|  |  |
| --- | --- |
| **NAMA**  **NIM**  **KELAS**  **MATKUL** | **: HENDRA USMAN**  **: D0221079**  **: F – INFORMATIKA**  **: BASIS DATA I** |

**SOAL:**

1. Apa perbedaan Normalisasi data dan Redudansi data? Jelaskan dan berikan contohnya.
2. Apa yang dimaksud Basisdata SQL dan NonSQL? Contoh aplikasinya apa?
3. Buatlah sketsa basis data tentang Penjualan di draw.io atau yang sejenis, copy gambarnya dan kirimkan.
4. Aplikasikan Basisdata pada no 3, seperti ditugas kemarin (tulis scriptnya).

**JAWAB:**

1. Apa perbedaan Normalisasi data dan Redudansi data? Jelaskan dan berikan contohnya.

Penjelasan dan Contoh:

* ***Normalisasi data*** adalah proses mengatur struktur tabel database agar sesuai dengan prinsip-prinsip normalisasi, yang bertujuan untuk menghindari redundansi data dan memastikan integritas data. Normalisasi data terbagi menjadi beberapa tingkatan, mulai dari normal form pertama sampai kelima. Semakin tinggi tingkat normalisasi, semakin ternormalisasi tabel tersebut, tetapi juga semakin sulit untuk melakukan query dan mengambil data dari tabel tersebut.

Berikut ini adalah contoh tabel yang telah ternormalisasi sampai dengan tingkat ketiga (Third Normal Form, 3NF):

Tabel Pelanggan:

| Kode Pelanggan | Nama Pelanggan |
| --- | --- |
| P001 | Hendra Usman |
| P002 | Fivin Arianti |
| P003 | Hendri Ali |

Tabel Order:

| Kode Order | Kode Pelanggan | Tanggal Order |
| --- | --- | --- |
| O001 | P001 | 2022-12-28 |
| O002 | P001 | 2022-12-27 |
| O003 | P002 | 2022-02-28 |

Tabel Order Detail:

| Kode Order | Kode Produk | Jumlah | Harga |
| --- | --- | --- | --- |
| O001 | P01 | 2 | 100 |
| O001 | P02 | 1 | 50 |
| O002 | P03 | 3 | 200 |
| O003 | P01 | 5 | 100 |

Dalam contoh tabel di atas, terdapat 3 tabel yang ternormalisasi sampai dengan tingkat 3NF, yaitu tabel Pelanggan, Order, dan Order Detail. Tabel Pelanggan berisi data pelanggan, tabel Order berisi data order yang dibuat oleh pelanggan, dan tabel Order Detail berisi detail produk yang dipesan dalam suatu order. Dengan demikian, data tidak terduplikasi dan integritas data terjaga dengan baik.

* ***Redundansi (redudansi) data*** adalah ketika data yang sama terdapat lebih dari satu kali dalam suatu tabel atau basis data. Redundansi data dapat menyebabkan masalah seperti ketidak akuratan data, ketidak efisienan penyimpanan data, dan masalah integritas data.

Contoh dari redundansi data adalah ketika tabel yang berisi data pelanggan memiliki kolom "nama pelanggan" dan "alamat pelanggan", tetapi juga memiliki kolom "nama pelanggan dan alamat" yang berisi nama dan alamat pelanggan yang sama yang terpisah dalam kolom-kolom terpisah. Dalam kasus ini, data nama dan alamat pelanggan tersebut terdapat lebih dari satu kali dalam tabel, sehingga mengakibatkan redundansi data.

Untuk menghindari redundansi data, salah satu cara yang dapat dilakukan adalah dengan melakukan normalisasi data. Misalnya, dalam contoh di atas, kolom "nama pelanggan dan alamat" dapat dihilangkan dan data nama pelanggan dan alamat pelanggan disimpan dalam tabel pelanggan yang terpisah, sehingga menghindari redundansi data dan memastikan integritas data.

1. Apa yang dimaksud Basisdata SQL dan NonSQL? Contoh aplikasinya apa?

Penjelasan dan Contoh:

* ***Basis data SQL*** (Structured Query Language) adalah jenis basis data yang menggunakan bahasa SQL untuk mengelola dan mengakses data. Basis data SQL biasanya menyimpan data dalam bentuk tabel yang terstruktur dengan jelas, dan menggunakan query SQL untuk mengakses dan memanipulasi data.

Contoh aplikasi basis data SQL adalah **MySQL**, **Oracle**, dan **Microsoft SQL Server**.

