

# **ALGORITMA DAN PEMROGRAMAN**

*Project Akhir*

**Dosen Pengampu:**

Ariawan Andi Suhandana, S.Kom. M.Ti



**Disusun Oleh:**

Kelompok 2

Kerta Wijaya	(2307421003)
Wanda Rizki Hamidah	(2307421006)
Nabiel Ischak	(2307421012)
Dara Samsara Ayu	(2307421013)
Reza Zidan Hanafi	(2307421014)
Pandora Ophelia Krstanto	(2307421018)

**TMJ 1A**

**PROGRAM STUDI TEKNIK MULTIMEDIA DAN JARINGAN**

**JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER**

**POLITEKNIK NEGERI JAKARTA**

**2023**

## **KATA PENGANTAR**

Dengan rahmat Tuhan Yang Maha Esa, kita panjatkan puji dan syukur ke hadirat-Nya yang telah melimpahkan berkat-Nya berupa pengetahuan dan kesempatan sehingga makalah mengenai yang berjudul “Project Akhir” dalam Algoritma dan Pemograman ini dapat diselesaikan dengan tepat waktu. Penulisan laporan ini dilakukan dalam rangka memenuhi tugas dari mata kuliah Algoritma dan Pemograman di Politeknik Negeri Jakarta.

Kami menyadari bahwa makalah ini masih jauh dari kata sempurna dikarenakan keterbatasan pengetahuan dan pengalaman dari kami. Oleh sebab itu, kami mengharapkan saran dan kritik dari pembaca agar kami dapat menjadikan makalah ini lebih sempurna lagi.

Akhir kata kami berharap kiranya makalah berjudul “Project Akhir” ini dapat memberikan manfaat maupun inspirasi terhadap pembaca.

Depok, 15 Januari 2024

Penyusun

## **LATAR BELAKANG**

Minimarket adalah sebuah jenis usaha yang menggabungkan antara konsep swalayan dalam skala kecil dengan target pasar yang sama dengan target pasar tradisional. Minimarket merupakan salah satu bentuk toko kelontong yang menjual berbagai produk, seperti makanan, perabot rumah tangga dan barang lainnya. Dalam beberapa tahun terakhir, minimarket menjadi tempat belanja yang sering digunakan oleh masyarakat karena mudah dengannya untuk berbelanja dan ketersediaan barang yang bervariasi. Namun, dalam beberapa kasus, pengolahan data transaksi dan pembuatan laporan di minimarket masih menjadi tantangan yang perlu diatasi, ditambah dengan semakin canggihnya teknologi di era sekarang maka dari itu kita disini membuat sebuah program yang memudahkan setiap kasir dalam transaksi pembelian suatu barang, dengan membuat sebuah program aplikasi yang memiliki fitur pendataan barang-barang yang dijual dan sekaligus proses penjumlahan total barang dan harga setiap pembelian.

## **PROBLEM YANG INGIN DIATASI**

Beberapa masalah yang ada dalam pengolahan data transaksi dan pembuatan laporan di minimarket, seperti ketidakakpastian dalam mengelola stok barang, kesulitan dalam membuat laporan transaksi dan ketergantungan pada sistem manual dalam mengelola transaksi penjualan dan pembelian.

## **TUJUAN**

Mengembangkan program kasir minimarket yang mampu mengelola transaksi penjualan dan pembelian dengan lebih efisien dan akurat, serta membantu minimarket dalam mengelola inventaris dan laporan dengan lebih rapih.

## **FITUR**

- a. Dual login: Aplikasi memiliki login khusus untuk admin dan kasir.
- b. Transaksi penjalan dan pembelian: Mencakup pencatatan transaksi penjualan dan pembelian, dengan menciptakan no transaksi, input produk dan mengatur metode pembayaran.
- c. Inventaris: Fitur untuk mengelola stok barang, seperti menambahkan produk, mengurangi stok, dan menghapus stok.
- d. Laporan: Menciptakan laporan transaksi untuk memudahkan analisis keuangan dan penjualan, seperti laporan penjualan, dan pembelian.
- e. Manajemen produk: Memungkinkan admin untuk menambah, mengedit, dan menghapus produk.

## ALGORITMA

Algoritma adalah independen terhadap bahasa pemrograman tertentu, artinya algoritma yang telah dibuat tidak boleh hanya dapat diterapkan pada bahasa pemrograman tertentu. Penulisan algoritma tidak terikat pada suatu aturan tertentu, tetapi harus jelas maksudnya untuk tiap langkah algoritmanya.

Algoritma merupakan urutan logis pengambilan keputusan untuk pemecahan masalah. Kata logis merupakan kata kunci. Langkah-langkah tersebut harus logis, ini berarti nilai kebenarannya harus dapat ditentukan, benar atau salah. Langkah-langkah yang tidak benar dapat memberikan hasil yang salah. Sebagai contoh tinjau persoalan mempertukarkan isi dua buah bejana, A dan B. Bejana A berisi larutan yang berwarna merah, sedangkan bejana B berisi air berwarna biru. Kita ingin mempertukarkan isi kedua bejana itu sedemikian sehingga bejana A berisi larutan berwarna biru dan bejana B berisi larutan berwarna merah.

