**UAS BAHASA PEMOGRAMAN KJ002**



**Dosen Pengampu Mata Kuliah :**

**RATNA YULIKA GO , S.Kom., M.T.I.**

**Nama dan NIM Anggota Kelompok :**

**Filbert Chiuman 20220801051**

**Vincent Etwin Mangapul 20220801267**

**Falkutas Ilmu Komputer**

**Teknik Informatika**

**Universitas Esa Unggul Jakarta Kebon Jeruk**

**Tahun 2024**

# **DAFTAR ISI**

**1. LIST REVISI ATAU PERMASALAHAN**………………....……..……..

A.) AKSESIBILITAS……………………………………………………... 3

B.) NOTIFIKASI ERROR………………………………………………... 3

C.) CATATAN SEMUA TRANSAKSI…………………………………… 3

D.) PENAMBAHAN HAPUS STOCK…………………………………... 3

E.) LOOPING…………………………………………………………….. 3

**2. TANGGAPAN ATAS PERMASALAHAN**…………………………..…

A.) AKSESIBILITAS…………………………..…………………………. 4

B.) NOTIFIKASI ERROR………………………………………………... 4

C.) CATATAN SEMUA TRANSAKSI………………………………...…. 4

D.) PENAMBAHAN HAPUS STOCK …………………………….....…. 4

E.) LOOPING…………………………………………………………….. 4

**3. PERUBAHAN PADA PROGRAM**…………………………………..…

A.) AKSESIBILITAS…………………………..…………………………. 5

B.) NOTIFIKASI ERROR………………………………………………... 7

C.) CATATAN SEMUA TRANSAKSI………………………………...…. 10

D.) PENAMBAHAN HAPUS STOCK …………………………….....…. 11

E.) LOOPING…………………………………………………………….. 12

# **1.) LIST REVISI ATAU PERMASALAHAN**

**A.) AKSESIBILITAS**

Permasalahan aksesibilitas sebelumnya termasuk dalam antarmuka pengguna yang tidak ramah, seperti:

- Tidak adanya opsi untuk kembali ke menu sebelumnya pada beberapa prompt.

- Penanganan input yang terbatas, seperti hanya menerima input numerik atau teks tanpa validasi yang tepat.

**B.) NOTIFIKASI ERROR**

Tidak ada pemberitahuan yang jelas ketika terjadi kesalahan input atau stok barang tidak mencukupi.

**C.) CATATAN SEMUA TRANSAKSI**

Sistem sebelumnya hanya mencetak struk ke satu file (struk.txt), dan tidak ada sistem untuk mencatat semua transaksi yang terjadi.

**D. PENAMBAHAN HAPUS STOCK**

Fitur untuk menghapus stok barang tidak ada dalam program sebelumnya.

**E. LOOPING**

Program sebelumnya tidak memberikan opsi untuk melanjutkan atau kembali pada beberapa loop, menyebabkan pengalaman pengguna yang kurang fleksibel.

# **2.) TANGGAPAN ATAS PERMASALAHAN**

**A.) AKSESIBILITAS**

Telah ditambahkan opsi untuk kembali pada beberapa prompt serta penanganan input yang lebih baik:

- Pada mode pemilik toko dan penjaga kasir, pengguna dapat mengetik "kembali" untuk kembali ke menu sebelumnya apabila typo dalam memilih fitur.

- Validasi input ditingkatkan untuk memastikan bahwa input numerik benar-benar numerik.

**B.) NOTIFIKASI ERROR**

Diperkenalkan pesan error yang lebih informatif untuk memberikan petunjuk yang jelas ketika terjadi kesalahan:

- Pesan kesalahan saat stok tidak mencukupi.

- Pemberitahuan saat input tidak valid atau pilihan tidak ada dalam daftar.

**C.) CATATAN SEMUA TRANSAKSI**

Setiap struk transaksi sekarang disimpan dengan nama file yang berbeda menggunakan counter (struk1.txt, struk2.txt, dll.). Juga, riwayat penjualan disimpan dalam file history.txt yang mencatat semua transaksi.

