

PT XYZ ingin membuat Aplikasi Sales, Dimana berisi data sales, produk PT XYZ, wilayah sales diseluruh provinsi Indonesia, nama klien, aktivitas sales ke klien, produk yang terjual harian di wilayah.

Dari deskripsi sebelumnya buatlah:

1. ERD table yang dibutuhkan

2. Data Manipulation Language (DML), Data Definition Language (DDL), dan Data Control Language (DCL).

3. Function, Procedure, Trigger

4. Tampilan Form yang dibuat

5. Graphic/Chart dari data yang di entri

## 🔧 Teknologi yang Digunakan

- PHP (Native, tanpa framework)
- MySQL / MariaDB
- phpMyAdmin (pengelolaan database)
- Chartjs

Struktur file sederhana:

/project-folder

```
|
|
|— db.php          # File koneksi database
|— index.php       # Dashboard grafik
|— penjualan.php   # Form input penjualan
|— produk.php      # CRUD Produk
|— sales.php       # CRUD Sales
|— klien.php       # CRUD Klien
|— aktivitas_sales.php # CRUD Aktivitas Sales
```

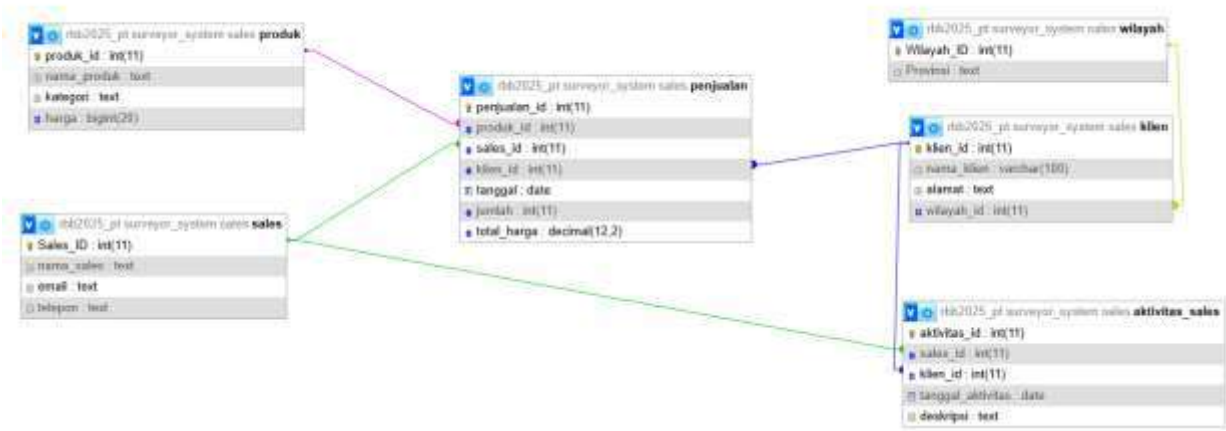
Future enhancement:

- Belum ada login atau kontrol hak akses
- Perlu validasi input lebih lanjut untuk digunakan secara publik.
- Disarankan untuk menambahkan Bootstrap agar tampilan lebih rapi dan responsif.
- Penambahan diagram pada dashboard yang dapat diakses
- Untuk penghapusan ditambahkan log otorisasi dan otentifikasi untuk pertanggungjawaban
- Penambahan navigator untuk pindah halaman

Installation & Running :

1. Pastikan sudah menginstall apache dan mysql server, untuk windows dapat digunakan xampp
2. Export rbb2025\_pt\_surveyor\_system\_sales.sql ke mysql anda
3. Copy paste isi dari [PHP Code] rbbsurveyor.zip ke PHP server anda. Bila menggunakan XAMPP di Windows maka akan ada di HTOCS
4. Bukan localhost/rbbsurveyor/index.php

