|  |  |  |
| --- | --- | --- |
| **MKK “DIAGRAM HUBUNGAN ANTAR ENTITAS (ERD)”** | | |
| **LKPD 1** | **:** | **KEGIATAN INDIVIDU** |
| **SEMESTER** | **:** | **GASAL** |
| **KELAS** | **:** | **XI RPL** |
| **TAHUN AJARAN** | **:** | **2025/2026** |

1. **TUJUAN PEMBELAJARAN**

Setelah melaksanakan kegiatan pembelajaran peserta didik mampu :

1. Mendiskripsikan definisi ERD
2. Menentukan simbol-simbol pada ERD
3. Menentukan jenis-jenis entitas, atribut, key, relasi, kardinalitas, dan batasan partisipan pada ERD
4. Mendiskripsikan Tipe Data
5. Memahami jenis-jenis tipe data
6. **ALAT DAN BAHAN**
7. Laptop/Komputer
8. Software membuat ERD (Microsoft Visio, Edraw, dan lain-lain)
9. Software XAMPP
10. Bahan Ajar
11. Internet
12. **KESELAMATAN KERJA**
    * Menyalakan dan mematikan komputer/laptop sesuai dengan prosedur !
    * Menggunakan internet dengan bijak !
13. **DASAR TEORI**
    * 1. **Pengertian ERD (Entity Relationship Diagram)**

Diagram relasi entitas atau *entity-relationship diagram* (ERD) adalah suatu diagram dalam bentuk gambar atau simbol yang mengidentifikasi tipe dari entitas di dalam suatu sistem yang diuraikan dalam data dengan atributnya, dan menjelaskan hubungan atau relasi diantara entitas tersebut.

* + 1. **Simbol/Notasi ERD**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Notasi** | **Komponen** | **Keterangan** |
| 1 |  | Entitas (*Entity*) | Obyek nyata atau konsep yang disimpan datanya, misalnya: buku, siswa, jurusan, dll |
| 2 |  | Entitas Lemah (*Weak Entity*) | Entitas yang keberadaannya tergantung pada entitas lain |
| 3 |  | Relasi (*Relationship*) | Hubungan antar entitas, misalnya: meminjam, membeli, mengajar, dll |
| 4 |  | *Identifying Relationship* | Relasi yang menghubungkan entitas lemah dengan entitas pendukungnya |
| 5 |  | Atribut | Karakteristik atau informasi dari entitas, misalnya: Nama, Alamat, NISN, dll |
| 6 |  | Atribut *primary key* | Atribut unik untuk identifikasi entitas, misalnya: NISN, id\_buku, id\_jurusan, dll |
| 7 |  | Atribut *multivalue* | Atribut yang mempunyai lebih dari satu nilai, misalnya: nomor\_telepon, keahlian |
| 8 |  | Atribut *composite* | Atribut yang terdiri dari beberapa bagian, misalnya: Nama (nama depan, nama belakang) |
| 9 |  | Atribut *derivatif* | Atribut yang tidak harus disimpan dalam basis data atau atribut yang dihasilkan atribut lain dari suatu relasi |
| 10 | R  E2  E1 | *Total participation* of E2 In R | Suatu entitas **wajib** berelasi dengan entitas lainnya, misalnya: entitas siswa wajib memiliki nilai. |
| 11 | E1  R  E2  1  N | *Cardinality ratio* 1 : N for E1 : E2 In R | Suatu entitas memiliki relasi *one to many*, misalnya: satu siswa memiliki banyak nilai |

* + 1. **Tipe Data**
       1. **Pengertian Tipe Data**

**Tipe data** adalah **jenis data** yang dapat disimpan dalam suatu kolom di dalam **basis data** (database). Tipe data menentukan **bentuk** dan **batasan nilai** yang bisa dimasukkan ke dalam kolom, seperti angka, teks, tanggal, atau boolean (benar/salah).

