**LAPORAN PROYEK UAS BASIS DATA WARUNG SEMBAKO AZMI**

“Disusun Guna Memenuhi Tugas Akhir Mata Kuliah Basis Data*”*

****

Oleh Kelompok 4:

Tri Anggoro Budi 4523210108

Fitria Dwi Anggraeni 4523210142

Zahra Tsabitah 4523210145

Dosen:

Adi Wahyu Pribadi , S.Si., M.Kom

**S1-Teknik Informatika**

**Fakultas Teknik Universitas Pancasila**

**2024/2025**

1. **Latar Belakang**

A person standing next to another person

Description automatically generatedA diagram of a company

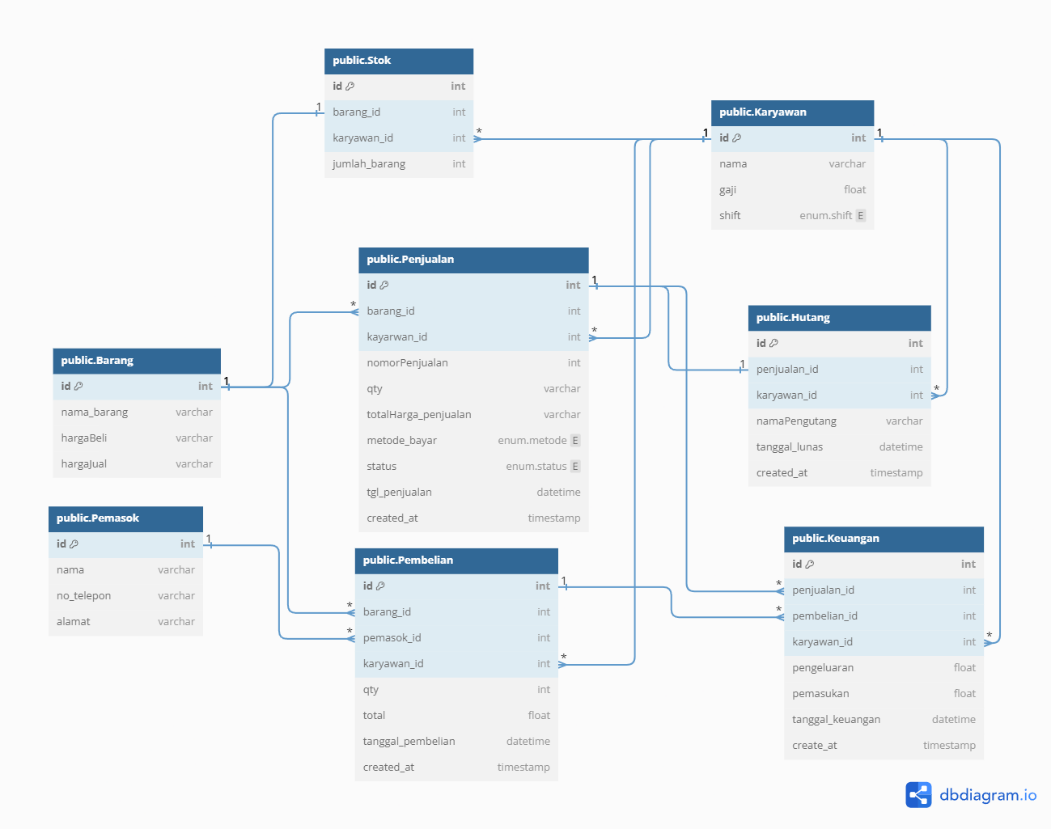
Description automatically generated

Gambar 1. Warung sembako Azmi dan alur proses bisnis

Warung Azmi adalah sebuah warung yang terletak di Jalan Pedati I, Jatinegara. Warung azmi merupakan warung sembako yang menyediakan kebutuhan sehari-hari seperti, sembako dan barang-barang lainnya. Diharapkan dengan mengambil topik ini dapat dilakukan implementasi nyata terhadap aktivitas/kegiatan warung dan catatatnnya yang dimana awalnya masih dilakukan secara manual dapat dijadikan versi digital. Untuk memudahkan dalam melakukan transaksi secara real time dan memiliki catatatan atau riwayat data pada setiap data.

* **Proses pembelian barang:** barang dibeli dari pemasok.
* **Proses penyetokan barang:** barang disimpan dan dilakukan pencatatan terhadap stok barang di warung untuk proses penjualan.
* **Proses penjualan barang:** menjual barang terhadap pembeli/konsumen, jika ada barang terjual maka akan dicatat dan stok akan dikurangi sesuai dengan jumlah barang yang terjual dan transaksi dicatat di buku penjualan atau sistem digital.
* **Proses penjagaan warung:** Selama warung beroperasi, warung akan dijaga oleh beberapa orang secara bergantian.

1. **Entity Relationship Diagram (ERD)**



Gambar 2. Entity Relationship Diagram

*Entity Relationship Diagram* (ERD) diatas adalah gambaran hubungan antar relasi pada tabel yang terdapat pada sistem informasi warung sembako, selain itu ERD ini juga digunakan untuk rancangan tabel dan relasi entitas sebelum membuat database. Terdapat tabel *master* diantaranya adalah tabel barangs, pemasoks, dan karyawans adalah tabel yang menyimpan data *master*, sedang kan tabel relasi seperti stoks, penjualans, pembelians, hutangs, dan keuangans.

Hubungan antar tabel didalam basis data yang dihubungkan melalui *Foreign key*, misalnya didalam tabel penjualans berelasi dengan tabel barangs dan karyawans, sedang kan tabel pembelians berhubungan dengan tabel barangs, pemasoks, dan karyawans. Tabel keuangans ialah tabel yang mencatat seluruh arus kas pada transaksi penjualan dan pembelian.

Dapat dilihat pada bagian entitas yang berwarna biru merupakan posisi atau entitas yang memiliki relasi dengan entitas lain pada tabel yang berbeda, Selain itu juga dapat dilihat pada bagian ujung garis terdapat angka 1 dan simbol \* yang menunjukan jenis relasi yang terjadi antar entitas. Angka 1 melambangkan relasi one dan simbol \* menunjukan relasi many, jadi jika pada tabel master terdapat angka 1 dan tabel relasi terdapat simbol \* berarti menunjukan relasi one to many.

1. **Penjelasan Entitas**
2. **Tabel Master**

Tabel master ialah tabel yang berisi informasi data utama dari seluruh tabel dan biasanya sebagai *primary key* (kunci utama) yang unik.

