LEMBAR KERJA PESERTA DIDIK
BASIS DATA

LEMBAR KERJA PERSERTA DIDIK
(LKPD)
Merancang Database

Nama Sekolah	:	SMK NEGERI 1 KATAPANG
Kelas/Semester	:	XI RPL 1 / 1
Kompetensi Keahlian	:	Rekayasa Perangkat Lunak (RPL)
Alokasi Waktu	:	6 X 45 Menit (270 Jam Pelajaran)
Tanggal	:	22 - 29 November 2024
Nama Kelompok	:	Acef Kiki S. Maulana Fajar
		Ananda Dwi Putra Masyuri
		Ananda Mahatva Yodha
		Daffa Jaya Perkasa
		Muhammad Rayhan Ali

DAFTAR ISI

BAB I
PENDAHULUAN

1.1. Tema

Kami mengambil tema yang bersangkutan dengan kedisiplinan sekolah. Yaitu kali mengambil tema Buku Saku yang dijadikan secara Online atau Electronic. Dengan Nama E-SAKU, kami menghadirkan inovasi yang bisa membuat sebuah revolusi besar untuk sekolah.

Dengan adanya E-SAKU ini, kami memiliki tujuan yaitu untuk membantu pihak sekolah dalam mengelola data pelanggaran siswa secara efektif, serta memberikan transparansi kepada siswa dan orang tua terkait riwayat pelanggaran yang telah dilakukan. Dengan adanya buku saku elektronik ini, diharapkan siswa lebih termotivasi untuk menjaga kedisiplinan mereka, karena adanya sistem yang mencatat secara otomatis setiap pelanggaran yang dilakukan dan memperlihatkan konsekuensi yang akan dihadapi jika poin pelanggaran semakin tinggi

E-SAKU ini juga memiliki beberapa manfaat yang di antaranya:

1. Efisiensi dalam Pengelolaan Data

   Sistem ini memungkinkan pengelolaan data pelanggaran siswa secara otomatis dan terstruktur, mengurangi beban administrasi guru.

2. Akses Mudah

   Siswa, guru, dan orang tua dapat memantau poin pelanggaran dengan mudah kapan saja, sehingga lebih transparan.

3. Penyimpanan Aman

   Data pelanggaran disimpan secara digital, mengurangi risiko kehilangan atau kerusakan data.

4. Peningkatan Kesadaran Siswa

   Dengan adanya catatan pelanggaran yang transparan, siswa lebih terdorong untuk menghindari pelanggaran.

5. Keterlibatan Orang Tua

Orang tua dapat memantau perkembangan kedisiplinan anak mereka, sehingga dapat lebih proaktif dalam memberikan bimbingan.

1.2. Timeline

HARI	TANGGAL	AKTOR / PELAKSANA KEGIATAN	KEGIATAN PROJECT
Jum'at	22-11-2024	- Ananda Mahatva Yodha	Membuat Perancangan dan DML.
		- Muhammad Rayhan Ali	Membuat ERD E-Saku.
		- Ananda Dwi Putra Mayuri	Merancang DDL.
		-Acef Kiki S. Maulana Fajar	Membuat Laporan Harian.
Rabu	27-11-2024	- Ananda Mahatva Yodha	Melanjutkan Pembuatan Database Bagian DML.
		- Muhammad Rayhan Ali	Melanjutkan Membuat ERD.
		- Daffa Jaya Perkasa - Ananda Dwi Putra Mayuri	Membuat Struktur Bagian DDL.
		-Acef Kiki S. Maulana Fajar	Membuat Laporan Pembuatan & Perancangan Database Project E-Saku.
Jum'at	29-11-2024	Seluruh Anggota	Mengumpulkan Laporan Masing-Masing.

1.3. Analisis Kasus

Dalam lingkungan sekolah, menjaga ketertiban dan kedisiplinan siswa adalah bagian penting dari upaya menciptakan suasana belajar yang kondusif. Salah satu cara untuk memonitor perilaku siswa adalah melalui pencatatan poin pelanggaran yang diberikan ketika seorang siswa melanggar aturan sekolah.

