

LAPORAN PRAKTIKUM
POSTTEST 8
ALGORITMA PEMROGRAMAN DASAR



Disusun oleh:
Syarifah Anargya Rizky (2509106007)
Kelas A1'25

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

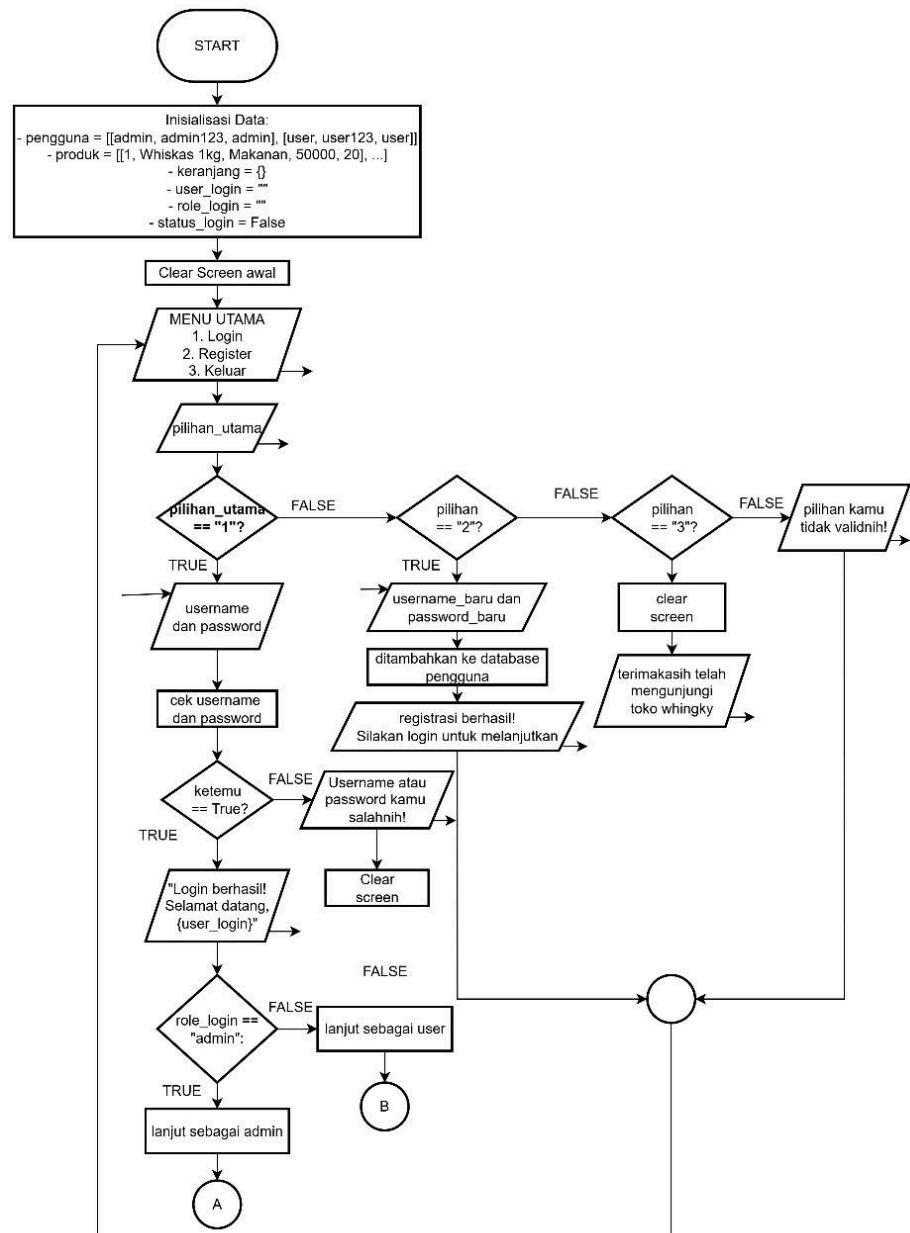
DAFTAR ISI

DAFTAR ISI.....	1
1. Flowchart.....	3
Gambar 1.1 Flowchart bagian awal program toko makanan kucing whinky.....	3
Gambar 1.2 Flowchart bagian on page A untuk program toko makanan kucing whinky...	4
Gambar 1.3 Flowchart bagian on page C untuk program toko makanan kucing whinky...	5
Gambar 1.4 Flowchart bagian on page D untuk program toko makanan kucing whinky...	6
Gambar 1.5 Flowchart bagian on page E untuk program toko makanan kucing whinky...	7
Gambar 1.6 Flowchart bagian on page F untuk program toko makanan kucing whinky....	8
Gambar 1.7 Flowchart bagian on page B untuk program toko makanan kucing whinky...	9
Gambar 1.8 Flowchart bagian on page G untuk program toko makanan kucing whinky.	10
Gambar 1.9 Flowchart bagian on page H untuk program toko makanan kucing whinky.	11
Gambar 1.10 Flowchart bagian on page I untuk program toko makanan kucing whinky	12
Gambar 1.11 Flowchart bagian on page J untuk program toko makanan kucing whinky	13
2. Deskripsi Singkat Program.....	14
3. Source Code.....	15
3.1 bagian Data.py.....	15
3.2 bagian Functions.py.....	16
3.3 bagian Main.py.....	19
4. Hasil Output.....	29
4.1 Sebagai Admin.....	29
Gambar 4.1.1 Ketika login sebagai admin di pilihan 1 (login).....	29
Gambar 4.1.2 tampilan layar bila jadi admin.....	29
Gambar 4.1.3 output pilihan 1 (melihat semua produk).....	30
Gambar 4.1.4 output pilihan 2 (menambahkan produk baru pada etalase toko).....	30
Gambar 4.1.5 output pilihan 1 (udah ditambahin si bedak kutu).....	30
Gambar 4.1.6 output pilihan 3 (jadi bisa memperbarui nama/ kategori/ harga/ stok)	31
Gambar 4.1.7 output pilihan 4 (admin dapat menghapus apapun yang ada di etalase)	31
Gambar 4.1.8 output pilihan 5 (admin bisa mencari produk lebih cepat dengan fitur ini).....	32
4.2 sebagai user.....	32
Gambar 4.2.1 output pilihan 2 sedari awal dengan tujuan daftar akun baru.....	32
Gambar 4.2.1.a output pilihan 2 sedari awal dengan tujuan daftar akun baru.....	32
Gambar 4.2.2 Ketika login sebagai user (kalau registrasi itu akunnya nanti jadi user) di pilihan 1 (login).....	33
Gambar 4.2.3 output pilihan 1 (melihat semua produk).....	33
Gambar 4.2.4 output pilihan 2 (fitur mencari produk).....	33
Gambar 4.2.5 output pilihan 3 (jadi user bisa nambahin barang kekeranjang sepuasnya *asalkan stoknya cukup yaa).....	34
Gambar 4.2.6 output pilihan 4 (lihat isi keranjang dan dihitungkan subtotal belanjanya berapa).....	34
Gambar 4.2.7 output pilihan 5 (fitur menghapus produk pilihan yang ada di keranjang).....	34

