Stok Per Kategori (Agregasi)

```
● ● ●
1 CREATE OR ALTER VIEW V_LaporanStokKategori
2 AS
3 SELECT
4     k.KategoriId,
5     k.NamaKategori,
6     COUNT(i.InventarisId) AS JumlahSubkategori,
7     SUM(i.JumlahStok) AS TotalStok
8     FROM KategoriInventaris k
9     LEFT JOIN Inventaris i
10        ON k.KategoriId = i.KategoriId
11    GROUP BY
12        k.KategoriId,
13        k.NamaKategori;
14
15    SELECT *
16    FROM V_LaporanStokKategori
```



	KategoriId	NamaKategori	JumlahSubkategori	TotalStok
1	1	Peralatan Olahraga	5	50
2	2	Peralatan Musik	4	13
3	3	Peralatan Laboratorium	5	200
4	4	Peralatan Seni Rupa	4	110
5	5	Peralatan UKS	4	14
6	6	Peralatan Masak	4	22
7	7	Peralatan Elektronik	5	36

### Penjelasan:

View **V\_LaporanStokKategori** digunakan untuk menampilkan laporan jumlah inventaris berdasarkan kategori. Data diambil dari tabel KategoriInventaris dan Inventaris dengan tujuan mengetahui jumlah jenis barang serta total stok fisik pada setiap kategori.

View ini menggunakan LEFT JOIN agar seluruh kategori tetap ditampilkan meskipun belum memiliki data inventaris. Proses agregasi dilakukan menggunakan fungsi COUNT untuk menghitung jumlah jenis barang dan SUM untuk menghitung total stok inventaris. Data dikelompokkan berdasarkan kategori sehingga setiap baris mewakili satu kategori inventaris.

## 2. View Ranking Murid Berdasarkan Frekuensi Peminjaman (Agregasi)

```
● ● ●
1 CREATE VIEW V_RankingPeminjam
2 AS
3 SELECT
4     m.MuridId,
5     m.NamaMurid,
6     m.Kelas,
7     COUNT(p.PeminjamanId) AS TotalKaliPinjam
8 FROM Murid m
9 LEFT JOIN Peminjaman p
10    ON m.MuridId = p.MuridId
11 GROUP BY
12     m.MuridId,
13     m.NamaMurid,
14     m.Kelas;
15
16 SELECT *
17 FROM V_RankingPeminjam
```

The screenshot shows the SQL Server Management Studio interface. At the top, there are three colored status indicators (red, yellow, green). Below them is the SQL query for creating the view. At the bottom, there are two tabs: 'Results' and 'Messages'. The 'Results' tab displays the output of the query, which is a table showing student names, their class, and the total number of loans they have taken.

	MuridId	NamaMurid	Kelas	TotalKaliPinjam
1	1	Ilham Saugi	XII	1
2	2	Muhamad Nur Salam	XII	1
3	3	Afsal Prima Maula	X	1
4	4	Billy Wicaksono	XI	1
5	5	Muhamad Phatur Rahman	XI	1
6	6	Yanto Santoso	XII	1
7	7	Riski Pratama	XII	1
8	8	Muhammad Raehan	X	1
9	9	Akbar Putra Pamungkas	XI	1
10	10	Siti Maimunah	XI	1
11	11	Sabrina Azizah	XII	1
12	12	Ali Fauzi	XII	1
13	13	Muhammad Thoriq	X	1
14	14	Rifky Pratama	XI	1
15	15	Khoirun Rozan	XI	1
16	16	P. P. L.	VII	1

### Penjelasan:

View **V\_RankingPeminjam** digunakan untuk menampilkan daftar murid berdasarkan frekuensi peminjaman inventaris yang dilakukan. Data diambil dari tabel Murid dan Peminjaman untuk menghitung jumlah transaksi peminjaman setiap murid.

View ini menggunakan LEFT JOIN agar seluruh data murid tetap ditampilkan, termasuk murid yang belum pernah melakukan peminjaman. Proses agregasi dilakukan dengan fungsi COUNT untuk menghitung total peminjaman, kemudian data dikelompokkan berdasarkan identitas murid sehingga setiap baris merepresentasikan satu murid beserta jumlah peminjamannya.

### 3. View Barang yang Paling Sering Dipinjam (Agregasi)

```
● ● ●
1 CREATE OR ALTER VIEW V_BarangTerlaris AS
2 SELECT
3     i.InventarisId,
4     i.NamaInventaris,
5     SUM(dp.JumlahPinjam) AS TotalUnitDipinjam
6 FROM Inventaris i
7 JOIN DetailPeminjaman dp
8     ON i.InventarisId = dp.InventarisId
9 GROUP BY
10    i.InventarisId,
11    i.NamaInventaris;
12
13 SELECT *
14 FROM V_BarangTerlaris
```



	InventarisId	NamaInventaris	TotalUnitDipinjam
1	3	Bola Voli Molten	2
2	4	Net Badminton	3
3	5	Raket Tenis Meja	2
4	6	Gitar Akustik Yamaha	5
5	7	Keyboard Roland	1
6	8	Drum Set Pearl	1
7	9	Microphone Shure	1
8	10	Mikroskop Cahaya	1
9	11	Tabung Reaksi	3
10	12	Bunsen Burner	4
11	13	Gelas Ukur 500ml	1
12	14	Pipet Tetes	2
13	15	Kanvas Lukis A3	1
14	16	Cat Akrilik Set	1
15	17	Kuas Lukis Set	2

### Penjelasan:

View **V\_BaRangTerlaris** digunakan untuk menampilkan daftar inventaris yang paling sering dipinjam berdasarkan total unit peminjaman. Data diperoleh dari tabel Inventaris dan DetailPeminjaman.