1. **Tabel Barang**

Digunakan untuk menyimpan informasi mengenai tentang barang-barang yang dijual di warung. Dan dengan atribut:

|  |  |
| --- | --- |
| **Nama** | **Tipe data** |
| id\_barang | Int |
| nama | Varchar |
| Harga\_beli | Varchar |
| Harga\_jual | Varchar |
| create\_at | TIMESTAMP |

1. **Tabel Pemasok:**

Tabel ini digunakan untuk menyimpan informasi mengenai data tentang pemasok barang dengan atribut:

|  |  |
| --- | --- |
| **Nama** | **Tipe data** |
| Id\_pemasok | Int |
| Nama | Int |
| No\_telephone | Int |
| alamat | int |
| create\_at | TIMESTAMP |

1. **Tabel Karyawan:**

Tabel ini digunakan untuk menyimpan informasi mengenai karyawan yang bekerja di warung dengan isi atributnya:

|  |  |
| --- | --- |
| **Nama** | **Tipe data** |
| Id\_karyawan | int |
| Nama | varchar |
| Gaji | float |
| shift | enum.shift |
| create\_at | TIMESTAMP |

1. **Tabel Relasi**

Tabel relasi ialah tabel yang berisi data informasi mengenai yang berhubungan 1 atau 2 dari tabel master. Dan biasanya di hubungkan dengan *foreign key* yang direlasikan dari tabel master.

1. **Tabel Hutang**

Tabel ini digunakan untuk mencatat informasi transaksi hutang dari pelanggan dan dengan atribut:

|  |  |
| --- | --- |
| **Nama** | **Tipe data** |
| id\_hutang | int |
| id\_penjualan | int |
| id\_karyawan | int |
| namaPengutang | varchar |
| tanggal\_lunas | datetime |
| create\_at | TIMESTAMP |

1. **Tabel Stok**

Tabel stok ini digunakan untuk memantau jumlah stok barang. Dan dengan atribut

|  |  |
| --- | --- |
| **nama** | **Tipe data** |
| id\_stok | int |
| id\_barang | int |
| id\_karyawan | int |
| stok\_awal | int |
| stok\_akhir | int |
| create\_at | TIMESTAMP |

1. **Tabel Pembelian**

Tabel pembelian ini digunakan untuk mencatat pembelian barang dari pemasok. Dengan atribut:

|  |  |
| --- | --- |
| **Nama** | **Tipe data** |
| pembelian\_id | int |
| id\_barang | int |
| id\_pemasok | int |
| id\_karyawan | int |
| qty | int |
| total | float |
| tanggal\_pembelian | datetime |
| create\_at | TIMESTAMP |

1. **Tabel Penjualan**

Pada tabel penjualan ini digunakan untuk mencatat penjualan barang pada pelanggan dengan atribut:

|  |  |
| --- | --- |
| **Nama** | **Tipe data** |
| penjualan\_id | int |
| barang\_id | int |
| id\_karyawan | int |
| nomor\_penjualan | int |
| qty | int |
| totalHarga\_penjualan | Varchar |
| metode\_bayar | enum.metode |
| status | enum.status |
| tanggal\_penjualan | datetime |
| create\_at | TIMESTAMP |

1. **Tabel Keuangan**

Pada tabel keuangan ini digunakan untuk mencatat pengeluaran dan pemasukan keuangan pada warung Azmi. Dengan atribut:

|  |  |
| --- | --- |
| **Nama** | **Tipe data** |
| keuangan\_id | int |
| id\_penjualan | int |
| id\_pembelian | int |
| karyawan\_id | int |
| total\_gaji | float |
| pengeluaran | float |
| pemasukan | float |
| tanggal\_keuangan | datetime |
| create\_at | TIMESTAMP |

1. **Antar Tabel**
2. Tabel Barang berkaitan dengan tabel Stok, tabel Pembelian, dan tabel Penjualan melalui ***id\_barang***.
3. Tabel Pemasok berkaitan dengan tabel Pembelian melalui ***id\_pemasok***.
4. Tabel Karyawan berkaitan dengan semua tabel relasi karena karyawan bertugas mencatat seluruh aktivitas (Pembelian, penjualan, Stok, Keuangan, dan Hutang).
5. Tabel Penjualan berkaitan dengan tabel Hutang dan tabel Keuangan melalui ***id\_penjualan***.
6. Tabel Pembelian berkaitan dengan tabel Keuangan melalui ***id\_pembelian***.
7. **Penjelasan Atribut**

* ***id\_barang****,* ***id\_pemasok****,* ***id\_karyawan****,* ***id\_hutang****,* ***id\_stok****,* ***pembelian\_id****,* ***penjualan\_id****,* dan***keuangan\_id***: Id digunakan untuk mengidentifikasi setiap entitas atau setiap transaksi
* ***nama\_barang****,* ***nama****,* ***nama\_pengutang****,* ***nama\_penjualan***: ini merupakan Nama barang, nama Pemasok, dan nama pengutang.
* ***harga\_jual****,* ***harga\_beli, totalHarga\_penjualan, tota*l**, ***pengeluaran****,* ***pemasukan****,* dan ***total\_gaji***: nilai atau total terkait barang, transaksi dan keuangan.
* ***no\_telepone****,* dan***alamat****:* Informasi kontak dan Alamat dari si pemasok.
* ***stok\_awal****,* ***stok\_akhir***dan***qty***: Informasi mengenai jumlah barang yang terkait stok atau transaksi.
* ***tanggal\_lunas****,* ***tanggal\_pembelian****,* ***tanggal\_penjualan***dan***tanggal\_keuangan***: Informasi tanggal yang terkait aktivitas atau transaksi.
* ***gaji****,* ***shift****:* Informasi terkait besaran gaji dan jadwal kerja karyawan.
* ***metode\_bayar***dan***status****:* Informasi terkait metode pembayaran dan status transaksi.

1. **Penjelasan Relasi**
2. **One-to-one**

One-to-one ialah relasi dimana satu baris dari tabel hanya memiliki satu *foreign key*.

* **BarangStok**: Satu barang dapat memiliki 1 penyimpanan data stok.
* **PenjualanHutang**: Setiap satu transaksi penjualan ada dalam satu catatan hutang, tetapi satu catatan hutang dapat terkait dengan satu transaksi penjualan.

1. **One-to-many**

One-to-many ialah relasi dimana satu tabel memiliki satu, dua atau lebih *foreign key*.

* **BarangPembelian**: Setiap barang yang dijual di warung sembako dapat dipesan melalui satu atau lebih transaksi.
* **PemasokPembelian**: Setiap pemasok dapat terlibat dalam satu atau lebih transaksi pembelian yang dilakukan oleh warung.
* **KaryawanPembelian**: Satu karyawan bisa menangani banyak pembelian, tetapi setiap pembelian hanya ditangani oleh satu karyawan.
* **BarangPenjualan**: Satu barang dapat dijual dalam satu atau lebih transaksi penjualan.
* **Karyawan Penjualan**: Satu karyawan bisa menangani banyak pembelian, tetapi setiap pembelian hanya ditangani oleh satu karyawan.
* **Karyawan Hutang**: Setiap hutang terkait oleh satu karyawan, tetapi satu karyawan dapat bertanggung jawab atas banyak perhutangan.
* **Penjualan Hutang**: Satu transaksi penjualana dapat ada dalam satu catatan hutang, tetapi satu catatan hutang dapat terkait dengan 1 transaksi penjualan.
* **Penjualan Keuangan**: Satu catatan keuangan dapat mencangkup banyak transaksi penjualan, tetapi setiap transaksi penjualan hanya terkait dengan 1 catatan keuangan.
* **Pembelian Keuangan**: Satu catatan keuangan dapat mencangkup banyak transaksi pembelian, namun setiap transaksi pembelian hanya terkait dengan satu catatan keuangan.
* **Karyawan Keuangan**: Setiap catatan keuangan dapat terkait pada satu karyawan yang bertanggung jawab, namun satu karyawan dapat memiliki banyak catatan keuangan.