**D. PENAMBAHAN HAPUS STOCK**

Ditambahkan fitur untuk mengurangi stok barang tanpa menghapus barang dari inventaris, serta opsi untuk menghapus barang ketika stok mencapai nol.

**E. LOOPING**

Ditambahkan opsi untuk melanjutkan atau kembali pada beberapa loop untuk meningkatkan fleksibilitas pengguna:

- Opsi untuk menginput barang lagi, menghapus barang lagi, dan menambah item ke keranjang lagi setelah setiap tindakan.

# **3.) PERUBAHAN PADA PROGRAM**

**A.) AKSESIBILITAS**

Perubahan terkait aksesibilitas berfokus pada penambahan opsi untuk kembali ke menu sebelumnya.

**- Sebelumnya :**

def owner\_mode():

while True:

# ...

if choice == "1":

num\_items = int(input("Berapa banyak barang yang ingin diinput? "))

# ...

# ...

elif choice == "3":

item = input("Masukkan nama barang yang ingin dihapus: ")

# ...

def cashier\_mode():

while True:

# ...

if choice == "1":

item\_index = int(input("Masukkan nomor barang yang ingin ditambahkan: ")) - 1

# ...

**- Sesudahnya :**

def owner\_mode():

while True:

# ...

if choice == "1":

while True:

num\_items\_input = input("Berapa banyak barang yang ingin diinput? (atau ketik 'kembali' untuk kembali): ").strip()

if num\_items\_input.lower() == 'kembali':

break

try:

num\_items = int(num\_items\_input)

if num\_items < 0:

raise ValueError

for \_ in range(num\_items):

item = input("Masukkan nama barang: ").strip()

price = float(input("Masukkan harga barang: "))

stock = int(input("Masukkan stok barang: "))

inventory[item] = {"price": price, "stock": stock}

save\_inventory()

print("Barang-barang telah ditambahkan.")

except ValueError:

print("Input tidak valid. Silakan masukkan jumlah yang benar.")

continue\_input = input("Apakah Anda ingin menginput barang lagi? (ya/tidak): ").strip().lower()

if continue\_input != 'ya':

break

# ...

elif choice == "3":

while True:

item = input("Masukkan nama barang yang ingin dihapus (atau ketik 'kembali' untuk kembali): ").strip()

if item.lower() == 'kembali':

break

if item in inventory:

del inventory[item]

save\_inventory()

print("Barang telah dihapus.")

else:

print("Barang tidak ditemukan dalam inventaris.")

continue\_delete = input("Apakah Anda ingin menghapus barang lain? (ya/tidak): ").strip().lower()

if continue\_delete != 'ya':

break

def cashier\_mode():

while True:

# ...

if choice == "1":

while True:

print("\n-- Daftar Barang --")

for i, (item, details) in enumerate(inventory.items(), start=1):

print(f"{i}. {item}: Harga = Rp{details['price']}, Stok = {details['stock']}")

item\_index = input("Masukkan nomor barang yang ingin ditambahkan (atau ketik 'kembali' untuk kembali): ").strip()

if item\_index.lower() == 'kembali':

break

item\_index = int(item\_index) - 1

if 0 <= item\_index < len(inventory):

item = list(inventory.keys())[item\_index]

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

if quantity <= inventory[item]["stock"]:

inventory[item]["stock"] -= quantity

cart.append((item, quantity, inventory[item]["price"]))

print("Item telah ditambahkan ke keranjang.")

if inventory[item]["stock"] == 0:

print(f"Stok {item} telah habis, menghapus dari inventaris.")

del inventory[item]

save\_inventory()

else:

print("Stok tidak mencukupi.")

else:

print("Pilihan tidak valid.")