View ini menggunakan INNER JOIN sehingga hanya barang yang pernah dipinjam yang akan ditampilkan. Proses agregasi dilakukan dengan fungsi SUM untuk menghitung total jumlah unit yang dipinjam dari setiap barang. Data kemudian dikelompokkan berdasarkan identitas inventaris sehingga setiap baris menunjukkan satu jenis barang beserta total peminjamannya.

#### 4. View Barang Per Lokasi (Agregasi)

```
● ● ●

1 CREATE OR ALTER VIEW V_AsetPerLokasi AS
2 SELECT
3     l.LokasiId,
4     l.NamaLokasi,
5     COUNT(i.InventarisId) AS JenisBarang,
6     SUM(i.JumlahStok) AS TotalAset
7 FROM LokasiPenyimpanan l
8 LEFT JOIN Inventaris i
9     ON l.LokasiId = i.LokasiId
10 GROUP BY
11     l.LokasiId,
12     l.NamaLokasi;
13
14 SELECT *
15 FROM V_AsetPerLokasi
```

	LokasiId	NamaLokasi	JenisBarang	TotalAset
1	1	Ruang Lukis	4	110
2	2	Lab Komputer	5	36
3	3	Ruang Olahraga	5	50
4	4	Lab Kimia	5	200
5	5	Ruang Musik	4	13
6	6	Ruang UKS	4	14
7	7	Kantin	4	22

### Penjelasan:

View **V\_AsetPerLokasi** digunakan untuk menampilkan informasi jumlah inventaris berdasarkan lokasi penyimpanan. Data diambil dari tabel LokasiPenyimpanan dan Inventaris untuk mengetahui jumlah jenis barang serta total aset pada setiap lokasi.

View ini menggunakan LEFT JOIN agar seluruh lokasi tetap ditampilkan, termasuk lokasi yang belum memiliki inventaris. Proses agregasi dilakukan menggunakan fungsi COUNT untuk menghitung jumlah jenis barang dan SUM untuk menghitung total stok fisik. Data dikelompokkan berdasarkan lokasi sehingga setiap baris merepresentasikan satu lokasi penyimpanan.

## 5. View Status yang Meminjam (Agregasi)

```
● ● ●

1 --view status yang meminjam
2 CREATE OR ALTER VIEW V_MonitoringStatus AS
3 SELECT
4     StatusPeminjaman,
5     COUNT(*) AS JumlahPeminjaman
6 FROM Peminjaman
7 GROUP BY StatusPeminjaman;
8
9 SELECT *
10 FROM V_MonitoringStatus;
```



	StatusPeminjaman	JumlahPeminjaman
1	Dipinjam	18
2	Selesai	9
3	Terlambat	3

### Penjelasan:

View **V\_MonitoringStatus** digunakan untuk menampilkan ringkasan jumlah peminjaman berdasarkan status peminjaman. Data diambil dari tabel Peminjaman untuk mengetahui distribusi transaksi pada setiap status, seperti *Dipinjam*, *Selesai*, atau *Terlambat*.

View ini menggunakan fungsi agregasi COUNT untuk menghitung jumlah transaksi pada masing-masing status. Data kemudian dikelompokkan berdasarkan status peminjaman sehingga setiap baris merepresentasikan satu kategori status beserta total peminjamannya.

## 6. View Data Barang Lengkap (Join)

```
CREATE OR ALTER VIEW V_InventarisLengkap AS
SELECT
    i.KodeInventaris,
    i.NamaInventaris,
    k.NamaKategori,
    l.NamaLokasi,
    ko.NamaKondisi,
    i.JumlahStok
FROM Inventaris i
JOIN KategoriInventaris k ON i.KategoriID = k.KategoriID
JOIN LokasiPenyimpanan l ON i.LokasiId = l.LokasiID
JOIN KondisiInventaris ko ON i.KondisiId = ko.KondisiID;
SELECT *
FROM V_InventarisLengkap;
```

KodeInventaris	NamaInventaris	NamaKategori	NamaLokasi	NamaKondisi	JumlahStok
P0001	Bola Basket Spalding	Peralatan Olahraga	Ruang Olahraga	Baik	8
P0002	Matras Yoga	Peralatan Olahraga	Ruang Olahraga	Baik	10
P0003	Bola Vol Molten	Peralatan Olahraga	Ruang Olahraga	Baik	12
P0004	Net Badminton	Peralatan Olahraga	Ruang Olahraga	Rusak Ringan	5
P0005	Raket Tenis Meja	Peralatan Olahraga	Ruang Olahraga	Baik	15
PM001	Gitar Akustik Yamaha	Peralatan Musik	Ruang Musik	Baik	4
PM002	Keyboard Roland	Peralatan Musik	Ruang Musik	Baik	2
PM003	Drum Set Pearl	Peralatan Musik	Ruang Musik	Rusak Ringan	1
PM004	Microphone Shure	Peralatan Musik	Ruang Musik	Baik	6
PL001	Mikroskop Cahaya	Peralatan Laboratorium	Lab Kimia	Baik	15
PL002	Tabung Reaksi	Peralatan Laboratorium	Lab Kimia	Rusak Ringan	100
PL003	Bunsen Burner	Peralatan Laboratorium	Lab Kimia	Baik	10
PL004	Gelas Ukur 500ml	Peralatan Laboratorium	Lab Kimia	Baik	25
PL005	Pipet Tetes	Peralatan Laboratorium	Lab Kimia	Rusak Ringan	50

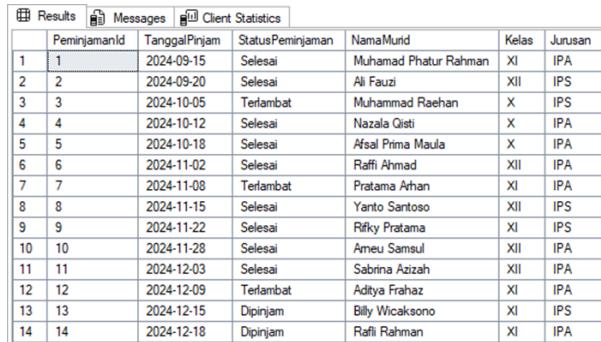
### Penjelasan:

View **V\_InventarisLengkap** digunakan untuk menampilkan informasi inventaris secara lengkap dan mudah dibaca. Data diambil dari tabel Inventaris yang digabungkan dengan tabel KategoriInventaris, LokasiPenyimpanan, dan KondisiInventaris.