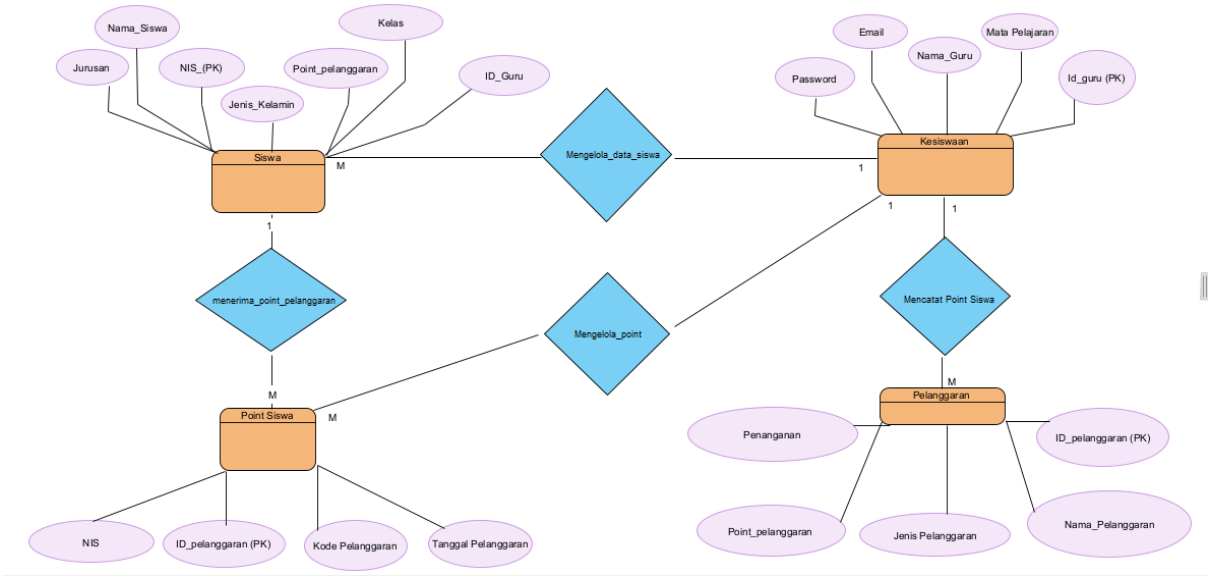
Selama ini, banyak sekolah menggunakan metode pencatatan manual melalui buku saku fisik atau laporan tertulis untuk memonitor pelanggaran siswa. Namun, metode ini sering kali tidak efisien dan memiliki beberapa keterbatasan, seperti risiko kehilangan catatan, lupa dalam membawa buku saku ke sekolah, dan resiko buku saku rusak.

Dengan adanya perkembangan teknologi digital, dibutuhkan suatu sistem yang lebih efisien, terstruktur, dan mudah diakses oleh pihak terkait, seperti siswa, guru, dan orang tua. Program Buku Saku Elektronik Pelanggaran Siswa hadir sebagai solusi untuk menjawab kebutuhan ini.

Program ini memungkinkan pencatatan poin pelanggaran siswa secara elektronik, di mana setiap kali seorang siswa melakukan pelanggaran, poin tersebut langsung ditambahkan ke dalam buku saku elektronik mereka. Dengan sistem ini, catatan pelanggaran siswa menjadi lebih mudah dipantau, aman, dan dapat diakses oleh siswa dan pihak yang berwenang kapan saja.

BAB II
PERANCANGAN

2.1. *Entity Relationship Diagram* (ERD)



2.2. Kardinalitas dan Relasi

- Keterangan
  1. 1 = One
  2. M = Many
  3. 1:1 = One to One
  4. 1:M = One to Many
  5. M:1 = Many to One
  6. M:M = Many to Many
- Kardinalitas
  1. Siswa 1:M Kesiswaan
  2. Siswa 1:M Point Siswa
  3. Point Siswa M:1 Kesiswaan
  4. Kesiswaan 1:M Pelanggaran
- Relasi

  Garis lurus satu arah.

2.3. Struktur Basis Data

1. Tabel Siswa

Tabel ini menyimpan data siswa.

- **id_siswa** (PK) - ID Siswa (Primary Key)
- **nama_siswa** - Nama lengkap siswa
- **jurusan** - Jurusan yang diambil siswa
- **nis** - Nomor Induk Siswa Nasional (NIS)
- **jenis_kelamin** - Jenis kelamin siswa
- **kelas** - Kelas tempat siswa bersekolah
- **id_guru** (FK) - ID Guru yang mengajar siswa (Foreign Key yang merujuk ke Tabel Guru)

2. Tabel Guru

Tabel ini menyimpan data guru.

- **id_guru** (PK) - ID Guru (Primary Key)
- **nama_guru** - Nama lengkap guru
- **email** - Email guru
- **password** - Password untuk login guru
- **mata_pelajaran** - Mata pelajaran yang diajarkan oleh guru

3. Tabel Point_Pelanggaran

Tabel ini menyimpan jenis pelanggaran yang ada di sekolah.

- **id_pelanggaran** (PK) - ID Pelanggaran (Primary Key)
- **kode_pelanggaran** - Kode unik untuk pelanggaran
- **nama_pelanggaran** - Nama pelanggaran
- **jenis_pelanggaran** - Jenis pelanggaran

4. Tabel Kesiswaan

Tabel ini mengelola data yang berkaitan dengan kegiatan kesiswaan dan administrasi.

