**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 |

**LINK SUMBER**

https://repository.nusamandiri.ac.id/repo/files/225635/download/Modul-Dasar-Pemrograman-2018---Program-Kasir.pdf

**PEMBAGIAN TUGAS**

Reza Zidan Hanafi (2307421014) : Source code (2), flowchart dan printscreen .

Kerta Wijaya (2307421003) : Flowchart, Source code (2).

Dara Samsara Ayu (2307421013) : Algoritma, source code(3).

Nabiel Ischak (2307421012) : Dokumentasi penjelasan dan problem .

Pandora Ophelia Krstanto (2307421018) : Tujuan, fitur dan source code (3).

Wanda Rizki Hamidah (2307421006) : Latar Belakang, Algoritma dan source code(1).

**LATAR BELAKANG**

Program kasir cake adalah program yang dibuat untuk membantu proses transaksi penjualan pada sebuah toko kue. Program ini dibuat menggunakan bahasa pemrograman Python dan menggunakan modul datetime dan sqlite3. Program ini memiliki beberapa fitur seperti login kasir, riwayat transaksi, lihat daftar harga, dan exit.

**PROBLEM YANG INGIN DIATASI**

Program kasir cake dibuat untuk mengatasi masalah dalam proses transaksi penjualan pada sebuah toko kue yang masih menggunakan sistem manual. Dengan adanya program ini, diharapkan proses transaksi penjualan menjadi lebih efisien dan akurat.

**BATASAN MASALAH**

Ada dua Batasan masalah

1. Ukuran dari kue, di referensi kami ada penjelasan dari ukuran kue dari small, medium, dan large, tapi kita tidak menggunakannya

2. Menyimpan riwayat transaksi dengan struktur sendiri karna di jurnal tidak dijelaskan dengan detail, maka kita simpan dengan menggunakan sqlite.

**TUJUAN**

Tujuan dari pembuatan program kasir cake adalah untuk membantu proses transaksi penjualan pada sebuah toko kue menjadi lebih efisien dan akurat. Program ini juga dapat membantu dalam mengelola riwayat transaksi dan daftar harga produk yang dijual.

**FITUR**

Program kasir sebuah cake shop/Toko Kue yang meliputi:

1. Login kasir yang mengharuskan kasir memasukkan kode kasirnya
2. Transaksi cake yang akan menentukan harga cake, jumlah pembayaran hingga kembalian
3. Melihat riwayat transaksi, dimana riwayat transaksi dapat dilihat, atau dihapus
4. Melihat daftar harga cake

**ALGORITMA**

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

**Algoritma/Bahasa Manusia:**

1. Program dimulai dengan menampilkan menu utama, yang berisi opsi untuk login kasir, melihat riwayat transaksi, melihat daftar harga, dan keluar.
2. Jika pengguna memilih nomor 1, maka akan diminta untuk memasukkan kode kasir. Berdasarkan kode kasir yang dimasukkan, program akan menentukan nama kasir.
3. Setelah login, program akan menampilkan daftar barang yang tersedia. Pengguna dapat memasukkan kode barang dan jumlah yang ingin dibeli. Setiap transaksi akan disimpan dalam list transaksi.
4. Setelah selesai berbelanja, program akan menampilkan detail transaksi, termasuk nama toko, alamat, nama kasir, waktu, daftar barang yang dibeli beserta harganya, total harga, dan pilihan untuk membayar tunai.
5. Jika pengguna memilih nomor 2, program akan menampilkan riwayat transaksi dari database, termasuk detail transaksi, total transaksi, total penjualan, dan total uang yang diterima.
6. Jika pengguna memilih nomor 3, program akan menampilkan daftar harga menu kopi berdasarkan daftar barang yang tersedia.
7. Jika pengguna memilih nomor 4, program akan keluar.
8. Jika pengguna memasukkan nomor selain 1, 2, 3, atau 4, program akan menampilkan pesan "Nomor tidak valid".