View ini bertujuan menggantikan tampilan data berbasis ID menjadi informasi deskriptif berupa nama kategori, lokasi, dan kondisi barang. Dengan demikian, view ini memudahkan proses monitoring dan pelaporan inventaris karena seluruh informasi penting ditampilkan dalam satu tampilan terintegrasi.

## 7. View Orang yang Meminjam (Join)

```
1 CREATE OR ALTER VIEW V_InfoPeminjam AS
2 SELECT
3     p.PeminjamanId,
4     p.TanggalPinjam,
5     p.StatusPeminjaman,
6     m.NamaMurid,
7     m.Kelas,
8     m.Jurusan
9 FROM Peminjaman p
10 JOIN Murid m ON p.MuridId = m.MuridId;
11
12 SELECT *
13 FROM V_InfoPeminjam;
```



	PeminjamanId	TanggalPinjam	StatusPeminjaman	NamaMurid	Kelas	Jurusan
1	1	2024-09-15	Selesai	Muhamad Phatur Rahman	XI	IPA
2	2	2024-09-20	Selesai	Ali Fauzi	XII	IPS
3	3	2024-10-05	Terlambat	Muhammad Rahehan	X	IPS
4	4	2024-10-12	Selesai	Nazala Gisti	X	IPA
5	5	2024-10-18	Selesai	Afsal Prima Maula	X	IPA
6	6	2024-11-02	Selesai	Raffi Ahmad	XII	IPA
7	7	2024-11-08	Terlambat	Pratama Arhan	XI	IPA
8	8	2024-11-15	Selesai	Yanto Santoso	XII	IPS
9	9	2024-11-22	Selesai	Rifky Pratama	XI	IPS
10	10	2024-11-28	Selesai	Ameu Samsul	XII	IPA
11	11	2024-12-03	Selesai	Sabrina Azizah	XII	IPA
12	12	2024-12-09	Terlambat	Aditya Frahaz	XI	IPA
13	13	2024-12-15	Dipinjam	Billy Wicaksono	XI	IPS
14	14	2024-12-18	Dipinjam	Rafli Rahman	XI	IPA

### Penjelasan:

View **V\_InfoPeminjam** digunakan untuk menampilkan informasi peminjaman beserta identitas murid yang melakukan peminjaman. Data diambil dari tabel Peminjaman dan Murid untuk menggabungkan informasi transaksi dengan profil peminjam.

View ini menggunakan INNER JOIN agar setiap data peminjaman yang ditampilkan selalu memiliki data murid yang valid. Dengan adanya view ini, proses pemantauan aktivitas peminjaman dapat dilakukan dengan lebih mudah karena informasi peminjaman dan identitas murid ditampilkan secara terintegrasi.

## 8. View Barang Apa Saja yang Dipinjam (Join)

```
● ● ●  
1 CREATE OR ALTER VIEW V_DetailBarangPinjam AS  
2 SELECT  
3     dp.PeminjamanId,  
4     i.KodeInventaris,  
5     i.NamaInventaris,  
6     dp.JumlahPinjam  
7 FROM DetailPeminjaman dp  
8 JOIN Inventaris i  
9     ON dp.InventarisId = i.InventarisId;  
10  
11 SELECT *  
12 FROM V_DetailBarangPinjam;
```

	PeminjamanId	KodeInventaris	NamaInventaris	JumlahPinjam
1	1	PO003	Bola Voli Molten	2
2	2	PSR001	Kanvas Lukis A3	1
3	3	PM002	Keyboard Roland	1
4	4	PU004	Tempat Tidur Pasien	1
5	5	PL002	Tabung Reaksi	3
6	6	PO005	Raket Tenis Meja	2
7	7	PSR004	Easel Kayu	1
8	8	PM004	Microphone Shure	1
9	9	PMK004	Rice Cooker Miyako	1
10	10	PL005	Pipet Tetes	2
11	11	PM001	Gitar Akustik Yamaha	4
12	12	PE002	Proyektor Epson	1
13	13	PU001	Stetoskop	5
14	14	PM003	Drum Set Pearl	1

### Penjelasan:

View **V\_DetailBarangPinjam** digunakan untuk menampilkan rincian barang yang dipinjam dalam setiap transaksi peminjaman. Data diambil dari tabel **DetailPeminjaman** dan **Inventaris** untuk menggabungkan informasi transaksi dengan data deskriptif barang.

View ini menampilkan informasi berupa ID peminjaman, kode inventaris, nama barang, serta jumlah unit yang dipinjam. Dengan adanya view ini, pengguna dapat mengetahui secara detail barang apa saja yang terlibat dalam setiap transaksi peminjaman inventaris.