## 1. ERD table yang dibutuhkan



### Entitas dan Relasi:

#### 1. Sales

- sales\_id (PK)
- nama\_sales
- email
- telepon

#### 2. Produk

- produk\_id (PK)
- nama\_produk
- kategori
- harga

#### 3. Wilayah

- wilayah\_id (PK)
- nama\_provinsi

#### 4. Klien

- klien\_id (PK)
- nama\_klien

- alamat
- wilayah\_id (FK)

#### **5. Aktivitas\_Sales**

- aktivitas\_id (PK)
- sales\_id (FK)
- klien\_id (FK)
- tanggal\_aktivitas
- deskripsi

#### **6. Penjualan**

- penjualan\_id (PK)
- produk\_id (FK)
- sales\_id (FK)
- klien\_id (FK)
- tanggal
- jumlah
- total\_harga

Relasi:

- 1 Sales bisa punya banyak Aktivitas dan Penjualan
- 1 Produk bisa dijual banyak kali
- 1 Klien bisa dihubungi banyak kali oleh Sales
- 1 Wilayah bisa punya banyak Klien

## 2. Data Manipulation Language (DML), Data Definition Language (DDL), dan Data Control Language (DCL).

Data Definition Language (DDL) : Digunakan untuk membuat dan mengatur struktur database: tabel, kolom, indeks, dll. Perintah : CREATE, ALTER, DROP, TRUNCATE

-- Create tables

```
CREATE TABLE `wilayah` (  
  `Wilayah_ID` int(11) NOT NULL AUTO_INCREMENT,  
  `Provinsi` text NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
CREATE TABLE `produk` (  
  `produk_id` int(11) NOT NULL AUTO_INCREMENT,  
  `nama_produk` text NOT NULL,  
  `kategori` text DEFAULT NULL,  
  `harga` bigint(20) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
CREATE TABLE `sales` (  
  `Sales_ID` int(11) NOT NULL AUTO_INCREMENT,  
  `nama_sales` text NOT NULL,  
  `email` text DEFAULT NULL,  
  `telepon` text DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
CREATE TABLE `klien` (  
  `klien_id` int(11) NOT NULL AUTO_INCREMENT,
```

```
`klien_id` int(11) NOT NULL AUTO_INCREMENT,  
`nama_klien` varchar(100) DEFAULT NULL,  
`alamat` text DEFAULT NULL,  
`wilayah_id` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
CREATE TABLE `aktivitas_sales` (  
`aktivitas_id` int(11) NOT NULL AUTO_INCREMENT,  
`sales_id` int(11) DEFAULT NULL,  
`klien_id` int(11) DEFAULT NULL,  
`tanggal_aktivitas` date DEFAULT NULL,  
`deskripsi` text DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
CREATE TABLE `penjualan` (  
`penjualan_id` int(11) NOT NULL AUTO_INCREMENT,  
`produk_id` int(11) DEFAULT NULL,  
`sales_id` int(11) DEFAULT NULL,  
`klien_id` int(11) DEFAULT NULL,  
`tanggal` date DEFAULT NULL,  
`jumlah` int(11) DEFAULT NULL,  
`total_harga` decimal(12,2) DEFAULT NULL,  
`notes` text DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

-- Indexes : PRIMARY KEY & RELATION KEY

```
ALTER TABLE `aktivitas_sales`  
ADD PRIMARY KEY (`aktivitas_id`),  
ADD KEY `sales_id` (`sales_id`),  
ADD KEY `klien_id` (`klien_id`);
```

```
ALTER TABLE `klien`  
ADD PRIMARY KEY (`klien_id`),  
ADD KEY `wilayah_id` (`wilayah_id`);
```

```
ALTER TABLE `penjualan`  
ADD PRIMARY KEY (`penjualan_id`),  
ADD KEY `produk_id` (`produk_id`),  
ADD KEY `sales_id` (`sales_id`),  
ADD KEY `klien_id` (`klien_id`);
```