* + - 1. **Fungsi Tipe Data**

1. Mengontrol **jenis data** yang boleh dimasukkan (contoh: hanya angka, hanya teks).
2. Membantu **menghemat ruang penyimpanan** di database.
3. Menjaga **integritas dan validitas data**.
4. Mempermudah pengolahan dan pencarian data sesuai jenisnya.
   * + 1. **Jenis-Jenis Tipe Data**

Tipe data numerik (angka)

Digunakan untuk menyimpan data angka, baik bilangan bulat maupun pecahan/desimal.

| **Tipe Data** | **Keterangan** |
| --- | --- |
| TINYINT | Bilangan bulat sangat kecil (−128 s/d 127) |
| SMALLINT | Bilangan bulat kecil (−32.768 s/d 32.767) |
| MEDIUMINT | Bilangan bulat sedang |
| INT / INTEGER | Bilangan bulat standar (−2.147.483.648 s/d 2.147.483.647) |
| BIGINT | Bilangan bulat besar |
| DECIMAL(p,s) | Bilangan desimal dengan presisi tetap, cocok untuk keuangan |
| FLOAT(p) | Bilangan pecahan presisi tunggal |
| DOUBLE | Bilangan pecahan presisi ganda |

Tipe data string (teks)

Digunakan untuk menyimpan teks, huruf, dan simbol.

| **Tipe Data** | **Keterangan** |
| --- | --- |
| CHAR(n) | Teks tetap (fixed-length), maksimal 255 karakter |
| VARCHAR(n) | Teks variabel (variable-length), maksimal 65.535 karakter |
| TEXT | Teks panjang (hingga 65.535 karakter) |
| TINYTEXT, MEDIUMTEXT, LONGTEXT | Versi lain dari TEXT dengan kapasitas berbeda |
| ENUM | Pilihan data tertentu, misal: ENUM('L','P') untuk jenis kelamin |
| SET | Menyimpan satu atau lebih nilai dari daftar |

Tipe data datetime (tanggal dan waktu)

Digunakan untuk menyimpan informasi waktu dan tanggal.

| **Tipe Data** | **Format** | **Keterangan** |
| --- | --- | --- |
| DATE | YYYY-MM-DD | Menyimpan tanggal |
| TIME | HH:MM:SS | Menyimpan jam |
| DATETIME | YYYY-MM-DD HH:MM:SS | Gabungan tanggal dan waktu |
| TIMESTAMP | YYYY-MM-DD HH:MM:SS | Waktu sistem (otomatis update) |
| YEAR | YYYY | Menyimpan tahun (4 digit) |

Tipe data BLOB

**BLOB** adalah singkatan dari **Binary Large Object** yaitu tipe data di MySQL yang digunakan untuk menyimpan **data biner dalam jumlah besar**, seperti:

* Gambar (JPEG, PNG)
* Video
* File PDF, DOCX
* Data terenkripsi

| **Tipe Data BLOB** | **Ukuran Maksimum** | **Keterangan** |
| --- | --- | --- |
| TINYBLOB | 255 byte | Untuk file berukuran sangat kecil. |
| BLOB | 65.535 byte (~64 KB) | Digunakan untuk file ukuran kecil hingga sedang. |
| MEDIUMBLOB | 16.777.215 byte (~16 MB) | Cocok untuk file gambar atau audio sedang. |
| LONGBLOB | 4.294.967.295 byte (~4 GB) | Digunakan untuk menyimpan video, file besar, dll. |

### ****Catatan Penting tentang BLOB****

* Data BLOB **tidak bisa dibaca langsung** seperti teks.
* Biasanya digunakan bersama **aplikasi frontend** (PHP, Java, Python) untuk mengunggah dan menampilkan file.
* Untuk performa, sering disarankan menyimpan file di server dan hanya menyimpan **path atau link file** di database.

### ****Kapan Menggunakan BLOB?**** Gunakan BLOB jika:

* File tidak dapat dikonversi menjadi teks.
* Ingin menyimpan file langsung di database, bukan di sistem file.
* Sistem memerlukan keamanan ekstra (data sensitif).
  + - 1. **Contoh Penggunaan Tipe Data dalam MySQL**

| **Jenis** | **Tipe Data** | **Contoh Nilai** | **Digunakan Untuk** |
| --- | --- | --- | --- |
| **Angka** | INT, DECIMAL, FLOAT | 10, 3.14 | Umur, Harga, Nilai |
| **Teks** | CHAR, VARCHAR, TEXT | "Andi", "Jl. Merdeka" | Nama, Alamat |
| **Tanggal/Waktu** | DATE, TIME, DATETIME | "2025-07-24" | Tanggal lahir, Waktu transaksi |
| **Boolean** | BOOLEAN | TRUE, FALSE | Status aktif, login |
| **Pilihan** | ENUM | 'L', 'P' | Jenis Kelamin, Status |

