TUGAS PBOP

“RELASI DATABASE”

Nama :

1. Yoga rusdi hantoro_5230411266

FAKULTAS SAINS DAN TEKNOLOGI

UNIV TEKNOLOGI YOGYAKARTA

2024/2025

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa atas limpahan rahmatNya sehingga penulis dapat mengatasi segala rintangan dan kesulitan sampai akhirnya dapat menyelesaikan laporan ini, penulis tidak lupa mengucapkan terimakasih atas bantuan dan kerjasama dari berbagai pihak. Ucapan terima kasih ini penulis haturkan Semoga amal dan kebaikan saudara-saudara yang mendapatkan balasan yang setimpal dari Tuhan Yang Maha Esa. Penulis menyadari segala kekurangan dan ketidaksempurnaan laporan ini, dengan segala kerendahan hati penulis dengan senang hati menerima kritik dan saran yang sifatnya membangun guna perbaikan dan kesempurnaan laporan ini. Semoga laporan ini dapat bermanfaat bagi pihak yang berkepentingan.

DAFTAR ISI

PENDAHULUAN

Latar Belakang

Basis data adalah penyimpanan data yang terstruktur (*database*). Seharusnya, kamu juga sudah familier dengan istilah itu, tetapi mungkin belum tahu kenapa kita membutuhkannya. Nah, simak analogi di bawah ini untuk lebih mengerti tentang basis data.

Model relasional mengatur data ke dalam tabel — dengan setiap baris mewakili setiap catatan dan setiap kolom terdiri dari atribut yang berisi nilai. Struktur basis data *tabular* ini mempermudah membangun hubungan di antara titik-titik data sehingga informasi dapat diakses dengan berbagai cara yang berbeda tanpa harus mengatur ulang data itu sendiri.