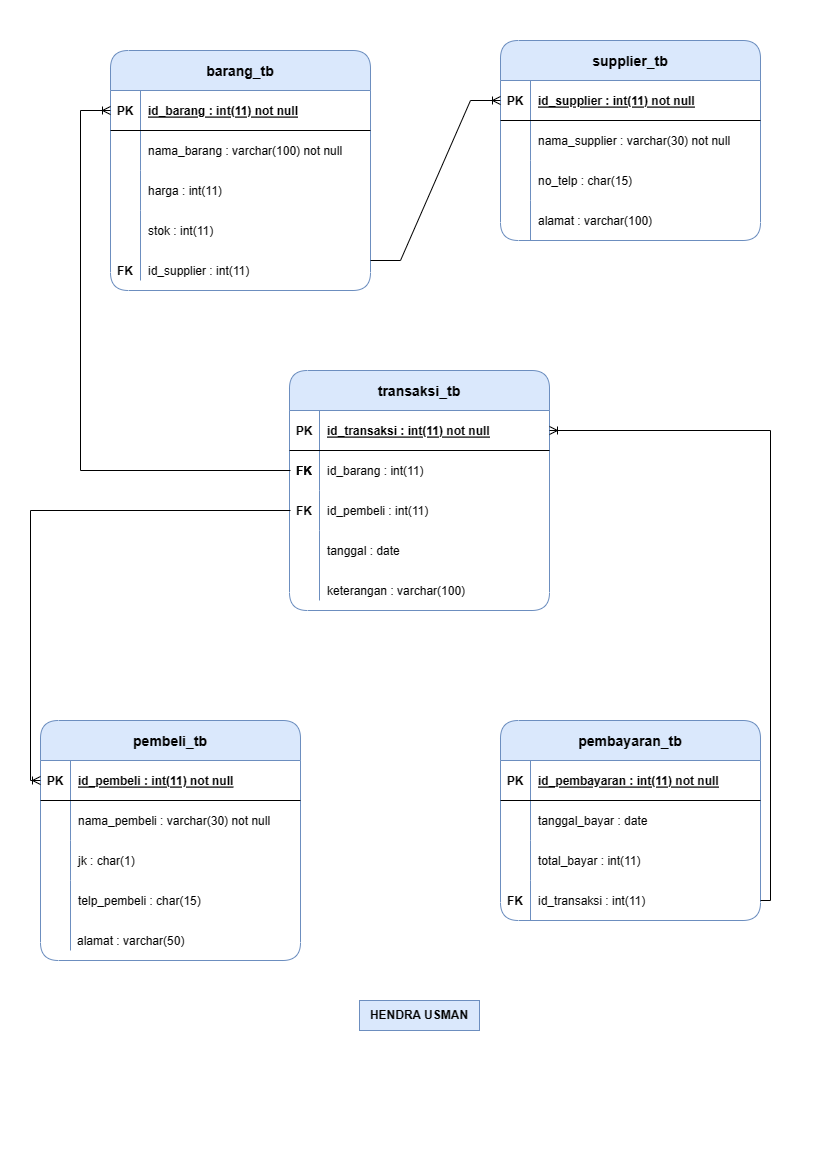
* ***Basis data Non-SQL***, atau NoSQL, adalah jenis basis data yang tidak menggunakan SQL untuk mengelola dan mengakses data. Basis data Non-SQL biasanya lebih fleksibel daripada basis data SQL karena tidak terikat oleh struktur tabel yang terbatas.

Contoh aplikasi basis data Non-SQL adalah **MongoDB**, **Cassandra**, dan **Redis**.

* Kedua jenis basis data ini dapat digunakan dalam berbagai macam aplikasi, termasuk aplikasi web, aplikasi mobile, dan aplikasi perusahaan. Namun, basis data SQL lebih sering digunakan untuk aplikasi yang memerlukan struktur data yang lebih ketat dan konsisten, sedangkan basis data Non-SQL lebih sering digunakan untuk aplikasi yang memerlukan fleksibilitas dan skalabilitas yang lebih tinggi.

1. Sketsa Penjualan (penjualan\_db)

*(link akses:* [*https://app.diagrams.net/?src=about#HMR-HENDRA%2Fdb1usb%2Fmaster%2Fretail.drawio*](https://app.diagrams.net/?src=about#HMR-HENDRA%2Fdb1usb%2Fmaster%2Fretail.drawio) *)*



1. Script Database Penjualan (penjualan\_db):

*( link akses:* [*db1usb/penjualan.sql at master · MR-HENDRA/db1usb (github.com)*](https://github.com/MR-HENDRA/db1usb/blob/master/DATABASE/penjualan.sql)*)*

* **Membuat Database *penjualan\_db***

CREATE DATABASE penjualan\_db;

* **Masuk ke *penjualan\_db***

\c penjualan\_db;

* **Membuat Tabel *supplier\_tb***

CREATE TABLE supplier\_tb (

    id\_supplier INT PRIMARY KEY NOT NULL,

    nama\_supplier VARCHAR (30) NOT NULL,

    no\_telp CHAR (15),

alamat VARCHAR (100)

);

* **Membuat Tabel *barang\_tb***

CREATE TABLE barang\_tb (

    id\_barang INT PRIMARY KEY NOT NULL,

    nama\_barang VARCHAR (100) NOT NULL,

    harga INT,

    stok INT,

    id\_supplier INT

);