Berikut adalah algoritma dari project akhir yang merupakan langkah-langkah membuat transaksi penjual dan pembelian:

1. Import library sqlite3
2. Buat koneksi ke database dengan nama "db\_kasir.db"
3. Buat cursor untuk mengakses database
4. Buat tabel "barang" dan "transaksi" jika belum ada
5. Buat fungsi "get\_barang" untuk mendapatkan data barang berdasarkan kode barang
6. Buat fungsi "lihat\_pembelian" untuk menampilkan data pembelian pelanggan
7. Tampilkan menu login sebagai admin atau kasir
8. Jika login sebagai admin, tampilkan menu CRUD
9. Jika pilihan menu adalah 1, tambahkan data barang baru ke dalam tabel "barang"
10. Jika pilihan menu adalah 2, tampilkan semua data barang dari tabel "barang"
11. Jika pilihan menu adalah 3, ubah data barang yang sudah ada di tabel "barang"
12. Jika pilihan menu adalah 4, hapus data barang yang sudah ada di tabel "barang"
13. Jika pilihan menu adalah 5, tampilkan data pembelian pelanggan dari tabel "transaksi"
14. Jika pilihan menu adalah 6, logout dari akun admin

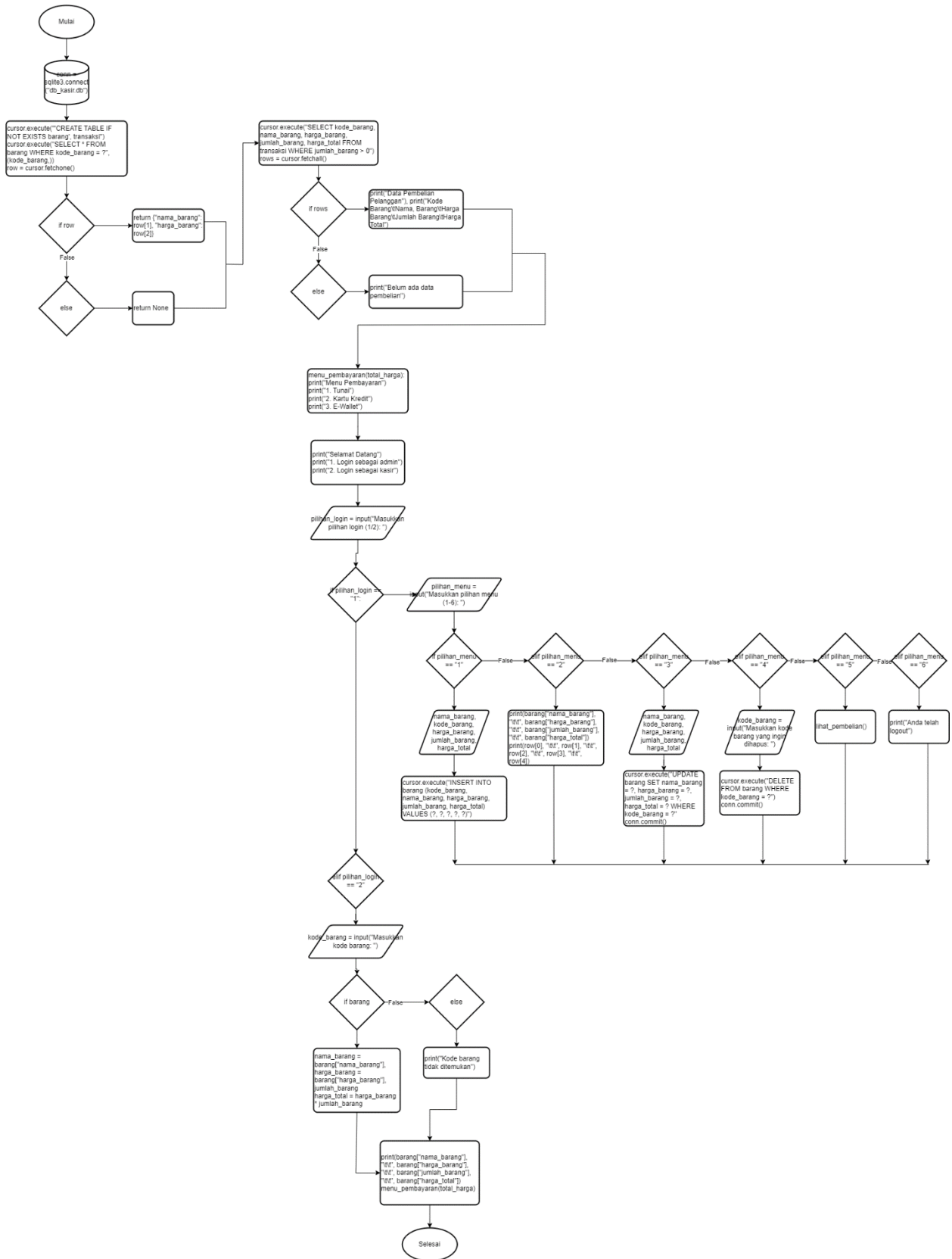
15. Jika login sebagai kasir, buat keranjang belanja
16. Dapatkan data barang berdasarkan kode barang yang dimasukkan oleh kasir
17. Jika kode barang ditemukan, tampilkan nama barang, harga barang, dan minta input jumlah barang yang ingin dibeli
18. Hitung harga total dari barang yang dibeli dan tambahkan ke keranjang belanja
19. Simpan data pembelian ke dalam tabel "transaksi"
20. Tampilkan semua barang yang sudah dibeli beserta harga satuan, jumlah, dan harga total
21. Tampilkan total harga dari semua barang yang sudah dibeli
22. Tutup koneksi ke database.

## FLOWCHART

*Flowchart* (Bagian alir) terbagi menjadi 2 bagian yaitu *flowchart program* dan *flowchart system*. *Flochart program* adalah Bagan alir program adalah suatu bagan yang menggambarkan arus logika dari data yang akan diproses dalam suatu program dari awal sampai akhir. Bagan alir program merupakan alat yang berguna bagi programmer untuk mempersiapkan program yang rumit. Bagan alir terdiri dari simbol-simbol yang mewakili fungsi-fungsi langkah program dan garis alir (flow lines) menunjukkan urutan dari symbol yang akan dikerjakan.