```
ALTER TABLE `produk`  
ADD PRIMARY KEY (`produk_id`);
```

```
ALTER TABLE `sales`  
ADD PRIMARY KEY (`Sales_ID`);
```

```
ALTER TABLE `wilayah`  
ADD PRIMARY KEY (`Wilayah_ID`);
```

--- DROP tidak dilakukan karena itu akan menghapus seluruh table

--- TRUNCATE tidak dilakukan karena itu akan menghapus seluruh isi table

--- Database yg baik adalah yang memiliki change log, sehingga seluruh transaksi akan dilakukan melalui frontend dan tercatat flag deleted, bukan melakukan DROP/TRUNCATE yang tidak memiliki rekam jejak akuntabilitas.

**Data Manipulation Language (DML) : Digunakan untuk memanipulasi data di dalam tabel, seperti menambahkan, membaca, mengubah, atau menghapus data. Perintah : INSERT, SELECT, UPDATE, DELETE**

```
INSERT INTO `wilayah` (`Wilayah_ID`, `Provinsi`) VALUES
```

```
(1, 'Nanggroe Aceh Darussalam'),
```

```
(2, 'Sumatera Utara'),
```

```
(3, 'Sumatera Selatan'),
```

```
(4, 'Sumatera Barat'),
```

```
(5, 'Bengkulu'),
```

```
(6, 'Riau'),
```

```
(7, 'Kepulauan Riau'),
```

```
(8, 'Jambi'),
```

```
(9, 'Lampung'),
```

```
(10, 'Bangka Belitung'),
```

```
(11, 'Kalimantan Barat'),
```

```
(12, 'Kalimantan Timur'),
```

```
(13, 'Kalimantan Selatan'),
```

```
(14, 'Kalimantan Tengah'),
```

```
(15, 'Kalimantan Utara'),
```

```
(16, 'Banten'),
```

```
(17, 'DKI Jakarta'),
```

```
(18, 'Jawa Barat'),
```



(19, 'Jawa Tengah'),  
(20, 'Daerah Istimewa Yogyakarta'),  
(21, 'Jawa Timur'),  
(22, 'Bali'),  
(23, 'Nusa Tenggara Timur'),  
(24, 'Nusa Tenggara Barat'),  
(25, 'Gorontalo'),  
(26, 'Sulawesi Barat'),  
(27, 'Sulawesi Tengah'),  
(28, 'Sulawesi Utara'),  
(29, 'Sulawesi Tenggara'),  
(30, 'Sulawesi Selatan'),  
(31, 'Maluku Utara'),  
(32, 'Maluku'),  
(33, 'Papua Barat'),  
(34, 'Papua'),  
(35, 'Papua Tengah'),  
(36, 'Papua Pegunungan'),  
(37, 'Papua Selatan'),  
(38, 'Papua Barat Daya');

-- Insert sample sales

INSERT INTO Sales VALUES (1, 'Sales1', 'sales1@ptxyz.co.id', '08123456789');  
INSERT INTO Sales VALUES (2, 'Sales2', 'sales2@ptxyz.co.id', '08123456789');  
INSERT INTO Sales VALUES (3, 'Sales3', 'sales3@ptxyz.co.id', '08123456789');

-- Insert sample product

```
INSERT INTO Produk (`nama_produk`, `kategori`, `harga`) VALUES ('Produk1', 'Kategori1', 1000000);
```

```
INSERT INTO Produk (`nama_produk`, `kategori`, `harga`) VALUES ('Produk2', 'Kategori1', 1100000);
```

```
INSERT INTO Produk (`nama_produk`, `kategori`, `harga`) VALUES ('Produk3', 'Kategori1', 1200000);
```

```
INSERT INTO Produk (`nama_produk`, `kategori`, `harga`) VALUES ('Produk4', 'Kategori2', 1300000);
```

```
INSERT INTO Produk (`nama_produk`, `kategori`, `harga`) VALUES ('Produk5', 'Kategori2', 1400000);
```