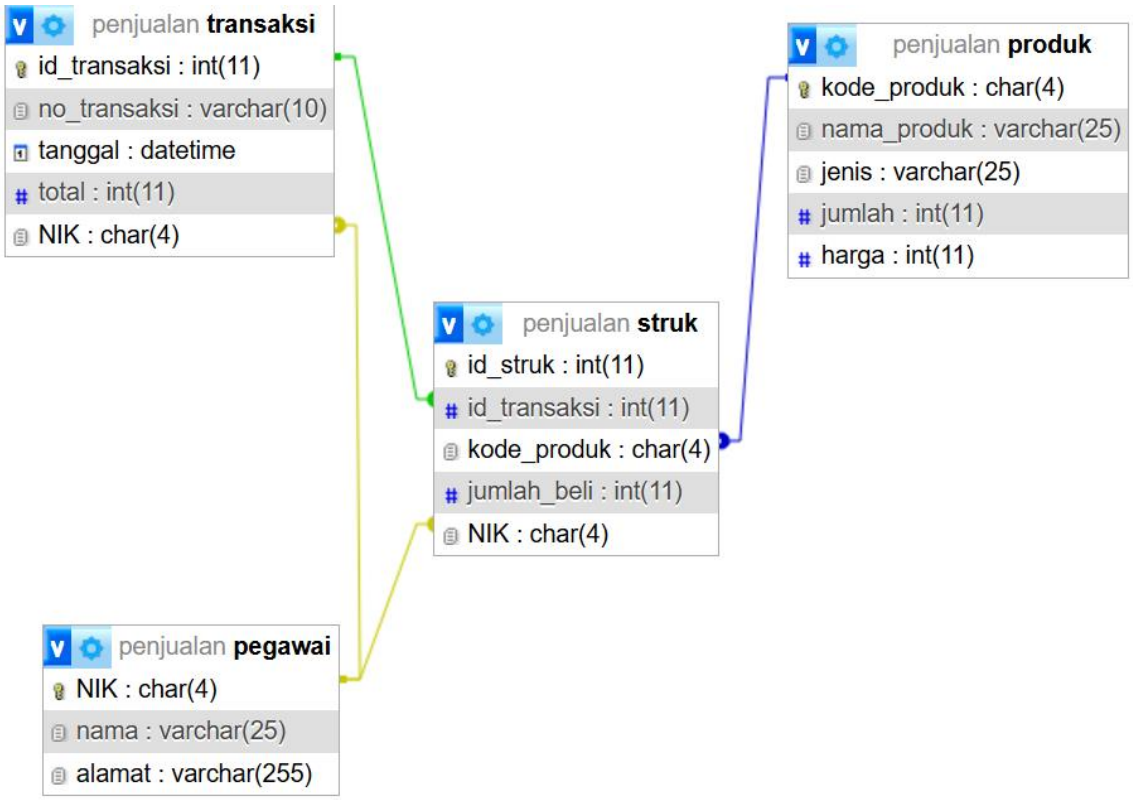
Tujuan

Tujuan dibuatnya laporan ini adalah untuk menuntaskan tugas Pemrograman Berbasis objek pertemuan 9.dan untuk mendalami dan memelajari relasi database dan penggunaan dalam program python

BAB II

1.Relasi Database

Pada relasi tabel ini terdapat 4 tabel yang digunakan yaitu transaksi,prudk ,strukdann pegawai.pada relasi database ini struk sebagai tabel yang menampung segala kode unik untuk disimpan.dan



2. Program membuat Database dan Tabel

Pada program ini tentu saja diawali dengan import mysql.connector untuk menghubungkan database yaitu sql dengan program.Pada sub program create_connection digunakan untuk menghubungan program dengan database yang digunakan dan pada program ini saya menggunakan database penjualan.

Pada Sub Program initialize_database pada sub program ini digunakan untuk membuat database penjualan dan table table yang digunakan dan pada sub program ini jikan dijalankan juga ada validasi jika database telah dibuat dan pada table tertabat 4 tabel yang berisi yaitu pegawai,struk,transaksi dan produk.

```
import mysql.connector

def create_connection(database=True):
    if database:
        return mysql.connector.connect(
            host="localhost",
```

```
            user="root",
            password="",
            database="penjualan"
        )
    else:
        return mysql.connector.connect(
            host="localhost",
            user="root",
            password=""
        )

def initialize_database():
    conn = create_connection(database=False)
    cur = conn.cursor()
    cur.execute("CREATE DATABASE IF NOT EXISTS penjualan")
    cur.execute("USE penjualan")

    cur.execute("""
    CREATE TABLE IF NOT EXISTS pegawai (
        NIK CHAR(4) NOT NULL PRIMARY KEY,
        nama VARCHAR(25),
        alamat VARCHAR(255)
    )
    """)

    cur.execute("""
    CREATE TABLE IF NOT EXISTS produk (
        kode_produk CHAR(4) NOT NULL PRIMARY KEY,
        nama_produk VARCHAR(25),
        jenis VARCHAR(25),
        jumlah INT,
        harga INT
    )
    """)

    cur.execute("""
    CREATE TABLE IF NOT EXISTS transaksi (
        id_transaksi INT AUTO_INCREMENT PRIMARY KEY,
        no_transaksi VARCHAR(10),
        tanggal DATETIME,
        total INT,
        NIK CHAR(4),
        FOREIGN KEY (NIK) REFERENCES pegawai(NIK)
    )
    """)

    cur.execute("""
    CREATE TABLE IF NOT EXISTS struk (
```

```
    id_struk INT AUTO_INCREMENT PRIMARY KEY,
    id_transaksi INT,
    kode_produk CHAR(4),
    jumlah_beli INT,
    NIK CHAR(4),
    FOREIGN KEY (id_transaksi) REFERENCES transaksi(id_transaksi),
    FOREIGN KEY (kode_produk) REFERENCES produk(kode_produk),
    FOREIGN KEY (NIK) REFERENCES pegawai(NIK)
  )
  """)
```