*Flowchart system* adalah Bagan alir sistem berbeda dengan bagan alir program. Bagan alir program sifatnya lebih terperinci tentang langkah-langkah proses di dalam program dari awal sampai akhir. Bagan alir sistem hanya menggambarkan arus data dari sistem. Simbol symbol yang digunakan pada bagan alir sistem ada yang sama dan ada yang berbeda dengan simbol-simbol yang digunakan pada bagan alir program.

Berikut adalah *flowchart* dari project akhir yang merupakan bagan alir membuat transaksi penjual dan pembeli:





## CODE DAN PENJELASAN

```
1 import sqlite3
2 conn = sqlite3.connect("db_kasir.db")
3 cursor = conn.cursor()
4 cursor.execute('CREATE TABLE IF NOT EXISTS barang (kode_barang text, nama_barang text, harga_barang number, jumlah_barang number, harga_total number)')
5 cursor.execute('CREATE TABLE IF NOT EXISTS transaksi (kode_barang text, nama_barang text, harga_barang number, jumlah_barang number, harga_total number)')
```

`import sqlite3`: mengimpor modul `sqlite3` yang digunakan untuk mengakses database SQLite dari Python.

`conn = sqlite3.connect("db_kasir.db")`: membuat koneksi ke database dengan nama "db\_kasir.db". Jika database tersebut belum ada, maka akan dibuat secara otomatis.

`cursor = conn.cursor()`: membuat objek cursor yang digunakan untuk mengeksekusi perintah SQL pada database.

`cursor.execute("CREATE TABLE IF NOT EXISTS barang (kode_barang text, nama_barang text, harga_barang number, jumlah_barang number, harga_total number)")`: mengeksekusi perintah SQL untuk membuat tabel "barang" pada database. Tabel ini memiliki lima kolom yaitu `kode_barang`, `nama_barang`, `harga_barang`, `jumlah_barang`, dan `harga_total`. Jika tabel sudah ada, maka perintah ini tidak akan membuat tabel baru.

`cursor.execute("CREATE TABLE IF NOT EXISTS transaksi (kode_barang text, nama_barang text, harga_barang number, jumlah_barang number, harga_total number)")`: mengeksekusi perintah SQL untuk membuat tabel "transaksi" pada database. Tabel ini memiliki lima kolom yang sama dengan tabel "barang". Jika tabel sudah ada, maka perintah ini tidak akan membuat tabel baru.

```
1 def lihat_pembelian():
2     cursor.execute("SELECT kode_barang, nama_barang, harga_barang, jumlah_barang, harga_total FROM transaksi WHERE jumlah_barang > 0")
3     rows = cursor.fetchall()
4     if rows:
5         print("Data Pembelian Pelanggan")
6         print("Kode Barang\tNama Barang\tHarga Barang\tJumlah Barang\tHarga Total")
7         for row in rows:
8             print(row[0], "\t\t", row[1], "\t\t", row[2], "\t\t", row[3], "\t\t", row[4])
9     else:
10        print("Belum ada data pembelian")
```

Fungsi `lihat_pembelian()` pertama-tama mengeksekusi perintah SQL `SELECT` untuk memilih kolom `kode_barang`, `nama_barang`, `harga_barang`, `jumlah_barang`, dan `harga_total` dari tabel "transaksi" dimana `jumlah_barang` lebih dari 0.

Hasil dari perintah SELECT disimpan dalam variabel rows menggunakan metode fetchall() untuk mengambil semua baris hasil query.

Selanjutnya, dilakukan pengecekan apakah terdapat data pembelian dalam rows. Jika ada, maka data akan ditampilkan dalam format tabel dengan menggunakan perulangan for. Setiap baris data ditampilkan dengan menggunakan fungsi print() dan dipisahkan dengan tab (\t). Jika tidak ada data pembelian, akan ditampilkan pesan "Belum ada data pembelian".

```
1  def menu_pembayaran(total_harga):
2      print("Menu Pembayaran")
3      print("1. Tunai")
4      print("2. Kartu Debit")
5      print("3. E-Wallet")
6      metode_pembayaran = input("Pilih metode pembayaran (1/2/3): ")
7      if metode_pembayaran == "1":
8          uang_tunai = int(input("Masukkan jumlah uang tunai: "))
9          if uang_tunai >= total_harga:
10             kembalian = uang_tunai - total_harga
11             print("Kembalian: ", kembalian)
12             print("Terima kasih!")
13         else:
14             print("Uang tunai kurang")
15     elif metode_pembayaran == "2":
16         print("Pembayaran dengan kartu debit transaksi berhasil")
17     elif metode_pembayaran == "3":
18         print("Pembayaran dengan e-wallet transaksi berhasil")
19     else:
20         print("Pilihan tidak valid")
```

Fungsi menu\_pembayaran(total\_harga) digunakan untuk memungkinkan pengguna memilih metode pembayaran dan melakukan pembayaran berdasarkan total harga pembelian.

Fungsi ini mencetak menu pembayaran yang terdiri dari opsi "Tunai", "Kartu Kredit", dan "E-Wallet".

Pengguna diminta untuk memilih metode pembayaran melalui input.

Berdasarkan pilihan pengguna, dilakukan pengecekan menggunakan struktur kondisional if-elif-else untuk menentukan tindakan selanjutnya.

```
1 print("Selamat Datang")
2 print("1. Login sebagai admin")
3 print("2. Login sebagai kasir")
4 pilihan_login = input("Masukkan pilihan login (1/2): ")
```

Kode ini mencetak pesan "Selamat Datang" dan opsi untuk melakukan login sebagai admin atau kasir.