**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) menunjukan 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:

A black and white diagram

Description automatically generated

**CODE DAN PENJELASAN**

A screenshot of a computer

Description automatically generated

Kode yang diberikan mengimpor dua modul, yaitu datetime dan sqlite3. Selain itu, kode tersebut juga mendefinisikan sebuah kamus (dictionary) daftar\_barang yang berisi informasi tentang berbagai kue, seperti nama dan harganya. Kunci (key) dari kamus tersebut adalah kode produk untuk setiap kue. Tujuan dari kode ini tidak jelas tanpa konteks tambahan. Modul datetime menyediakan kelas untuk bekerja dengan tanggal dan waktu, sementara modul sqlite3 menyediakan cara untuk bekerja dengan basis data SQLite di Python

A screen shot of a computer program

Description automatically generated

Navigasi: Menampilkan pesan "Navigasi = Menu utama" untuk menunjukkan bahwa kode ini adalah bagian dari menu utama.

Nama toko dan alamat toko: Menyimpan nama toko dan alamat toko dalam variabel nama\_toko dan alamat\_toko, lalu menampilkan pesan yang mengandung nama toko dan alamat toko.

Waktu: Mengambil waktu saat ini menggunakan modul datetime dan menyimpan waktu tersebut dalam variabel waktu. Lalu, menampilkan pesan yang mengandung waktu dalam format "dd/mm/yyyy HH:MM:SS".

List: Menampilkan daftar menu utama yang terdiri dari beberapa pilihan, seperti "Login kasir", "Riwayat transaksi", "Lihat daftar harga", dan "Exit".

A screen shot of a computer program

Description automatically generated

Kode ini meminta pengguna untuk memasukkan nomor menu, dan jika nomor menu yang dimasukkan adalah "1", maka program akan meminta pengguna untuk memasukkan kode kasir. Jika kode kasir yang dimasukkan adalah "001", program akan menyimpan nama kasir sebagai "Reza Zidan". Jika kode kasir yang dimasukkan adalah "002", program akan menyimpan nama kasir sebagai "Kerta Wijaya". Jika kode kasir yang dimasukkan tidak sesuai, program akan menyimpan nama kasir sebagai "Kasir Tidak Dikenal", menampilkannya, dan keluar dari program.

A screen shot of a computer program

Description automatically generated

Kode ini membuat sebuah list kosong bernama transaksi dan memulai sebuah loop while True yang akan berjalan sampai kode kasir memasukkan "x" untuk batal. Dalam loop tersebut, kode barang yang dimasukkan oleh kasir akan diinputkan dan diperiksa apakah ada dalam daftar barang. Jika kode barang ada dalam daftar barang, maka program akan menampilkan nama barang dan harga, dan meminta kasir untuk memasukkan jumlah barang yang akan dibeli. Jumlah barang yang dimasukkan akan disimpan dalam list transaksi dalam bentuk dictionary yang mengandung kode barang, nama barang, harga satuan, dan jumlah.

Kode ini menggunakan modul datetime untuk mengambil waktu saat ini dan menyimpan waktu tersebut dalam variabel waktu. Lalu, menampilkan pesan yang mengandung waktu dalam format "dd/mm/yyyy HH:MM:SS".

A computer screen shot of a code

Description automatically generated

Kode ini adalah kode Python yang mencetak receipt. Pertama, kode mencetak nama toko, alamat toko, nama kasir, dan waktu transaksi. Kemudian, kode mencetak tabel dengan kolom "Nama Barang", "Ukuran", "QTY", "Harga", dan "Total Harga". Selanjutnya, kode mencetak garis bawah yang terdiri dari 60 karakter "-" untuk membatasi tabel dari kolom lain. Kode ini menggunakan f-strings untuk memformat output dengan spasi dan aligment yang sesuai.

A screen shot of a computer program

Description automatically generated

Mengambil data dari transaksi: Program mengambil data dari sebuah transaksi yang disimpan dalam sebuah list dengan nama transaksi. Setiap item dalam list berisi informasi tentang nama barang, jumlah yang dibeli, harga satuan, dan total harga item.

Menghitung total harga item: Dalam perulangan, program menghitung total harga item dengan mengkali harga satuan dengan jumlah yang dibeli. Hasilnya disimpan dalam variabel total\_harga\_item.

Menampilkan data transaksi: Program menampilkan data transaksi dengan menggunakan format tabel. Nama barang, jumlah, harga satuan, dan total harga item ditampilkan dalam kolom yang sesuai.

Menghitung total harga transaksi: Setelah menghitung total harga item, program menghitung total harga transaksi dengan menambahkan total harga item dari setiap item ke variabel total\_harga.