1. **Many-to-many**

Many-to-many ialah relasi dimana satu atribut memiliki hubungan dari satu atau lebih banyak atribut. Relasi many to many biasanya direalisasikan dengan relasi one to many dan tabel penghubung nya yaitu tabel relasi. Contohnya pada topik ini, relasi many to many terjadi pada atribut yang memiliki hubungan dengan atribut yang ada pada tabel pembelian, penjualan, dan keuangan.

1. **Proses Pembuatan Database**

|  |  |
| --- | --- |
| **No** | **Gambar + Query + Penjelasan** |
| 1. Membuat Database | ***CREATE DATABASE warung\_azmi CHARACTER SET utf8mb4 COLLATE utf8mb4\_general\_ci;***  “Membuat database baru dengan nama database warung\_azmi, menggunakan *CHARACTER SET* *utf8mb4* untuk menentukan teks yang disimpan menggunakan format utf-8 untuk semua karakter Unicode, termasuk emoji. Dan *COLLATE utf8mb4\_general\_ci* itu merupakan perbandingan umum yang dimana termasuk lebih cepat tetapi kurang detail dan Ci (*Case-insentive*) pencarian huruf besar dan kecil sama saja.” |
| 1. Membuat Tabel Barangs | ***CREATE TABLE barangs (***  ***id INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,***  ***nama VARCHAR(100) NOT NULL, harga\_beli DOUBLE NOT NULL,***  ***harga\_jual DOUBLE NOT NULL, create\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;***  “Membuat tabel barang pada database dengan atribut id sebagai *PRIMARY KEY* default dan sudah *Auto\_increment,*  nama, harga\_beli dan harga\_jual, dan ada create\_at menggunakan *CURRENT\_TIMESTAMP* dengan sesuai dengan waktu pembuatan database. Menggunakan penyimpanan InnoDB, format utf-8 semua Unicode termasuk emoji, dengan menggunakan perbandingan dan pencarian huruf besar dan kecil sama saja.” |
| 1. Membuat Tabel karyawans | ***CREATE TABLE karyawans(***  ***id INT PRIMARY KEY NOT NULL AUTO\_INCREMENT, nama VARCHAR(100) NOT NULL, gaji DOUBLE NOT NULL, shift ENUM(‘pagi’,’siang’,’malam’), create\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;***  “Membuat tabel karyawan pada database dengan atribut id sebagai *PRIMARY KEY*, nama, gaji, shift, dan create\_at menggunakan CURRENT\_TIMESTAMP sesuai dengan waktu pada pembuatan database. Menggunakan penyimpanan InnoDB, format utf-8 semua Unicode termasuk emoji, dengan menggunakan perbandingan dan pencarian huruf besar dan kecil sama saja.” |
| 1. Membuat Tabel pemasoks | ***CREATE TABLE pemasoks (***  ***id INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,***  ***nama VARCHAR(100) NOT NULL, telpon VARCHAR(50) NOT NULL, alamat TEXT NOT NULL, created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;***  “Membuat tabel pemasoks pada database dengan atribut id sebagai *PRIMARY KEY*, alamat, dan created\_at menggunakan *CURRENT\_TIMESTAMP* yang sesuai dengan pembuatan database. Menggunakan penyimpanan InnoDB, format utf-8 semua Unicode termasuk emoji, dengan menggunakan perbandingan dan pencarian huruf besar dan kecil sama saja.” |
| 1. Membuat Tabel pembelians | ***CREATE TABLE pembelians (***  ***id INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,***  ***barang\_id INT NOT NULL, pemasok\_id INT NOT NULL, karyawan\_id INT NOT NULL, jumlah INT NOT NULL, total\_harga DOUBLE NOT NULL,***  ***tanggal DATE NOT NULL, created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,***  ***FOREIGN KEY (barang\_id) REFERENCES barangs (id) ON DELETE CASCADE,***  ***FOREIGN KEY (pemasok\_id) REFERENCES pemasoks (id) ON DELETE CASCADE,***  ***FOREIGN KEY (karyawan\_id) REFERENCES karyawans (id) ON DELETE CASCADE) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;***  “ Membuat tabel pembelians pada database dengan menggunakan atribut id sebagai *PRIMARY KEY*, jumlah, dan total\_harga, tanggal menggunakan date dan create\_at menggunakan default *CURRENT\_TIMESTAMP*. Adanya *foreign key* pada barang\_id, pemasok\_id, dan karyawan\_id dengan menunjuk pada id dan *DELETE CASCADE* penghapusan data otomatis. Menggunakan penyimpanan InnoDB, format utf-8 semua Unicode termasuk emoji, dengan menggunakan perbandingan dan pencarian huruf besar dan kecil sama saja.” |
| 1. Membuat tabel stoks | ***CREATE TABLE stoks (***  ***id INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,***  ***barang\_id INT NOT NULL, karyawan\_id INT NOT NULL, jumlah\_barang INT NOT NULL DEFAULT 0, created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,***  ***FOREIGN KEY (barang\_id) REFERENCES barangs (id) ON DELETE CASCADE,***  ***FOREIGN KEY (karyawan\_id) REFERENCES karyawans (id) ON DELETE CASCADE) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4***  ***COLLATE=utf8mb4\_general\_ci;***  “ Membuat tabel stocks pada database dengan atribut id memakai int dan *AUTO\_INCREMENT* sebagai *PRIMARY KEY*, jumlah\_barang int, dan created\_at dengan sesuai pada waktu pembuatan database. Adanya *foreign key* pada barang\_id dan karyawan\_id dengan menunjuk pada id dan *DELETE CASCADE* sebagai penghapusan otomatis. Menggunakan penyimpanan InnoDB, format utf-8 semua Unicode termasuk emoji, dengan menggunakan perbandingan dan pencarian huruf besar dan kecil sama saja.” |