continue\_add = input("Apakah Anda ingin menambah item lain? (ya/tidak): ").strip().lower()

if continue\_add != 'ya':

break

# ...

elif choice == "4":

while True:

total = sum(quantity \* price for \_, quantity, price in cart)

ppn = total \* 0.1

total\_with\_ppn = total + ppn

print(f"Total yang harus dibayar (dengan PPN): Rp{total\_with\_ppn}")

amount = input("Masukkan nominal pembayaran (atau ketik 'kembali' untuk kembali): ").strip()

if amount.lower() == 'kembali':

break

amount = float(amount)

kembalian = amount - total\_with\_ppn

if amount >= total\_with\_ppn:

print("Pembayaran berhasil. Menunggu konfirmasi Penjaga Kasir.")

if confirm\_payment():

print("Pembayaran telah dikonfirmasi. Struk sedang dicetak...")

print\_receipt(cart, total, ppn, total\_with\_ppn, kembalian, amount)

cart.clear() # Clear the cart after successful payment

else:

print("Pembayaran ditolak oleh kasir toko.")

else:

print("Nominal pembayaran kurang. Pembayaran gagal.")

break

# ...

**B.) NOTIFIKASI ERROR**

Penambahan notifikasi error saat input tidak valid atau stok tidak mencukupi.

**- Sebelumnya :**

def owner\_mode():

while True:

# ...

if choice == "1":

num\_items = int(input("Berapa banyak barang yang ingin diinput? "))

# ...

# ...

elif choice == "3":

item = input("Masukkan nama barang yang ingin dihapus: ")

# ...

def cashier\_mode():

while True:

# ...

if choice == "1":

item\_index = int(input("Masukkan nomor barang yang ingin ditambahkan: ")) - 1

# ...

**- Sesudahnya :**

def owner\_mode():

while True:

# ...

if choice == "1":

while True:

num\_items\_input = input("Berapa banyak barang yang ingin diinput? (atau ketik 'kembali' untuk kembali): ").strip()

if num\_items\_input.lower() == 'kembali':

break

try:

num\_items = int(num\_items\_input)

if num\_items < 0:

raise ValueError

for \_ in range(num\_items):

item = input("Masukkan nama barang: ").strip()

price = float(input("Masukkan harga barang: "))

stock = int(input("Masukkan stok barang: "))

inventory[item] = {"price": price, "stock": stock}

save\_inventory()

print("Barang-barang telah ditambahkan.")

except ValueError:

print("Input tidak valid. Silakan masukkan jumlah yang benar.")

continue\_input = input("Apakah Anda ingin menginput barang lagi? (ya/tidak): ").strip().lower()

if continue\_input != 'ya':

break

# ...

elif choice == "3":

while True:

item = input("Masukkan nama barang yang ingin dihapus (atau ketik 'kembali' untuk kembali): ").strip()

if item.lower() == 'kembali':

break

if item in inventory:

del inventory[item]

save\_inventory()

print("Barang telah dihapus.")

else:

print("Barang tidak ditemukan dalam inventaris.")

continue\_delete = input("Apakah Anda ingin menghapus barang lain? (ya/tidak): ").strip().lower()

if continue\_delete != 'ya':

break

def cashier\_mode():

while True:

# ...

if choice == "1":

while True:

print("\n-- Daftar Barang --")

for i, (item, details) in enumerate(inventory.items(), start=1):

print(f"{i}. {item}: Harga = Rp{details['price']}, Stok = {details['stock']}")

item\_index = input("Masukkan nomor barang yang ingin ditambahkan (atau ketik 'kembali' untuk kembali): ").strip()

if item\_index.lower() == 'kembali':

break

item\_index = int(item\_index) - 1

if 0 <= item\_index < len(inventory):

item = list(inventory.keys())[item\_index]

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

if quantity <= inventory[item]["stock"]:

inventory[item]["stock"] -= quantity

cart.append((item, quantity, inventory[item]["price"]))

print("Item telah ditambahkan ke keranjang.")

if inventory[item]["stock"] == 0:

print(f"Stok {item} telah habis, menghapus dari inventaris.")

del inventory[item]

save\_inventory()

else:

print("Stok tidak mencukupi.")

else:

print("Pilihan tidak valid.")