Menghitung kembalian: Program meminta pembeli untuk memasukkan uang tunai yang akan dibayar. Kemudian, program menghitung kembalian dengan mengurangi total harga dari uang tunai yang dibayar.

Menyimpan data transaksi ke database: Program menggunakan SQLite untuk membuat tabel transactions jika belum ada. Setelah itu, program menyimpan data transaksi ke dalam tabel, termasuk tanggal transaksi, nama kasir, nama barang, jumlah yang dibeli, harga satuan, dan total harga transaksi.

Menampilkan kembalian: Program menampilkan kembalian yang akan dikembalikan ke pembeli.

Menampilkan pesan terima kasih: Program menampilkan pesan terima kasih kepada pembeli.

A screen shot of a computer

Description automatically generated

Mendapatkan tanggal dan waktu saat ini: Baris pertama kode mengambil tanggal dan waktu saat ini dalam format tertentu menggunakan modul waktu dan metode strftime. Hasilnya disimpan dalam variabel date.

Iterasi setiap item dalam transaksi: Selanjutnya, program melakukan iterasi untuk setiap item dalam list transaksi.

Mengambil informasi item transaksi: Untuk setiap item, program mengambil informasi seperti nama barang, jumlah, harga satuan, dan menghitung total harga dari item tersebut.

Memasukkan data ke dalam database: Setelah mendapatkan informasi item transaksi, program memasukkan data tersebut ke dalam tabel transactions dalam database SQLite menggunakan perintah conn.execute. Data yang dimasukkan meliputi tanggal, nama kasir, nama barang, jumlah, harga satuan, dan total harga.

Mengkomit perubahan dan menutup koneksi: Setelah semua data transaksi dimasukkan, program melakukan commit untuk menyimpan perubahan ke dalam database dan menutup koneksi ke database.

A screen shot of a computer program

Description automatically generated

Memeriksa nomor menu: Program memeriksa nomor menu yang dipilih oleh pengguna. Jika nomor menu adalah "2", maka program akan menampilkan riwayat transaksi.

Menampilkan informasi toko: Program menampilkan informasi toko seperti nama dan alamat.

Menampilkan waktu saat ini: Program menampilkan waktu saat ini dalam format tertentu menggunakan modul waktu dan metode strftime.

Menampilkan riwayat transaksi: Program menampilkan riwayat transaksi dengan mengambil data dari tabel transactions dalam database SQLite menggunakan perintah SELECT. Data yang diambil meliputi tanggal, nama kasir, nama barang, jumlah, harga satuan, dan total harga. Hasilnya disimpan dalam variabel transactions.

Menghitung total transaksi: Setelah mendapatkan data transaksi, program menghitung total transaksi dengan menjumlahkan total harga dari setiap transaksi. Hasilnya disimpan dalam variabel total\_transaksi.

Menghitung total penjualan: Program juga menghitung total penjualan dengan menghitung jumlah barang yang terjual. Hasilnya disimpan dalam variabel total\_penjualan.

Menghitung total uang yang diterima: Program menghitung total uang yang diterima dari seluruh transaksi. Hasilnya disimpan dalam variabel total\_uang\_diterima.

A screen shot of a computer program

Description automatically generated

Iterasi setiap transaksi: Program melakukan iterasi untuk setiap transaksi dalam variabel transactions.

Mengambil informasi transaksi: Untuk setiap transaksi, program mengambil informasi seperti tanggal, nama kasir, nama barang, jumlah, harga satuan, dan total harga.

Menampilkan informasi transaksi: Program menampilkan informasi transaksi dengan menggunakan format tabel. Nama kasir, tanggal, nama barang, jumlah, harga satuan, dan total harga ditampilkan dalam kolom yang sesuai.

Menghitung jumlah harga total: Program menghitung jumlah harga total dari setiap transaksi dengan menjumlahkan total harga dari setiap item.

Menghitung jumlah bayar: Program menghitung jumlah bayar dari setiap transaksi dengan menjumlahkan total harga dari setiap item.

Menampilkan jumlah harga total dan jumlah bayar: Program menampilkan jumlah harga total dan jumlah bayar dari setiap transaksi.