| 1. Membuat tabel penjualans | ***CREATE TABLE penjualans (***  ***id INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,***  ***barang\_id INT NOT NULL, karyawan\_id INT NOT NULL,***  ***nomor\_penjualan INT NOT NULL, jumlah INT NOT NULL, total\_harga DOUBLE NOT NULL, status ENUM('lunas', 'hutang') NOT NULL, metode\_bayar ENUM('cash', 'qris') DEFAULT NULL, tanggal DATE NOT NULL, created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,***  ***FOREIGN KEY (barang\_id) REFERENCES barangs (id) ON DELETE CASCADE,***  ***FOREIGN KEY (karyawan\_id) REFERENCES karyawans (id) ON DELETE CASCADE) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4=COLLATE=utf8mb4\_general\_ci;***  “ Membuat tabel pada penjualans dengan atribut id sebagai *PRIMARY KEY*, nomor\_penjualan dan jumlah, total\_harga, status dan metode\_bayar, tanggal Date default, dan created\_at memakai *CURRENT\_TIMESTAMP* dengan sesuai pada waktu pembuatan database. Adanya *foreign key* pada barang\_id dan karyawan\_id dengan menunjuk pada id dan *DELETE CASCADE* dengan penghapusan otomatis. Menggunakan penyimpanan InnoDB, format utf-8 semua Unicode termasuk emoji, dengan menggunakan perbandingan dan pencarian huruf besar dan kecil sama saja.” |
| 1. Membuat tabel hutangs | ***CREATE TABLE hutangs (***  ***id INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,***  ***penjualan\_id INT NOT NULL, karyawan\_id INT NOT NULL, nama\_penghutang VARCHAR(100) NOT NULL, tanggal\_lunas DATE NOT NULL, created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,***  ***FOREIGN KEY (penjualan\_id) REFERENCES penjualans (id) ON DELETE CASCADE,***  ***FOREIGN KEY (karyawan\_id) REFERENCES karyawans (id) ON DELETE CASCADE) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4***  ***COLLATE=utf8mb4\_general\_ci;***  “ Membuat tabel hutangs dengan atribut id sebagai *PRIMARY KEY*, nama\_penghutang, tanggal\_lunas date default, dan created\_at memakai *CURRENT\_TIMESTAMP* dengan sesuai pada waktu pembuatan database. Adanya *foreign key* pada penjualan\_id dan karyawan\_id dengan menunjuk pada id dan *DELETE CASCADE* sebagai penghapusan otomatis. Menggunakan penyimpanan InnoDB, format utf-8 semua Unicode termasuk emoji, dengan menggunakan perbandingan dan pencarian huruf besar dan kecil sama saja.” |
| 1. Membuat tabel keuangans | ***CREATE TABLE keuangans (***  ***id INT PRIMARY KEY NOT NULL AUTO\_INCREMENT, pembelian\_id INT NOT NULL, penjualan\_id INT NOT NULL, karyawan\_id INT NOT NULL, pengeluaran DOUBLE NOT NULL DEFAULT 0, pemasukan DOUBLE NOT NULL DEFAULT 0, tanggal DATE NOT NULL, created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,***  ***FOREIGN KEY (pembelian\_id) REFERENCES pembelians (id) ON DELETE CASCADE,***  ***FOREIGN KEY (penjualan\_id) REFERENCES penjualans (id) ON DELETE CASCADE,***  ***FOREIGN KEY (karyawan\_id) REFERENCES karyawans (id) ON DELETE CASCADE) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4***  ***COLLATE=utf8mb4\_general\_ci;***  “ Membuat tabel keuangans dengan atribut id sebagai *PRIMARY KEY*, pengeluaran dan pemasukan memakai *double default*, tanggal memakai date dan created\_at memakai *CURRENT\_TIMESTAMP* dengan otomatis waktu pembuatan database. Adanya *foreign key* pada pembelian\_id, penjualan\_id, dan karyawan\_id dengan menunjuk pada id dan *DELETE CASCADE* sebagai penghapusan otomatis. Menggunakan penyimpanan InnoDB, format utf-8 semua Unicode termasuk emoji, dengan menggunakan perbandingan dan pencarian huruf besar dan kecil sama saja.” |
| 1. Mengisi field tabel barangs | ***INSERT INTO barangs (nama, harga\_beli, harga\_jual) VALUES ('Beras Premium', 10000, 12000), ('Minyak Goreng', 14000, 16000), ('Gula Pasir', 12000, 15000), ('Tepung Terigu', 9000, 12000), ('Kopi Instan', 2500, 3000), ('Susu Bubuk', 45000, 55000), ('Mie Instan', 2500, 3000), ('Cokelat Batang', 15000, 18000), ('Keju Parut', 35000, 45000), ('Sabun Mandi', 7000, 8500), ('Shampo Sachet', 2000, 2500), ('Tisu Gulung', 12000, 15000), ('Air Mineral Galon', 15000, 20000), ('Roti Tawar', 12000, 14000), ('Sosis Ayam', 40000, 50000), ('Kentang Goreng', 25000, 32000), ('Telur Ayam', 20000, 25000), ('Kecap Manis', 12000, 15000), ('Saos Sambal', 10000, 12000), ('Biskuit Kaleng', 40000, 50000);***  “ Mengisi field pada tabel barangs dengan sesuai pada isi struktur pada tabel barangs yaitu ada ide, nama, harga\_beli, dan harga\_jual. Karena menggunakan *AUTO\_INCREMENT* tidak ada struktur id\_barang pada query diatas, MySql sudah otomatis penomoran”. |
| 1. Mengisi field tabel pembelians | ***INSERT INTO pembelians (barang\_id, pemasok\_id, karyawan\_id, jumlah, total\_harga, tanggal) VALUES***  ***(1, 1, 1, 5, 500000, '2024-12-01'), (2, 2, 2, 3, 300000, '2024-12-02'), (3, 3, 3, 7, 700000, '2024-12-03'), (4, 4, 4, 2, 200000, '2024-12-04'), (5, 5, 5, 10, 1000000, '2024-12-05'), (6, 6, 6, 6, 600000, '2024-12-06'), (7, 7, 7, 8, 800000, '2024-12-07'), (8, 8, 8, 4, 400000, '2024-12-08'), (9, 9, 9, 12, 1200000, '2024-12-09'), (10, 10, 10, 9, 900000, '2024-12-10'), (11, 1, 1, 3, 300000, '2024-12-11'), (12, 2, 2, 5, 500000, '2024-12-12'), (13, 3, 3, 6, 600000, '2024-12-13'), (14, 4, 4, 7, 700000, '2024-12-14'), (15, 5, 5, 2, 200000, '2024-12-15'), (16, 6, 6, 8, 800000, '2024-12-16'), (17, 7, 7, 4, 400000, '2024-12-17'), (18, 8, 8, 9, 900000, '2024-12-18'), (19, 9, 9, 6, 600000, '2024-12-19'),***  ***(20, 10, 10, 10, 1000000, '2024-12-20');***  “ Mengisi pada tabel pembelians dengan value dengan **id\_barang**, **id\_pemasok**, **id\_karyawan**, **jumlah**,**total\_harga** dan **tanggal**. Karena adanya *Foreign key* pada barang\_id, karyawan\_id, pemasok\_id jadi hanya menyebutkan id nya saja tidak perlu menginput nama nya lagi”. |