- **id_siswa** (FK) - ID Siswa (Foreign Key yang merujuk ke Tabel Siswa)
- **id_guru** (FK) - ID Guru (Foreign Key yang merujuk ke Tabel Guru)
- **mengelola_data_siswa** - Menunjukkan apakah guru tersebut memiliki akses untuk mengelola data siswa
- **mengelola_point** - Menunjukkan apakah guru tersebut memiliki akses untuk mengelola point pelanggaran siswa

5. Tabel Menerima_Point_Pelanggaran

Tabel ini menyimpan informasi pelanggaran yang diterima oleh siswa.

- **id_siswa** (FK) - ID Siswa (Foreign Key yang merujuk ke Tabel Siswa)
- **id_pelanggaran** (FK) - ID Pelanggaran (Foreign Key yang merujuk ke Tabel Point_Pelanggaran)
- **tanggal_pelanggaran** - Tanggal saat pelanggaran terjadi

6. Tabel Mencatat_Point_Siswa

Tabel ini menyimpan catatan point pelanggaran yang diterima siswa.

- **id_siswa** (FK) - ID Siswa (Foreign Key yang merujuk ke Tabel Siswa)
- **id_pelanggaran** (FK) - ID Pelanggaran (Foreign Key yang merujuk ke Tabel Point_Pelanggaran)
- **point_pelanggaran** - Jumlah point pelanggaran yang diberikan
- **tanggal_pelanggaran** - Tanggal saat point pelanggaran dicatat

Relasi Antar Tabel:

- **Siswa** memiliki relasi satu ke banyak dengan **Menerima_Point_Pelanggaran** dan **Mencatat_Point_Siswa**.
- **Guru** dapat mengelola data **Siswa** dan **Point_Pelanggaran** melalui relasi pada **Kesiswaan**.
- **Point_Pelanggaran** memiliki relasi banyak ke banyak dengan **Siswa** melalui tabel **Menerima_Point_Pelanggaran** dan **Mencatat_Point_Siswa**.
- **Siswa** memiliki relasi banyak ke satu dengan **Guru** melalui kelas yang dikelola oleh guru tersebut.

BAB III
PENGUJIAN

Pengujian pada tugas basis data ini didahului dengan beberapa tahapan. Para anggota kelompok terlebih dahulu melakukan analisis untuk menentukan tema program yang sesuai. Tema terkait kedisiplinan menjadi pilihan bagi para anggota kelompok kami. Kemudian kami perdiskusi satu sama lain untuk membahas tema lebih spesifik sehingga terbentuk rumusan tujuan dari tugas ini yaitu membuat sebuah database untuk mencatat point-point pelanggaran siswa secara digital.

Sebelum melakukan praktek, bagian perancangan perlu membuat sebuah rancangan dari data yang akan dibuat. Pembuatan rancangan data tersebut dilakukan didalam *software* Visual Paradigm. Perancang berdiskusi dengan guru yang bersangkutan dan aggota kelompok lainnya untuk menentukan entitas, atribut, kardinalitas dan relasi yang cocok. Proses perancangan memakan waktu 2 hari di hari yang berbeda, karena ada beberapa hal yang masih kurang tepat.

3.1. Implementasi

- **Membuat Database**

Pertama pembuatan data pertama-tama kita perlu membuat database itu sendiri dengan cara :
Create database [nama database];

```
MariaDB [buku_saku]> create database Nekat_saku;
Query OK, 1 row affected (0.003 sec)
```

- **Membuat Table Database**

Selanjutnya untuk menyimpan data kita perlu kita perlu sebuah table dengan cara:
Create table [nama table] ([nama field] typedata [jumlah tipe data]……N);

Dalam pembuatan table bisa saja ditambah primary key dan auto increment. Fungsi dari masing masing sytax tersebut yaitu :

- **Primary key**

Fungsinya menjadi kunci untuk menyambungkan table satu ke table lain dengan syarat isi dari record harus berbeda satu sama lain. Primary key hanya bisa digunakan satu field dalam satu table.

- **Auto Increment**

Fungsinya untuk menambah otomatis data, namun dengan syarat typedata harus menggunakan intcrement (int).