Pengguna diminta untuk memilih opsi login melalui input.

Pilihan login yang dimasukkan oleh pengguna disimpan dalam variabel `pilihan_login` menggunakan fungsi `input()`.

```
1 if pilihan_login == "1":
2     while True:
3         print("Menu CRUD")
4         print("1. Tambah barang")
5         print("2. Lihat barang")
6         print("3. Ubah barang")
7         print("4. Hapus barang")
8         print("5. Data Pembelian")
9         print("6. Logout")
10        pilihan_menu = input("Masukkan pilihan menu (1-6): ")
```

Pada baris pertama, dilakukan pengecekan apakah `pilihan_login` sama dengan "1". Jika benar, maka blok kode di dalamnya akan dieksekusi.

Selanjutnya, terdapat sebuah loop `while True`, yang berarti loop akan terus berjalan selama kondisinya benar (`True`).

Di dalam loop, dicetak menu CRUD dan opsi-opsinya, kemudian pengguna diminta untuk memasukkan pilihan menu melalui input.

```

1  if pilihan_menu == "1":
2      nama_barang = input("Masukkan nama barang: ")
3      kode_barang = input("Masukkan kode barang: ")
4      harga_barang = input("Masukkan harga barang: ")
5      jumlah_barang = input("Masukkan jumlah barang: ")
6      harga_total = int(harga_barang) * int(jumlah_barang)
7      cursor.execute("INSERT INTO barang (kode_barang, nama_barang, harga_barang, jumlah_barang, harga_total) VALUES (?, ?, ?, ?, ?)", (kode_barang, nama_barang, harga_barang, jumlah_barang, harga_total))
8
9      print("Data barang berhasil ditambahkan")

```

Pada baris pertama, dilakukan pengecekan apakah pilihan\_menu sama dengan "1". Jika benar, maka blok kode di dalamnya akan dieksekusi.

Setelah mendapatkan input dari pengguna untuk setiap kolom barang (nama, kode, harga, jumlah, dan harga total), menghitung harga total dengan mengalikan harga barang dan jumlah barang.

Menggunakan fungsi cursor.execute() untuk menjalankan perintah SQL untuk menambahkan data ke tabel "barang" dalam database.

Menggunakan fungsi conn.commit() untuk menyimpan perubahan ke database.

Menampilkan pesan "Data barang berhasil ditambahkan" sebagai konfirmasi kepengguna.

```

1  elif pilihan_menu == "2":
2      cursor.execute("SELECT * FROM barang")
3      rows = cursor.fetchall()
4      if rows:
5          print("Kode Barang\tNama Barang\tHarga Barang\tJumlah Barang\tHarga Total")
6          for row in rows:
7              print(row[0], "\t\t", row[1], "\t\t", row[2], "\t\t", row[3], "\t\t", row[4])
8      else:
9          print("Belum ada data barang")

```

Pada blok kode ini, terdapat kondisi elif yang mengecek apakah pilihan\_menu sama dengan "2". Jika benar, maka blok kode di dalamnya akan dieksekusi.

Selanjutnya, dilakukan eksekusi perintah SQL SELECT \* FROM barang menggunakan objek cursor untuk mengambil semua data dari tabel "barang".

Hasil dari perintah SELECT disimpan dalam variabel rows menggunakan metode fetchall() untuk mengambil semua baris hasil query.

Selanjutnya, dilakukan pengecekan apakah terdapat data barang dalam rows. Jika ada, maka data akan ditampilkan dalam format yang sesuai dengan kolom-kolom yang ada. Jika tidak ada data barang, akan dicetak pesan "Belum ada data barang".

```
1 elif pilihan_menu == "3":
2     kode_barang = input("Masukkan kode barang yang ingin diubah: ")
3     cursor.execute("SELECT * FROM barang WHERE kode_barang = ?", (kode_barang,))
4     row = cursor.fetchone()
5     if row:
6         print("Data Barang Lama:")
7         print("Kode Barang:", row[0])
8         print("Nama Barang:", row[1])
9         print("Harga Barang:", row[2])
10        print("Jumlah Barang:", row[3])
11        print("Harga Total:", row[4])
12        nama_barang = input("Masukkan nama barang baru: ")
13        harga_barang = input("Masukkan harga barang baru: ")
14        jumlah_barang = input("Masukkan jumlah barang baru: ")
15        harga_total = int(harga_barang) * int(jumlah_barang)
16        cursor.execute("UPDATE barang SET nama_barang = ?, harga_barang = ?, jumlah_barang = ?, harga_total = ? WHERE kode_barang = ?", (nama_barang, harga_barang, jumlah_barang, harga_total, kode_barang))
17        conn.commit()
18        print("Data barang berhasil diubah")
19    else:
20        print("Data barang tidak ditemukan")
```

Pada blok kode ini, terdapat kondisi elif yang mengecek apakah pilihan\_menu sama dengan "3". Jika benar, maka blok kode di dalamnya akan dieksekusi.

Selanjutnya, pengguna diminta untuk memasukkan kode barang yang ingin diubah melalui input.

Dilakukan eksekusi perintah SQL SELECT untuk mengambil data barang yang sesuai dengan kode barang yang dimasukkan oleh pengguna.

Hasil dari perintah SELECT disimpan dalam variabel row menggunakan metode fetchone() untuk mengambil satu baris hasil query.

Selanjutnya, dilakukan pengecekan apakah terdapat data barang dalam row. Jika ada, maka data akan ditampilkan dalam format yang sesuai dengan kolom-kolom yang ada. Pengguna diminta untuk memasukkan data barang baru melalui input.

Menggunakan fungsi cursor.execute() untuk menjalankan perintah SQL untuk mengubah data barang dalam tabel "barang" dalam database.