## 9. View Melihat Lokasi Barang (Join)

```
● ● ●  
1 CREATE OR ALTER VIEW V_LokasiBarang AS  
2 SELECT  
3     l.NamaLokasi,  
4     l.Keterangan AS InfoRuangan,  
5     i>NamaInventaris,  
6     i.JumlahStok  
7 FROM LokasiPenyimpanan l  
8 JOIN Inventaris i ON l.LokasiId = i.LokasiId;  
9  
10 SELECT *  
11 FROM V_LokasiBarang;  
12
```

Results				
	NamaLokasi	InfoRuangan	Namalnventaris	JumlahStok
1	Ruang Olahraga	Samping Lapangan	Bola Basket Spalding	8
2	Ruang Olahraga	Samping Lapangan	Matras Yoga	10
3	Ruang Olahraga	Samping Lapangan	Bola Voli Molten	12
4	Ruang Olahraga	Samping Lapangan	Net Badminton	5
5	Ruang Olahraga	Samping Lapangan	Raket Tenis Meja	15
6	Ruang Musik	Lantai 2 Gedung A	Gitar Akustik Yamaha	4
7	Ruang Musik	Lantai 2 Gedung A	Keyboard Roland	2
8	Ruang Musik	Lantai 2 Gedung A	Drum Set Pearl	1
9	Ruang Musik	Lantai 2 Gedung A	Microphone Shure	6
10	Lab Kimia	Lantai 1 Tengah	Mikroskop Cahaya	15
11	Lab Kimia	Lantai 1 Tengah	Tabung Reaksi	100
12	Lab Kimia	Lantai 1 Tengah	Bunsen Burner	10
13	Lab Kimia	Lantai 1 Tengah	Gelas Ukur 500ml	25
14	Lab Kimia	Lantai 1 Tengah	Pipet Tetes	50

### Penjelasan:

View **V\_LokasiBarang** digunakan untuk menampilkan informasi lokasi penyimpanan setiap barang inventaris. Data diambil dari tabel LokasiPenyimpanan dan Inventaris untuk mengaitkan data barang dengan lokasi penyimpanannya.

View ini menampilkan nama lokasi, informasi ruangan, nama barang, serta jumlah stok yang tersedia pada lokasi tersebut. Dengan adanya view ini, pengguna dapat dengan mudah mengetahui posisi penyimpanan dan ketersediaan inventaris di setiap lokasi.

## 10. View Mengetahui Kondisi Barang (Join)

```
● ● ●  
1 CREATE OR ALTER VIEW V_KondisiBarang AS  
2 SELECT  
3     ko.NamaKondisi,  
4     i.NamaInventaris,  
5     i.JumlahStok,  
6     i.TanggalPengadaan  
7 FROM KondisiInventaris ko  
8 JOIN Inventaris i ON ko.KondisiId = i.KondisiId;  
9  
10 SELECT *  
11 FROM V_KondisiBarang;
```

The screenshot shows a SQL query results window with three tabs at the top: 'Results' (selected), 'Messages', and 'Client Statistics'. The 'Results' tab displays a table with 14 rows of data. The columns are labeled: NamaKondisi, NamaInventaris, JumlahStok, and TanggalPengadaan. The data is as follows:

	NamaKondisi	NamaInventaris	JumlahStok	TanggalPengadaan
1	Baik	Bola Basket Spalding	8	2024-09-15
2	Baik	Matras Yoga	10	2023-10-10
3	Baik	Bola Voli Molten	12	2023-09-01
4	Rusak Ringan	Net Badminton	5	2024-03-20
5	Baik	Raket Tenis Meja	15	2024-07-10
6	Baik	Gitar Akustik Yamaha	4	2024-08-12
7	Baik	Keyboard Roland	2	2023-08-20
8	Rusak Ringan	Drum Set Pearl	1	2024-02-15
9	Baik	Microphone Shure	6	2024-05-18
10	Baik	Mikroskop Cahaya	15	2022-12-05
11	Rusak Ringan	Tabung Reaksi	100	2022-11-20
12	Baik	Bunsen Burner	10	2023-05-15
13	Baik	Gelas Ukur 500ml	25	2023-08-22
14	Rusak Ringan	Pipet Tetes	50	2024-01-10

### Penjelasan:

View **V\_KondisiBarang** digunakan untuk menampilkan daftar inventaris berdasarkan kondisi fisiknya. Data diambil dari tabel KondisiInventaris dan Inventaris untuk mengelompokkan barang sesuai dengan klasifikasi kondisi seperti *Baik*, *Rusak Ringan*, atau *Rusak Berat*.

View ini menampilkan informasi nama barang, jumlah stok, serta tanggal pengadaan berdasarkan kondisi masing-masing barang. Dengan adanya view ini, pihak sekolah dapat melakukan pemantauan kondisi inventaris secara lebih terstruktur untuk keperluan perawatan dan pengelolaan aset.

## 11. View Mengetahui Murid yang Sedang Meminjam (Join)

```
-- View murid yang sedang meminjam inventaris
CREATE OR ALTER VIEW V_MuridSedangMeminjam AS
SELECT
    p.PeminjamanId,
    m.NamaMurid,
    m.Kelas,
    p.TanggalPinjam,
    p.StatusPeminjaman
FROM Peminjaman p
JOIN Murid m
    ON p.MuridId = m.MuridId
WHERE p.StatusPeminjaman = 'Dipinjam';

SELECT *
FROM V_MuridSedangMeminjam;
```

	PeminjamanId	NamaMurid	Kelas	TanggalPinjam	StatusPeminjaman
1	13	Billy Wicaksono	XI	2024-12-15	Dipinjam
2	14	Rafii Rahman	XI	2024-12-18	Dipinjam
3	15	Riski Pratama	XII	2024-12-22	Dipinjam
4	16	Reza Putra	XII	2024-12-26	Dipinjam
5	17	Prima Maula	X	2024-12-29	Dipinjam
6	18	Muhamad Nur Salam	XII	2025-01-03	Dipinjam
7	19	Muhammad Thoriq	X	2025-01-07	Dipinjam
8	20	Vicka Aulia	XII	2025-01-10	Dipinjam
9	21	Akbar Putra Pamungkas	XI	2025-01-14	Dipinjam
10	22	Farel Rahmat	XI	2025-01-17	Dipinjam
11	23	Ilham Saugi	XII	2025-01-21	Dipinjam
12	24	Khoirun Rozan	XI	2025-01-24	Dipinjam
13	25	Daniel Aditya	XII	2025-01-28	Dipinjam
14	26	Siti Maimunah	XI	2025-02-01	Dipinjam
15	27	Bahlil Ilahi	X	2025-02-05	Dipinjam
16	28	Areva Munira	XI	2025-02-09	Dipinjam
17	29	Rafii Khoirun	XI	2025-02-12	Dipinjam
18	30	Denis Figo	XII	2025-02-16	Dipinjam