TABLE SISWA

```
MariaDB [Nekat_saku]> create table siswa (NIS int (10) primary key,nama_siswa text, JK text, Kelas var
char (20), jurusan text, Id_guru int (10),point int (100));
Query OK, 0 rows affected (0.015 sec)

MariaDB [Nekat_saku]> desc siswa;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| NIS        | int(10)     | NO   | PRI | NULL    |       |
| nama_siswa | text        | YES  |     | NULL    |       |
| JK         | text        | YES  |     | NULL    |       |
| Kelas      | varchar(20) | YES  |     | NULL    |       |
| jurusan    | text        | YES  |     | NULL    |       |
| Id_guru    | int(10)     | YES  |     | NULL    |       |
| point      | int(100)    | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
7 rows in set (0.031 sec)
```

TABLE KESISWAAAN

```
MariaDB [Nekat_saku]> create table kesiswaan (Id_guru int (10) primary key,nama_guru text, mata_pelaja
ran text, email varchar (20), Password varchar (20));
Query OK, 0 rows affected (0.014 sec)

MariaDB [Nekat_saku]> desc kesiswaan;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| Id_guru       | int(10)     | NO   | PRI | NULL    |       |
| nama_guru     | text        | YES  |     | NULL    |       |
| mata_pelajaran | text       | YES  |     | NULL    |       |
| email         | varchar(20) | YES  |     | NULL    |       |
| Password      | varchar(20) | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
5 rows in set (0.019 sec)
```

POINT SISWA

```
MariaDB [Nekat_saku]> create table point_siswa(Id_pelanggaran int (10) primary key, NIS int (10), tang
gal_pelanggaran varchar (30));
Query OK, 0 rows affected (0.018 sec)

MariaDB [Nekat_saku]> desc point_siswa;
+--------------------+-------------+------+-----+---------+-------+
| Field              | Type        | Null | Key | Default | Extra |
+--------------------+-------------+------+-----+---------+-------+
| Id_pelanggaran     | int(10)     | NO   | PRI | NULL    |       |
| NIS                | int(10)     | YES  |     | NULL    |       |
| tanggal_pelanggaran | varchar(30) | YES  |     | NULL    |       |
+--------------------+-------------+------+-----+---------+-------+
3 rows in set (0.030 sec)
```

PELANGGARAN

```
MariaDB [Nekat_saku]> create table pelanggaran(id_pelanggaran int (10) primary key, jenis_pelanggaran
text, nama_pelanggaran text, point int (100), penanganan varchar (100));
Query OK, 0 rows affected (0.018 sec)

MariaDB [Nekat_saku]> desc pelanggaran
    -> ;
+------------------+--------------+------+-----+---------+-------+
| Field            | Type         | Null | Key | Default | Extra |
+------------------+--------------+------+-----+---------+-------+
| id_pelanggaran   | int(10)      | NO   | PRI | NULL    |       |
| jenis_pelanggaran | text         | YES  |     | NULL    |       |
| nama_pelanggaran  | text         | YES  |     | NULL    |       |
| point            | int(100)     | YES  |     | NULL    |       |
| penanganan       | varchar(100) | YES  |     | NULL    |       |
+------------------+--------------+------+-----+---------+-------+
5 rows in set (0.026 sec)
```

- **Menghapus table data**

Selanjutnya kita dapat menghapus table data dengan menggunakan syntax drop table [nama table];

```
MariaDB [Nekat_saku]> drop table pelanggaran;
Query OK, 0 rows affected (0.015 sec)

MariaDB [Nekat_saku]> desc pelanggaran;
ERROR 1146 (42S02): Table 'nekat_saku.pelanggaran' doesn't exist
```

- **Mengganti nama field**

Kemudian untuk mengganti nama field, kita dapat menggunakan perintah syntax alter table [nama table] CHANGE [nama field sebelumnya] [nama field baru] [type data (jumlah)];

Dalam syntax ini juga dapat ditambah primary key atau auto increment dengan cara menulis fungsi setelah jumlah tipe data.

```
MariaDB [Nekat_saku]> alter table pelanggaran change id_pelanggaran kode_pelangaran int (10);
Query OK, 0 rows affected (0.018 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [Nekat_saku]> desc pelanggaran;
+------------------+--------------+------+-----+---------+-------+
| Field            | Type         | Null | Key | Default | Extra |
+------------------+--------------+------+-----+---------+-------+
| kode_pelangaran  | int(10)      | NO   | PRI | NULL    |       |
| jenis_pelanggaran | text         | YES  |     | NULL    |       |
| nama_pelanggaran  | text         | YES  |     | NULL    |       |
| point            | int(100)     | YES  |     | NULL    |       |
| penanganan       | varchar(100) | YES  |     | NULL    |       |
+------------------+--------------+------+-----+---------+-------+
5 rows in set (0.019 sec)
```