Menampilkan baris kosong: Program menampilkan baris kosong untuk memisahkan informasi transaksi dari informasi transaksi berikutnya.

A computer screen shot of a program

Description automatically generated

Menambahkan transaksi: Program menambahkan transaksi ke dalam variabel transactions.

Menambahkan total transaksi: Program menambahkan 1 ke variabel total\_transaksi untuk menghitung total transaksi.

Menambahkan total penjualan: Program menambahkan total harga dari transaksi ke variabel total\_penjualan untuk menghitung total penjualan.

Menambahkan total uang yang diterima: Program menambahkan total harga dari transaksi ke variabel total\_uang\_diterima untuk menghitung total uang yang diterima.

Menampilkan total transaksi, total penjualan, dan total uang yang diterima: Program menampilkan total transaksi, total penjualan, dan total uang yang diterima dengan menggunakan format tabel.

Menampilkan baris kosong: Program menampilkan baris kosong untuk memisahkan informasi transaksi dari informasi transaksi berikutnya.

A screen shot of a computer code

Description automatically generated

Memeriksa nomor menu: Program memeriksa nomor menu yang dipilih oleh pengguna. Jika nomor menu adalah "3", maka program akan menampilkan daftar harga menu kue.

Menampilkan daftar harga menu kue: Program menampilkan daftar harga menu kue dengan menggunakan format tabel. Kode barang, nama barang, dan harga ditampilkan dalam kolom yang sesuai.

Keluar dari program: Jika pengguna memilih nomor menu "4", maka program akan keluar.

Menampilkan pesan kesalahan: Jika pengguna memilih nomor menu selain "3" dan "4", maka program akan menampilkan pesan "Nomor tidak valid".

**CODE**

import datetime

import sqlite3

daftar\_barang = {

"B001": {"nama": "Black Forest", "harga": 110000},

"B002": {"nama": "Cheese Cake", "harga": 200000},

"B003": {"nama": "Choco Brownies", "harga": 160000},

"B004": {"nama": "Fruit Cake", "harga": 180000},

"B005": {"nama": "Greentea Cake", "harga": 160000},

"B006": {"nama": "Lapis Surabaya", "harga": 160000},

"B007": {"nama": "Oreo Cake", "harga": 160000},

"B008": {"nama": "Rainbow Cake", "harga": 160000},

"B009": {"nama": "Red Velvet Cake", "harga": 180000},

"B0010": {"nama": "Tiramisu Cake", "harga": 160000},

}

print("Navigasi = Menu utama\n")

nama\_toko = "Toko Cake"

alamat\_toko = "Jl. Sudirman No. 123"

print(f"{nama\_toko}\n{alamat\_toko}\n")

waktu = datetime.datetime.now()

print(f"Waktu: {waktu.strftime('%d/%m/%Y %H:%M:%S')}\n")

print("List:")

print("1. Login kasir")

print("2. Riwayat transaksi")

print("3. Lihat daftar harga")

print("4. Exit")

nomor\_menu = input("Masukkan nomor menu yang dipilih: ")

if nomor\_menu == "1":

kode\_kasir = input("Masukkan kode kasir: ")

if kode\_kasir == "001":

nama\_kasir = "Reza Zidan"

elif kode\_kasir == "002":

nama\_kasir = "Kerta Wijaya"

else:

nama\_kasir = "Kasir Tidak Dikenal"

print(f"\nNavigasi = Menu Utama -> Login Kasir -> Transaksi\n")

input("Tekan enter untuk memulai transaksi...")

transaksi = []

while True:

kode\_barang = input("Masukkan kode barang (klik enter untuk selesai, x untuk batal): ")

if kode\_barang.lower() == "x":

break

elif kode\_barang in daftar\_barang:

print(f"{daftar\_barang[kode\_barang]['nama']} - Rp {daftar\_barang[kode\_barang]['harga']}")