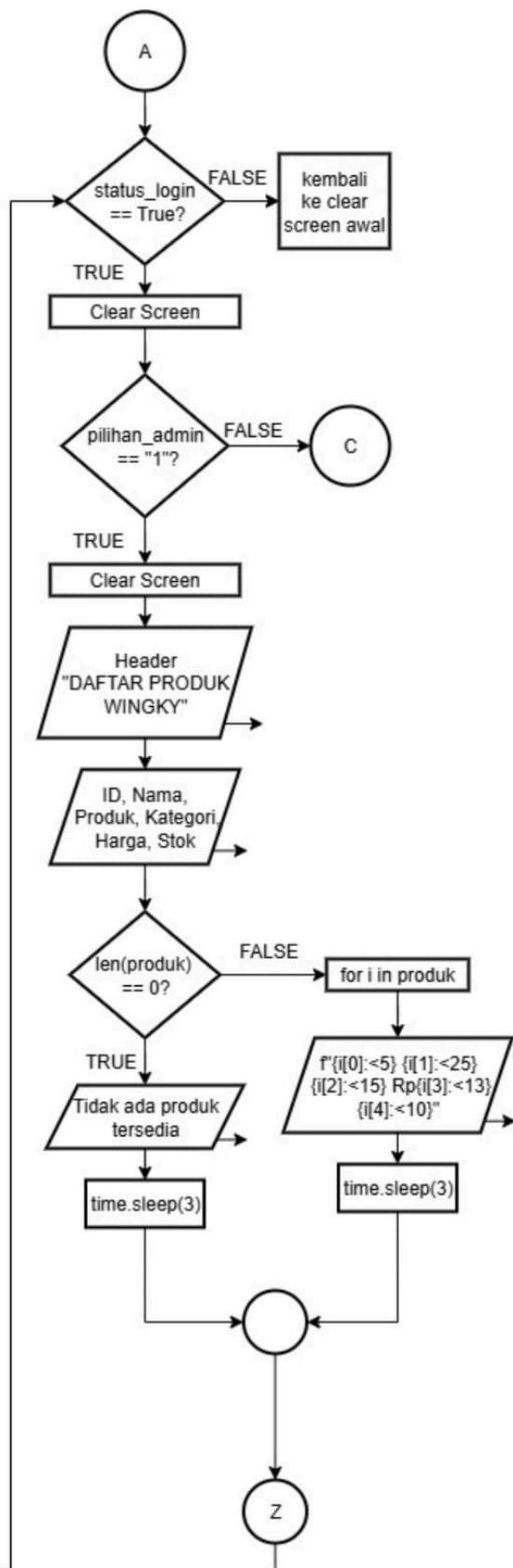
Gambar 4.2.8 output pilihan 6 (fitur checkout).....	35
Gambar 4.2.9 jika pengguna memilih pilihan 3 sejak awal.....	35
5. Langkah-langkah GIT.....	35
5.1 GIT Add.....	36
Gambar 5.1.1 Penggunaan Git Add.....	36
5.2 GIT Commit.....	36
Gambar 5.2.1 Penggunaan Git Commit.....	36
5.3 GIT Push.....	36
Gambar 5.3.1 penggunaan Git Push untuk menghubungkan repository lokal ke github.....	36

1. Flowchart

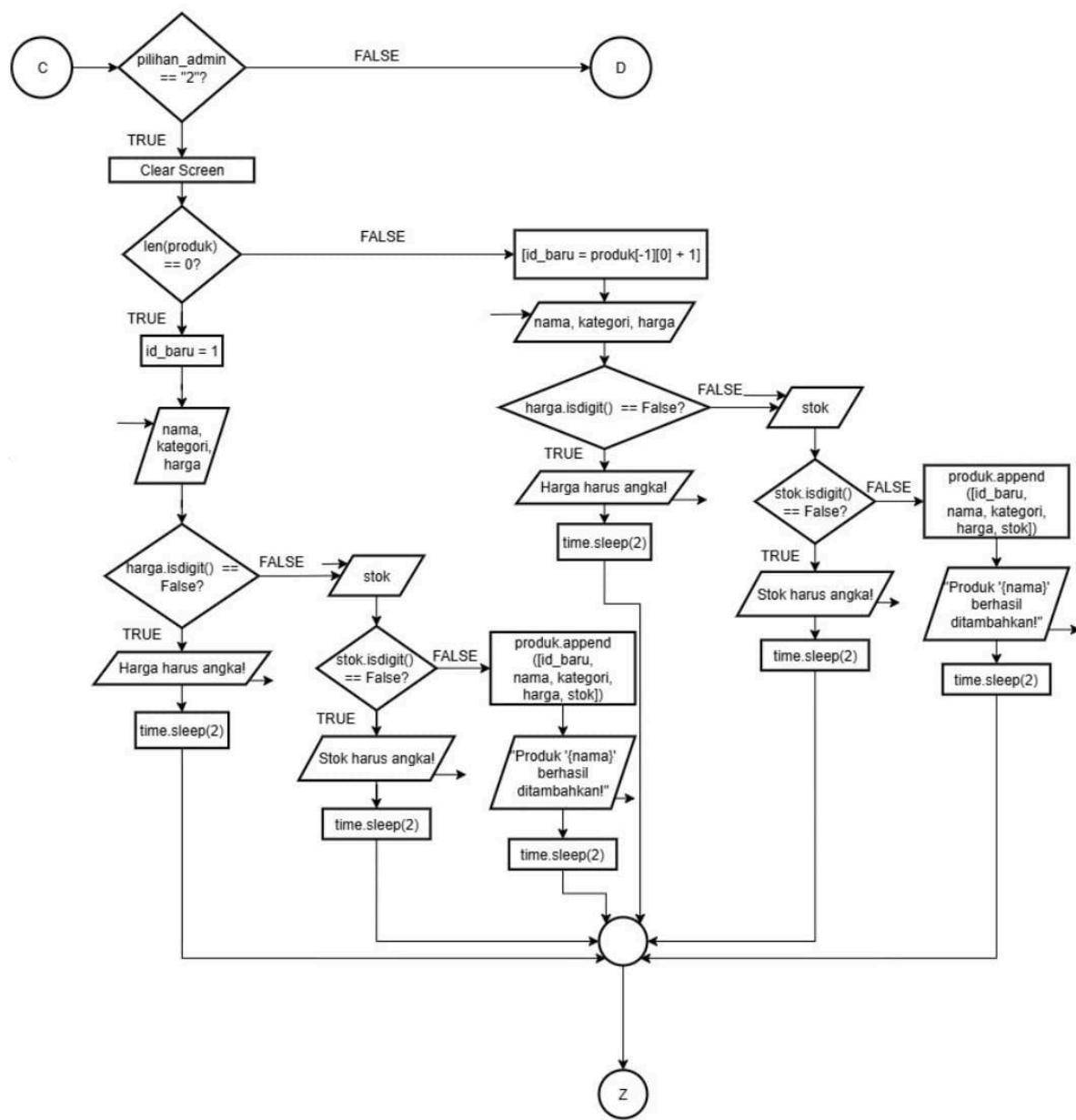
Flowchart merupakan diagram yang biasa digunakan dalam dunia informatika dengan tujuan agar dapat menggambarkan dengan lebih jelas dari suatu langkah-langkah, keputusan, dan alur kerja dari suatu proses dalam sistem. Flowchart terdiri dari berbagai simbol-simbol dengan makna tersendiri yang digunakan untuk menyederhanakan rangkaian prosedur agar memudahkan pemahaman terhadap informasi yang ingin disampaikan.