- **Menambah Field dalam table data**

Jika ingin menambah field dalam table data, kita dapat menggnakan perintah syntax ALTER TABLE [nama tabel] ADD COLUMN [nama field yang ditambah] [type data (jumlah)];

```
MariaDB [Nekat_saku]> ALTER TABLE point_siswa ADD COLUMN kode_pelanggaran INT(10);
Query OK, 0 rows affected (0.009 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [Nekat_saku]> desc point_siswa;
+---------------------+-------------+------+-----+---------+-------+
| Field               | Type        | Null | Key | Default | Extra |
+---------------------+-------------+------+-----+---------+-------+
| Id_pelanggaran      | int(10)     | NO   | PRI | NULL    |       |
| NIS                 | int(10)     | YES  |     | NULL    |       |
| tanggal_pelanggaran | varchar(30) | YES  |     | NULL    |       |
| kode_pelanggaran    | int(10)     | YES  |     | NULL    |       |
+---------------------+-------------+------+-----+---------+-------+
4 rows in set (0.030 sec)
```

- **Menambah isi tabel data dan melihat isi tabel data**

Untuk menambah isi tabel data, kita dapat menggunakan syntax INSERT INTO [nama tabel] VALUES (“isi”,”isi”, ……. N);

Untuk menampilkan isi database yang telah diisi kita gunakan syntax
Select [kondisi] from [nama table];
Untuk menampilkan seluruh isi dari table, pada kondisi kita gunakan “ * “.
Jadi seperti ini
Select * from siswa;
Sedangkan jika kita ingin menampilkan tabel tertentu dari table, pada kondisi kita ganti jadi nama field.
Select nama,kelas,point from siswa;

TABEL PELANGGARAN

```
MariaDB [nekat_saku]> insert into pelanggaran (kode_pelangaran, jenis_pelanggaran, nama_pelanggaran, point, penanganan) values
    -> (1, 'Ringan', 'Tidak memakai seragam lengkap', 5, 'Peringatan'),
    -> (2, 'Sedang', 'Datang terlambat', 2, 'Diberi tugas'),
    -> (3, 'Berat', 'Berkelahi', 50, 'Skorsing 3 hari'),
    -> (4, 'Berat', 'Membawa barang terlarang', 100, 'Dikeluarkan dari sekolah');
Query OK, 4 rows affected (0.009 sec)
Records: 4  Duplicates: 0  Warnings: 0

MariaDB [nekat_saku]> select * from pelanggaran
    -> ;
+----------------+-----------------+-----------------------------+-------+-------------------------+
| kode_pelangaran | jenis_pelanggaran | nama_pelanggaran            | point | penanganan              |
+----------------+-----------------+-----------------------------+-------+-------------------------+
|              1 | Ringan          | Tidak memakai seragam lengkap |     5 | Peringatan              |
|              2 | Sedang          | Datang terlambat            |     2 | Diberi tugas            |
|              3 | Berat           | Berkelahi                   |    50 | Skorsing 3 hari         |
|              4 | Berat           | Membawa barang terlarang    |   100 | Dikeluarkan dari sekolah |
+----------------+-----------------+-----------------------------+-------+-------------------------+
4 rows in set (0.001 sec)
```

TABEL SISWA

```
MariaDB [nekat_saku]> insert into siswa (NIS, nama_siswa, JK, Kelas, jurusan, Id_guru, point) values
    -> (12345, 'Ali Maulana', 'Laki-laki', 'XII IPA 1', 'IPA', 1, 10),
    -> (12346, 'Siti Aisyah', 'Perempuan', 'XII IPS 2', 'IPS', 2, 0),
    -> (12347, 'Arif syarifudin', 'Laki-laki', 'XII IPA 3', 'IPA', 1, 12),
    -> (12348, 'Dewi Pratiwi', 'Perempuan', 'XII IPS 1', 'IPS', 3, 15);
Query OK, 4 rows affected (0.008 sec)
Records: 4  Duplicates: 0  Warnings: 0

MariaDB [nekat_saku]> select * from siswa;
+-------+----------------+-----------+-----------+---------+---------+-------+
| NIS   | nama_siswa     | JK        | Kelas     | jurusan | Id_guru | point |
+-------+----------------+-----------+-----------+---------+---------+-------+
| 12345 | Ali Maulana    | Laki-laki | XII IPA 1 | IPA     |       1 |    10 |
| 12346 | Siti Aisyah    | Perempuan | XII IPS 2 | IPS     |       2 |     0 |
| 12347 | Arif syarifudin | Laki-laki | XII IPA 3 | IPA     |       1 |    12 |
| 12348 | Dewi Pratiwi   | Perempuan | XII IPS 1 | IPS     |       3 |    15 |
+-------+----------------+-----------+-----------+---------+---------+-------+
4 rows in set (0.002 sec)
```