3. CRUD Pada Tabel Pegawai

Pada bagian pegawai ini berisi sub program Create,Read,Update dan Delete.pada bagian pegawai berisi atribut NIK,nama,Alamat sebagai identias setiap pegawai.Untuk setiap verivikasi CRUD yang digunakan adalah atribut NiK sebagai indentitas.

```
def create_pegawai():
    NIK = input("Masukkan NIK: ")
    if not NIK or len(NIK) != 4:
        print("NIK harus berisi 4 karakter.")
        return
    nama = input("Masukkan Nama: ")
    if not nama:
        print("Nama tidak boleh kosong.")
        return
    alamat = input("Masukkan Alamat: ")
    if not alamat:
        print("Alamat tidak boleh kosong.")
        return
    conn = create_connection()
    cur = conn.cursor()
    cur.execute("INSERT INTO pegawai (NIK, nama, alamat) VALUES (%s, %s, %s)",
(NIK, nama, alamat))
    conn.commit()
    cur.close()
    conn.close()
    print("Pegawai berhasil ditambahkan.")

def read_pegawai():
    conn = create_connection()
    cur = conn.cursor()
    cur.execute("SELECT * FROM pegawai")
    results = cur.fetchall()
    for row in results:
        print(row)
    cur.close()
    conn.close()

def update_pegawai():
```

```
    NIK = input("Masukkan NIK Pegawai yang akan diupdate: ")
    if not NIK or len(NIK) != 4:
        print("NIK harus berisi 4 karakter.")
        return
    nama = input("Masukkan Nama baru: ")
    if not nama:
        print("Nama tidak boleh kosong.")
        return
    alamat = input("Masukkan Alamat baru: ")
    if not alamat:
        print("Alamat tidak boleh kosong.")
        return
    conn = create_connection()
    cur = conn.cursor()
    cur.execute("UPDATE pegawai SET nama=%s, alamat=%s WHERE NIK=%s", (nama,
alamat, NIK))
    conn.commit()
    cur.close()
    conn.close()
    print("Data Pegawai berhasil diupdate.")

def delete_pegawai():
    NIK = input("Masukkan NIK Pegawai yang akan dihapus: ")
    if not NIK or len(NIK) != 4:
        print("NIK harus berisi 4 karakter.")
        return
    conn = create_connection()
    cur = conn.cursor()
    cur.execute("DELETE FROM pegawai WHERE NIK=%s", (NIK,))
    conn.commit()
    cur.close()
    conn.close()
    print("Pegawai berhasil dihapus.")
```

4.CRUD pada tabel Produk

Pad aprogam produk terdapat juga kemiripan dengan program pada tabel yang lain namun terdapat perbedaan juga di dalam bagian ini.

Pada tabel berisi atribut kode produk,nama produk,jenis,jumlah,dan harga produk.Pada tabel ini primary key adalah Kode_produk sebagai atribut unuik yang tidak dapat disamakan dengan atribut yang lain.

```
def create_produk():
    kode_produk = input("Masukkan Kode Produk: ")
    if not kode_produk or len(kode_produk) != 4:
        print("Kode produk harus berisi 4 karakter.")
        return
    nama_produk = input("Masukkan Nama Produk: ")
    if not nama_produk:
```

```
        print("Nama produk tidak boleh kosong.")
        return
    jenis = input("Masukkan Jenis Produk: ")
    if not jenis:
        print("Jenis produk tidak boleh kosong.")
        return
    jumlah = input("Masukkan Jumlah: ")
    if not jumlah.isdigit() or int(jumlah) <= 0:
        print("Jumlah harus berupa angka positif.")
        return
    harga = input("Masukkan Harga: ")
    if not harga.isdigit() or int(harga) <= 0:
        print("Harga harus berupa angka positif.")
        return
    conn = create_connection()
    cur = conn.cursor()
    cur.execute("INSERT INTO produk (kode_produk, nama_produk, jenis, jumlah,
harga) VALUES (%s, %s, %s, %s, %s)",
                (kode_produk, nama_produk, jenis, int(jumlah), int(harga)))
    conn.commit()
    cur.close()
    conn.close()
    print("Produk berhasil ditambahkan.")

def read_produk():
    conn = create_connection()
    cur = conn.cursor()
    cur.execute("SELECT * FROM produk")
    results = cur.fetchall()
    for row in results:
        print(row)
    cur.close()
    conn.close()

def update_produk():
    kode_produk = input("Masukkan Kode Produk yang akan diupdate: ")
    if not kode_produk or len(kode_produk) != 4:
        print("Kode produk harus berisi 4 karakter.")
        return
    nama_produk = input("Masukkan Nama Produk baru: ")
    if not nama_produk:
        print("Nama produk tidak boleh kosong.")
        return
    jenis = input("Masukkan Jenis Produk baru: ")
    if not jenis:
        print("Jenis produk tidak boleh kosong.")
        return
    jumlah = input("Masukkan Jumlah baru: ")
```

```
    if not jumlah.isdigit() or int(jumlah) <= 0:
        print("Jumlah harus berupa angka positif.")
        return
    harga = input("Masukkan Harga baru: ")
    if not harga.isdigit() or int(harga) <= 0:
        print("Harga harus berupa angka positif.")
        return
    conn = create_connection()
    cur = conn.cursor()
    cur.execute("UPDATE produk SET nama_produk=%s, jenis=%s, jumlah=%s,
harga=%s WHERE kode_produk=%s",
                (nama_produk, jenis, int(jumlah), int(harga), kode_produk))
    conn.commit()
    cur.close()
    conn.close()
    print("Produk berhasil diupdate.")
```