continue\_add = input("Apakah Anda ingin menambah item lain? (ya/tidak): ").strip().lower()

if continue\_add != 'ya':

break

# ...

elif choice == "4":

while True:

total = sum(quantity \* price for \_, quantity, price in cart)

ppn = total \* 0.1

total\_with\_ppn = total + ppn

print(f"Total yang harus dibayar (dengan PPN): Rp{total\_with\_ppn}")

amount = input("Masukkan nominal pembayaran (atau ketik 'kembali' untuk kembali): ").strip()

if amount.lower() == 'kembali':

break

amount = float(amount)

kembalian = amount - total\_with\_ppn

if amount >= total\_with\_ppn:

print("Pembayaran berhasil. Menunggu konfirmasi Penjaga Kasir.")

if confirm\_payment():

print("Pembayaran telah dikonfirmasi. Struk sedang dicetak...")

print\_receipt(cart, total, ppn, total\_with\_ppn, kembalian, amount)

cart.clear() # Clear the cart after successful payment

else:

print("Pembayaran ditolak oleh kasir toko.")

else:

print("Nominal pembayaran kurang. Pembayaran gagal.")

break

# ...

**C.) CATATAN SEMUA TRANSAKSI**

Penambahan fitur untuk mencatat semua transaksi dalam file terpisah.

**- Sebelumnya :**

Kode sebelumnya tidak memiliki fungsi `save\_history`

**- Sesudahnya :**

def save\_history(receipt\_text):

history\_file = "history.txt"

with open(history\_file, "a") as file:

file.write(receipt\_text)

file.write("\n---\n")

def cashier\_mode():

while True:

# ...

elif choice == "4":

while True:

total = sum(quantity \* price for \_, quantity, price in cart)

ppn = total \* 0.1

total\_with\_ppn = total + ppn

print(f"Total yang harus dibayar (dengan PPN): Rp{total\_with\_ppn}")

amount = input("Masukkan nominal pembayaran (atau ketik 'kembali' untuk kembali): ").strip()

if amount.lower() == 'kembali':

break

amount = float(amount)

kembalian = amount - total\_with\_ppn

if amount >= total\_with\_ppn:

print("Pembayaran berhasil. Menunggu konfirmasi Penjaga Kasir.")

if confirm\_payment():

print("Pembayaran telah dikonfirmasi. Struk sedang dicetak...")

receipt\_text = generate\_receipt\_text(cart, total, ppn, total\_with\_ppn, kembalian, amount)

save\_receipt(receipt\_text)

save\_history(receipt\_text)

cart.clear() # Clear the cart after successful payment

else:

print("Pembayaran ditolak oleh kasir toko.")

else:

print("Nominal pembayaran kurang. Pembayaran gagal.")

break

# ...

**D. PENAMBAHAN HAPUS STOCK**

Penambahan fungsi untuk menghapus barang dari stok.

**- Sebelumnya :**

# Kode sebelumnya hanya memiliki opsi untuk menghapus item dalam `owner\_mode`

def owner\_mode():

while True:

# ...

elif choice == "3":

item = input("Masukkan nama barang yang ingin dihapus: ")

# ...

**- Sesudahnya :**

def owner\_mode():

while True:

# ...

elif choice == "3":

while True:

item = input("Masukkan nama barang yang ingin dihapus (atau ketik 'kembali' untuk kembali): ").strip()

if item.lower() == 'kembali':

break

if item in inventory:

del inventory[item]

save\_inventory()

print("Barang telah dihapus.")

else:

print("Barang tidak ditemukan dalam inventaris.")

continue\_delete = input("Apakah Anda ingin menghapus barang lain? (ya/tidak): ").strip().lower()

if continue\_delete != 'ya':

break

**E. LOOPING**

Penambahan loop untuk menambah dan menghapus barang serta konfirmasi pembayaran.