### Penjelasan:

View **V\_MuridSedangMeminjam** digunakan untuk menampilkan daftar murid yang sedang melakukan peminjaman inventaris. View ini menggabungkan data transaksi peminjaman dengan data identitas murid sehingga informasi yang ditampilkan mencakup nama murid, kelas, tanggal peminjaman, dan status peminjaman. Data yang ditampilkan difilter hanya untuk peminjaman dengan status “Dipinjam”, sehingga memudahkan petugas dalam memantau peminjaman yang masih aktif.

## 12. View Mengetahui Inventaris yang Tersedia (Seleksi/Filter)

```
-- View inventaris yang tersedia
CREATE OR ALTER VIEW V_InventarisTersedia AS
SELECT
    InventarisId,
    KodeInventaris,
    NamaInventaris,
    JumlahStok
FROM Inventaris
WHERE JumlahStok > 0;

SELECT *
FROM V_InventarisTersedia;
```

	InventarisId	KodeInventaris	NamaInventaris	JumlahStok
1	3	PO001	Bola Basket Spalding	8
2	4	PO002	Matras Yoga	10
3	5	PO003	Bola Voli Molten	12
4	6	PO004	Net Badminton	5
5	7	PO005	Raket Tenis Meja	15
6	8	PM001	Gitar Akustik Yamaha	4
7	9	PM002	Keyboard Roland	2
8	10	PM003	Drum Set Pearl	1
9	11	PM004	Microphone Shure	6
10	12	PL001	Mikroskop Cahaya	15
11	13	PL002	Tabung Reaksi	100
12	14	PL003	Bunsen Burner	10
13	15	PL004	Gelas Ukur 500ml	25
14	16	PL005	Pipet Tetes	50
15	17	PSR001	Kanvas Lukis A3	50
16	18	PSR002	Cat Akrilik Set	20
17	19	PSR003	Kuas Lukis Set	30
18	20	PSR004	Easel Kayu	10
19	21	PU001	Stetoskop	5
20	22	PU002	Timbangan Badan	2
21	23	PU003	Tensimeter Digital	3
22	24	PU004	Tempat Tidur Pasien	4

### Penjelasan:

View **V\_InventarisTersedia** berfungsi untuk menampilkan daftar inventaris yang masih tersedia untuk dipinjam. View ini menampilkan informasi utama barang seperti kode inventaris, nama inventaris, dan jumlah stok. Data yang ditampilkan hanya inventaris dengan jumlah stok lebih dari nol, sehingga membantu petugas mengetahui barang yang siap digunakan atau dipinjam.

### 13. View Mengetahui Kondisi Barang yang Baik (Join)

```
-- View inventaris dengan kondisi baik
CREATE OR ALTER VIEW V_InventarisKondisiBaik AS
SELECT
    i.KodeInventaris,
    i.NamaInventaris,
    ko>NamaKondisi,
    i.JumlahStok
FROM Inventaris i
JOIN KondisiInventaris ko
    ON i.KondisiId = ko.KondisiId
WHERE ko>NamaKondisi = 'Baik';

SELECT *
FROM V_InventarisKondisiBaik;
```

	KodeInventaris	NamaInventaris	NamaKondisi	JumlahStok
1	PO001	Bola Basket Spalding	Baik	8
2	PO002	Matras Yoga	Baik	10
3	PO003	Bola Voli Molten	Baik	12
4	PO005	Raket Tenis Meja	Baik	15
5	PM001	Gitar Akustik Yamaha	Baik	4
6	PM002	Keyboard Roland	Baik	2
7	PM004	Microphone Shure	Baik	6
8	PL001	Mikroskop Cahaya	Baik	15
9	PL003	Bunsen Burner	Baik	10
10	PL004	Gelas Ukur 500ml	Baik	25
11	PSR001	Kanvas Lukis A3	Baik	50
12	PSR002	Cat Akrilik Set	Baik	20
13	PSR004	Easel Kayu	Baik	10
14	PU001	Stetoskop	Baik	5
15	PU003	Tensimeter Digital	Baik	3
16	PMK001	Kompor Gas 2 Tungku	Baik	5
17	PMK003	Blender Philips	Baik	3
18	PMK004	Rice Cooker Miyako	Baik	4
19	PE001	Laptop ASUS ROG	Baik	15
20	PE005	Kamera Canon 80D	Baik	3

#### Penjelasan:

View **V\_InventarisKondisiBaik** digunakan untuk menampilkan inventaris yang berada dalam kondisi fisik baik. View ini menggabungkan data inventaris dengan data kondisi inventaris untuk menampilkan nama barang, kondisi, dan jumlah stok. Informasi ini berguna untuk memastikan barang yang layak pakai dan aman digunakan dalam kegiatan peminjaman.