5.CRUD pada Tabel Transaksi

Pada tabel transaksi terdapat atribut no transaksii,tanggal,total,dan NIK.pada tabel ini primary yaitu atribut id transaksi yang sudah tergenerate pada database dan juga.dan pad transaksi ini akan terhubung dengan struk.

```
def create_transaksi():
    no_transaksi = input("Masukkan Nomor Transaksi: ")
    tanggal = input("Masukkan Tanggal (YYYY-MM-DD): ")
    try:
        datetime.datetime.strptime(tanggal, '%Y-%m-%d')
    except ValueError:
        print("Format tanggal salah. Harus YYYY-MM-DD.")
        return
    total = input("Masukkan Total Transaksi: ")
    if not total.isdigit() or int(total) <= 0:
        print("Total harus berupa angka positif.")
        return
    NIK = input("Masukkan NIK Pegawai: ")
    if not NIK or len(NIK) != 4:
        print("NIK harus berisi 4 karakter.")
        return
    conn = create_connection()
    cur = conn.cursor()
    cur.execute("INSERT INTO transaksi (no_transaksi, tanggal, total, NIK)
VALUES (%s, %s, %s, %s)",
                (no_transaksi, tanggal, int(total), NIK))
    conn.commit()
```

```
    cur.close()
    conn.close()
    print("Transaksi berhasil ditambahkan.")

def read_transaksi():
    conn = create_connection()
    cur = conn.cursor()
    cur.execute("SELECT * FROM transaksi")
    results = cur.fetchall()
    for row in results:
        print(row)
    cur.close()
    conn.close()

def update_transaksi():
    id_transaksi = input("Masukkan ID Transaksi yang akan diupdate: ")
    if not id_transaksi.isdigit():
        print("ID Transaksi harus berupa angka.")
        return
    no_transaksi = input("Masukkan Nomor Transaksi baru: ")
    tanggal = input("Masukkan Tanggal baru (YYYY-MM-DD): ")
    try:
        datetime.datetime.strptime(tanggal, '%Y-%m-%d')
    except ValueError:
        print("Format tanggal salah. Harus YYYY-MM-DD.")
        return
    total = input("Masukkan Total baru: ")
    if not total.isdigit() or int(total) <= 0:
        print("Total harus berupa angka positif.")
        return
    NIK = input("Masukkan NIK Pegawai: ")
    if not NIK or len(NIK) != 4:
        print("NIK harus berisi 4 karakter.")
        return
    conn = create_connection()
    cur = conn.cursor()
    cur.execute("UPDATE transaksi SET no_transaksi=%s, tanggal=%s, total=%s,
NIK=%s WHERE id_transaksi=%s",
                (no_transaksi, tanggal, int(total), NIK, id_transaksi))
    conn.commit()
    cur.close()
    conn.close()
    print("Transaksi berhasil diupdate.")

def delete_transaksi():
    id_transaksi = input("Masukkan ID Transaksi yang akan dihapus: ")
    if not id_transaksi.isdigit():
        print("ID Transaksi harus berupa angka.")
```

```
        return
    conn = create_connection()
    cur = conn.cursor()
    cur.execute("DELETE FROM transaksi WHERE id_transaksi=%s",
(id_transaksi,))
    conn.commit()
    cur.close()
    conn.close()
    print("Transaksi berhasil dihapus.")
```

6. CRUD untuk Tabel Struk

CRUD pada tabel struk ini tentu saja terdapat validasi pada setiap sub program.