```
INSERT INTO Produk (`nama_produk`, `kategori`, `harga`) VALUES ('Produk6', 'Kategori2', 1500000);
```

-- Insert sample klien

```
INSERT INTO `klien` (`nama_klien`, `alamat`, `wilayah_id`) VALUES ('PT Klien1', 'Jl. Merdeka No.1', 1);
```

```
INSERT INTO `klien` (`nama_klien`, `alamat`, `wilayah_id`) VALUES ('PT Klien2', 'Jl. Indonesia No.2', 2);
```

```
INSERT INTO `klien` (`nama_klien`, `alamat`, `wilayah_id`) VALUES ('PT Klien3', 'Jl. Asia-Afrika No.3', 3);
```

-- Insert aktivitas sales

```
INSERT INTO `aktivitas_sales` (`sales_id`, `klien_id`, `tanggal_aktivitas`, `deskripsi`) VALUES (1, 1, '2025-06-20', 'Presentasi produk');
```

```
INSERT INTO `aktivitas_sales` (`sales_id`, `klien_id`, `tanggal_aktivitas`, `deskripsi`) VALUES (1, 2, '2025-06-21', 'Presentasi produk');
```

```
INSERT INTO `aktivitas_sales` (`sales_id`, `klien_id`, `tanggal_aktivitas`, `deskripsi`) VALUES (2, 2, '2025-06-22', 'Demo produk');
```

```
INSERT INTO `aktivitas_sales` (`sales_id`, `klien_id`, `tanggal_aktivitas`,  
`deskripsi`) VALUES (2, 3, '2025-06-23', 'Tanda-tangan deal penjualan');
```

-- Insert penjualan

```
INSERT INTO `penjualan` (`produk_id`, `sales_id`, `klien_id`, `tanggal`, `jumlah`)  
VALUES (1, 2, 3, '2025-06-24', 3);
```

```
INSERT INTO `penjualan` (`produk_id`, `sales_id`, `klien_id`, `tanggal`, `jumlah`,  
`total_harga`, `notes`) VALUES (2, 2, 3, '2025-06-24', 3, 3200000, 'Diskon 100rb dari  
negosiasi');
```

```
INSERT INTO `penjualan` (`produk_id`, `sales_id`, `klien_id`, `tanggal`,  
`jumlah`) VALUES (6, 2, 3, '2025-06-25', 3);
```

Data Control Language (DCL) : Digunakan untuk mengatur hak akses pengguna ke data di database. Perintah : GRANT, REVOKE

-- Membuat user baru

```
CREATE USER 'admin1ITSurveyor'@'localhost' IDENTIFIED BY 'a4dm1sur2132h_23';
```

-- Memberikan hak akses

```
GRANT SELECT, INSERT, UPDATE ON *.* TO 'admin1ITSurveyor'@'localhost';
```

### 3. Function, Procedure, Triger

Function : Melihat penjualan per sales untuk hari ini

DELIMITER \$\$

```
CREATE FUNCTION TotalPenjualanSales(sales INT)
```

```
RETURNS DECIMAL(12,2)
```

```
DETERMINISTIC
```

```
BEGIN
```

```
    DECLARE total DECIMAL(12,2);
```

```
    SELECT SUM(total_harga) INTO total
```

```
    FROM Penjualan
```

```
    WHERE sales_id = sales AND tanggal = CURDATE();
```

```
    RETURN IFNULL(total, 0);
```

```
END$$
```

DELIMITER ;

**PEMANGGILAN FUNCTION:** *SELECT TotalPenjualanSales(1) AS total\_penjualan;*

*Penjelasan:*

*1 adalah parameter sales (misal sales\_id = 1)*


*AS total\_penjualan memberikan alias nama kolom hasil output supaya lebih jelas*

**Trigger :** Menghitung invoice penjualan dengan mengalikan jumlah barang yang dibeli dengan harga barang hanya bila notes tidak kosong.

*Penjelasan: Notes memiliki nilai adalah pada saat scenario penjualan dengan diskon atau nota tertentu. Pada saat kondisi ini, maka pada system sederhana ini perhitungan dapat dilakukan manual.*

DELIMITER \$\$