TABEL POINT SISWA

```
MariaDB [nekat_saku]> insert into point_siswa (Id_pelanggaran, NIS, tanggal_pelanggaran, kode_pelanggaran) values
    -> (1, 12345, '2024-11-01', 1),
    -> (2, 12348, '2024-11-02', 2),
    -> (3, 12347, '2024-11-03', 3),
    -> (4, 12348, '2024-11-04', 4);
Query OK, 4 rows affected (0.004 sec)
Records: 4  Duplicates: 0  Warnings: 0

MariaDB [nekat_saku]> select * from point_siswa;
+----------------+-------+---------------------+------------------+
| Id_pelanggaran | NIS   | tanggal_pelanggaran | kode_pelanggaran |
+----------------+-------+---------------------+------------------+
|              1 | 12345 | 2024-11-01          |                1 |
|              2 | 12348 | 2024-11-02          |                2 |
|              3 | 12347 | 2024-11-03          |                3 |
|              4 | 12348 | 2024-11-04          |                4 |
+----------------+-------+---------------------+------------------+
4 rows in set (0.001 sec)
```

- **Menampilkan isi tabel dengan tabel lain**

Pada kali ini kita akan menampilkan data dari tabel satu dengan tabel lainnya dengan syntax Select [nama tabel].[nama field], [nama tabel].[nama field] FROM [tabel yang digunakan], [tabel yang digunakan] WHERE [nama tabel].[foreign key] = [nama tabel].[foreign key];

```
MariaDB [nekat_saku]> select siswa.nis, siswa.nama_siswa, siswa.JK, siswa.point, kesiswaan.nama_guru from siswa, kesiswaan where siswa.id_guru = kesiswaan.id_guru;
+-------+----------------+-----------+-------+-------------+
| nis   | nama_siswa     | JK        | point | nama_guru   |
+-------+----------------+-----------+-------+-------------+
| 12345 | Ali Maulana    | Laki-laki |    10 | Budi Santoso |
| 12346 | Siti Aisyah    | Perempuan |     0 | Siti Rahma   |
| 12347 | Arif syarifudin | Laki-laki |    12 | Budi Santoso |
| 12348 | Dewi Pratiwi   | Perempuan |    15 | Agus Salim   |
+-------+----------------+-----------+-------+-------------+
4 rows in set (0.001 sec)
```

- **Fungsi Join**

Adapaun cara lain untuk menyambungkan tabel satu dengan tabel lain menggunakan fungsi join. Fungsi join dibagi menjadi 3 yaitu INNER JOIN, LEFT JOIN, RIGHT JOIN.

INNER JOIN : membandingkan record di setiap table untuk dicek apakah nilai sama atau tidak

Cara Penulisan :SELECT fields FROM table1 INNER JOIN table2 ON table1.field = table2.field;

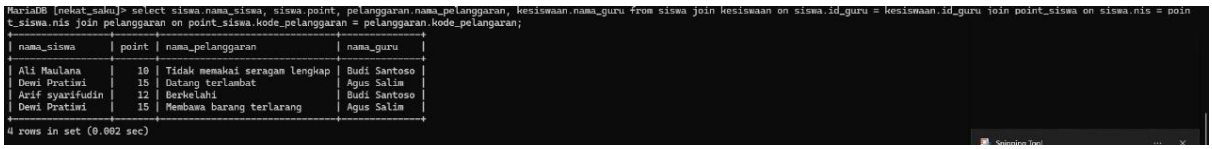
LEFT JOIN : menghasilkan nilai berdasarkan table kiri (table1) dan nilai yang sama di table kanan (table2)

Cara Penulisan : SELECT fields FROM table1 LEFT JOIN table2 ON table1.field = table2.field;

RIGHT JOIN : hampir sama seperti LEFT JOIN hanya yang menjadimaster adalah table kanan (table2)

Cara Penulisan : SELECT fields FROM table1 RIGHT JOIN table2 ON table1.field = table2.field;

- **INNER JOIN**



```
MariaDB [nekat_saku]> select siswa.nama_siswa, siswa.point, pelanggaran.nama_pelanggaran, kesiswaan.nama_guru from siswa join kesiswaan on siswa.id_guru = kesiswaan.id_guru join point_siswa on siswa.nis = poin
t_siswa.nis join pelanggaran on point_siswa.kode_pelanggaran = pelanggaran.kode_pelangaran;
+--------------+-------+-----------------------------+-------------+
| nama_siswa   | point | nama_pelanggaran            | nama_guru   |
+--------------+-------+-----------------------------+-------------+
| Ali Maulana  |    10 | Tidak memakai seragam lengkap | Budi Santoso |
| Dewi Pratiwi |    15 | Datang terlambat            | Agus Salim  |
| Arif syarifudin |  12 | Berkelahi                   | Budi Santoso |
| Dewi Pratiwi |    15 | Membawa barang terlarang    | Agus Salim  |
+--------------+-------+-----------------------------+-------------+
4 rows in set (0.002 sec)
```