**- Sebelumnya :**

def owner\_mode():

while True:

# ...

if choice == "1":

num\_items = int(input("Berapa banyak barang yang ingin diinput? "))

# ...

# ...

elif choice == "3":

item = input("Masukkan nama barang yang ingin dihapus: ")

# ...

def cashier\_mode():

while True:

# ...

if choice == "1":

item\_index = int(input("Masukkan nomor barang yang ingin ditambahkan: ")) - 1

# ...

**- Sesudahnya :**

def owner\_mode():

while True:

# ...

if choice == "1":

while True:

num\_items\_input = input("Berapa banyak barang yang ingin diinput? (atau ketik 'kembali' untuk kembali): ").strip()

if num\_items\_input.lower() == 'kembali':

break

try:

num\_items = int(num\_items\_input)

if num\_items < 0:

raise ValueError

for \_ in range(num\_items):

item = input("Masukkan nama barang: ").strip()

price = float(input("Masukkan harga barang: "))

stock = int(input("Masukkan stok barang: "))

inventory[item] = {"price": price, "stock": stock}

save\_inventory()

print("Barang-barang telah ditambahkan.")

except ValueError:

print("Input tidak valid. Silakan masukkan jumlah yang benar.")

continue\_input = input("Apakah Anda ingin menginput barang lagi? (ya/tidak): ").strip().lower()

if continue\_input != 'ya':

break

# ...

elif choice == "3":

while True:

item = input("Masukkan nama barang yang ingin dihapus (atau ketik 'kembali' untuk kembali): ").strip()

if item.lower() == 'kembali':

break

if item in inventory:

del inventory[item]

save\_inventory()

print("Barang telah dihapus.")

else:

print("Barang tidak ditemukan dalam inventaris.")

continue\_delete = input("Apakah Anda ingin menghapus barang lain? (ya/tidak): ").strip().lower()

if continue\_delete != 'ya':

break

def cashier\_mode():

while True:

# ...

if choice == "1":

while True:

print("\n-- Daftar Barang --")

for i, (item, details) in enumerate(inventory.items(), start=1):

print(f"{i}. {item}: Harga = Rp{details['price']}, Stok = {details['stock']}")

item\_index = input("Masukkan nomor barang yang ingin ditambahkan (atau ketik 'kembali' untuk kembali): ").strip()

if item\_index.lower() == 'kembali':

break

item\_index = int(item\_index) - 1

if 0 <= item\_index < len(inventory):

item = list(inventory.keys())[item\_index]

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

if quantity <= inventory[item]["stock"]:

inventory[item]["stock"] -= quantity

cart.append((item, quantity, inventory[item]["price"]))

print("Item telah ditambahkan ke keranjang.")

if inventory[item]["stock"] == 0:

print(f"Stok {item} telah habis, menghapus dari inventaris.")

del inventory[item]

save\_inventory()

else:

print("Stok tidak mencukupi.")

else:

print("Pilihan tidak valid.")

continue\_add = input("Apakah Anda ingin menambah item lain? (ya/tidak): ").strip().lower()

if continue\_add != 'ya':

break

# ...

elif choice == "4":

while True:

total = sum(quantity \* price for \_, quantity, price in cart)

ppn = total \* 0.1

total\_with\_ppn = total + ppn

print(f"Total yang harus dibayar (dengan PPN): Rp{total\_with\_ppn}")

amount = input("Masukkan nominal pembayaran (atau ketik 'kembali' untuk kembali): ").strip()

if amount.lower() == 'kembali':

break

amount = float(amount)

kembalian = amount - total\_with\_ppn

if amount >= total\_with\_ppn:

print("Pembayaran berhasil. Menunggu konfirmasi Penjaga Kasir.")

if confirm\_payment():

print("Pembayaran telah dikonfirmasi. Struk sedang dicetak...")

receipt\_text = generate\_receipt\_text(cart, total, ppn, total\_with\_ppn, kembalian, amount)

save\_receipt(receipt\_text)

save\_history(receipt\_text)

cart.clear() # Clear the cart after successful payment

else:

print("Pembayaran ditolak oleh kasir toko.")

else:

print("Nominal pembayaran kurang. Pembayaran gagal.")

break

# ...