#### 14. View Mengetahui Kondisi Barang Rusak Ringan (Join)

```
-- View inventaris dengan kondisi rusak ringan
CREATE OR ALTER VIEW V_InventarisKondisiRusakRingan AS
SELECT
    i.KodeInventaris,
    i.NamaInventaris,
    ko.NamaKondisi,
    i.JumlahStok
FROM Inventaris i
JOIN KondisiInventaris ko
    ON i.KondisiId = ko.KondisiId
WHERE ko>NamaKondisi = 'Rusak Ringan';

SELECT *
FROM V_InventarisKondisiRusakRingan;
```

Results				
	KodeInventaris	NamaInventaris	NamaKondisi	JumlahStok
1	PO004	Net Badminton	Rusak Ringan	5
2	PM003	Drum Set Pearl	Rusak Ringan	1
3	PL002	Tabung Reaksi	Rusak Ringan	100
4	PL005	Pipet Tetes	Rusak Ringan	50
5	PSR003	Kuas Lukis Set	Rusak Ringan	30
6	PU002	Timbangan Badan	Rusak Ringan	2
7	PU004	Tempat Tidur Pasien	Rusak Ringan	4
8	PMK002	Panci Set Stainless	Rusak Ringan	10
9	PE002	Proyektor Epson	Rusak Ringan	12
10	PE003	Printer HP Laserjet	Rusak Ringan	4

#### Penjelasan:

View **V\_InventarisKondisiRusakRingan** berfungsi untuk menampilkan daftar inventaris dengan kondisi rusak ringan. View ini membantu petugas dalam mengidentifikasi barang yang masih dapat digunakan namun memerlukan perawatan atau perbaikan ringan. Data yang ditampilkan mencakup kode barang, nama inventaris, kondisi, dan jumlah stok.

#### 15. View Mengetahui Kondisi Barang Rusak Berat (Join)

```
-- View inventaris dengan kondisi rusak berat
CREATE OR ALTER VIEW V_InventarisKondisiRusakBerat AS
SELECT
    i.KodeInventaris,
    i.NamaInventaris,
    ko.NamaKondisi,
    i.JumlahStok
FROM Inventaris i
JOIN KondisiInventaris ko
    ON i.KondisiId = ko.KondisiId
WHERE ko>NamaKondisi = 'Rusak Berat';

SELECT *
FROM V_InventarisKondisiRusakBerat;
```

Results				
	KodeInventaris	NamaInventaris	NamaKondisi	JumlahStok
1	PE004	Scanner Brother	Rusak Berat	2

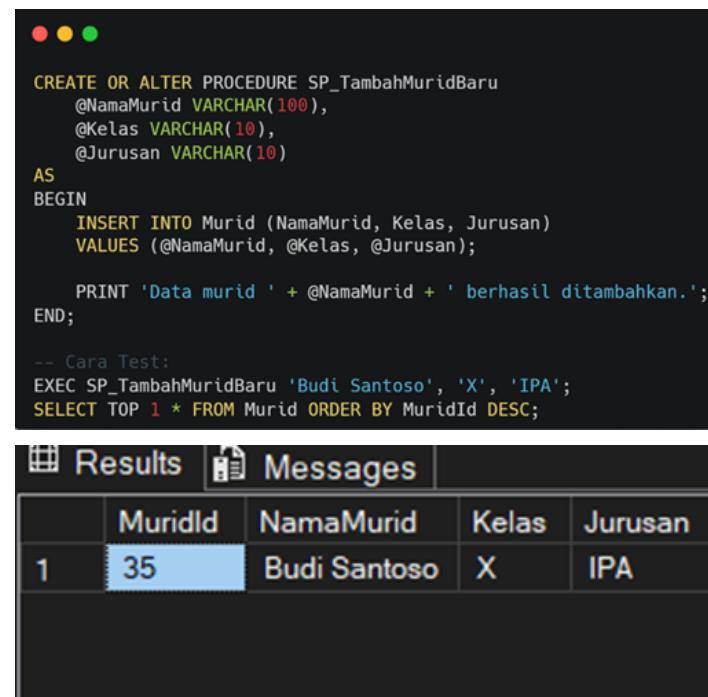
### Penjelasan:

View **V\_InventarisKondisiRusakBerat** digunakan untuk menampilkan inventaris yang berada dalam kondisi rusak berat. Informasi yang ditampilkan membantu pihak pengelola dalam mengambil keputusan terkait penghapusan, penggantian, atau pengadaan ulang inventaris. View ini menampilkan data nama barang, kondisi, dan jumlah stok berdasarkan klasifikasi kondisi rusak berat.

### 2.3.3 Pembuatan Procedure

Procedure merupakan kumpulan perintah SQL yang disimpan dalam database untuk menangani logika bisnis yang kompleks. Pada sistem ini, terdapat 3 prosedur utama yang dikembangkan untuk menyederhanakan proses input data murid, pengelolaan stok (restock), dan validasi pengembalian barang secara otomatis.

#### 1. Procedure Menambah Murid Baru



```
CREATE OR ALTER PROCEDURE SP_TambahMuridBaru
    @NamaMurid VARCHAR(100),
    @Kelas VARCHAR(10),
    @Jurusan VARCHAR(10)
AS
BEGIN
    INSERT INTO Murid (NamaMurid, Kelas, Jurusan)
    VALUES (@NamaMurid, @Kelas, @Jurusan);

    PRINT 'Data murid ' + @NamaMurid + ' berhasil ditambahkan.';

END;

-- Cara Test:
EXEC SP_TambahMuridBaru 'Budi Santoso', 'X', 'IPA';
SELECT TOP 1 * FROM Murid ORDER BY MuridId DESC;
```

	MuridId	NamaMurid	Kelas	Jurusan
1	35	Budi Santoso	X	IPA

### Penjelasan:

Stored Procedure **SP\_TambahMuridBaru** berfungsi untuk menyederhanakan proses input data siswa baru ke dalam database. Dengan prosedur ini, pengguna tidak perlu menulis query INSERT manual yang panjang berulang kali. Prosedur menerima parameter nama, kelas, dan jurusan, lalu secara

otomatis menyimpannya ke tabel Murid dan memberikan notifikasi bahwa data berhasil ditambahkan.