* + - 1. **Tips Memilih Tipe Data**

1. Gunakan **VARCHAR** jika panjang teks bisa bervariasi.
2. Gunakan **DECIMAL** untuk data keuangan agar lebih akurat.
3. Gunakan **ENUM** untuk data pilihan tetap seperti jenis kelamin, status, dll.
4. Gunakan **INT** untuk ID atau data angka umum.
5. **TUGAS INDIVIDU**
6. Jelaskan yang anda ketahui tentang ERD serta apa manfaat dari pembuatan ERD?
7. Sebutkan simbol pada ERD serta beri penjelasannya?
8. Jelaskan yang dimaksud dengan entitas dan sebutkan jenis-jenisnya disertai penjelasannya serta contoh?
9. Jelaskan yang demaksud dengan atribut dan sebutkan jenis-jenisnya disertai penjelasannya serta contoh?
10. Jelaskan yang demaksud dengan key dan sebutkan jenis-jenisnya disertai penjelasannya serta contoh?
11. Jelaskan yang demaksud dengan relasi dan berikan contohnya?
12. Jelaskan yang demaksud dengan kardinaliatas dan sebutkan jenis-jenisnya disertai penjelasannya serta contoh?
13. Jelaskan yang demaksud dengan batasan partisipan dan sebutkan jenis-jenisnya disertai penjelasannya serta contoh?
14. Carilah contoh gambar ERD sederhana serta beri penjelasaanya?
15. Studi kasus:

**Sebuah sistem akademik mencatat data siswa, guru, dan mata pelajaran. Siswa dapat menerima lebih dari satu mata pelajaran dan satu mata pelajaran bisa diampu oleh lebih dari satu guru.**

**Pertanyaan:**Buatlah desain **struktur ERD** berdasarkan deskripsi tersebut, sebutkan entitas, relasi, dan jenis relasinya!

1. Studi kasus:

Tabel produk memiliki kolom: id\_produk, nama\_produk, harga, tanggal\_produksi, foto\_produk.

**Pertanyaan:**  
Evaluasilah pemilihan tipe data berikut:

* id\_produk: INT
* nama\_produk: VARCHAR(100)
* harga: FLOAT
* tanggal\_produksi: DATE
* foto\_produk: TEXT

Apakah pemilihan tipe data sudah tepat? Jika tidak, jelaskan alasannya dan berikan alternatif yang lebih tepat!

1. Buatlah **tabel MySQL dan struktur ERD** sederhana untuk sistem kasir mini market yang menyimpan data pelanggan, produk, dan transaksi penjualan.
2. Apa dampaknya jika seorang perancang database **tidak memperhatikan pemilihan tipe data dengan tepat** dalam pembuatan tabel basis data?
3. Rancanglah sebuah tabel karyawan yang menyimpan:

* ID karyawan
* Nama lengkap
* Jenis kelamin
* Tanggal lahir
* Foto karyawan

Tentukan nama kolom, tipe data, dan alasan pemilihannya!

1. Studi kasus:

Sebuah toko buku online ingin membuat sistem informasi untuk mencatat transaksi penjualan buku. Dalam sistem ini, data yang perlu dicatat meliputi:

* **Data Pelanggan** yang berisi informasi: ID pelanggan, nama lengkap, alamat email, dan nomor telepon.
* **Data Buku** yang berisi: ID buku, judul buku, pengarang, harga, dan stok.
* **Data Transaksi** yang mencatat siapa membeli buku apa, kapan pembelian dilakukan, dan berapa jumlah buku yang dibeli.
* Satu pelanggan bisa melakukan beberapa transaksi, dan satu transaksi bisa melibatkan lebih dari satu buku.

**Pertanyaan:**

Identifikasi entitas yang terdapat dalam studi kasus di atas!