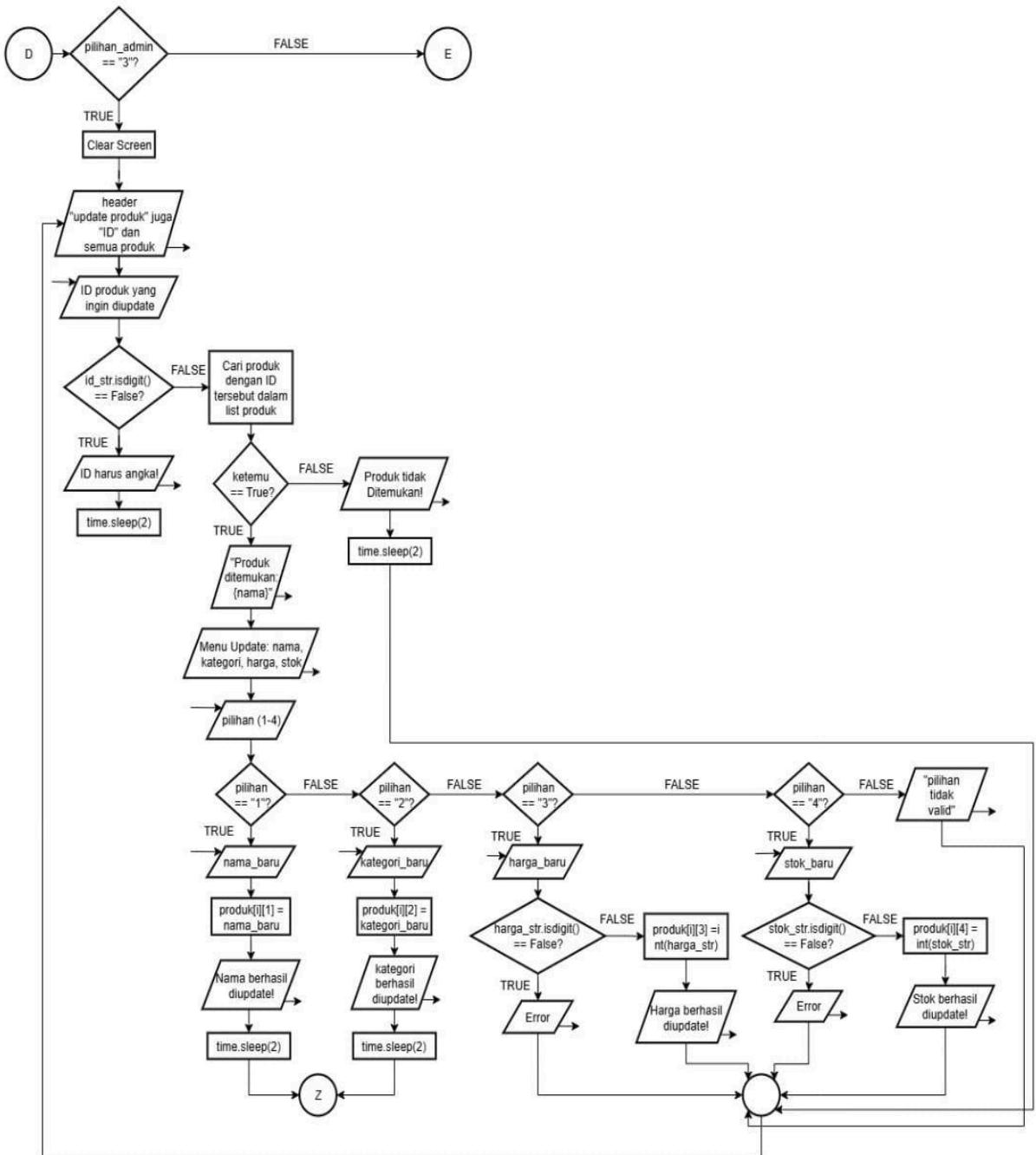
Gambar 1.1 Flowchart bagian awal program toko makanan kucing whinky



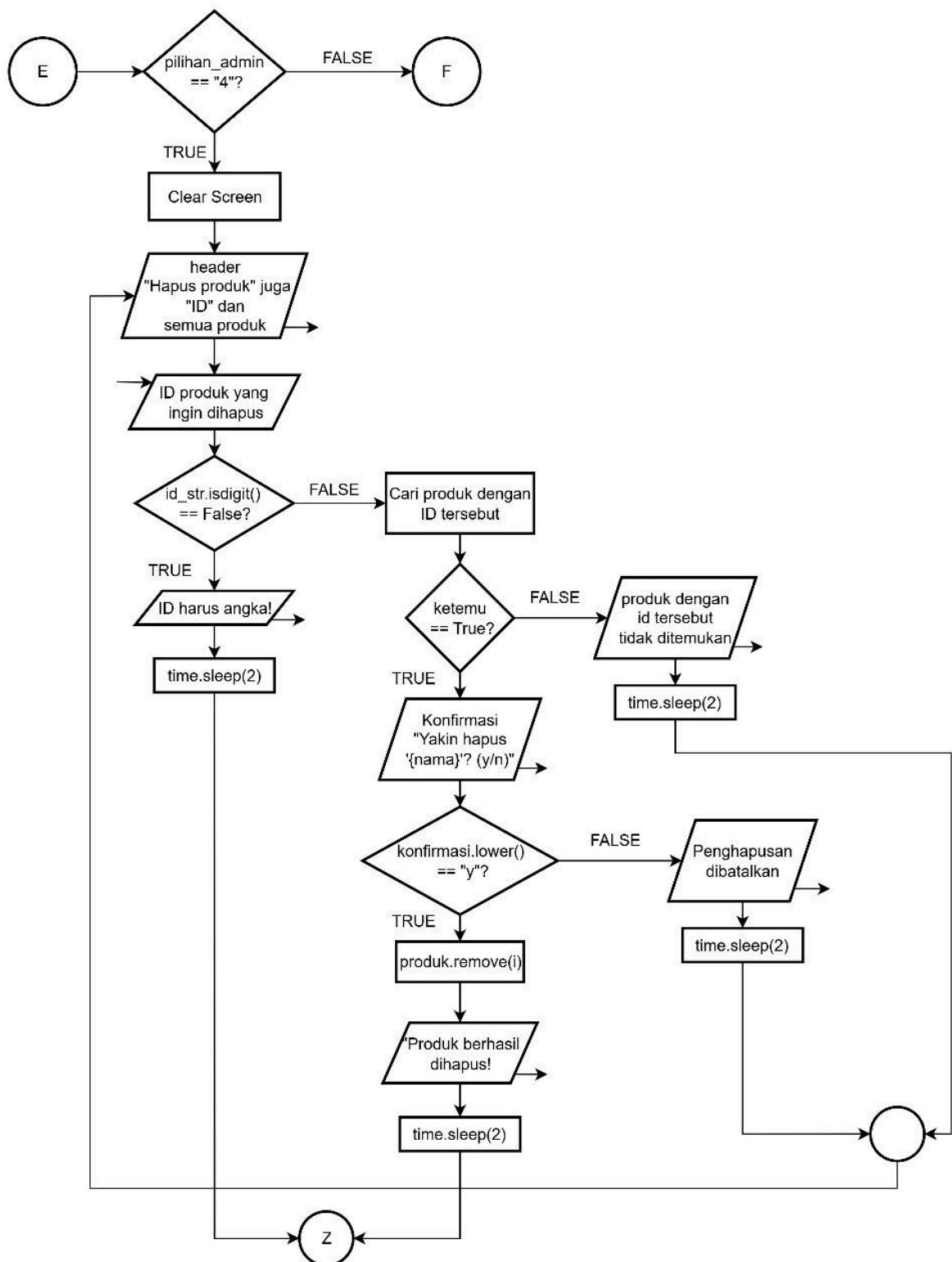
Gambar 1.2 Flowchart bagian on page A untuk program toko makanan kucing whinky