# Input jumlah barang

jumlah = int(input("Masukkan jumlah: "))

transaksi.append({"kode\_barang": kode\_barang, "nama\_barang": daftar\_barang[kode\_barang]['nama'], "harga\_satuan": daftar\_barang[kode\_barang]['harga'], "jumlah": jumlah})

print(f"\n{'Nama Toko':<15} {nama\_toko}")

print(f"{'Alamat':<15} {alamat\_toko}\n")

print(f"{'Nama Kasir':<15} {nama\_kasir}")

print(f"{'Waktu':<15} {waktu.strftime('%d/%m/%Y %H:%M:%S')}\n")

print(f"{'Nama Barang':<20} {'Ukuran':<10} {'QTY':<10} {'Harga':<10} {'Total Harga':<10}")

print("-" \* 60)

total\_harga = 0

for item in transaksi:

total\_harga\_item = item['harga\_satuan'] \* item['jumlah']

print(f"{item['nama\_barang']:<20} {'':<10} {item['jumlah']:<10} {item['harga\_satuan']:<10} {total\_harga\_item:<10}")

total\_harga += total\_harga\_item

print("-" \* 60)

print(f"{'Jumlah Harga':<50} {total\_harga:>10}")

print("-" \* 60)

uang\_tunai = int(input("Bayar tunai: "))

kembalian = uang\_tunai - total\_harga

print("-" \* 60)

print(f"{'Kembalian':<50} {kembalian:>10}")

print("-" \* 60)

print("Terima kasih telah berbelanja di Toko Cake!")

conn = sqlite3.connect('transactions.db')

conn.execute('''CREATE TABLE IF NOT EXISTS transactions

(id INTEGER PRIMARY KEY AUTOINCREMENT,

date TEXT NOT NULL,

cashier TEXT NOT NULL,

item TEXT NOT NULL,

quantity INTEGER NOT NULL,

price INTEGER NOT NULL,

total INTEGER NOT NULL);''')

date = waktu.strftime('%d/%m/%Y %H:%M:%S')

for item in transaksi:

item\_name = item['nama\_barang']

quantity = item['jumlah']

price = item['harga\_satuan']

total = item['harga\_satuan'] \* item['jumlah']

conn.execute(f"INSERT INTO transactions (date, cashier, item, quantity, price, total) VALUES ('{date}', '{nama\_kasir}', '{item\_name}', {quantity}, {price}, {total})")

conn.commit()

conn.close()

elif nomor\_menu == "2":

print("\nNavigasi = Menu Utama -> Riwayat transaksi -> Lihat riwayat transaksi\n")

print(f"{nama\_toko}\n{alamat\_toko}\n")

print(f"Waktu: {waktu.strftime('%d/%m/%Y %H:%M:%S')}\n")

print("Riwayat transaksi")

print("-" \* 7)

conn = sqlite3.connect('transactions.db')

cursor = conn.execute("SELECT date, cashier, item, quantity, price, total FROM transactions")

transactions = cursor.fetchall()

total\_transaksi = 0

total\_penjualan = 0

total\_uang\_diterima = 0

for transaction in transactions:

date = transaction[0]

cashier = transaction[1]

item = transaction[2]

quantity = transaction[3]

price = transaction[4]

total = transaction[5]

print("\n")

print("-" \* 9)

print(f"{'':<1}{'nama kasir:':<20}{cashier}")

print(f"{'':<1}{'waktu:':<20}{date}")

print("\n")

print(f"{'':<1}{'nama barang':<20}{'QTY':<10}{'harga satuan':<15}{'total harga':<15}")

print(f"{'':<1}{'-----':<20}{'----':<10}{'------------':<15}{'-----------':<15}")

print(f"{'':<1}{item:<20}{quantity:<10}{price:<15}{total:<15}")

print(f"{'':<1}{'jumlah harga total:':<50}{total:<10}")

print(f"{'':<1}{'jumlah bayar:':<50}{total:<10}")

print(f"{'':<1}{'':<50}{'':<10}")

total\_transaksi += 1

total\_penjualan += total

total\_uang\_diterima += total

print("-" \* 7)

print(f"{'':<1}{'Total transaksi:':<50}{total\_transaksi:<10}")

print(f"{'':<1}{'Total penjualan:':<50}{total\_penjualan:<10}")

print(f"{'':<1}{'Total uang yang diterima:':<50}{total\_uang\_diterima:<10}")

print("-" \* 7)

elif nomor\_menu == "3":

print("Daftar Harga Menu Kue")

print(f"Kode Barang {'':<8} | Nama Barang {'':<18} | Harga")

for kode, barang in daftar\_barang.items():

print(f"{kode:<20} | {barang['nama']:<30} | {barang['harga']}")