Menggunakan fungsi conn.commit() untuk menyimpan perubahan ke database.

Menampilkan pesan "Data barang berhasil diubah" sebagai konfirmasi kepengguna.

```

1 elif pilihan_menu == "4":
2     kode_barang = input("Masukkan kode barang yang ingin dihapus: ")
3     cursor.execute("DELETE FROM barang WHERE kode_barang = ?", (kode_barang,))
4     conn.commit()
5     print("Data barang berhasil dihapus")
6 elif pilihan_menu == "5":
7     lihat_pembelian()
8 elif pilihan_menu == "6":
9     print("Anda telah logout")
10    while True:
11        pilihan_kembali = input("Kembali ke menu CRUD? (y/n): ")
12        if pilihan_kembali == "y":
13            break
14        elif pilihan_kembali == "n":
15            while True:
16                pilihan_logout = input("Logout? (y/n): ")
17                if pilihan_logout == "y":
18                    print("Anda telah logout")
19                    break
20                elif pilihan_logout == "n":
21                    break
22            break
23        else:
24            print("Pilihan tidak valid")
25    break
26 else:
27    print("Pilihan tidak valid")

```

elif pilihan\_menu == "4"::: Jika pilihan menu adalah 4, maka akan dijalankan kode ini.

kode\_barang = input("Masukkan kode barang yang ingin dihapus: "): Mengambil input kode barang yang ingin dihapus dari pengguna.

cursor.execute("DELETE FROM barang WHERE kode\_barang = ?", (kode\_barang,)):  
Menjalankan query SQL untuk menghapus data dengan kode barang yang diberikan.

conn.commit(): Menyimpan perubahan ke database.

print("Data barang berhasil dihapus"): Menampilkan pesan "Data barang berhasil dihapus" jika operasi hapus berhasil.

elif pilihan\_menu == "5"::: Jika pilihan menu adalah 5, maka akan dijalankan kode lihat\_pembelian().

elif pilihan\_menu == "6"::: Jika pilihan menu adalah 6, maka akan dijalankan kode ini.

print("Anda telah logout"): Menampilkan pesan "Anda telah logout".

while True:: Melakukan loop sampai pengguna memilih y atau n untuk kembali ke menu CRUD atau logout.

pilihan\_kembali = input("Kembali ke menu CRUD? (y/n): "): Mengambil input dari pengguna untuk kembali ke menu CRUD atau logout.

if pilihan\_kembali == "y":: Jika pilihan kembali adalah y, maka akan keluar dari loop dan kembali ke menu CRUD.

elif pilihan\_kembali == "n": Jika pilihan kembali adalah n, maka akan dijalankan kode di dalam loop ini.

while True:: Melakukan loop sampai pengguna memilih y atau n untuk logout.

pilihan\_logout = input("Logout? (y/n): "): Mengambil input dari pengguna untuk logout.

if pilihan\_logout == "y":: Jika pilihan logout adalah y, maka akan keluar dari loop dan kembali ke menu CRUD.

elif pilihan\_logout == "n": Jika pilihan logout adalah n, maka akan kembali ke menu CRUD.

break: Keluar dari loop sampai logout atau kembali ke menu CRUD.

else: print("Pilihan tidak valid"): Jika pilihan tidak valid (yaitu tidak ada input y atau input tidak valid), maka menampilkan pesan ini.

else: print("Pilihan tidak valid"): Jika tidak ada pilihan menu yang valid (1, 2, 3, 4, 5, atau 6), maka menampilkan pesan ini.

```
elif pilihan_login == "2":
    keranjang = []
    while True:
        kode_barang = input("Masukkan kode barang: ")
        barang = get_barang(kode_barang)
        if barang:
            nama_barang = barang["nama_barang"]
            harga_barang = barang["harga_barang"]
            print("Nama Barang:", nama_barang)
            print("Harga Barang:", harga_barang)
            jumlah_barang = int(input("Masukkan jumlah barang yang ingin dibeli: "))
            harga_total = harga_barang * jumlah_barang
            print("Harga Total:", harga_total)
            keranjang.append({"nama_barang": nama_barang, "harga_barang": harga_barang, "jumlah_barang": jumlah_barang})
            cursor.execute("INSERT INTO transaksi (kode_barang, nama_barang, harga_barang, jumlah_barang, conn.commit())")
        else:
            print("Kode barang tidak ditemukan")
    pilihan_tambah = input("Tambah barang? (y/n): ")
    if pilihan_tambah == "n":
        break
```

Pada bagian ini, jika pengguna memilih login sebagai "kasir" (pilihan "2"), maka program akan menjalankan kode di dalamnya.

Sebuah list kosong dengan nama keranjang dibuat untuk menyimpan barang-barang yang akan dibeli.

Selama loop while True, pengguna diminta untuk memasukkan kode barang.

Fungsi get\_barang(kode\_barang) digunakan untuk mendapatkan informasi tentang barang berdasarkan kode yang dimasukkan pengguna.

Jika barang dengan kode tersebut ditemukan, program akan menampilkan informasi tentang barang tersebut (nama dan harga), dan meminta pengguna untuk memasukkan jumlah barang yang ingin dibeli.

Setelah itu, program akan menghitung harga total untuk barang tersebut, dan menambahkannya ke dalam keranjang.

Selain itu, program juga akan menambahkan informasi transaksi ke dalam database menggunakan perintah SQL INSERT.

Jika barang dengan kode tersebut tidak ditemukan, program akan menampilkan pesan "Kode barang tidak ditemukan".