| 1. Mengisi field tabel pemasoks | ***INSERT INTO pemasoks (nama, telpon, alamat) VALUES***  ***('Ali', '081234567890', 'Jl. Merdeka No. 10, Jakarta'), ('Budi', '082345678901', 'Jl. Pahlawan No. 5, Bandung'), ('Citra', '083456789012', 'Jl. Raya No. 20, Surabaya'), ('Dani', '084567890123', 'Jl. Suka No. 30, Yogyakarta'), ('Eka', '085678901234', 'Jl. Cendana No. 15, Semarang'),***  ***('Fikri', '086789012345', 'Jl. Anggrek No. 25, Malang'), ('Gina', '087890123456', 'Jl. Mawar No. 35, Medan'), ('Hadi', '088901234567', 'Jl. Kembang No. 40, Makassar'), ('Indah', '089012345678', 'Jl. Sumber No. 45, Bali'), ('Joko', '081112233445', 'Jl. Senopati No. 50, Jakarta'), ('Kiki', '082223344556', 'Jl. Gunung No. 60, Bandung'),***  ***('Lina', '083334455667', 'Jl. Laut No. 70, Surabaya'), ('Mira', '084445566778', 'Jl. Teluk No. 80, Yogyakarta'), ('Nina', '085556677889', 'Jl. Jaya No. 90, Semarang'), ('Omar', '086667788990', 'Jl. Hijau No. 100, Malang'), ('Putri', '087778899001', 'Jl. Taman No. 110, Medan'),***  ***('Rudi', '088889900112', 'Jl. Rimba No. 120, Makassar'), ('Siti', '089990011223', 'Jl. Pelita No. 130, Bali'), ('Tono', '081223344556', 'Jl. Sejahtera No. 140, Jakarta'), ('Umar', '082334455667', 'Jl. Persada No. 150, Bandung');***  “ Mengisi isi field pada tabel pemasoks dengan value yaitu nama, telpon, dan alamat. Karena menggunakan *AUTO\_INCREMENT* jadi id\_pemasok tidak dimasukkan secara manual karena jika sudah ada Auto\_Increment maka sudah jdi otomatis penomorannya”. |
| 1. Mengisi field tabel karyawans | ***INSERT INTO karyawans (nama, gaji, shift) VALUES***  ***('Andi', 3000000, 'pagi'), ('Budi', 3000000, 'siang'), ('Citra', 3000000, 'malam'), ('Dani', 3000000, 'pagi'), ('Eka', 3000000, 'siang'), ('Fikri', 3000000, 'malam'), ('Gina', 3000000, 'pagi'), ('Hadi', 3000000, 'siang'), ('Indah', 3000000, 'malam'), ('Joko', 3000000, 'pagi'), ('Kiki', 3000000, 'siang'), ('Lina', 3000000, 'malam'), ('Mira', 3000000, 'pagi'), ('Nina', 3000000, 'siang'), ('Omar', 3000000, 'malam'), ('Putri', 3000000, 'pagi'), ('Rudi', 3000000, 'siang'), ('Siti', 3000000, 'malam'), ('Tono', 3000000, 'pagi'), ('Umar', 3000000, 'siang');***  “ Mengisi field pada tabel karyawans dengan value nama, gaji, dan shift saja. Tidak ada id\_karyawan karena sudah otomatis jika sudah menggunakan *AUTO\_INCREMENT*”. |
| 1. Mengisi field tabel stoks | ***INSERT INTO stoks (barang\_id, karyawan\_id, jumlah\_barang) VALUES***  ***(1, 1, 100), (2, 2, 200), (3, 3, 150), (4, 4, 50), (5, 5, 300), (6, 6, 120), (7, 7, 250),(8, 8, 180), (9, 9, 90), (10, 10, 210), (11, 11, 130), (12, 12, 75), (13, 13, 160),(14, 14, 240), (15, 15, 110), (16, 16, 190), (17, 17, 220), (18, 18, 80), (19, 19, 140), (20, 20, 160);***  “ Mengisi field pada tabel stoks dengan value barang\_id, karyawan\_id, dan jumlah\_barang. Tidak ada id\_stok karena sudah menggunakan *AUTO\_INCREMENT* pada MySql dan sudah otomatis terupdate pada id\_stok nya. Adanya *Foreign key* pada barang\_id menunjuk pada id\_barang dan karyawan\_id menunjuk pada id masing-masing”. |
| 1. Mengisi field tabel penjualans | ***INSERT INTO penjualans (barang\_id, karyawan\_id, nomor\_penjualan, jumlah, total\_harga, status, metode\_bayar, tanggal)VALUES***  ***(1, 1, 1001, 5, 500000.00, 'lunas', 'cash', '2024-12-22'), (2, 2, 1002, 10, 1000000.00, 'hutang', NULL, '2024-12-23'), (3, 3, 1003, 3, 300000.00, 'lunas', 'cash', '2024-12-24'), (4, 4, 1004, 8, 800000.00, 'hutang', NULL, '2024-12-25'), (5, 5, 1005, 2, 200000.00, 'lunas', 'qris', '2024-12-26'), (6, 6, 1006, 4, 400000.00, 'hutang', NULL, '2024-12-27'), (7, 7, 1007, 6, 600000.00, 'lunas', 'cash', '2024-12-28'), (8, 8, 1008, 9, 900000.00, 'hutang', NULL, '2024-12-29'),***  ***(9, 9, 1009, 1, 100000.00, 'lunas', 'cash', '2024-12-30'), (10, 10, 1010, 7, 700000.00, 'hutang', NULL, '2024-12-31'), (11, 11, 1011, 5, 500000.00, 'lunas', 'qris', '2025-01-01'), (12, 12, 1012, 10, 1000000.00, 'hutang', NULL, '2025-01-02'), (13, 13, 1013, 3, 300000.00, 'lunas', 'cash', '2025-01-03'), (14, 14, 1014, 8, 800000.00, 'hutang', NULL, '2025-01-04'), (15, 15, 1015, 2, 200000.00, 'lunas', 'cash', '2025-01-05'),***  ***(16, 16, 1016, 4, 400000.00, 'hutang', NULL, '2025-01-06'), (17, 17, 1017, 6, 600000.00, 'lunas', 'qris', '2025-01-07'), (18, 18, 1018, 9, 900000.00, 'hutang', NULL, '2025-01-08'), (19, 19, 1019, 1, 100000.00, 'lunas', 'qris', '2025-01-09'), (20, 20, 1020, 7, 700000.00, 'hutang', NULL, '2025-01-10');***  “ Mengisi field pada tabel penjualans dengan value barang\_id, karyawan\_id, nomor\_penjualan, jumlah, total\_harga, status, metode\_bayar, dan tanggal. Dengan adanya *foreign key*  pada barang\_id dan karyawan\_id jadi penyebutan hanya menggunakan id tidak menginput nama lagi”. |
| 1. Mengisi field tabel hutangs | ***INSERT INTO hutangs (penjualan\_id, karyawan\_id, nama\_penghutang, tanggal\_lunas) VALUES***  ***(2, 2, 'Ahmad Zaki', '2025-12-22'), (4, 4, 'Budi Santoso', '2025-12-23'), (6, 6, 'Citra Dewi', '2025-12-24'), (8, 8, 'Dewi Susanti', '2025-12-25'),(10, 10, 'Eko Wijaya', '2025-12-26'),(12, 12, 'Faisal Rahman', '2025-12-27'),(14, 14, 'Gina Kurniawati', '2025-12-28'),(16, 16, 'Hendra Setiawan', '2025-12-29'),(18, 18, 'Ika Lestari', '2025-12-30'),(20, 20, 'Joko Prabowo', '2025-12-31'),(1, 1, 'Kiki Setyani', '2025-01-01'),(3, 3, 'Lina Maharani', '2025-01-02'),(5, 5, 'Maya Sari', '2025-01-03'),(7, 7, 'Nina Salim', '2025-01-04'),(9, 9, 'Oka Laksana', '2025-01-05'),***  ***(11, 11, 'Putu Suryani', '2025-01-06'),(13, 13, 'Qori Zainuddin', '2025-01-07'),(15, 15, 'Rina Anggraeni', '2025-01-08'),(17, 17, 'Satria Gunawan', '2025-01-09'),(19, 19, 'Teguh Priyono', '2025-01-10');***  “ Mengisi field pada tabel hutangs dengan value penjualan\_id, karyawan\_id, nama\_penghutang, dan tanggal\_lunas. Adanya *foreign key* pada penjualan\_id dan karyawan\_id jadi penyebutan hanya menggunakan id tidak menggunakan nama”. |