## 2. Procedure Restock Barang

```
CREATE OR ALTER PROCEDURE SP_RestockBarang
    @KodeInventaris VARCHAR(30),
    @JumlahTambah INT
AS
BEGIN
    UPDATE Inventaris
    SET JumlahStok = JumlahStok + @JumlahTambah,
        TanggalPengadaan = GETDATE()
    WHERE KodeInventaris = @KodeInventaris;
END;

-- Cara Test:
EXEC SP_RestockBarang 'P0001', 5;
SELECT KodeInventaris, NamaInventaris, JumlahStok FROM Inventaris WHERE KodeInventaris = 'P0001';
```

	KodeInventaris	NamaInventaris	JumlahStok
1	P0001	Bola Basket Spalding	23

### Penjelasan:

Stored Procedure **SP\_RestockBarang** digunakan untuk menangani proses pengadaan barang atau penambahan stok (*restock*). Prosedur ini secara otomatis memperbarui jumlah stok barang berdasarkan kode inventaris yang diinputkan. Selain menambah jumlah, prosedur ini juga memperbarui kolom TanggalPengadaan ke waktu saat ini (*real-time*), sehingga data administrasi tetap akurat sesuai aktivitas terakhir.

## 3. Procedure Proses Pengembalian

```
CREATE OR ALTER PROCEDURE SP_ProsesPengembalian
    @PeminjamanId INT
AS
BEGIN
    UPDATE Peminjaman
    SET TanggalKembali = GETDATE(),
        StatusPeminjaman = 'Selesai'
    WHERE PeminjamanId = @PeminjamanId;
END;

-- Cara Test:
EXEC SP_ProsesPengembalian 13;
SELECT PeminjamanId, StatusPeminjaman, TanggalKembali FROM Peminjaman WHERE PeminjamanId = 13;
```

	PeminjamanId	StatusPeminjaman	TanggalKembali
1	13	Selesai	2025-12-31

### Penjelasan:

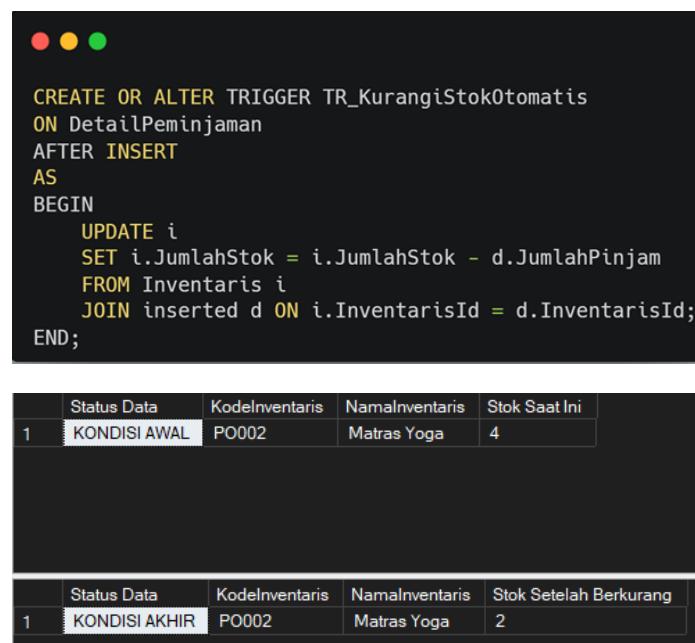
Stored Procedure **SP\_ProsesPengembalian** berfungsi untuk mempermudah admin dalam menutup transaksi peminjaman. Ketika siswa mengembalikan barang, prosedur ini akan mengubah status peminjaman menjadi 'Selesai' dan secara otomatis mengisi kolom TanggalKembali dengan tanggal hari ini. Hal

ini mencegah kesalahan input tanggal manual dan menjaga integritas data riwayat peminjaman.

#### 2.3.4 Pembuatan Trigger

Trigger diimplementasikan sebagai mekanisme otomasi dan validasi data yang berjalan di latar belakang (*background process*). Terdapat 2 trigger yang dibuat untuk menjaga konsistensi jumlah stok dan mencegah kesalahan manipulasi data inventaris.

##### 1. Trigger Pengurangan Stok Otomatis



The screenshot shows a MySQL Workbench interface. At the top, there are three colored dots (red, yellow, green). Below them is a code editor containing the following SQL script:

```
CREATE OR ALTER TRIGGER TR_KurangiStokOtomatis
ON DetailPeminjaman
AFTER INSERT
AS
BEGIN
    UPDATE i
    SET i.JumlahStok = i.JumlahStok - d.JumlahPinjam
    FROM Inventaris i
    JOIN inserted d ON i.InventarisId = d.InventarisId;
END;
```

Below the code editor are two tables. The first table, labeled "Status Data" at the top, has four columns: KodeInventaris, NamalInventaris, and Stok Saat Ini. It contains one row with values P0002, Matras Yoga, and 4. The second table, labeled "Status Data" at the bottom, has four columns: KodeInventaris, NamalInventaris, and Stok Setelah Berkurang. It also contains one row with values P0002, Matras Yoga, and 2.

##### Penjelasan :

Trigger **TR\_KurangiStokOtomatis** adalah fitur otomatisasi yang berjalan ketika ada data baru ditambahkan ke tabel *DetailPeminjaman*. Trigger ini berfungsi untuk memotong stok barang di tabel induk (*Inventaris*) sesuai dengan jumlah barang yang dipinjam oleh siswa. Dengan adanya trigger ini, data stok barang akan selalu *update* secara otomatis (real-time) tanpa perlu dilakukan pengurangan manual oleh petugas.