Setelah itu, pengguna akan ditanya apakah ingin menambahkan barang lain ke dalam keranjang. Jika pengguna memilih "n", loop akan dihentikan.

```
1 print("Nama Barang\tHarga Satuan\tJumlah\tHarga Total")
2 total_harga = 0
3 for barang in keranjang:
4     print(barang["nama_barang"], "\t\t", barang["harga_barang"], "\t\t", barang["jumlah_barang"], "\t\t", barang["harga_total"])
5     total_harga += barang["harga_total"]
6 print("Total Harga:", total_harga)
7 menu_pembayaran(total_harga)
8 conn.close()
```

print("Nama Barang\tHarga Satuan\tJumlah\tHarga Total"): Baris ini mencetak header untuk daftar barang yang akan ditampilkan.

total\_harga = 0: Inisialisasi variabel total\_harga dengan nilai 0 untuk menyimpan total harga dari seluruh barang yang akan ditampilkan.

for barang in keranjang:: Melakukan iterasi untuk setiap barang dalam keranjang.



`print(barang["nama_barang"], "\t\t", barang["harga_barang"], "\t\t", barang["jumlah_barang"], "\t\t", barang["harga_total"]):` Baris ini mencetak detail dari setiap barang seperti nama barang, harga satuan, jumlah, dan harga total.

`total_harga += barang["harga_total"]:` Menambahkan `harga_total` dari setiap barang ke dalam `total_harga`.

`print("Total Harga:", total_harga):` Mencetak total harga dari seluruh barang yang ada dalam keranjang.

`menu_pembayaran(total_harga):` Memanggil fungsi `menu_pembayaran` dengan parameter `total_harga` untuk menampilkan menu pembayaran.

`conn.close():` Menutup koneksi ke database setelah semua operasi selesai.

## CODE

```
import sqlite3

conn = sqlite3.connect("db_kasir.db")

cursor = conn.cursor()

cursor.execute("CREATE TABLE IF NOT EXISTS barang (kode_barang text, nama_barang text, harga_barang number, jumlah_barang number, harga_total number)")

cursor.execute("CREATE TABLE IF NOT EXISTS transaksi (kode_barang text, nama_barang text, harga_barang number, jumlah_barang number, harga_total number)")

def get_barang(kode_barang):

    cursor.execute("SELECT * FROM barang WHERE kode_barang = ?", (kode_barang,))

    row = cursor.fetchone()

    if row:

        return {"nama_barang": row[1], "harga_barang": row[2]}

    else:

        return None

def lihat_pembelian():

    cursor.execute("SELECT kode_barang, nama_barang, harga_barang, jumlah_barang, harga_total FROM transaksi WHERE jumlah_barang > 0")
```

```

rows = cursor.fetchall()
if rows:
    print("Data Pembelian Pelanggan")
    print("Kode Barang\tNama Barang\tHarga Barang\tJumlah Barang\tHarga Total")
    for row in rows:
        print(row[0], "\t\t", row[1], "\t\t", row[2], "\t\t", row[3], "\t\t", row[4])
else:
    print("Belum ada data pembelian")

```

```

def menu_pembayaran(total_harga):
    print("Menu Pembayaran")
    print("1. Tunai")
    print("2. Kartu Kredit")
    print("3. E-Wallet")
    metode_pembayaran = input("Pilih metode pembayaran (1/2/3): ")
    if metode_pembayaran == "1":
        uang_tunai = int(input("Masukkan jumlah uang tunai: "))
        if uang_tunai >= total_harga:
            kembalian = uang_tunai - total_harga
            print("Kembalian: ", kembalian)
            print("Terima kasih!")
        else:
            print("Uang tunai kurang")
    elif metode_pembayaran == "2":
        print("Pembayaran dengan kartu kredit")
    elif metode_pembayaran == "3":
        print("Pembayaran dengan e-wallet")
    else:

```

```

        print("Pilihan tidak valid")

print("Selamat Datang")

print("1. Login sebagai admin")
print("2. Login sebagai kasir")

pilihan_login = input("Masukkan pilihan login (1/2): ")

if pilihan_login == "1":
    while True:
        print("Menu CRUD")
        print("1. Tambah barang")
        print("2. Lihat barang")
        print("3. Ubah barang")
        print("4. Hapus barang")
        print("5. Data Pembelian")
        print("6. Logout")

        pilihan_menu = input("Masukkan pilihan menu (1-6): ")

        if pilihan_menu == "1":
            nama_barang = input("Masukkan nama barang: ")
            kode_barang = input("Masukkan kode barang: ")
            harga_barang = input("Masukkan harga barang: ")
            jumlah_barang = input("Masukkan jumlah barang: ")
            harga_total = int(harga_barang) * int(jumlah_barang)

            cursor.execute("INSERT INTO barang (kode_barang, nama_barang, harga_barang,
            jumlah_barang, harga_total) VALUES (?, ?, ?, ?, ?)", (kode_barang, nama_barang,
            harga_barang, jumlah_barang, harga_total))

            conn.commit()

            print("Data barang berhasil ditambahkan")

        elif pilihan_menu == "2":
            cursor.execute("SELECT * FROM barang")

            rows = cursor.fetchall()