| 1. Mengisi field tabel keuangans | ***INSERT INTO keuangans (pembelian\_id, penjualan\_id, karyawan\_id, pengeluaran, pemasukan, tanggal) VALUES***  ***(1, 1, 1, 300000.00, 500000.00, '2024-12-01'),***  ***(2, 2, 2, 500000.00, 700000.00, '2024-12-02'),***  ***(3, 3, 3, 200000.00, 400000.00, '2024-12-03'),***  ***(4, 4, 4, 400000.00, 600000.00, '2024-12-04'),***  ***(5, 5, 5, 250000.00, 550000.00, '2024-12-05'),***  ***(6, 6, 6, 300000.00, 700000.00, '2024-12-06'),***  ***(7, 7, 7, 500000.00, 1000000.00, '2024-12-07'),***  ***(8, 8, 8, 200000.00, 450000.00, '2024-12-08'),***  ***(9, 9, 9, 400000.00, 800000.00, '2024-12-09'),***  ***(10, 10, 10, 300000.00, 650000.00, '2024-12-10'),***  ***(11, 11, 11, 350000.00, 750000.00, '2024-12-11'),***  ***(12, 12, 12, 450000.00, 850000.00, '2024-12-12'),***  ***(13, 13, 13, 500000.00, 1000000.00, '2024-12-13'),***  ***(14, 14, 14, 250000.00, 550000.00, '2024-12-14'),***  ***(15, 15, 15, 300000.00, 600000.00, '2024-12-15'),***  ***(16, 16, 16, 400000.00, 700000.00, '2024-12-16'),***  ***(17, 17, 17, 350000.00, 750000.00, '2024-12-17'),***  ***(18, 18, 18, 200000.00, 500000.00, '2024-12-18'),***  ***(19, 19, 19, 300000.00, 600000.00, '2024-12-19'),***  ***(20, 20, 20, 500000.00, 800000.00, '2024-12-20');***  “ Mengisi field pada tabel keuangans dengan value pembelian\_id, penjualan\_id, karyawan\_id, pengeluaran, pemasukan,dan tanggal. Adanya *foreign key* pada pembelian\_id, penjualan\_id, dan karyawan\_id dengan menyebutkan menggunakan id dan tidak menggunakan nama. Dan sudah menggunakan AUTO\_INCREMENT, jadi sudah tidak memasukkan ide secara manual karena sudah secara otomatis. |
| 1. *Select*, *Update,* dan *Delete* dari tabel barangs. | 1. ***Select***       ***SELECT \* FROM barangs***  “Melihat seluruh isi field pada tabel barags”   1. ***Update***       **UPDATE barangs SET nama = 'Indomie kari ayam', harga\_beli = 2450, harga\_jual = 3500 WHERE id = 1;**  “ Mengupdate isi field pada tabel barangs dengan mengubah nama dan harga\_jual Dimana yang diubah adalah id\_barang ke-1”.   1. ***Delete***       **DELETE FROM barangs WHERE id = 20;**  “Menghapus isi field pada tabel barangs Dimana id\_barang ke-20”. |
| 1. *Select, Update,* dan *Delete* dari tabel karyawans | 1. ***Select***       ***SELECT \* FROM karyawans;***  “Melihat seluruh isi field pada tabel karyawans”.   1. ***Update***       ***UPDATE karyawans SET nama = ‘Roesdi Imoet’,  shift = ‘malam’ WHERE id = 1;***  “Mengupdate isi field pada tabel karyawan yaitu nama dan shift nya Dimana yang diubah id\_karyawan ke-1”.   1. ***Delete***       ***DELETE FROM karyawans WHERE id = 20;***  “Menghapus isi field pada id\_karyawan ke-20”. |
| 1. *Select, Update,* dan *Delete* dari tabel pemasoks | 1. ***Select***       ***SELECT\* FROM pemasoks;***  “ Melihat seluruh isi field pada tabel pemasoks”   1. ***Update***       ***UPDATE pemasoks SET nama = "PT Sinar Gurih", telpon = "021222333" WHERE id = 1;***  “ Mengupdate tabel pemasoks pada id ke-1 dengan mengubah nama dan no\_telephone”.   1. ***Delete***       ***DELETE FROM pemasoks WHERE id = 1;***  “ Menghapus dari tabel pemasoks Dimana yang dihapus adalah id ke-1”. |
| 1. *Select, Update,* dan *Delete* dari tabel pembelians | 1. ***Select***       ***SELECT \* FROM pembelians;***  “ Melihat seluruh isi tabel dari pembelians”.   1. ***Update***       ***UPDATE pembelians SET barang\_id = 1, pemasok\_id = 3 WHERE id = 2;***  “ Mengupdate pada tabel pembelians dengan mengubah barang\_id dan pemasok\_id Dimana yang diuabh yaitu id pembelians ke-2”.   1. ***Delete***       ***DELETE FROM pembelians WHERE id = 19;***  “Meghapus data dari tabel pembelians Dimana yang dihapus yaitu data yang ke -19”. |
| 1. *Select, Update,* dan *Delete* dari tabel stoks | 1. ***Select***       ***SELECT \* FROM stoks;***  **“** Melihat seluruh isi tabel stoks"   1. ***Update***       ***UPDATE stoks SET karyawan\_id = 19 WHERE id = 1;***  “ Mengupdate data dari tabel stoks yaitu karyawan\_id dimana yang diubah yaitu data ke-1”.   1. ***Delete***       ***DELETE FROM stoks WHERE id = 15;***  “ Menghapus data dari tabel stoks yaitu data ke-15”. |
| 1. *Select, Update,* dan *Delete* dari tabel penjualans | 1. ***Select***       ***SELECT \* FROM penjualans;***  “ Melihat seluruh isi pada tabel penjualans”   1. ***Update***       ***UPDATE penjualans SET karyawan\_id = 1, status = "lunas" WHERE id = 2;***  “Mengupdate data pada tabel penjualans yaitu karyawan\_id dan status Dimana yang diubah yaitu data id ke -2”.   1. ***Delete***       ***DELETE FROM penjualans WHERE id = 2***  “ Menghapus data dari tabel penjualans yaitu data ide ke-2”. |
| 1. *Select, Update,* dan *Delete* dari tabel hutangs | 1. ***Select***       ***SELECT \* FROM hutangs;***  “ Melihat seluruh isi pada tabel hutangs”.   1. ***Update***       ***UPDATE hutangs SET karyawan\_id = 1, nama\_penghutang = 'Jin Hidayah Wahid' WHERE id = 4;***  “ Mengupdate pada tabel hutangs yaitu karyawan\_id, nama\_penghutang pada data id ke-4”.   1. ***Delete***       ***DELETE FROM hutangs WHERE id = 19;***  “ Menghapus data dari tabel hutangs yaitu ide ke-19”. |
| 1. *Select, Update,* dan *Delete* dari tabel keuangans | 1. ***Select***       ***SELECT \* FROM keuangans;***  “ Melihat isi seluruh tabel keuangans”   1. ***Update***       ***UPDATE keuangans SET karyawan\_id = 16, pemasukan = 1000000, pengeluaran = 90000000 WHERE id = 10;***  “ Mengupdate data tabel keungans yaitu karyawan\_id, pemasukan dan pengeluaran yaitu data ke-10”.   1. ***Delete***       ***DELETE FROM keuangans WHERE id = 10;***  “ Menghapus data tabel keuangans yaitu data id ke-10”. |