- **LEFT JOIN**

```
MariaDB [nekat_saku]> select siswa.NIS, siswa.nama_siswa,siswa.jk, siswa.point, kesiswaan.nama_guru fr
om siswa LEFT JOIN kesiswaan ON siswa.id_guru = kesiswaan.id_guru;
+-------+-----------------+-----------+-------+--------------+
| NIS   | nama_siswa      | jk        | point | nama_guru    |
+-------+-----------------+-----------+-------+--------------+
| 12345 | Ali Maulana     | Laki-laki |    10 | Budi Santoso |
| 12346 | Siti Aisyah     | Perempuan |     0 | Siti Rahma   |
| 12347 | Arif syarifudin | Laki-laki |    12 | Budi Santoso |
| 12348 | Dewi Pratiwi    | Perempuan |    15 | Agus Salim   |
+-------+-----------------+-----------+-------+--------------+
4 rows in set (0.002 sec)
```

- **RIGHT JOIN**

```
MariaDB [nekat_saku]> select siswa.NIS, siswa.nama_siswa,siswa.jk, siswa.point, kesiswaan.nama_guru fr
om siswa RIGHT JOIN kesiswaan ON siswa.id_guru = kesiswaan.id_guru;
+-------+-----------------+-----------+-------+--------------+
| NIS   | nama_siswa      | jk        | point | nama_guru    |
+-------+-----------------+-----------+-------+--------------+
| 12345 | Ali Maulana     | Laki-laki |    10 | Budi Santoso |
| 12346 | Siti Aisyah     | Perempuan |     0 | Siti Rahma   |
| 12347 | Arif syarifudin | Laki-laki |    12 | Budi Santoso |
| 12348 | Dewi Pratiwi    | Perempuan |    15 | Agus Salim   |
+-------+-----------------+-----------+-------+--------------+
4 rows in set (0.001 sec)
```

- **Ascending dan Descending**

Kemudian kita bisa mengurutkan data dari terbesar hingga terkecil(Z – A) dan terkecil hingga terbesar (Z – A) menggunakan syntax :

Terkecil hingga Terbesar (A – Z)

select [kondisi] from [tabel] order by [nama field] asc;

Terbesar hingga Terkecil (Z – A)

select [kondisi] fro [tabel] order by [nama field] desc;

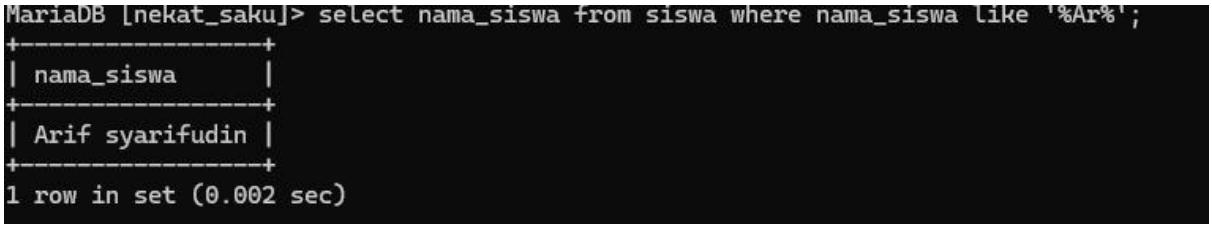
```
MariaDB [nekat_saku]> select * from siswa order by point asc;
+-------+-----------------+-----------+-----------+---------+---------+-------+
| NIS   | nama_siswa      | JK        | Kelas     | jurusan | Id_guru | point |
+-------+-----------------+-----------+-----------+---------+---------+-------+
| 12346 | Siti Aisyah     | Perempuan | XII IPS 2 | IPS     |       2 |     0 |
| 12345 | Ali Maulana     | Laki-laki | XII IPA 1 | IPA     |       1 |    10 |
| 12347 | Arif syarifudin | Laki-laki | XII IPA 3 | IPA     |       1 |    12 |
| 12348 | Dewi Pratiwi    | Perempuan | XII IPS 1 | IPS     |       3 |    15 |
+-------+-----------------+-----------+-----------+---------+---------+-------+
4 rows in set (0.001 sec)

MariaDB [nekat_saku]> select * from siswa order by point desc;
+-------+-----------------+-----------+-----------+---------+---------+-------+
| NIS   | nama_siswa      | JK        | Kelas     | jurusan | Id_guru | point |
+-------+-----------------+-----------+-----------+---------+---------+-------+
| 12348 | Dewi Pratiwi    | Perempuan | XII IPS 1 | IPS     |       3 |    15 |
| 12347 | Arif syarifudin | Laki-laki | XII IPA 3 | IPA     |       1 |    12 |
| 12345 | Ali Maulana     | Laki-laki | XII IPA 1 | IPA     |       1 |    10 |
| 12346 | Siti Aisyah     | Perempuan | XII IPS 2 | IPS     |       2 |     0 |
+-------+-----------------+-----------+-----------+---------+---------+-------+
4 rows in set (0.001 sec)
```