```

```

if rows:

    print("Kode Barang\tNama Barang\tHarga Barang\tJumlah Barang\tHarga Total")

    for row in rows:

        print(row[0], "\t\t", row[1], "\t\t", row[2], "\t\t", row[3], "\t\t", row[4])

    else:

        print("Belum ada data barang")

elif pilihan_menu == "3":

    kode_barang = input("Masukkan kode barang yang ingin diubah: ")

    cursor.execute("SELECT * FROM barang WHERE kode_barang = ?", (kode_barang,))

    row = cursor.fetchone()

    if row:

        print("Data Barang Lama:")

        print("Kode Barang:", row[0])

        print("Nama Barang:", row[1])

        print("Harga Barang:", row[2])

        print("Jumlah Barang:", row[3])

        print("Harga Total:", row[4])

        nama_barang = input("Masukkan nama barang baru: ")

        harga_barang = input("Masukkan harga barang baru: ")

        jumlah_barang = input("Masukkan jumlah barang baru: ")

        harga_total = int(harga_barang) * int(jumlah_barang)

        cursor.execute("UPDATE barang SET nama_barang = ?, harga_barang = ?,
jumlah_barang = ?, harga_total = ? WHERE kode_barang = ?", (nama_barang, harga_barang,
jumlah_barang, harga_total, kode_barang))

        conn.commit()

        print("Data barang berhasil diubah")

    else:

        print("Data barang tidak ditemukan")

elif pilihan_menu == "4":

```

```

kode_barang = input("Masukkan kode barang yang ingin dihapus: ")
cursor.execute("DELETE FROM barang WHERE kode_barang = ?", (kode_barang,))
conn.commit()

print("Data barang berhasil dihapus")
elif pilihan_menu == "5":
    lihat_pembelian()
elif pilihan_menu == "6":
    print("Anda telah logout")
    while True:
        pilihan_kembali = input("Kembali ke menu CRUD? (y/n): ")
        if pilihan_kembali == "y":
            break
        elif pilihan_kembali == "n":
            while True:
                pilihan_logout = input("Logout? (y/n): ")
                if pilihan_logout == "y":
                    print("Anda telah logout")
                    break
                elif pilihan_logout == "n":
                    break
            break
        else:
            print("Pilihan tidak valid")
            break
    else:
        print("Pilihan tidak valid")
elif pilihan_login == "2":
    keranjang = []

```

```

while True:

    kode_barang = input("Masukkan kode barang: ")

    barang = get_barang(kode_barang)

    if barang:

        nama_barang = barang["nama_barang"]
        harga_barang = barang["harga_barang"]
        print("Nama Barang:", nama_barang)
        print("Harga Barang:", harga_barang)
        jumlah_barang = int(input("Masukkan jumlah barang yang ingin dibeli: "))
        harga_total = harga_barang * jumlah_barang
        print("Harga Total:", harga_total)

        keranjang.append({"nama_barang": nama_barang, "harga_barang": harga_barang,
"jumlah_barang": jumlah_barang, "harga_total": harga_total})

        cursor.execute("INSERT INTO transaksi (kode_barang, nama_barang, harga_barang,
jumlah_barang, harga_total) VALUES (?, ?, ?, ?, ?)", (kode_barang, nama_barang,
harga_barang, jumlah_barang, harga_total))

        conn.commit()

    else:

        print("Kode barang tidak ditemukan")

        pilihan_tambah = input("Tambah barang? (y/n): ")

        if pilihan_tambah == "n":

            break

print("Nama Barang\tHarga Satuan\tJumlah\tHarga Total")

total_harga = 0

for barang in keranjang:

    print(barang["nama_barang"], "\t\t", barang["harga_barang"], "\t\t",
barang["jumlah_barang"], "\t\t", barang["harga_total"])

    total_harga += barang["harga_total"]

```

```
print("Total Harga:", total_harga)

menu_pembayaran(total_harga)

conn.close()
```

## OUTPUT

```
Selamat Datang
1. Login sebagai admin
2. Login sebagai kasir
Masukkan pilihan login (1/2):
```

Pada gambar diatas adalah tampilan utama untuk memilih user. Jika ingin login sebagai admin maka diminta untuk mengetikan 1 dan jika ingin login sebagai kasir maka diminta untuk mengetikan 2.

```
Masukkan pilihan login (1/2): 1
Menu CRUD
1. Tambah barang
2. Lihat barang
3. Ubah barang
4. Hapus barang
5. Data Pembelian
6. Logout
Masukkan pilihan menu (1-6):
```

Pada gambar diatas adalah tampilan jika memilih login sebagai admin. Pada tampilan admin itu tersedia beberapa menu, yaitu:

1. untuk menambahkan barang
2. untuk melihat barang
3. untuk mengubah barang
4. untuk menghapus barang
5. untuk melihat data pembelian
6. untuk keluar dari user admin

```
Masukkan pilihan menu (1-6): 1
Masukkan nama barang: Sabun
Masukkan kode barang: S1010
Masukkan harga barang: 25000
Masukkan jumlah barang: 35
Data barang berhasil ditambahkan
```

Gambar diatas adalah tampilan ketika admin memilih menu 1, yaitu untuk menambahkan barang. Pada tampilan diatas admin menambahkan nama barang, kode barang, harga barang, lalu menambahkan jumlah barang. Setelah selesai, maka barang yang sudah ditambahkan akan masuk ke menu 2

```

Menu CRUD
1. Tambah barang
2. Lihat barang
3. Ubah barang
4. Hapus barang
5. Data Pembelian
6. Logout
Masukkan pilihan menu (1-6): 2
Kode Barang      Nama Barang      Harga Barang      Jumlah Barang      Harga Total
S1010            Sabun            25000             35                 875000

```

Gambar diatas adalah tampilan ketika admin memilih menu 2, yaitu untuk melihat barang yang sudah ditambahkan tadi. Dapat dilihat barang yang sudah ditambahkan tadi sudah tersedia.

```

Masukkan pilihan menu (1-6): 3
Masukkan kode barang yang ingin diubah: S1010
Data Barang Lama:
Kode Barang: S1010
Nama Barang: Sabun
Harga Barang: 25000
Jumlah Barang: 35
Harga Total: 875000
Masukkan nama barang baru: Tissue
Masukkan harga barang baru: 20000
Masukkan jumlah barang baru: 30
Data barang berhasil diubah
Menu CRUD
1. Tambah barang
2. Lihat barang
3. Ubah barang
4. Hapus barang
5. Data Pembelian
6. Logout
Masukkan pilihan menu (1-6): 2
Kode Barang      Nama Barang      Harga Barang      Jumlah Barang      Harga Total
S1010            Tissue          20000             30                 600000