1. **Transaksi Menggunakan metode ACID**

Transaksi ialah sekumpulan perintah SQL yang dieksekusi secara bersamaan untuk memastikan integritas nya data. **ACID** adalah singkatan dari *Atomicityy, Consistency, isolation,* dan *Durability*. **Atomicity**, *yaitu ketunggaan dimana transaksi dianggap satu unit, jika satu operasi gagal maka transaksi batal menggunakan perintah* ***ROLLBACK***. **Consistency***, yaitu konsistensi. Dimana transaksi dipastikan data tetap konsisten*. **Isolation***, yaitu pengisolasian. Dimana transaksi bejalan secara independent, tidak dipengaruhi oleh trnasaksi lainnya.* **Durability***, yaitu perubahan data menjadi permanen dengan perintah melakukan* ***COMMIT***. Adanya metode **ACID** ini dapat meningkatkan keamanan data dan menghindari kesalahan. Memungkinkan user menambahkan, mengecek, dan mengupdate database secara bersamaan. Jika pada pengecekan barang kurang maka bisa dibatalkan transaksi nya menggunakan **ROLLBACK** tetapi jika ada dapat dilanjutkan transaksinya dengan mengupdate data nya dan menyimpan data dengan dilakukan **COMMIT**.

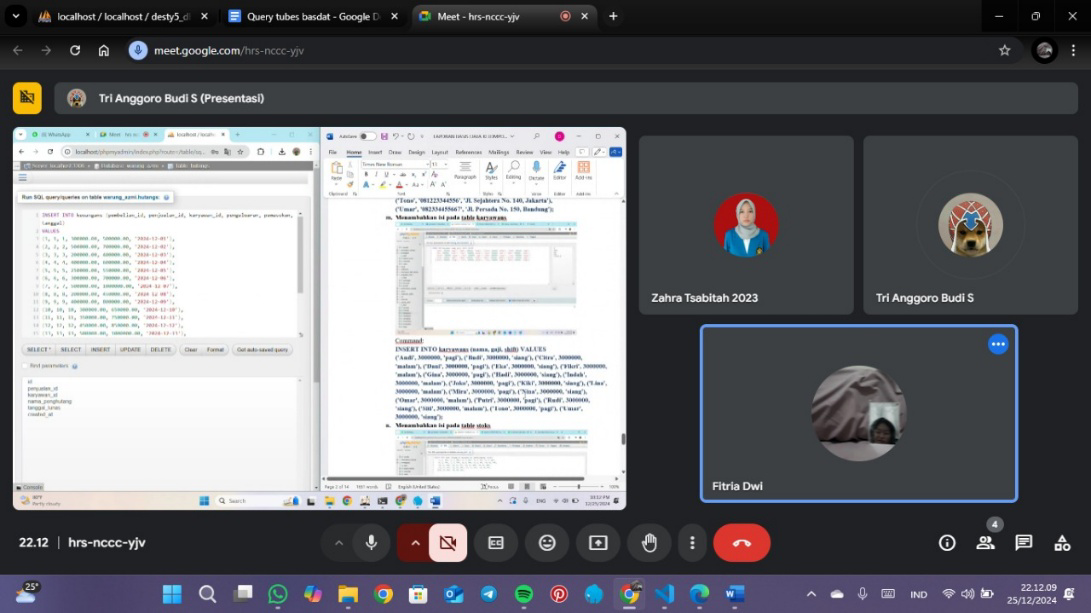
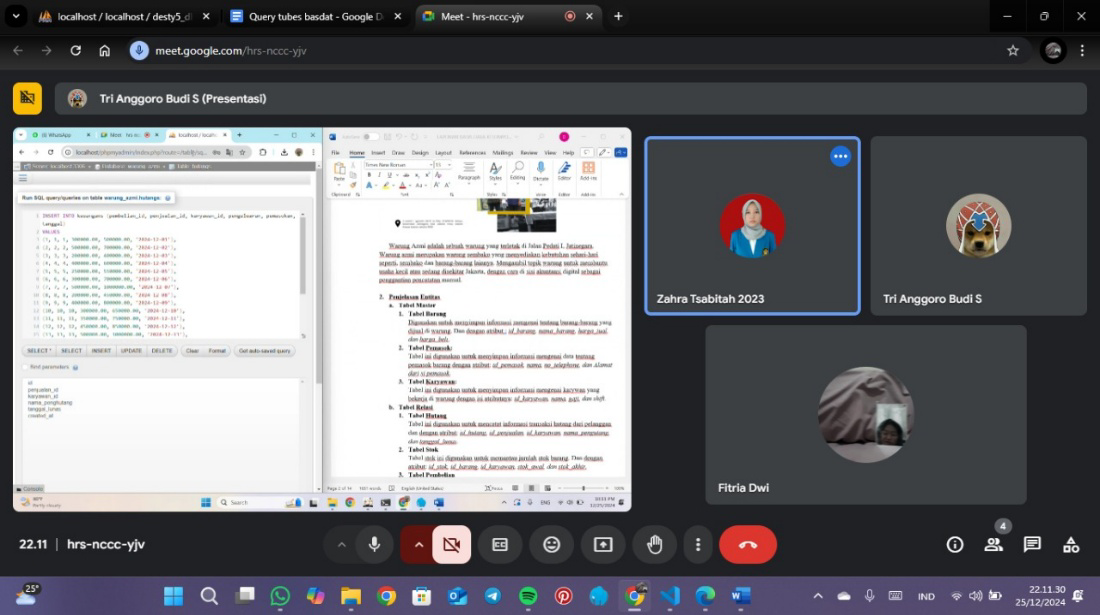
|  |  |
| --- | --- |
| **No** | **Gambar + Query + Penjelasan** |
| 1. Membuat Transaksi | ***START TRANSACTION;***  ***INSERT INTO pembelians (id, barang\_id, pemasok\_id, karyawan\_id, jumlah, total\_harga, tanggal) VALUES (1, 1, 4, 10, 20, 200000, '2024-12-29');***  ***SELECT jumlah\_barang,***  ***IF ( jumlah\_barang = 0, 'Kosong',***  ***IF ( jumlah\_barang < 50, 'Low',***  ***IF ( jumlah\_barang < 100, 'Mid', 'Over' ))) AS Kategori***  ***FROM stoks WHERE barang\_id = 1;***  ***UPDATE stoks SET jumlah\_barang = jumlah\_barang + 20 WHERE barang\_id = 1;***  ***SELECT \* FROM stoks WHERE barang\_id = 2;***  ***COMMIT;***  “ Memulai transaksi dengan metode ACID, dengan menambahkan pada tabel pembelians dengan query insert, dan memeriksa dengan query select atribut **jumlah\_barang** pada tabel stoks dengan memakai *foreign key* **barang\_id** ke-2. Maka dilanjutkan dengan Update tabel stoks melalui atribut **jumlah\_barang** dan dilakukan lagi pengecheckan dengan *SELECT* tabel stoks dengan *foreign key* **barang\_id** ke-1. Dan di lakukan *COMMIT* untuk menyimpan data perubahan secara permanen”. |
| 1. Membuat transaksi batal | ***START TRANSAKCTION;***  ***INSERT INTO pembelians (id, barang\_id, pemasok\_id, karyawan\_id, jumlah, total\_harga, tanggal) VALUES (2, 2, 4, 10, 20, 200000, ‘2025-01-12’);***  ***SELECT \* FROM pembelians;***  ***SELECT jumlah\_barang,***  ***IF (jumlah\_barang = 0, ‘kosong’,***  ***IF (jumlah\_barang < 50, ‘Low’,***  ***IF (jumlah\_barang < 100, ‘Mid’, ‘Over’ ))) AS Kategori FROM***  ***Stoks WHERE barang\_id = 1;***  ***ROLLBACK;***  “Memulai transaksi dengan metode ACID, dengan menambahkan pada tabel pembelians dengan query insert, dan memeriksa dengan query select atribut **jumlah\_barang** pada tabel stoks dengan memakai *foreign key* **barang\_id** ke-1. Dari gambar diatas bahawa jumlah\_barang = **Over** maka dilakukan perintah **ROLLBACK**. Yaitu dibatalkan transaksi nya”. |