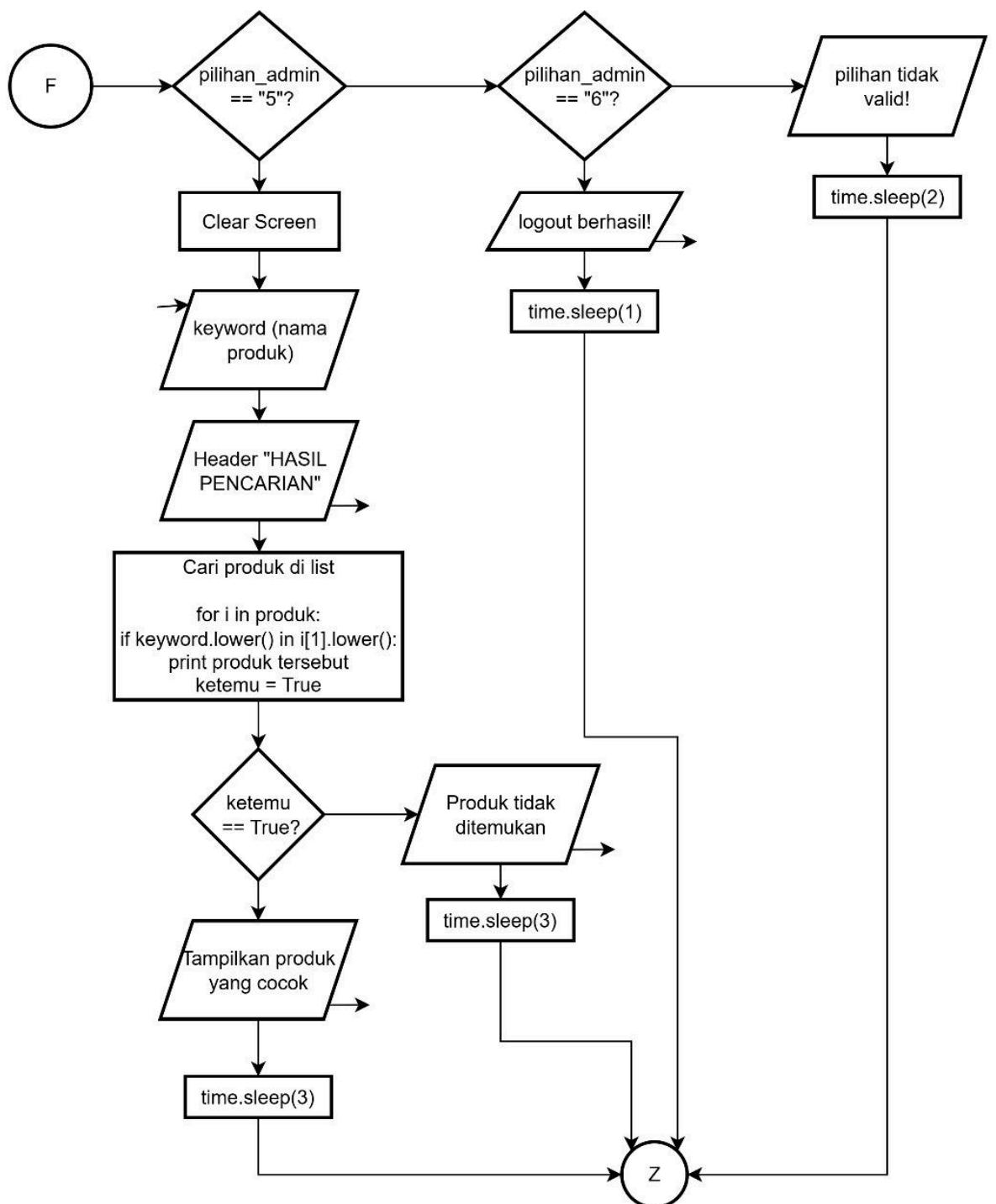
Gambar 1.3 Flowchart bagian on page C untuk program toko makanan kucing whinky



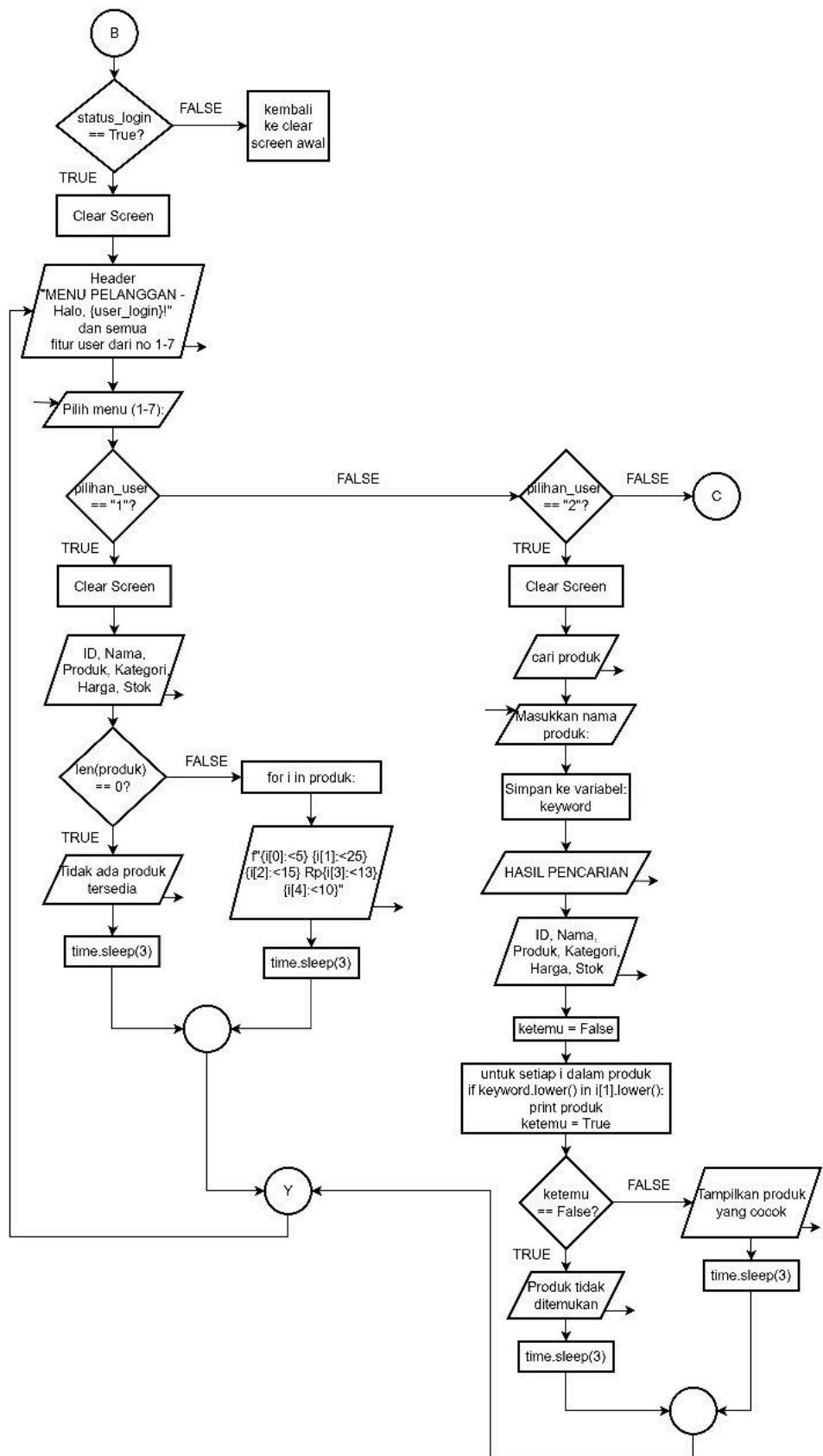
Gambar 1.4 Flowchart bagian on page D untuk program toko makanan kucing whinky