- **Fungsi Like**

Pada fungsi like kita dapat memungkinkan mencari pola sesuai keinginan kita, dengan operator fungsi like dibawah ini:

Like Operator	Deskripsi
WHERE nama LIKE 'a%'	Mencari nilai apapun pada field nama yang dimulai huruf “a”
WHERE nama LIKE '%a'	Mencari nilai apapun pada field nama yang diakhiri huruf “a”
WHERE nama LIKE '%or%'	Mencari nilai apapun pada field nama yang di dalamnya terdapat huruf “or”
WHERE nama LIKE '_r%'	Mencari nilai apapun pada field nama yang karakter ke-duanya huruf “r”
WHERE nama LIKE 'a_%'	Mencari nilai apapun pada field nama yang dimulai dengan huruf “a” dan panjangnya minimal 2 karakter
WHERE nama LIKE 'a__%'	Mencari nilai apapun pada field nama yang dimulai dengan huruf “a” dan panjangnya minimal 3 karakter
WHERE nama LIKE 'a%o'	Mencari nilai apapun pada field nama yang dimulai huruf “a” dan diakhiri huruf “o”

Lalu untuk penulisan syntax fungsi like yaitu : select [kondisi] from [tabel] where [field] LIKE [operator];



- **Agregat**

Ageragat adalah fungsi untuk melakukan perhitungan dari suatu data. Adapun beberapa jenis agregat seperti :
-AVG() : untuk mencari rata rata
-MAX() : untuk mencari nilai terbesar
-MIN() : untuk mencari nilai terkecil
-SUM() : untuk menjumlahkan
-COUNT(): untuk menghitung jumlah record
Cara penulisannya yaitu : select [agregat] ([nama field]) FROM [nama tabel];

```
MariaDB [nekat_saku]> select AVG(point) from siswa;
+------------+
| AVG(point) |
+------------+
|     9.2500 |
+------------+
1 row in set (0.001 sec)

MariaDB [nekat_saku]> select MAX(point) from siswa;
+------------+
| MAX(point) |
+------------+
|         15 |
+------------+
1 row in set (0.001 sec)

MariaDB [nekat_saku]> select MIN(point) from siswa;
+------------+
| MIN(point) |
+------------+
|          0 |
+------------+
1 row in set (0.001 sec)

MariaDB [nekat_saku]> select SUM(point) from siswa;
+------------+
| SUM(point) |
+------------+
|         37 |
+------------+
1 row in set (0.001 sec)

MariaDB [nekat_saku]> select COUNT(point) from siswa;
+--------------+
| COUNT(point) |
+--------------+
|            4 |
+--------------+
1 row in set (0.000 sec)
```

- **Penggunaan alias**

Kemudian ada penggunaan alias, dimana alias ini digunakan untuk penulisan kode suapaya lebih singkat. Kami akan memberikan contoh fungsi alias didalam agregat dengan syntax :
SELECT [agregat] ([nama field]) AS “[nama alias]” FROM [table];

- Dengan Alias

```
MariaDB [nekat_saku]> select AVG(point) AS "rata rata point pelanggaran siswa" FROM siswa;
+---------------------------------+
| rata rata point pelanggaran siswa |
+---------------------------------+
|                          9.2500 |
+---------------------------------+
1 row in set (0.001 sec)
```

- Tanpa Alias

```
MariaDB [nekat_saku]> select AVG(point) from siswa;
+-------------+
| AVG(point) |
+-------------+
|     9.2500 |
+-------------+
1 row in set (0.001 sec)
```

- **Perintah Group By**

Group by digunakan untuk mengelompokan data yang sama berdasarkab satu atau lebih kolom.
Dengan syntax : select [kondisi] from [table] group by [field dituju];

```
MariaDB [nekat_saku]> select * from siswa group by jurusan;
+-------+-------------+-----------+-----------+---------+---------+-------+
| NIS   | nama_siswa  | JK        | Kelas     | jurusan | Id_guru | point |
+-------+-------------+-----------+-----------+---------+---------+-------+
| 12345 | Ali Maulana | Laki-laki | XII IPA 1 | IPA     |       1 |    10 |
| 12346 | Siti Aisyah | Perempuan | XII IPS 2 | IPS     |       2 |     0 |
+-------+-------------+-----------+-----------+---------+---------+-------+
2 rows in set (0.019 sec)
```