1. **User Management dengan privileges**

Konsep pada **Data Control Language (DCL)** digunakan unyuk manajemen pengguna. Konsep ini memungkinkan pemberian akses kepada setiap pengguna dengan perintah **Grant**, yang digunakan untuk memberikan hak akses kepada pengguna database, serta **Revoke**, yang digunakan untuk mencabut hak akses yang sebelumnya telah diberikan. Penerapan konsep ini yaitu menyederhanakan pengelolaan database saat bekerja dalam tim, sehingga setiap anggota hanya dapat melakukan Tindakan sesuai dengan hak akses yang diberikan. Akses ini dapat dikontrol lebih ketat dengan menggunakan Alamat IP dari setiap perangkat pengguna. Akan tetapi ini untuk saat ini, akan dicontohkan melalui IP local.

|  |  |
| --- | --- |
| **No** | **Gambar + Query + Penjelasan** |
| 1. Membuat user *management* dengan *Privileged* | 1. **User “Budi”**     **CREATE USER 'budi'@'localhost' IDENTIFIED BY 'budi123';**  “ Membuat User ‘budi’ sebagai localhost dengan menggunakan nama ‘budi123’ “.   1. **User “Fitri”**     **CREATE USER 'fitri'@'localhost' IDENTIFIED BY 'fitri123';**  “ Membuat User ‘fitri’ yang terhubung localhost menggunakan nama User ‘fitri123’ “.   1. **User “Zahra”**     **CREATE USER 'zahra'@'localhost' IDENTIFIED BY 'zahra123';**  “ Membuat User ‘zahra’ sebagai localhost menggunakan nama identitas user ‘zahra123’ “. |
| 1. Memberikan hak akses user | 1. **User ‘budi’**     ***GRANT INSERT ON warung\_azmi.\* TO 'budi'@'localhost';***  “ Memberikan akses menginput record seluruh tabel pada database warung\_azmi“.   1. **User ‘fitri’**     ***GRANT ALL PRIVILEGES ON warung\_azmi.\* TO 'fitri'@'localhost';***  “ Memberikan seluruh akses di seluruh tabel pada database warung\_azmi “.   1. **User ‘zahra’**     ***GRANT SELECT ON warung\_azmi.\* TO 'zahra'@'localhost';***  “ Memberikan akses menampilkan seluruh tabel pada database warung\_azmi”. |
| 1. Melihat akses user | 1. **User ‘budi’**       ***SHOW GRANTS FOR 'budi'@'localhost';***  “Menampilkan akses pada user ‘budi’ ”.   1. **User ‘fitri’**       ***SHOW GRANTS FOR 'fitri'@'localhost';***  “ Menampilkan akses pada user ‘fitri’ ”.   1. **User ‘zahra’**       ***SHOW GRANTS FOR 'zahra'@'localhost';***  “ Menampilkan akses pada user ‘zahra’ |
| 1. Mencob hak akses user | 1. **User ‘budi’**     “ Mencoba login dengan user ‘budi’ “.    “ Dari gambar diatas akses tidak bisa karena user ‘budi’ hanya bisa menginput record data ke seluruh tabel di database warung\_azmi”.      “ Dari gambar diatas user ‘budi’ berhasil untuk menampilkan data seluruh tabel di database.”   1. **User ‘fitri’**           “ Dari gambar diatas pada user ‘fitri’ diberikan seluruh untuk dapat menginput, memanipulasi dan menampilkan data di seluruh tabel pada database warung\_azmi”.   1. **User ‘zahra’**         “ Dari gambar diatas berhasil user ‘zahra’ mengakses untuk menampilkan data dari seluruh tabel di database”.    “ Dari gambar diatas user ‘zahra’ gagal untuk menginput record karena user hanya diberikan akses menampilkan data tidak untuk menginput record di database”. |

1. **Progress kerja kelompok**

****

Gambar 3. Bukti pengerjaan tugas Bersama secara online melalui aplikasi Gmeet

Kami melakukan kerja kelompok dari membuat database hingga laporan melalui Gmeet/online secara berkala. Gambar diatas adalah bukti progress atau perkembangan kami. Bukti pengerjaan lainnya kami simpan pada repository github, dapat diakses melalui link dibawah ini.

**Link Github:** [**https://github.com/zahrtsa/warung\_azmi**](https://github.com/zahrtsa/warung_azmi)