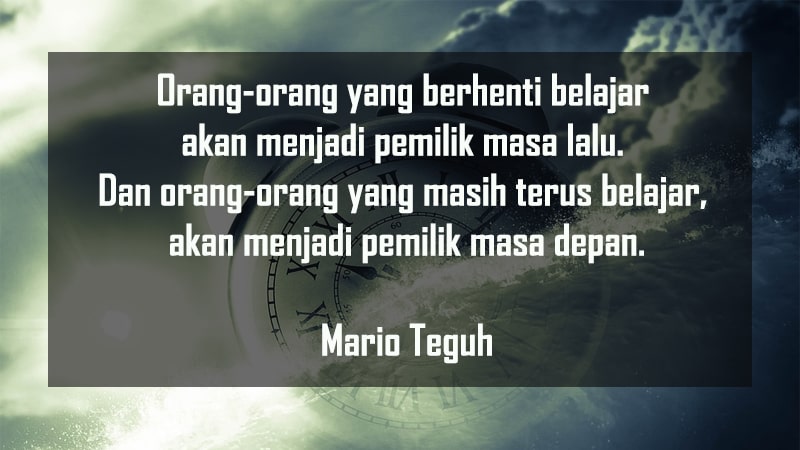
Buatlah ERD sederhana berdasarkan studi kasus tersebut, lengkap dengan relasi antar entitasnya dan jenis relasinya (1:1, 1:N, M:N)!

Tentukan struktur tabel dan tipe data yang sesuai untuk masing-masing entitas tersebut!

Mengapa penggunaan tipe data DECIMAL(10,2) lebih disarankan dibanding FLOAT untuk menyimpan harga buku?

1. Sebutkan dan jelaskan **semua** jenis tipe data yang ada di MySQL?

* **Setelah mengerjakan segera diupload di Teams**



**JAWABAN**

1. ERD adalah gambar atau diagram yang digunakan untuk menggambarkan hubungan antar data (entitas) dalam suatu sistem. ERD menunjukkan entitas, atribut (ciri-ciri entitas), dan hubungan antar entitas tersebut.

**Manfaat ERD:**

* Membantu **memahami struktur data** dalam sistem.
* Memudahkan dalam **merancang database** secara sistematis.
* Memperjelas hubungan antar data, seperti siapa yang berhubungan dengan siapa.
* Membantu komunikasi antara **programmer, analis sistem, dan pengguna**.
* Sebagai acuan dalam **pembuatan tabel database**.

2.

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Notasi** | **Komponen** | **Keterangan** |
| 1 |  | Entitas (*Entity*) | Obyek nyata atau konsep yang disimpan datanya, misalnya: buku, siswa, jurusan, dll |
| 2 |  | Entitas Lemah (*Weak Entity*) | Entitas yang keberadaannya tergantung pada entitas lain |
| 3 |  | Relasi (*Relationship*) | Hubungan antar entitas, misalnya: meminjam, membeli, mengajar, dll |
| 4 |  | *Identifying Relationship* | Relasi yang menghubungkan entitas lemah dengan entitas pendukungnya |
| 5 |  | Atribut | Karakteristik atau informasi dari entitas, misalnya: Nama, Alamat, NISN, dll |
| 6 |  | Atribut *primary key* | Atribut unik untuk identifikasi entitas, misalnya: NISN, id\_buku, id\_jurusan, dll |
| 7 |  | Atribut *multivalue* | Atribut yang mempunyai lebih dari satu nilai, misalnya: nomor\_telepon, keahlian |
| 8 |  | Atribut *composite* | Atribut yang terdiri dari beberapa bagian, misalnya: Nama (nama depan, nama belakang) |
| 9 |  | Atribut *derivatif* | Atribut yang tidak harus disimpan dalam basis data atau atribut yang dihasilkan atribut lain dari suatu relasi |
| 10 | R  E2  E1 | *Total participation* of E2 In R | Suatu entitas **wajib** berelasi dengan entitas lainnya, misalnya: entitas siswa wajib memiliki nilai. |
| 11 | E1  R  E2  1  N | *Cardinality ratio* 1 : N for E1 : E2 In R | Suatu entitas memiliki relasi *one to many*, misalnya: satu siswa memiliki banyak nilai |

3.Entitas adalah objek nyata atau konsep yang informasinya disimpan dalam database. Entitas bisa berupa orang, benda, tempat, atau kejadian yang memiliki data.

1.**Entitas Kuat (Strong Entity):**

Entitas yang bisa berdiri sendiri tanpa tergantung entitas lain.**Contoh:** Siswa, Guru, Buku.