```
CREATE TRIGGER `HitungTotalHarga` BEFORE INSERT ON `penjualan` FOR EACH ROW
BEGIN
```

```
    DECLARE harga DECIMAL(12,2); --  Pindahkan ke atas
```

```
    IF NEW.notes IS NOT NULL THEN
```

```
        SELECT harga INTO harga FROM Produk WHERE produk_id = NEW.produk_id;
```

```
        SET NEW.total_harga = harga * NEW.jumlah;
```

```
    END IF;
```

```
END
```

```
$$
```

```
DELIMITER ;
```

**Procedure :** Melakukan kalkulasi otomatis pada invoice pada saat menambahkan produk terjual.

**Pemanggilan PROCEDURE:**

**Pemanggilan prosedur dengan notes berisi teks:**

```
CALL TambahPenjualan(1, 2, 3, '2025-06-23', 10, 'Kunjungan sales rutin');
```

**Pemanggilan prosedur dengan notes kosong atau NULL:**

```
CALL TambahPenjualan(1, 2, 3, '2025-06-23', 10, "");
```

*Penjelasan : Setiap kali terjadi penjualan dan data dimasukan ke tambah penjualan, hanya dengan memasukan kode produk, kode sales, dan kode klien, system dapat otomatis mendapatkan harga barang pada real-time disaat penjualan berdasarkan dari table produk. Dan, total harga dapat didapatkan secara real-time dengan mengalikan jumlah produk*

*terjual dengan harga barang. Namun, untuk penjualan dengan notes tertentu seperti diskon, maka total penjualan akan dikosongkan dan dapat diisi secara manual, seperti yang sudah ditentukan pada Trigger HitungTotalHarga.*

*Parameter urutannya sesuai definisi prosedur:*

- 1. pid (produk\_id)*
- 2. sid (sales\_id)*
- 3. kid (klien\_id)*
- 4. tgl (tanggal)*
- 5. jumlah (jumlah produk)*
- 6. pnotes (notes)*

DELIMITER \$\$

CREATE PROCEDURE TambahPenjualan(

IN pid INT,

IN sid INT,

IN kid INT,

IN tgl DATE,

IN jumlah INT,

IN pnotes TEXT

)

BEGIN

DECLARE harga\_produk DECIMAL(12,2);

IF pnotes IS NULL OR pnotes = " THEN

-- notes kosong, set total\_harga NULL

```
INSERT INTO Penjualan(produk_id, sales_id, klien_id, tanggal, jumlah, total_harga,  
notes)
```

```
VALUES (pid, sid, kid, tgl, jumlah, NULL, pnotes);
```

```
ELSE
```

```
-- notes ada isi, biarkan trigger hitung total_harga
```

```
INSERT INTO Penjualan(produk_id, sales_id, klien_id, tanggal, jumlah, notes)
```