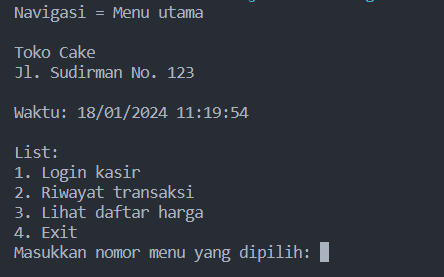
elif nomor\_menu == "4":

exit()

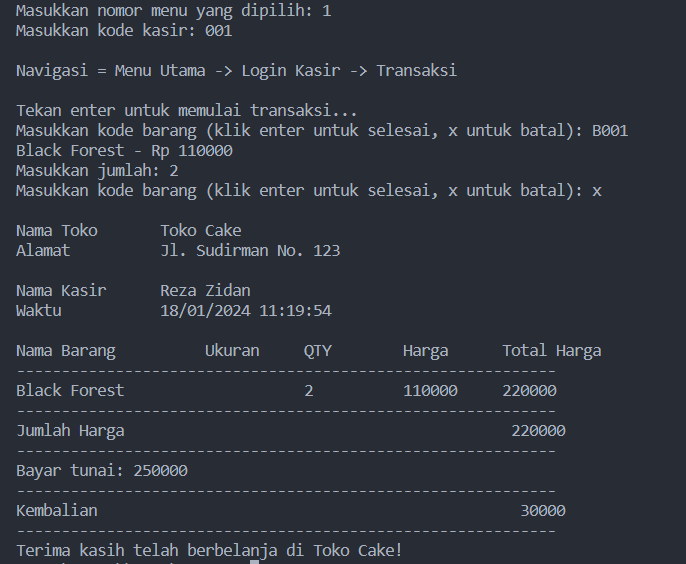
else:

print("Nomor tidak valid")

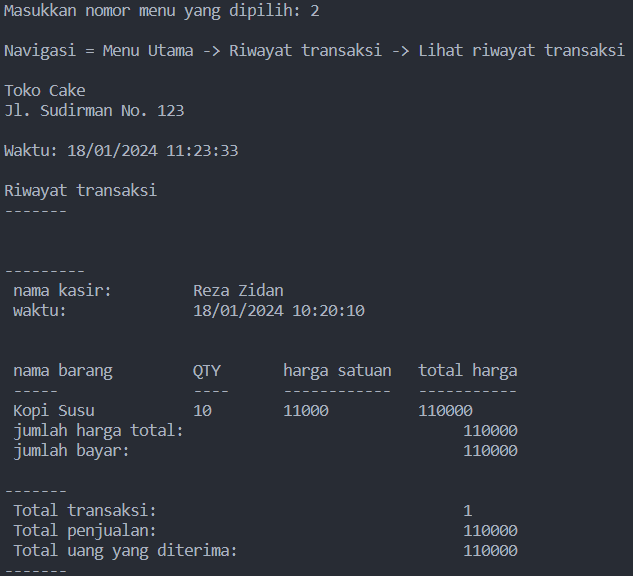
**OUTPUT**



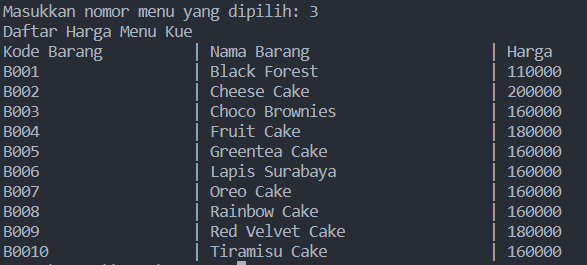
Pada gambar diatas adalah …



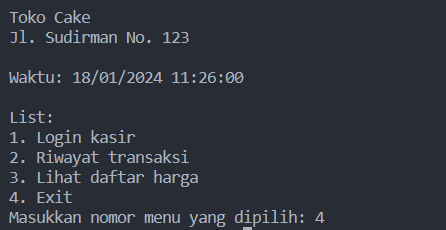
Pada gambar diatas adalah …



Gambar diatas adalah tampilan …



Gambar diatas adalah tampilan …



Gambar diatas adalah tampilan ketika …

**DOKUMENTASI FOTO**

****