2.**Entitas Lemah (Weak Entity):**

Entitas yang keberadaannya **bergantung pada entitas lain**. **Contoh:** Detail\_Transaksi (bergantung pada Transaksi), Anak (bergantung pada Orang Tua)

* + 1. Atribut adalah informasi atau ciri-ciri dari sebuah entitas. Misalnya, entitas **Siswa** bisa memiliki atribut **nama, alamat, NISN, dan tanggal lahir**.

### ****Jenis-Jenis Atribut:****

1.**Atribut Sederhana (Simple Attribute):**

Atribut yang tidak bisa dipecah lagi.**Contoh:** NISN, umur.

**2.Atribut Komposit (Composite Attribute):**

Atribut yang bisa dipecah menjadi bagian lebih kecil. **Contoh:** Nama lengkap → bisa dipecah menjadi nama depan dan nama belakang.

3.**Atribut Multivalued:**

Atribut yang bisa punya lebih dari satu nilai.**Contoh:** Nomor telepon (satu orang bisa punya lebih dari satu nomor).

**4.Atribut Kunci (Key Attribute):**

Atribut yang digunakan untuk **mengidentifikasi unik** setiap entitas.**Contoh:** NISN untuk entitas Siswa, ID\_Buku untuk entitas Buku.

**5.Atribut Derivatif (Derived Attribute):**

Atribut yang **didapat dari atribut lain**, tidak disimpan langsung.**Contoh:** Umur (dihitung dari tanggal lahir), total harga (jumlah × harga satuan).

5.Key adalah **atribut atau gabungan atribut** yang digunakan untuk **mengidentifikasi data secara unik** di dalam tabel database

### ****Jenis-Jenis Key:****

1. **Primary Key (Kunci Utama):**

Kunci utama yang **unik dan tidak boleh kosong (NULL)**.Digunakan untuk membedakan setiap baris data. **Contoh:** id\_siswa, id\_buku.

**2.Candidate Key:**

Atribut yang **bisa dijadikan** primary key karena nilainya unik. Tapi hanya **satu** yang dipilih sebagai primary key, sisanya disebut **alternate key**.**Contoh:** NISN dan No. KTP bisa dijadikan primary key untuk siswa.

3. **Alternate Key:**

Candidate key yang **tidak dipilih** sebagai primary key.**Contoh:** Jika NISN jadi primary key, maka No. KTP jadi alternate key.

4.**Foreign Key (Kunci Tamu):**

Kunci yang menghubungkan ke **primary key di tabel lain.**Digunakan untuk membentuk **relasi antar tabel**.**Contoh:** id\_siswa di tabel **nilai** adalah foreign key dari tabel **siswa**.

**5.Composite Key:**

Gabungan dari **dua atau lebih atribut** yang bersama-sama menjadi primary key.**Contoh:** id\_transaksi + id\_produk pada tabel detail\_transaksi.

6 .Relasi adalah **hubungan antar entitas** dalam sebuah sistem. Relasi menjelaskan bagaimana satu entitas **berkaitan** dengan entitas lainnya.

### ****Contoh Relasi:****

**Siswa – Meminjam – Buku**  
➤ Artinya: Satu siswa bisa meminjam satu atau lebih buku.  
➤ Relasi: **Meminjam**  
➤ Entitas yang terhubung: **Siswa** dan **Buku**

7.Kardinalitas adalah **jumlah hubungan** yang bisa terjadi antara satu entitas dengan entitas lainnya dalam suatu relasi. Dengan kata lain, ini menunjukkan **berapa banyak** entitas yang bisa saling terhubung.

1. **One to One (1 : 1)**

Satu entitas A hanya bisa berelasi dengan **satu entitas B**, dan sebaliknya. **Contoh:**  
➤ Setiap orang punya satu KTP, dan setiap KTP hanya untuk satu orang.

**2.One to Many (1 : N)**

Satu entitas A bisa berelasi dengan **banyak entitas B**, tapi entitas B hanya ke satu entitas A. **Contoh:**  
➤ Satu guru mengajar banyak siswa, tapi satu siswa hanya diajar oleh satu wali kelas.

**3.Many to Many (M : N)**

Banyak entitas A bisa berelasi dengan **banyak entitas B**.**Contoh:**  
➤ Siswa bisa mengambil banyak mata pelajaran, dan satu mata pelajaran bisa diambil banyak siswa.