```
VALUES (pid, sid, kid, tgl, jumlah, pnotes);
```

```
END IF;
```

```
END$$
```

```
DELIMITER ;
```

## 4. Tampilan Form yang dibuat

Dikarenakan waktu terbatas maka form CRUD hanya dibuat sederhana di PHP. Bila waktu memadai dapat digunakan framework yang lebih baik seperti Yii, ASP, Laravel, dsb.

1. Wilayah adalah data statis, sehingga dimasukan secara otomatis dari admin, sehingga tidak ada tampilannya di klien
2. Produk : **Produk.php** - Manajemen Data Produk  
CRUD data produk PT XYZ yang dijual oleh sales.
  - Field utama: produk\_id (auto increment), nama\_produk, kategori, harga.
  - Fitur: Tambah produk baru, edit, lihat daftar produk, dan hapus produk.
  - Digunakan sebagai referensi produk saat input penjualan.

### Data Produk

#### Tambah Produk

Nama Produk:

Kategori:

Harga:

---

#### Daftar Produk

ID	Nama Produk	Kategori	Harga	Aksi
1	Produk1	Kategori1	1,000,000	<a href="#">Edit</a>   <a href="#">Hapus</a>
2	Produk2	Kategori1	1,100,000	<a href="#">Edit</a>   <a href="#">Hapus</a>
3	Produk3	Kategori1	1,200,000	<a href="#">Edit</a>   <a href="#">Hapus</a>
4	Produk4	Kategori2	1,300,000	<a href="#">Edit</a>   <a href="#">Hapus</a>
5	Produk5	Kategori2	1,400,000	<a href="#">Edit</a>   <a href="#">Hapus</a>
6	Produk6	Kategori2	1,500,000	<a href="#">Edit</a>   <a href="#">Hapus</a>



### 3. Sales : **sales.php** - Manajemen Data Sales

- CRUD data sales PT XYZ.
- Field utama: Sales\_ID (auto increment), nama\_sales, email, telepon.
- Fitur: Tambah, lihat daftar, edit, hapus data sales.
- Digunakan untuk input dan update data sales yang melakukan penjualan.

## Data Sales

Nama:

Email:

Telepon:

ID	Nama	Email	Telepon	Aksi
1	Sales1	sales1@ptxyz.co.id	08123456789	<a href="#">Edit</a>   <a href="#">Hapus</a>
2	Sales2	sales2@ptxyz.co.id	08123456789	<a href="#">Edit</a>   <a href="#">Hapus</a>
3	Sales3	sales3@ptxyz.co.id	08123456789	<a href="#">Edit</a>   <a href="#">Hapus</a>

### 4. Klien : **klien.php** - Manajemen Data Klien

- CRUD data klien yang menjadi target sales.
- Field utama: klien\_id (auto increment), nama\_klien, alamat, wilayah\_id (relasi ke provinsi).
- Fitur: Tambah, lihat daftar, edit, hapus data klien.
- Wilayah diambil dari tabel wilayah untuk dropdown pilihan provinsi.

## Data Klien

Nama:

Alamat:

Wilayah:  ▼

ID	Nama	Alamat	Wilayah	Aksi
1	PT Klien1	Jl. Merdeka No.1	Nanggroe Aceh Darussalam	<a href="#">Edit</a>   <a href="#">Hapus</a>
2	PT Klien2	Jl. Indonesia No.2	Sumatera Utara	<a href="#">Edit</a>   <a href="#">Hapus</a>
3	PT Klien3	Jl. Asia-Afrika No.3	Sumatera Selatan	<a href="#">Edit</a>   <a href="#">Hapus</a>