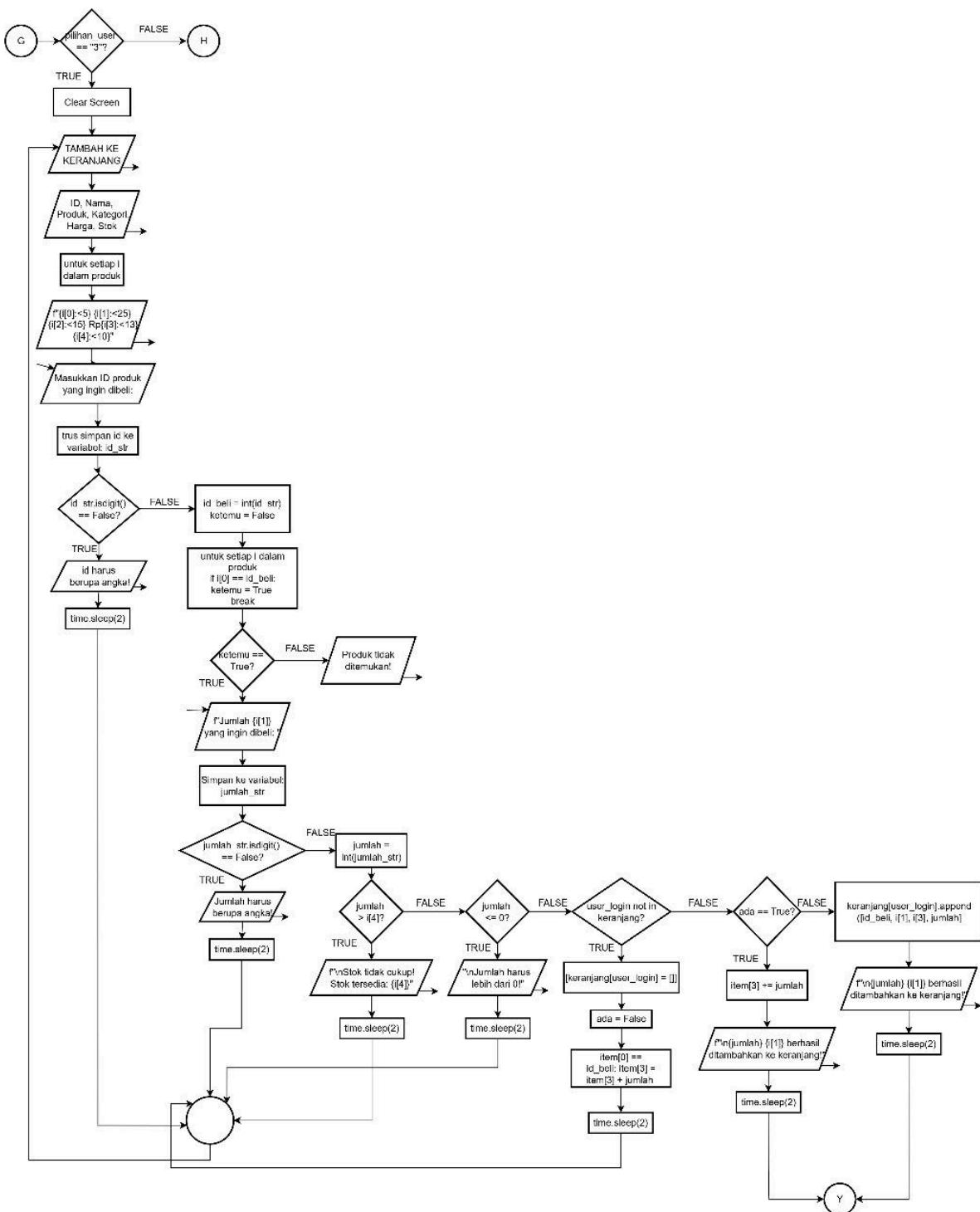
Gambar 1.5 Flowchart bagian on page E untuk program toko makanan kucing whinky



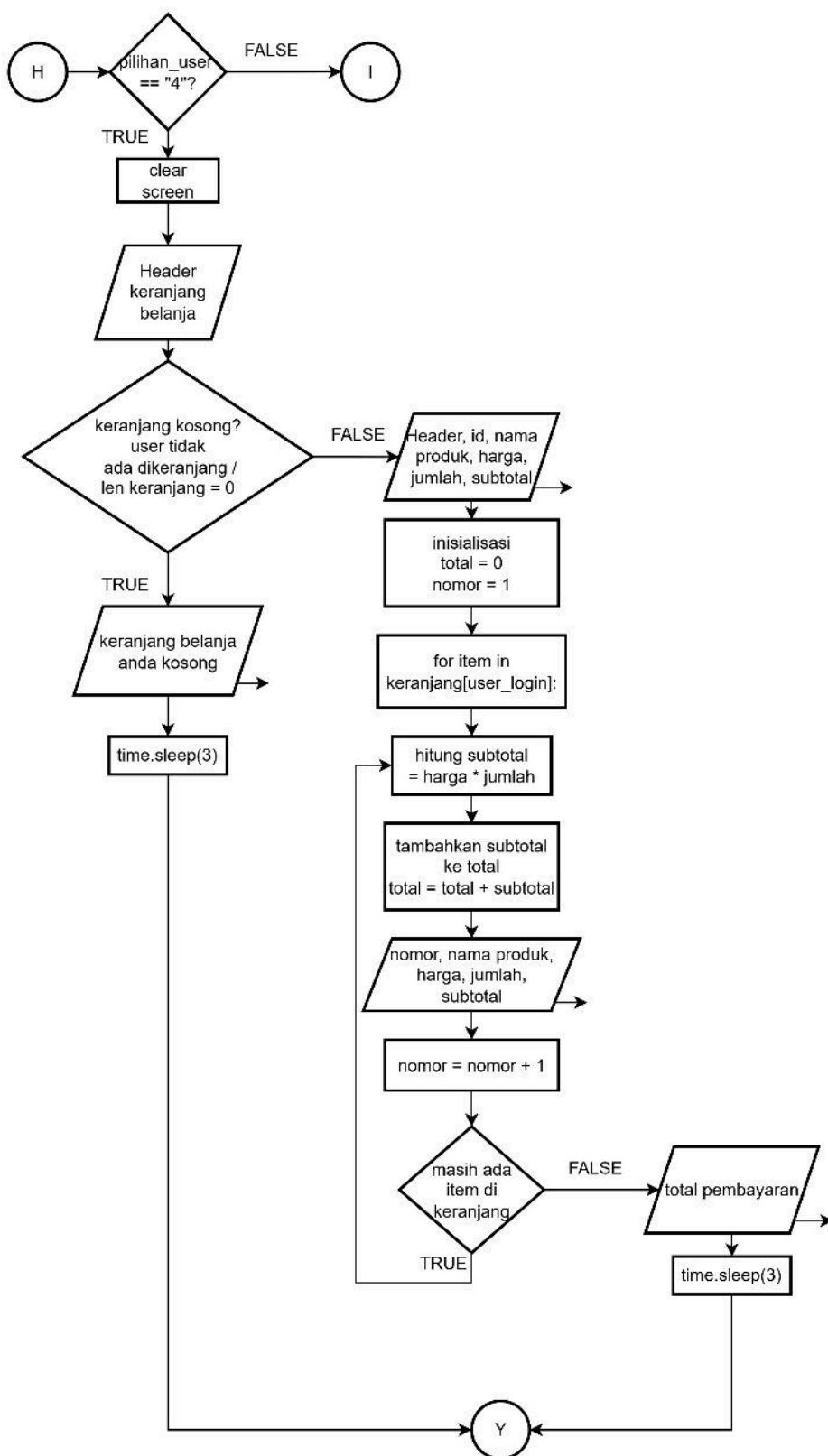
Gambar 1.6 Flowchart bagian on page F untuk program toko makanan kucing whinky



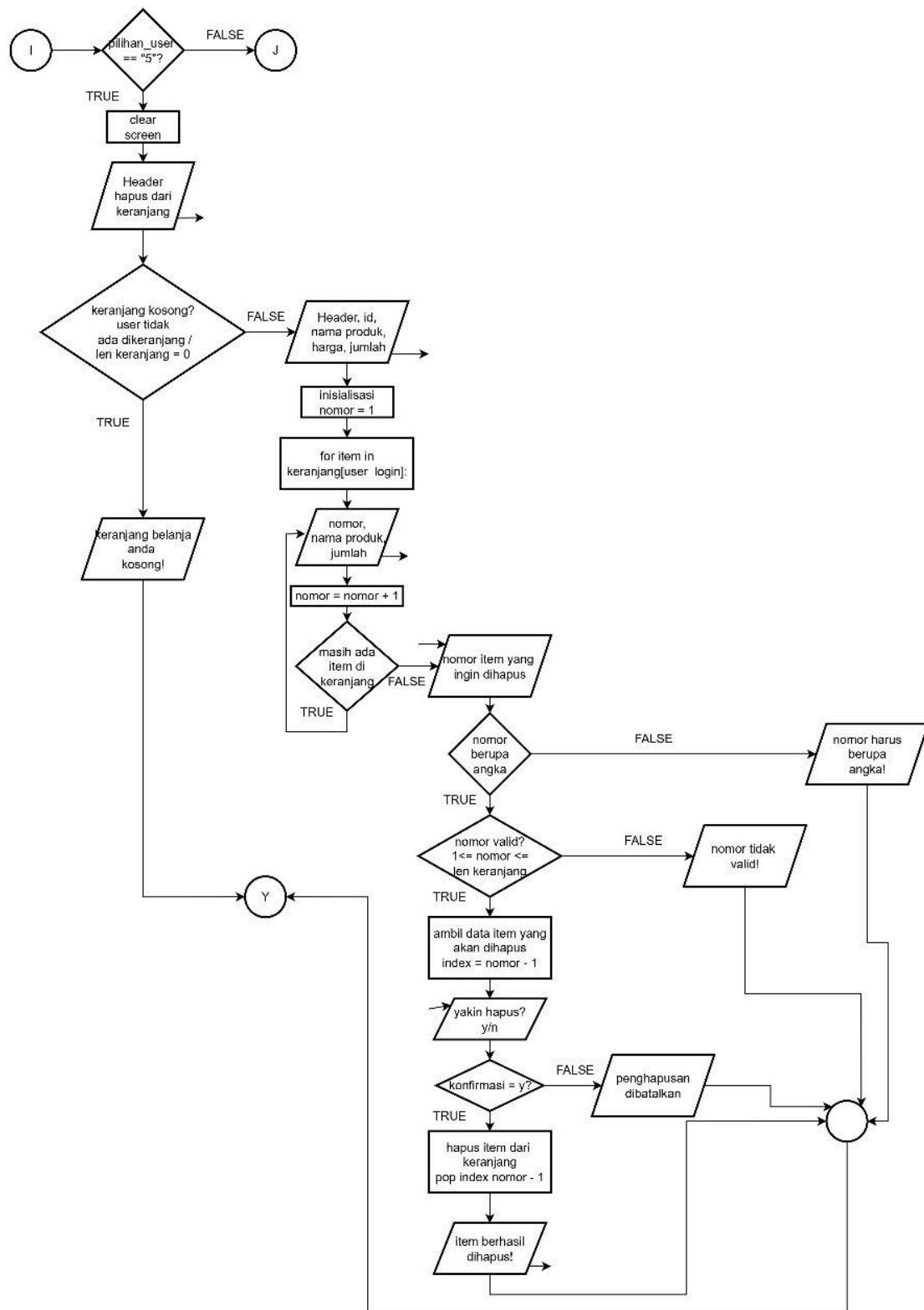
Gambar 1.7 Flowchart bagian on page B untuk program toko makanan kucing whinky