1. Batasan partisipan menjelaskan **seberapa wajib** sebuah entitas **terlibat dalam suatu relasi**. Ini menentukan apakah entitas **harus** ikut dalam relasi atau **boleh tidak**.

### ****Jenis-Jenis Batasan Partisipan:****

1. **Partisipasi Total (Total Participation):**

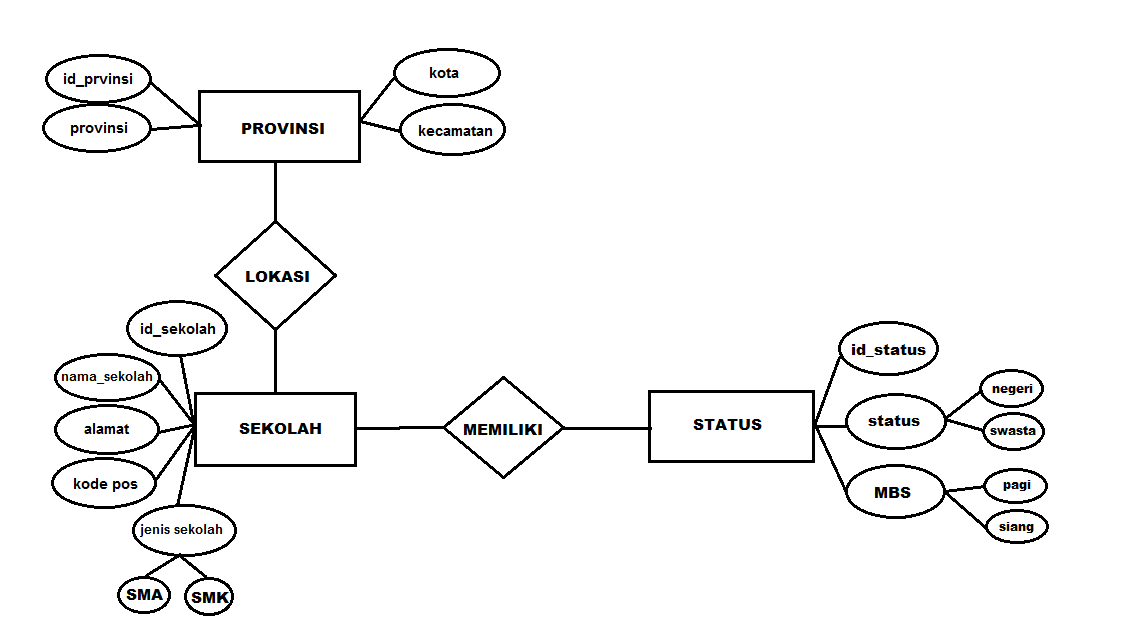
Semua entitas **harus ikut serta** dalam relasi.Digambarkan dengan **garis tebal** pada ERD.

**Contoh:**  
➤ Setiap siswa **wajib punya nilai** → entitas Siswa **wajib** terlibat dalam relasi Memiliki dengan entitas Nilai.

1. **Partisipasi Parsial (Partial Participation):**

Tidak semua entitas **harus ikut** dalam relasi. Digambarkan dengan **garis biasa (tipis)** pada ERD.

**Contoh:**  
➤ Tidak semua guru menjadi wali kelas → artinya **hanya sebagian** entitas Guru yang ikut dalam relasi Menjadi\_Wali\_Kelas.

9.

窗体顶端

Diagram ini menggambarkan sistem informasi data sekolah berdasarkan:

* **Lokasi (provinsi, kota, kecamatan)**
* **Identitas sekolah (nama, alamat, jenis sekolah)**
* **Status sekolah (negeri/swasta, pagi/siang)**

### 10.✅ ****Desain Struktur ERD:****

#### **Entitas:**

**SISWA**

id\_siswa (PK)

nama\_siswa

kelas

**GURU**

id\_guru (PK)

nama\_guru

mata\_pelajaran\_diampu

**MATA\_PELAJARAN**

id\_mapel (PK)

nama\_mapel

jumlah\_jam

### ****Relasi:****

**SISWA – MENGIKUTI – MATA\_PELAJARAN**

**Jenis Relasi:** Many to Many (M:N)

Karena satu siswa bisa mengikuti banyak mapel dan satu mapel bisa diikuti banyak siswa.