## 2. Trigger Validasi Stok Cukup

The screenshot shows two windows from SQL Server Management Studio. The top window displays the T-SQL code for creating a trigger:

```
CREATE OR ALTER TRIGGER TR_ValidasiStokCukup
ON DetailPeminjaman
AFTER INSERT
AS
BEGIN
    IF EXISTS (
        SELECT 1 FROM Inventaris i
        JOIN inserted d ON i.InventarisId = d.InventarisId
        WHERE i.JumlahStok < 0
    )
    BEGIN
        RAISERROR ('Stok barang tidak mencukupi.', 16, 1);
        ROLLBACK TRANSACTION;
    END
END;
```

The bottom window shows the 'Messages' tab with the following output:

```
13 Messages
Stok inventaris otomatis berkurang sesuai jumlah peminjaman.

(0 rows affected)
TEST BERHASIL: Transaksi digagalkan oleh Trigger.
Pesan Error SQL: Transaksi Dibatalkan: Stok barang tidak mencukupi untuk peminjaman ini.

Completion time: 2026-01-05T03:20:02.1161720+07:00
```

### Penjelasan:

Trigger **TR\_ValidasiStokCukup** berfungsi sebagai mekanisme keamanan data (*data integrity*). Trigger ini akan mencegah transaksi peminjaman jika jumlah barang yang dipinjam melebihi stok yang tersedia. Jika sistem mendeteksi bahwa stok akan menjadi negatif (kurang dari 0), trigger ini akan membatalkan transaksi (*Rollback*) dan memunculkan pesan peringatan. Ini mencegah terjadinya kesalahan pencatatan di mana barang yang dipinjam lebih banyak dari fisik yang ada.

## **BAB III**

## **PENUTUP**

### **3.1 Kendala Yang Dihadapi**

Selama proses pengerjaan, kelompok kami hanya mengalami kendala pada saat pengerjaan laporan akhir.

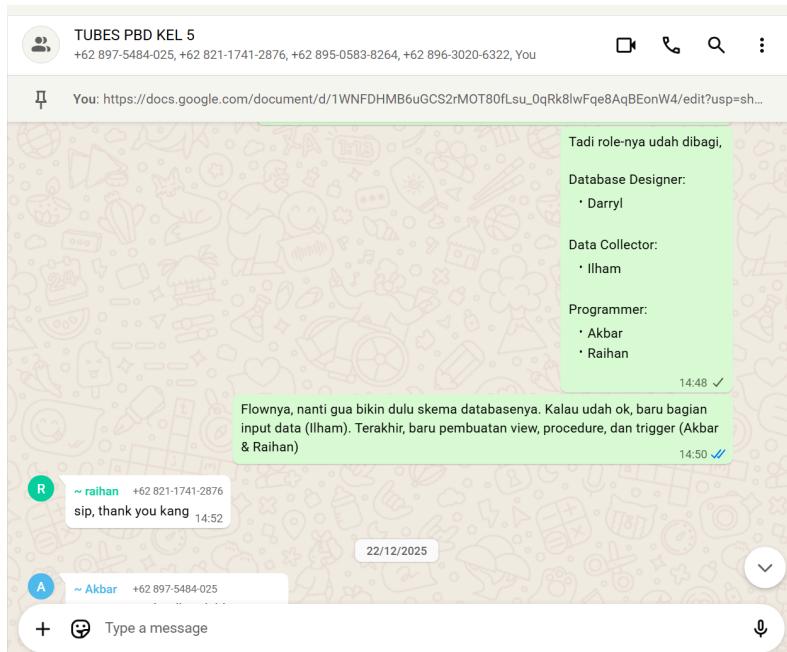
### **3.2 Solusi Kendala**

Kami memperbaiki komunikasi antar anggota supaya proses pengerjaan efisien.

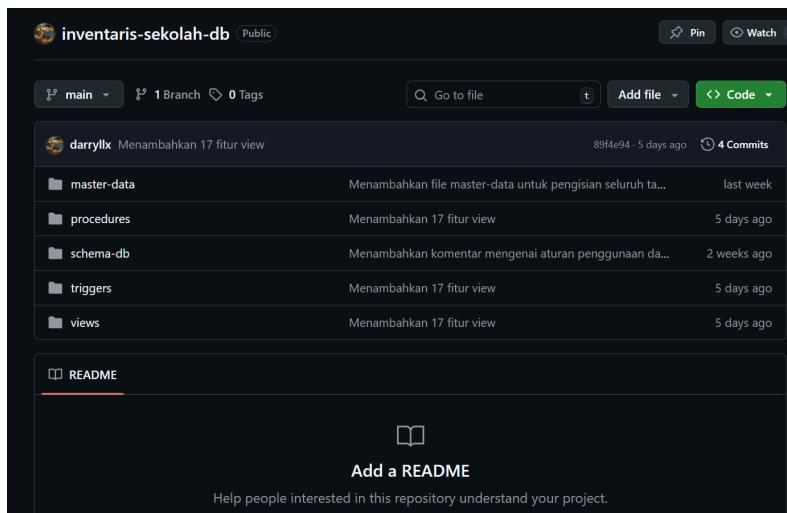
### **3.3 Kesimpulan**

Pengerjaan tugas ini membantu kami dalam mengembangkan kemampuan bekerja secara tim serta memahami secara menyeluruh proses perancangan dan pembangunan basis data, mulai dari tahap perencanaan hingga implementasi. Selain itu, melalui tugas ini kami berhasil mengimplementasikan berbagai materi penting dalam sistem basis data, seperti Data Definition Language (DDL), Data Manipulation Language (DML), pembuatan *view*, *procedure*, dan *trigger*, sehingga pemahaman kami terhadap konsep dan penerapannya menjadi lebih mendalam.

## LAMPIRAN



**Keterangan:** Pembagian role masing-masing anggota



**Keterangan:** Penggunaan Git & Github sebagai alat untuk manajemen perubahan kode SQL.