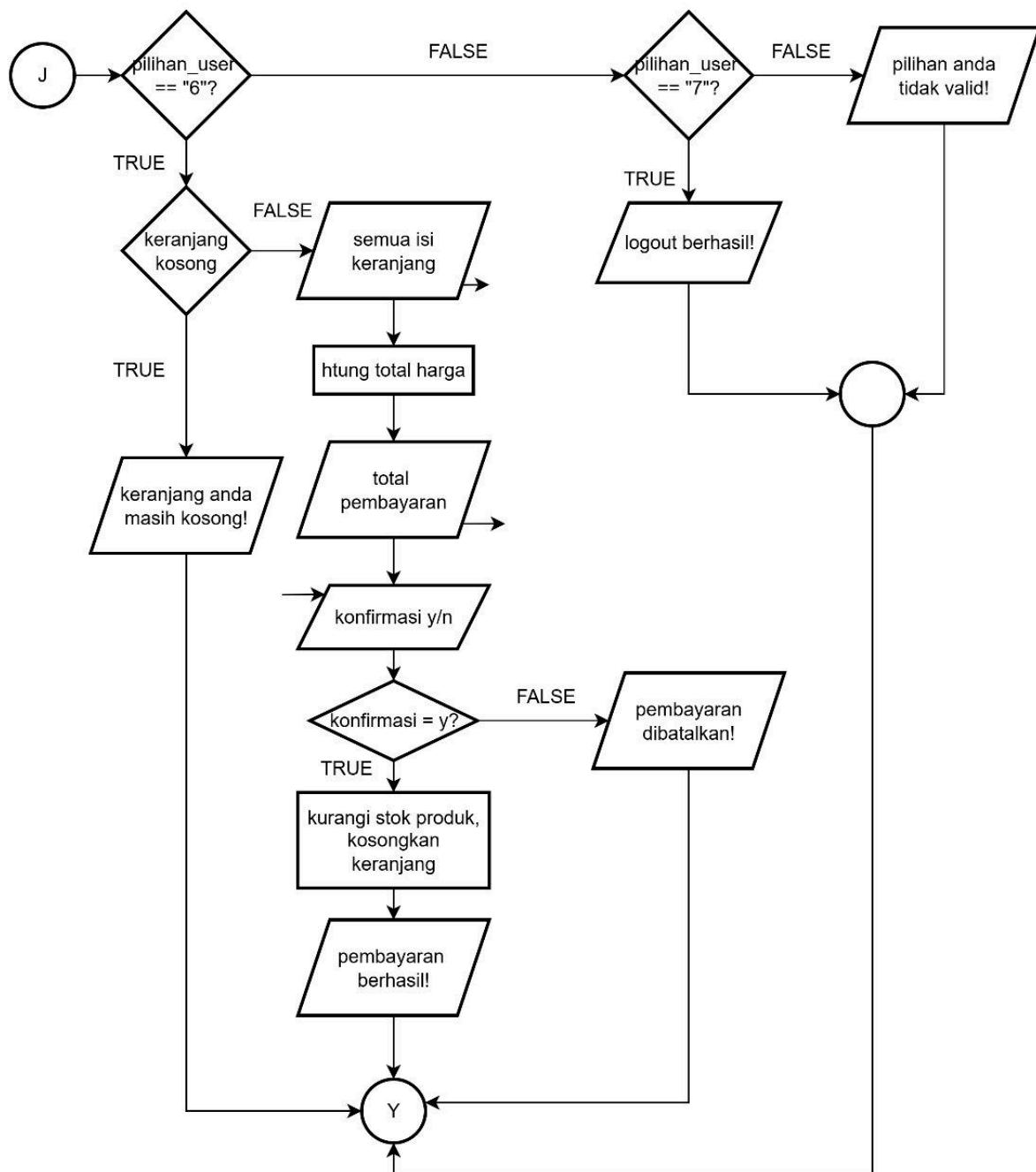
Gambar 1.8 Flowchart bagian on page G untuk program toko makanan kucing whinky



Gambar 1.9 Flowchart bagian on page H untuk program toko makanan kucing whinky



Gambar 1.10 Flowchart bagian on page I untuk program toko makanan kucing whinky



Gambar 1.11 Flowchart bagian on page J untuk program toko makanan kucing whinky

Jadi pertama tama yang muncul di tampilan program ini adalah ucapan selamat datang beserta tiga menu utama dari program ini yaitu 1. login (ini nanti bisa sebagai admin maupun user), 2. register (kita bisa buat nama username semau kita beserta passwordnya), 3. keluar. Lalu kita masuk ke pilihan 1, disitu nanti akan diminta input username dan password lalu program akan memproses buat dicek, kalau ketemu maka akan muncul output login berhasil selamat datang lalu program akan mengecek apakah yang memasukan inputan tadi merupakan admin atau user yang menentukan fitur kedepannya. jika pengguna memilih pilihan maka pengguna akan diminta untuk memberikan input berupa username dan

password baru sebagai data registrasi yang akan disimpan, lalu setelahnya akan muncul output registrasi berhasil, silahkan login untuk melanjutkan. Sedangkan di pilihan ketiga itu ya sudah jelas kalau hanya akan mendapatkan output terimakasih telah mengunjungi toko whinky atau jika pengguna iseng memilih pilihan selain 1-3 maka akan muncul output pilihan kamu tidak valid nih!.

Jika ternyata pengguna merupakan admin maka program akan memunculkan output dimana pengguna nanti diminta untuk memberikan inputan pilihan yang mau dilakukan oleh admin seperti pilihan 1 (lihat semua produk di pilihan ini nantinya program akan menampilkan tabel lengkap id nama produk stok nya dll), pilihan 2 (tambah produk, disini admin memiliki wewenang untuk menambahkan stok barang baru jadi nanti akan diminta input nama produk barunya, stoknya, harganya), pilihan 3 (update produk, nah disini admin dapat mengganti id nama jumlah stok harga dari semua barang yang sudah ada sebelumnya), pilihan 4 (menghapus produk, disini admin juga diberikan wewenang untuk menghapus etalase barang yang mana aja), pilihan 5 (cari produk, fitur ini sangat berguna apabila data barang nanti sudah mencapai ratusan atau ribuan karena dengan adanya ftur ini maka akan lebih cepat kita menemukan barang yang kita butuhkan atau kita cari cari.), dan yang terakhir fitur logout untuk keluar dari program.