➤ Dibuat tabel relasi tambahan: SISWA\_MAPEL

**SISWA\_MAPEL**

id\_siswa (FK)

id\_mapel (FK)

**GURU – MENGAJAR – MATA\_PELAJARAN**

**Jenis Relasi:** Many to Many (M:N)

Karena satu guru bisa mengajar banyak mapel, dan satu mapel bisa diajar oleh banyak guru.

➤ Dibuat tabel relasi tambahan: GURU\_MAPEL

**GURU\_MAPEL**

id\_guru (FK)

id\_mapel (FK

1. Evaluasi

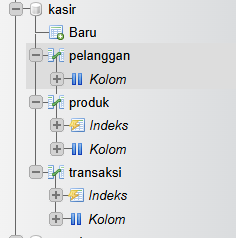
**-foto\_produk: TEXT**

**Kurang tepat**, karena TEXT menyimpan data teks, bukan file gambar.Kalau ingin **menyimpan file langsung di database**, sebaiknya gunakan BLOB.Tapi **lebih disarankan**: simpan **path file** saja, dan filenya diletakkan di folder server.**Alternatif yang lebih baik:** foto\_produk: VARCHAR(255) → untuk menyimpan **nama atau path file** ke gambar.

**-harga: FLOAT**

**Kurang tepat**, karena FLOAT bisa menyebabkan **pembulatan atau ketidakakuratan angka**, terutama untuk keuangan.**Alternatif yang lebih tepat:** DECIMAL(10,2)

12.



[PELANGGAN] --------< [TRANSAKSI] >-------- [PRODUK]

Entitas:

- PELANGGAN

- PRODUK

- TRANSAKSI (sebagai relasi sekaligus entitas penghubung)

1. **PELANGGAN**

**CREATE TABLE pelanggan (**

**id\_pelanggan INT PRIMARY KEY AUTO\_INCREMENT,**

**nama\_pelanggan VARCHAR(100),**

**no\_telepon VARCHAR(15),**

**alamat TEXT**

**);**

**2.**PRODUK****

**CREATE TABLE produk (**

**id\_produk INT PRIMARY KEY AUTO\_INCREMENT,**

**nama\_produk VARCHAR(100),**

**harga DECIMAL(10,2),**

**stok INT**

**);**

**3.TRANSAKSI**

**CREATE TABLE transaksi (**

**id\_transaksi INT PRIMARY KEY AUTO\_INCREMENT,**

**id\_pelanggan INT,**

**tanggal\_transaksi DATETIME,**

**FOREIGN KEY (id\_pelanggan) REFERENCES pelanggan(id\_pelanggan)**

**);**

13.**Pemborosan Ruang Penyimpanan (Storage):**

Contoh: Menggunakan TEXT untuk kolom nama yang seharusnya cukup pakai VARCHAR(50) akan menghabiskan memori lebih banyak.

**Kesalahan Data (Data Tidak Akurat):**

Misalnya: Menyimpan harga dengan FLOAT bisa membuat hasil perhitungan keuangan menjadi **tidak akurat** karena pembulatan.

**Kinerja Database Menurun:**

Tipe data yang besar dan tidak efisien dapat memperlambat proses pencarian (query) dan pemrosesan data.

**Relasi Antar Tabel Bisa Rusak:**

Jika tipe data foreign key berbeda dengan primary key tujuan, bisa menyebabkan **error** saat membuat relasi.

**Validasi Data Sulit:**

Jika tidak menggunakan tipe data yang tepat, seperti tanggal disimpan sebagai teks (VARCHAR), maka database tidak bisa **memvalidasi** atau menyaring data dengan benar.

**Masalah Keamanan dan Integritas Data:**

Salah memilih tipe data bisa memungkinkan data yang tidak sesuai masuk ke sistem (contoh: huruf dimasukkan ke kolom umur).

1. id\_karyawan INT PRIMARY KEY AUTO\_INCREMENT, -- ID unik tiap karyawan

nama\_lengkap VARCHAR(100), -- Menampung nama karyawan

jenis\_kelamin ENUM('L', 'P'), -- L = Laki-laki, P = Perempuan

tanggal\_lahir DATE, -- Format tanggal lahir (YYYY-MM-DD)

foto\_karyawan

#### 15.

**PELANGGAN**

**BUKU**

**TRANSAKSI**

**DETAIL\_TRANSAKSI** (tabel relasi untuk transaksi dengan banyak buku)

2.