Pada tabel struk berisi atribut id transaksi ,kode produk,jumlah beli dan NIK yang terhubung dengan semua tabel.

```
def create_struk():
    id_transaksi = input("Masukkan ID Transaksi: ")
    if not id_transaksi.isdigit():
        print("ID Transaksi harus berupa angka.")
        return
    kode_produk = input("Masukkan Kode Produk: ")
    if not kode_produk or len(kode_produk) != 4:
        print("Kode produk harus berisi 4 karakter.")
        return
    jumlah_beli = input("Masukkan Jumlah Beli: ")
    if not jumlah_beli.isdigit() or int(jumlah_beli) <= 0:
        print("Jumlah Beli harus berupa angka positif.")
        return
    NIK = input("Masukkan NIK Pegawai: ")
    if not NIK or len(NIK) != 4:
        print("NIK harus berisi 4 karakter.")
        return
    conn = create_connection()
    cur = conn.cursor()
    cur.execute("INSERT INTO struk (id_transaksi, kode_produk, jumlah_beli,
NIK) VALUES (%s, %s, %s, %s)",
                (id_transaksi, kode_produk, int(jumlah_beli), NIK))
    conn.commit()
    cur.close()
    conn.close()
    print("Struk berhasil ditambahkan.")
```

```python
def read_struk():
    conn = create_connection()
    cur = conn.cursor()
    cur.execute("SELECT * FROM struk")
    results = cur.fetchall()
    for row in results:
        print(row)
    cur.close()
    conn.close()

def update_struk():
    id_struk = input("Masukkan ID Struk yang akan diupdate: ")
    if not id_struk.isdigit():
        print("ID Struk harus berupa angka.")
        return
    id_transaksi = input("Masukkan ID Transaksi baru: ")
    if not id_transaksi.isdigit():
        print("ID Transaksi harus berupa angka.")
        return
    kode_produk = input("Masukkan Kode Produk baru: ")
    if not kode_produk or len(kode_produk) != 4:
        print("Kode produk harus berisi 4 karakter.")
        return
    jumlah_beli = input("Masukkan Jumlah Beli baru: ")
    if not jumlah_beli.isdigit() or int(jumlah_beli) <= 0:
        print("Jumlah Beli harus berupa angka positif.")
        return
    NIK = input("Masukkan NIK Pegawai: ")
    if not NIK or len(NIK) != 4:
        print("NIK harus berisi 4 karakter.")
        return
    conn = create_connection()
    cur = conn.cursor()
    cur.execute("UPDATE struk SET id_transaksi=%s, kode_produk=%s,
jumlah_beli=%s, NIK=%s WHERE id_struk=%s",
                (id_transaksi, kode_produk, int(jumlah_beli), NIK, id_struk))
    conn.commit()
    cur.close()
    conn.close()
    print("Struk berhasil diupdate.")

def delete_struk():
    id_struk = input("Masukkan ID Struk yang akan dihapus: ")
    if not id_struk.isdigit():
        print("ID Struk harus berupa angka.")
        return
    conn = create_connection()
```

```
    cur = conn.cursor()
    cur.execute("DELETE FROM struk WHERE id_struk=%s", (id_struk,))
    conn.commit()
    cur.close()
    conn.close()
    print("Struk berhasil dihapus.")
```