Jika ternyata pengguna merupakan user maka program akan memunculkan output pilihan 1-7 yang dapat user pilih melalui input pilihan. Pilihan 1 (melihat semua produk yang tersedia di toko hari ini jadi tau stoknya sisa berapa dengan harga berapa), pilihan 2 (cari produk, sama halnya dengan fitur yang ada di admin ini yang berguna agar memudahkan user mencari nama barang yang sedang diinginkan), pilihan 3 (menambahkan produk ke keranjang , jadi user benar benar dapat membeli produk melalui program kita), pilihan 4 (lihat keranjang, fitur ini bukan hanya akan mempermudah user dari segi bisa melihat barang apa saja yang sudah terambil akan tetapi juga dapat sekalian menghitungkan subtotal sementara dari belanjaannya user), pilihan 5 (hapus dari keranjang disini maksudnya apabila user tidak jadi membeli produk di toko whingky namun hanya beberapa fitur ini hadir untuk mengatasi permasalahan itu dimana dengan program ini user dapat memilih barang spesifik apa yang ingin mereka kembalikan ke etalase atau sama saja dengan tidak membeli), pilihan 6 (checkout untuk fitur ini sendiri sudah jelas digunakan untuk bagian pembayaran ya, dan ini juga akan di konfirmasikan lagi apakah sudah benar pesanannya serta langsung payment juga), dan yang terakhir itu udah pasti ada fitur logout untuk keluar dari program toko whinky

2. Deskripsi Singkat Program

Program sederhana mengenai toko makanan kucing winky ini dibuat dengan tujuan untuk mempermudah para pecinta kucing untuk dapat memenuhi kebutuhan pangan anak bulu mereka. Dengan adanya program sederhana berupa toko makanan winky ini para pengguna jadi tidak perlu cemas lagi untuk mencari toko makanan kucing terpercaya yang dapat menyediakan makanan kucing dengan berbagai merk yang mudah diakses hanya dari genggaman tangan saja.

3. Source Code

Berikut ini merupakan source code yang saya buat untuk program toko makanan kucing whinky secara sederhana dengan fitur lebih kompleks (berstandar CRUD). Di postest ke 8 ini saya hanya menyesuaikan beberapa bagian agar sesuai dengan modul terbaru juga perintah yang diminta (penggunaan pretty table serta library tambahan) serta membagi menjadi beberapa file (modul) yang berbeda sesuai dengan fungsionalitasnya. Disini saya membaginya menjadi 3 bagian yakni:

3.1 bagian Data.py

```
#Variabel global digunakan untuk menyimpan data pengguna, produk, dan keranjang belanja, serta status login.
# cara bacanya {username: {"password": password, "role": role}}
pengguna = {
    "gea": {
        "password": "pacar seokjin",
        "role": "admin"
    },
    "user": {
        "password": "user123",
        "role": "user"
    }
}

# cara bacanya: {id: {"nama": nama, "kategori": kategori, "harga": harga, "stok": stok}}
produk = {
    1: {"nama": "Whiskas 1kg", "kategori": "Makanan", "harga": 50000, "stok": 20},
    2: {"nama": "Royal Canin 2kg", "kategori": "Makanan", "harga": 150000, "stok": 15},
    3: {"nama": "Pasir Gumpal 5kg", "kategori": "Kebersihan", "harga": 45000, "stok": 30},
    4: {"nama": "Sisir Kucing", "kategori": "Grooming", "harga": 25000, "stok": 50},
    5: {"nama": "Mainan Bola", "kategori": "Mainan", "harga": 15000, "stok": 40}
}

# formatnya {username: {id_produk: {"nama": nama, "harga": harga, "jumlah": jumlah}}}
keranjang = {}

# Variabel yang digunakan untuk ngecek user yang login
user_login = ""
role_login = ""
status_login = False

# List kategori produk yang valid (Opsional: memudahkan validasi di Functions.py)
KATEGORI_VALID = ["Makanan", "Kebersihan", "Grooming", "Mainan"]
```

3.2 bagian Functions.py

```
# Berisi Kumpulan fungsi dan logika dari program
import os
import time
from prettytable import PrettyTable
import pwinput
#Import semua data dari Data.py
from Data import pengguna, produk, keranjang, user_login, role_login, status_login, KATEGORI_VALID

# Fungsi 1 yang pakai parameter
def validasi_input_angka(prompt, pesan_error="Input harus berupa angka!"):
    while True:
        try:
            input_str = input(prompt) #ini variabel lokal
            if not input_str:
                return None
            hasil = int(input_str) #ini variabel lokal
            return hasil
        except ValueError:
            print(f"\n{pesan_error}")
            time.sleep(4)
            return None

# Fungsi ke 2 yang pakai parameter juga
def hitung_total_keranjang_rekursif(list_items, index=0):
    if index >= len(list_items): #base case
        return 0

    item_sekarang = list_items[index]
    subtotal = item_sekarang["harga"] * item_sekarang["jumlah"]
    return subtotal + hitung_total_keranjang_rekursif(list_items, index + 1) #fungsi rekursif

# Fungsi 1 yang gaada parameternya
def tampilkan_header_utama():
    os.system('cls || clear')
    print("=" * 60)
    print("|  SELAMAT DATANG DI TOKO PERALATAN KUCING WINGKY  |")
    print("=" * 60)

# Fungsi ke 2 yang gaada parameternya
def generate_id_produk_baru():
    if len(produk) == 0:
        id_baru = 1
    else:
        id_baru = max(produk.keys()) + 1
    return id_baru

def tampilkan_daftar_produk(): #prosedur 1
    print("\n" + "=" * 59)
    print("|          DAFTAR PRODUK WINGKY          |")
```

```

print("=" * 59)

if len(produk) == 0:
    print("Tidak ada produk tersedia.")
else:
    table = PrettyTable()
    table.field_names = ["ID", "Nama Produk", "Kategori", "Harga", "Stok"]
    table.align["ID"] = "c"
    table.align["Nama Produk"] = "l"
    table.align["Kategori"] = "l"
    table.align["Harga"] = "r"
    table.align["Stok"] = "c"

for id_produk, data in produk.items(): #ini variabel lokal yang ada dalam loop
    table.add_row([
        id_produk,
        data['nama'],
        data['kategori'],
        f"Rp {data['harga']:,}",
        data['stok']
    ])

print(table)

print("=" * 59)

def tampilan_isi_keranjang(current_user): #prosedur ke 2
    global keranjang

    print("=" * 59)
    print("|          KERANJANG BELANJA          |")
    print("=" * 59)

    if current_user not in keranjang or len(keranjang[current_user]) == 0:
        print("\nKeranjang belanja kosong.")
        print("=" * 59)
        return 0
    else:
        table = PrettyTable()
        table.field_names = ["No", "Nama Produk", "Harga", "Jumlah", "Subtotal"]
        table.align["No"] = "c"
        table.align["Nama Produk"] = "l"
        table.align["Harga"] = "r"
        table.align["Jumlah"] = "c"
        table.align["Subtotal"] = "r"

        nomor = 1 #ini variabel lokal
        list_items = [] #ini variabel lokal

        for id_produk, item in keranjang[current_user].items(): #ini ada pakai variabel lokal
            subtotal = item["harga"] * item["jumlah"] #ini ada pakai variabel lokal