Buku

Detail\_Transaksi

Pelanggan

Transaksi

· **PELANGGAN – TRANSAKSI:** 1 : N (satu pelanggan bisa banyak transaksi)

· **TRANSAKSI – BUKU (via DETAIL\_TRANSAKSI):** M : N (satu transaksi bisa banyak buku, dan satu buku bisa muncul di banyak transaksi)

3.-- Tabel Pelanggan

TABLE pelanggan

id\_pelanggan INT PRIMARY KEY AUTO\_INCREMENT,

nama\_lengkap VARCHAR(100),

email VARCHAR(100),

no\_telepon VARCHAR(15)

-- Tabel Buku

TABLE buku

id\_buku INT PRIMARY KEY AUTO\_INCREMENT,

judul VARCHAR(100),

pengarang VARCHAR(100),

harga DECIMAL(10,2),

stok INT

-- Tabel Transaksi

TABLE transaksi

id\_transaksi INT PRIMARY KEY AUTO\_INCREMENT,

id\_pelanggan INT,

tanggal DATETIME,

FOREIGN KEY (id\_pelanggan) REFERENCES pelanggan(id\_pelanggan)

-- Tabel Detail Transaksi

TABLE detail\_transaksi

id\_detail INT PRIMARY KEY AUTO\_INCREMENT,

id\_transaksi INT,

id\_buku INT,

jumlah INT,

subtotal DECIMAL(10,2),

FOREIGN KEY (id\_transaksi) REFERENCES transaksi(id\_transaksi),

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

4.Karena FLOAT **bisa membulatkan angka**, sedangkan DECIMAL(10,2) lebih **presisi dan aman** untuk data uang.

16. 1. Tipe Data Numerik

| **Tipe Data** | **Keterangan** |
| --- | --- |
| TINYINT | Bilangan bulat sangat kecil (−128 s/d 127) |
| SMALLINT | Bilangan bulat kecil (−32.768 s/d 32.767) |
| MEDIUMINT | Bilangan bulat sedang |
| INT / INTEGER | Bilangan bulat standar (−2.147.483.648 s/d 2.147.483.647) |
| BIGINT | Bilangan bulat besar |
| DECIMAL(p,s) | Bilangan desimal dengan presisi tetap, cocok untuk keuangan |
| FLOAT(p) | Bilangan pecahan presisi tunggal |
| DOUBLE | Bilangan pecahan presisi ganda |

2.Tipe data string

| **Tipe Data** | **Keterangan** |
| --- | --- |
| CHAR(n) | Teks tetap (fixed-length), maksimal 255 karakter |
| VARCHAR(n) | Teks variabel (variable-length), maksimal 65.535 karakter |
| TEXT | Teks panjang (hingga 65.535 karakter) |
| TINYTEXT, MEDIUMTEXT, LONGTEXT | Versi lain dari TEXT dengan kapasitas berbeda |
| ENUM | Pilihan data tertentu, misal: ENUM('L','P') untuk jenis kelamin |

1. Tipe data (date time)

| **Tipe Data** | **Format** | **Keterangan** |
| --- | --- | --- |
| DATE | YYYY-MM-DD | Menyimpan tanggal |
| TIME | HH:MM:SS | Menyimpan jam |
| DATETIME | YYYY-MM-DD HH:MM:SS | Gabungan tanggal dan waktu |
| TIMESTAMP | YYYY-MM-DD HH:MM:SS | Waktu sistem (otomatis update) |
| YEAR | YYYY | Menyimpan tahun (4 digit) |

1. Tpe data BLOB

| **Tipe Data BLOB** | **Ukuran Maksimum** | **Keterangan** |
| --- | --- | --- |
| TINYBLOB | 255 byte | Untuk file berukuran sangat kecil. |
| BLOB | 65.535 byte (~64 KB) | Digunakan untuk file ukuran kecil hingga sedang. |
| MEDIUMBLOB | 16.777.215 byte (~16 MB) | Cocok untuk file gambar atau audio sedang. |
| LONGBLOB | 4.294.967.295 byte (~4 GB) | Digunakan untuk menyimpan video, file besar, dll. |