### 5. Aktivitas sales : **aktivitas\_sales.php** - Manajemen Aktivitas Sales

- CRUD aktivitas kunjungan atau presentasi sales ke klien.
- Field utama: aktivitas\_id (auto increment), sales\_id, klien\_id, tanggal\_aktivitas, deskripsi.
- Fitur: Tambah catatan aktivitas, edit, lihat, dan hapus.
- Mempermudah tracking aktivitas sales terkait klien dan tanggalnya

## Aktivitas Sales

Sales:  ▼

Klien:  ▼

Tanggal:

Deskripsi:

ID	Sales	Klien	Tanggal	Deskripsi	Aksi
4	Sales2	PT Klien3	2025-06-23	Tanda-tangan deal penjualan	<a href="#">Hapus</a>
3	Sales2	PT Klien2	2025-06-22	Demo produk	<a href="#">Hapus</a>
2	Sales1	PT Klien2	2025-06-21	Presentasi produk	<a href="#">Hapus</a>
1	Sales1	PT Klien1	2025-06-20	Presentasi produk	<a href="#">Hapus</a>

## 6. Transaksi : penjualan.php

Form penjualan adalah multi-input krn 1 transaksi dapat terdiri dari multi produk

### Fitur Utama:

- Input **Sales ID** dan **Klien ID** satu kali untuk semua entri.
- Input beberapa penjualan sekaligus dalam bentuk tabel dinamis.
- **Produk dipilih dari dropdown** (menampilkan nama\_produk tapi menyimpan produk\_id).
- **Harga dan Total** dihitung otomatis berdasarkan jumlah dan harga produk.
- tanggal otomatis dari CURDATE().
- notes otomatis NULL.

### Form Penjualan

Sales ID:   
Klien ID:

Produk	Jumlah	Harga	Total	Aksi
Produk1 ▾	1	<input type="text"/>	<input type="text"/>	Hapus

## 5. Graphic/Chart dari data yang di entri

**Tombol Navigasi untuk menuju menu input :**

- + Input Penjualan: menuju penjualan.php
- Produk, Sales, Klien, Aktivitas Sales: master data dari masing-masing data

### **Ide Dashboard Tambahan (Opsional Bila Waktu Memadai)**

Berikut ide-ide dashboard yang dapat dibuat dari data yang sudah tersedia:

#### **1. Penjualan Harian (Line Chart)**

- Menampilkan total penjualan per hari.
- Bisa difilter per bulan.

#### **2. Penjualan per Provinsi (Map atau Bar Chart)**

- Gunakan relasi klien → wilayah → provinsi.

#### **3. Top 5 Produk Terlaris**

- Urutkan berdasarkan SUM(jumlah) terbanyak.

#### **4. Aktivitas Sales per Hari**

- Line chart aktivitas kunjungan sales (aktivitas\_sales).

#### **5. Konversi Aktivitas → Penjualan**

- Perbandingan jumlah aktivitas vs jumlah penjualan per sales.

#### **6. Penjualan per Kategori Produk**

- Gunakan kolom kategori dari tabel produk.

#### **7. Penjualan Klien Tertinggi**

- Ranking klien berdasarkan total belanja.

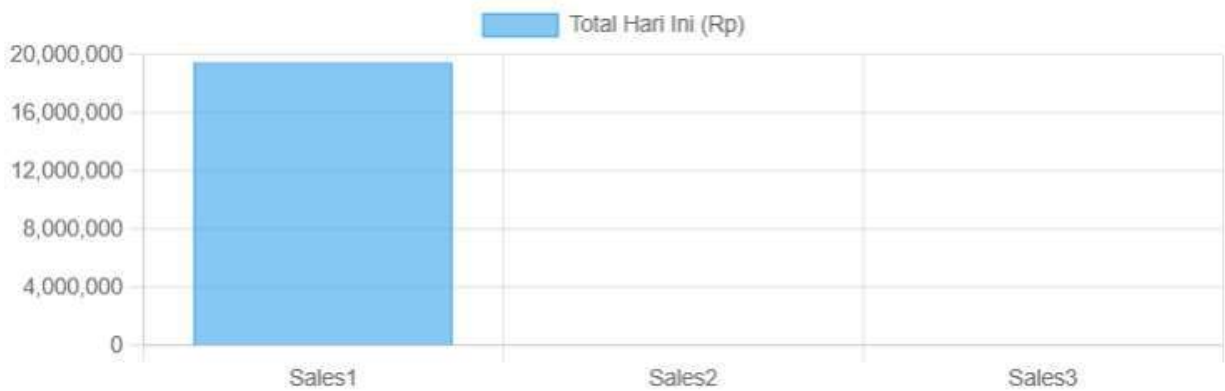
#### **8. Penjualan Bulanan per Sales**

- Menampilkan performa sales per bulan.

# Dashboard Penjualan

[+ Input Penjualan](#) | [Produk](#) | [Sales](#) | [Klien](#) | [Aktivitas Sales](#)

## Total Penjualan Hari Ini per Sales



## Jumlah Penjualan per Produk (Sepanjang Waktu)