7.Menu

Pada sub program menu terdapat 4 Sub program lagi yang berisi menu untuk proses menjalankan semua program yang lain berdasarkan tabel yang digunakan agar lebih tertata.

```
def menu_pegawai():
    while True:
        print("\n--- Menu Pegawai ---")
        print("1. Tambah Pegawai")
        print("2. Tampilkan Pegawai")
        print("3. Update Pegawai")
        print("4. Hapus Pegawai")
        print("0. Kembali ke Menu Utama")

        pilihan = input("Pilih operasi: ")

        if pilihan == '1':
            create_pegawai()
        elif pilihan == '2':
            read_pegawai()
        elif pilihan == '3':
            update_pegawai()
        elif pilihan == '4':
            delete_pegawai()
        elif pilihan == '0':
            break
        else:
            print("Pilihan tidak valid.")

def menu_produk():
    while True:
        print("\n--- Menu Produk ---")
        print("1. Tambah Produk")
        print("2. Tampilkan Produk")
        print("3. Update Produk")
        print("4. Hapus Produk")
        print("0. Kembali ke Menu Utama")

        pilihan = input("Pilih operasi: ")

        if pilihan == '1':
            create_produk()
```

```
        elif pilihan == '2':
            read_produk()
        elif pilihan == '3':
            update_produk()
        elif pilihan == '4':
            delete_produk()
        elif pilihan == '0':
            break
        else:
            print("Pilihan tidak valid.")

def menu_transaksi():
    while True:
        print("\n--- Menu Transaksi ---")
        print("1. Tambah Transaksi")
        print("2. Tampilkan Transaksi")
        print("3. Update Transaksi")
        print("4. Hapus Transaksi")
        print("0. Kembali ke Menu Utama")

        pilihan = input("Pilih operasi: ")

        if pilihan == '1':
            create_transaksi()
        elif pilihan == '2':
            read_transaksi()
        elif pilihan == '3':
            update_transaksi()
        elif pilihan == '4':
            delete_transaksi()
        elif pilihan == '0':
            break
        else:
            print("Pilihan tidak valid.")

def menu_struk():
    while True:
        print("\n--- Menu Struk ---")
        print("1. Tambah Struk")
        print("2. Tampilkan Struk")
        print("3. Update Struk")
        print("4. Hapus Struk")
        print("0. Kembali ke Menu Utama")

        pilihan = input("Pilih operasi: ")

        if pilihan == '1':
            create_struk()
```

```
        elif pilihan == '2':
            read_struk()
        elif pilihan == '3':
            update_struk()
        elif pilihan == '4':
            delete_struk()
        elif pilihan == '0':
            break
        else:
            print("Pilihan tidak valid.")

def main():
    while True:
        print("\n=== MENU UTAMA SISTEM MANAJEMEN PENJUALAN ===")
        print("1. Operasi Pegawai")
        print("2. Operasi Produk")
        print("3. Operasi Transaksi")
        print("4. Operasi Struk")
        print("0. Keluar")

        pilihan = input("Masukkan pilihan Anda: ")

        if pilihan == '1':
            menu_pegawai()
        elif pilihan == '2':
            menu_produk()
        elif pilihan == '3':
            menu_transaksi()
        elif pilihan == '4':
            menu_struk()
        elif pilihan == '0':
            print("Terima kasih telah menggunakan sistem ini.")
            break
        else:
            print("Pilihan tidak valid. Silakan coba lagi.")

if __name__ == "__main__":
    main()
```

PENUTUP

Setelah melalui serangkaian kegiatan dan analisis yang telah dijelaskan dalam laporan ini, dapat disimpulkan bahwa tujuan dari penelitian ini telah tercapai. Hasil yang diperoleh menunjukkan bahwa implementasi Relasi database. Selain itu, penelitian ini juga memberikan wawasan baru mengenai relasi database pada program python yang dapat menjadi dasar untuk penelitian lebih lanjut di masa depan.

Kami menyadari bahwa penelitian ini masih memiliki keterbatasan, seperti tidak kerapian program dan atribut yang sekiranya kurang, yang diharapkan dapat diperbaiki dalam penelitian selanjutnya.

Akhir kata, kami mengucapkan terima kasih kepada semua pihak yang telah membantu dalam pengerjaan Tugas ini, baik secara langsung maupun tidak langsung. Semoga hasil penelitian ini dapat memberikan manfaat.