* **Membuat Tabel *pembeli\_tb***

CREATE TABLE pembeli\_tb (

    id\_pembeli INT PRIMARY KEY NOT NULL,

    nama\_pembeli VARCHAR (30) NOT NULL,

    jk CHAR (1),

    telp\_pembeli CHAR (15),

    alamat\_pembeli VARCHAR (50)

);

* **Membuat Tabel *transaksi\_tb***

CREATE TABLE transaksi\_tb (

    id\_transaksi INT PRIMARY KEY NOT NULL,

    id\_barang INT,

    id\_pembeli INT,

    tanggal DATE,

    keterangan VARCHAR (100)

);

* **Membuat Tabel *pembayaran\_tb***

CREATE TABLE pembayaran\_tb (

    id\_pembayaran INT PRIMARY KEY NOT NULL,

    tanggal\_bayar DATE,

    total\_bayar INT,

    id\_transaksi INT

);

* **Membuat *Foreign Key* (FK) pada Tabel Tertentu**

ALTER TABLE barang\_tb ADD CONSTRAINT fk\_supplier\_barang FOREIGN KEY (id\_supplier) REFERENCES supplier\_tb(id\_supplier) ON DELETE CASCADE;

ALTER TABLE transaksi\_tb ADD CONSTRAINT fk\_barang\_transaksi FOREIGN KEY (id\_barang) REFERENCES barang\_tb(id\_barang) ON DELETE CASCADE;

ALTER TABLE transaksi\_tb ADD CONSTRAINT fk\_pembeli\_transaksi FOREIGN KEY (id\_pembeli) REFERENCES pembeli\_tb(id\_pembeli) ON DELETE CASCADE;

ALTER TABLE pembayaran\_tb ADD CONSTRAINT fk\_transaksi\_pembayaran FOREIGN KEY (id\_transaksi) REFERENCES transaksi\_tb(id\_transaksi) ON DELETE CASCADE;

* **Insert Data pada Tabel *supplier\_tb***

INSERT INTO supplier\_tb (id\_supplier, nama\_supplier, no\_telp, alamat) VALUES

('202201', 'PT. SEMESTA TECHNOLOGY', '08123456789', 'Jakarta Timur'),

('202202', 'PT. KARANG SAMUDERA', '08133456789', 'Bandung'),

('202203', 'PT. JAYA ABADI', '08143456789', 'Malang'),

('202204', 'PT. MINANG', '08153456789', 'Padang'),

('202205', 'PT. DAENG', '08163456789', 'Makassar'),

('202206', 'PT. MANDARQ', '08173456789', 'Majene');

* **Insert Data pada Tabel *barang\_tb***

INSERT INTO barang\_tb (id\_barang, nama\_barang, harga, stok, id\_supplier) VALUES

('202201001', 'Keyboard Rexus', '500000', '100', '202201'),

('202202001', 'Case Black Keyboard', '500000', '200', '202202'),

('202203001', 'Mouse Wireless VX-01', '300000', '100', '202203'),

('202204001', 'Headset Gaming', '200000', '1000', '202204'),

('202205001', 'Earphone Gaming', '200000', '1000', '202205'),

('202206001', 'Smart CCTV', '3000000', '5000', '202206');

* **Insert Data pada Tabel *pembeli\_tb***

INSERT INTO pembeli\_tb (id\_pembeli, nama\_pembeli, jk, telp\_pembeli, alamat\_pembeli) VALUES

('302201', 'Fivin', 'P', '08223456789', 'Babana'),

('302202', 'Via', 'P', '08323456789', 'Lembang'),

('302203', 'Nia', 'P', '08423456789', 'Tarailu'),

('302204', 'Arif', 'L', '08523456789', 'Baurung'),

('302205', 'Alim', 'L', '08623456789', 'Baurung'),

('302206', 'Harun', 'L', '08723456789', 'L. Maloga');

* **Insert Data pada Tabel *transaksi\_tb***

INSERT INTO transaksi\_tb (id\_transaksi, id\_barang, id\_pembeli, tanggal, keterangan) VALUES

('401', '202201001', '302201', '2022-12-28', 'no-ket'),

('402', '202202001', '302202', '2022-12-28', 'no-ket'),

('403', '202203001', '302203', '2022-12-28', 'no-ket'),

('404', '202204001', '302204', '2022-12-28', 'no-ket'),

('405', '202205001', '302205', '2022-12-28', 'no-ket'),

('406', '202206001', '302206', '2022-12-28', 'no-ket');

* **Insert Data pada Tabel *pembayaran\_tb***

INSERT INTO pembayaran\_tb (id\_pembayaran, tanggal\_bayar, total\_bayar, id\_transaksi) VALUES

('501', '2022-12-28','500000', '401'),

('502', '2022-12-28','500000', '402'),

('503', '2022-12-28','300000', '403'),

('504', '2022-12-28','200000', '404'),

('505', '2022-12-28','200000', '405'),

('506', '2022-12-28','3000000', '406');