```

Gambar diatas adalah tampilan ketika admin memilih menu 3. Yaitu untuk mengubah suatu barang yang sudah ditambahkan sebelumnya. Disini admin mengubah barang yang sebelumnya Sabun lalu diubah menjadi Tissue. Dan jika ingin melihat barang yang sudah diubah tadi admin perlu masuk ke menu 2 untuk melihat barang yang baru.



```

Menu CRUD
1. Tambah barang
2. Lihat barang
3. Ubah barang
4. Hapus barang
5. Data Pembelian
6. Logout
Masukkan pilihan menu (1-6): 4
Masukkan kode barang yang ingin dihapus: SR2020
Data barang berhasil dihapus

```

Gambar diatas adalah tampilan ketika admin memilih menu 4, yaitu untuk menghapus barang yang tersedia. Akan tetapi sebelum menghapus barang, diperlukan untuk masuk ke menu 2 untuk melihat barang apa yang tersedia, setelah tahu barang apa yang tersedia. Maka admin bisa langsung masuk kembali ke menu 4 untuk menghapus barang.

```

Masukkan pilihan menu (1-6): 2
Kode Barang    Nama Barang    Harga Barang    Jumlah Barang    Harga Total
S1010         Tissue         20000           30               600000
SR2020         Sari Roti      20000           20               400000
Menu CRUD
1. Tambah barang
2. Lihat barang
3. Ubah barang
4. Hapus barang
5. Data Pembelian
6. Logout
Masukkan pilihan menu (1-6): 4
Masukkan kode barang yang ingin dihapus: SR2020
Data barang berhasil dihapus
Menu CRUD
1. Tambah barang
2. Lihat barang
3. Ubah barang
4. Hapus barang
5. Data Pembelian
6. Logout
Masukkan pilihan menu (1-6): 2
Kode Barang    Nama Barang    Harga Barang    Jumlah Barang    Harga Total
S1010         Tissue         20000           30               600000

```

Seperti gambar di atas ini, pertama admin melihat barang yang tersedia, yaitu dengan masuk ke menu 2. Dan setelah itu admin masuk ke menu 4 yaitu untuk menghapus barang, di menu 4 admin diminta untuk memasukkan kode barang yang ingin di hapus. Lalu setelah admin memasukkan kode barang tersebut, maka admin dapat kembali ke menu 2 untuk melihat kembali apakah barang tersebut sudah terhapus atau belum.

```

Menu CRUD
1. Tambah barang
2. Lihat barang
3. Ubah barang
4. Hapus barang
5. Data Pembelian
6. Logout
Masukkan pilihan menu (1-6): 5
Data Pembelian Pelanggan

```

Kode Barang	Nama Barang	Harga Barang	Jumlah Barang	Harga Total
B1001	sampo	15000	2	30000
B1000	sabun	20000	1	20000
B1000	sabun	20000	1	20000
B1000	sabun	20000	1	20000
B1000	sabun	5000	5	25000
S1010	Tissue	20000	5	100000

Gambar diatas adalah tampilan ketika admin memilih menu 5, yaitu untuk melihat data pembelian/transaksi pelanggan. Terlihat diatas bahwa ada data pembelian/transaksi.

```

Menu CRUD
1. Tambah barang
2. Lihat barang
3. Ubah barang
4. Hapus barang
5. Data Pembelian
6. Logout
Masukkan pilihan menu (1-6): 6
Anda telah logout
Kembali ke menu CRUD? (y/n): n
Logout? (y/n): y
Anda telah logout

```

Gambar diatas adalah tampilan ketika admin memilih menu 6, yaitu untuk keluar dari user admin. Setelah admin mengetikan perintah n maka admin sudah keluar dari menu CRUD.

```
Selamat Datang
1. Login sebagai admin
2. Login sebagai kasir
Masukkan pilihan login (1/2): 2
Masukkan kode barang: S1010
Nama Barang: Tissue
Harga Barang: 20000
Masukkan jumlah barang yang ingin dibeli: 5
Harga Total: 100000
Tambah barang? (y/n): n
Nama Barang      Harga Satuan    Jumlah  Harga Total
Tissue           20000          5       100000
Total Harga: 100000
Menu Pembayaran
1. Tunai
2. Kartu Debit
3. E-Wallet
Pilih metode pembayaran (1/2/3): 1
Masukkan jumlah uang tunai: 100000
Kembalian: 0
Terima kasih!
```

Gambar diatas adalah tampilan ketika login sebagai kasir. Jika login sebagai kasir, maka disini kasir diminta untuk memasukan kode barang, jumlah barang yang ingin dibeli oleh pelanggan/pembeli. Lalu kasir di minta untuk memasukan metode pembayaran yang dipilih oleh pelanggan/pembeli.

## PEMBAGIAN TUGAS

Reza Zidan Hanafi	(2307421014) : Source code dan printscreen.
Kerta Wijaya	(2307421003) : Flowchart.
Dara Samsara Ayu	(2307421013) : Algoritma.
Nabiel Ischak	(2307421012) : Dokumentasi penjelasan.
Pandora Ophelia Krstanto	(2307421018) : Tujuan dan Fitur.
Wanda Rizki Hamidah	(2307421006) : Latar Belakang dan Problem.

## DOKUMENTASI FOTO