```

```

list_items.append(item)
table.add_row([
    nomor,
    item['nama'],
    f"Rp {item['harga'][1:]},",
    item['jumlah'],
    f"Rp {subtotal[1:]}"
])
nomor += 1

print(table)

#fungsi rekursif untuk hitung total
total = hitung_total_keranjang_rekursif(list_items) #ini ada pakai variabel lokal

print(f"{'TOTAL':>10} Rp {total:,}")
print("=" * 59)
return total

# Fungsi khusus untuk pencarian produk (fitur buat mempermudah cari produk)
def cari_produk_dan_tampilkan(keyword):
    print("\n" + "=" * 59)
    print("|          HASIL PENCARIAN          |")
    print("=" * 59)

    # Menggunakan PrettyTable untuk menampilkan hasil pencarian
    table = PrettyTable()
    table.field_names = ["ID", "Nama Produk", "Kategori", "Harga", "Stok"]
    table.align["ID"] = "c"
    table.align["Nama Produk"] = "l"
    table.align["Kategori"] = "l"
    table.align["Harga"] = "r"
    table.align["Stok"] = "c"

    ketemu = False
    for id_produk, data in produk.items():
        if keyword.lower() in data['nama'].lower():
            table.add_row([
                id_produk,
                data['nama'],
                data['kategori'],
                f"Rp {data['harga'][1:]},",
                data['stok']
            ])
            ketemu = True

    if ketemu:
        print(table)
    else:
        print("\nProduk tidak ditemukan.")
    print("=" * 59)

```

3.3 bagian Main.py

```
# Berisi program utama
from Functions import (
    tampilan_header_utama,
    validasi_input_angka,
    generate_id_produk_baru,
    tampilan_daftar_produk,
    tampilan_isi_keranjang,
    cari_produk_dan_tampilkan
)

from Data import (
    pengguna,
    produk,
    keranjang,
    user_login,
    role_login,
    status_login,
    KATEGORI_VALID
)

import os
import time
import pwinput
from prettytable import PrettyTable

def main():
    global user_login, role_login, status_login

    while True: #program utamanya
        tampilan_header_utama() #ini pakai fungsi 1
        print("\n1. Login")
        print("2. Register")
        print("3. Keluar")

        pilihan_utama = input("\nPilih menu (1-3): ")

        # Menu login
        if pilihan_utama == "1":
            tampilan_header_utama() # ini pakai fungsi 1
            print("\n----- LOGIN SECTION -----")

            try: # Error handling untuk bagian login
                username = input("Username: ") #ini variabel lokal
                if not username:
                    raise ValueError("Username tidak boleh kosong")

                password = pwinput.pwinput(prompt="Password: ", mask="*") #ini variabel lokal - menggunakan
                pwinput
                if not password:
```

```

raise ValueError("Password tidak boleh kosong")

# Verifikasi login dulu disini
if username in pengguna and pengguna[username]["password"] == password:
    user_login = username
    role_login = pengguna[username]["role"]
    status_login = True
    print(f"\nLogin berhasil! Selamat datang, {user_login}")
    time.sleep(4)

# Menu adminnya
if role_login == "admin":
    while status_login == True:
        os.system('cls || clear')
        print("=" * 50)
        print(f"|      MENU ADMIN - Halo, {user_login}!      |")
        print("=" * 50)
        print("\n1. Lihat Semua Produk")
        print("2. Tambah Produk")
        print("3. Update Produk")
        print("4. Hapus Produk")
        print("5. Cari Produk")
        print("6. Logout")

    pilihan_admin = input("\nPilih menu (1-6): ")

    # fitur lihat produk
    if pilihan_admin == "1":
        os.system('cls || clear')
        tampilkan_daftar_produk() # ini pakai prosedur 1
        time.sleep(10)

    # fitur tambah produk
    elif pilihan_admin == "2":
        os.system('cls || clear')
        print("=" * 50)
        print("|      TAMBAH PRODUK BARU      |")
        print("=" * 50)

    try: # Error handling untuk bagian tambah produk
        global produk

        id_baru = generate_id_produk_baru() # ini pakai fungsi 2

        nama = input("\nNama Produk: ")
        if not nama:
            raise ValueError("Nama produk tidak boleh kosong")

        kategori = input("Kategori (Makanan/Kebersihan/Grooming/Mainan): ")
        if not kategori:
            raise ValueError("Kategori tidak boleh kosong")

```

```

# fungsi buat validasi harus brupa angka
harga = validasi_input_angka("Harga: ", "Harga harus berupa angka!")
if harga is None or harga <= 0:
    raise ValueError("Harga harus lebih dari 0")

stok = validasi_input_angka("Stok: ", "Stok harus berupa angka!")
if stok is None or stok < 0:
    raise ValueError("Stok tidak boleh negatif")

produk[id_baru] = { #ini variabel lokal
    "nama": nama, #ini variabel lokal
    "kategori": kategori, #ini variabel lokal
    "harga": harga, #ini variabel lokal
    "stok": stok #ini variabel lokal
}
print(f"\nProduk '{nama}' berhasil ditambahkan!")
time.sleep(4)

except ValueError as e:
    print(f"\nError: {e}")
    time.sleep(4)

# fitur update produknya
elif pilihan_admin == "3":
    os.system('cls || clear')
    print("=" * 50)
    print("|           UPDATE PRODUK           |")
    print("=" * 50)

    # Menggunakan PrettyTable untuk menampilkan list produk
    table = PrettyTable()
    table.field_names = ["ID", "Nama Produk"]
    table.align["ID"] = "c"
    table.align["Nama Produk"] = "l"

    for id_produk, data in produk.items():
        table.add_row([id_produk, data['nama']])

    print(table)

    try: # Error handling untuk bagian fitur update
        id_update = validasi_input_angka("\nMasukkan ID produk yang ingin diupdate: ", "ID harus berupa angka!")#ini variabel lokal
        #line diatas ini pakai fungsi 1 dengan parameter
        if id_update is None:
            raise ValueError("Input dibatalkan")

        if id_update not in produk:
            raise KeyError("Produk dengan ID tersebut tidak ditemukan!")

```

```

print(f"\nProduk ditemukan: {produk[id_update]['nama']}")

print("\nPilih yang ingin diupdate:")
print("1. Nama")
print("2. Kategori")
print("3. Harga")
print("4. Stok")
pilih = input("Pilihan: ") #ini variabel lokal

if pilih == "1":
    nama_baru = input("Nama baru: ") #ini variabel lokal
    if not nama_baru:
        raise ValueError("Nama tidak boleh kosong")
    produk[id_update]["nama"] = nama_baru
    print("\nNama produk berhasil diupdate!")
elif pilih == "2":
    kategori_baru = input("Kategori baru: ") #ini variabel lokal
    if not kategori_baru:
        raise ValueError("Kategori tidak boleh kosong")
    produk[id_update]["kategori"] = kategori_baru
    print("\nKategori produk berhasil diupdate!")
elif pilih == "3":
    harga_baru = validasi_input_angka("Harga baru: ", "Harga harus berupa angka!")
    #line diatas merupakan variabel lokal dan menggunakan fungsi 1 dengan parameter
    if harga_baru is None or harga_baru <= 0:
        raise ValueError("Harga harus lebih dari 0")
    produk[id_update]["harga"] = harga_baru
    print("\nHarga produk berhasil diupdate!")
elif pilih == "4":
    stok_baru = validasi_input_angka("Stok baru: ", "Stok harus berupa angka!")
    #line diatasmerupakan variabel lokal dan menggunakan fungsi 1 dengan parameter
    if stok_baru is None or stok_baru < 0:
        raise ValueError("Stok tidak boleh negatif")
    produk[id_update]["stok"] = stok_baru
    print("\nStok produk berhasil diupdate!")
    time.sleep(4)
else:
    print("\nPilihan tidak valid!")
    time.sleep(4)

except (ValueError, KeyError) as e:
    print(f"\nError: {e}")
    time.sleep(4)

# fitur hapus produk
elif pilihan_admin == "4":
    os.system('cls || clear')
    print("=" * 50)
    print("|          HAPUS PRODUK          |")
    print("=" * 50)

# Menggunakan PrettyTable untuk menampilkan list produk

```

```

table = PrettyTable()
table.field_names = ["ID", "Nama Produk"]
table.align["ID"] = "c"
table.align["Nama Produk"] = "l"

for id_produk, data in produk.items():
    table.add_row([id_produk, data['nama']])

print(table)

try: # Error handling untuk fitur hapus produk
    id_hapus = validasi_input_angka("\nMasukkan ID produk yang ingin dihapus: ", "ID harus berupa angka!")
    #line diatas ini menggunakan fungsi 1 dengan parameter
    if id_hapus is None:
        raise ValueError("Input dibatalkan")

    if id_hapus not in produk:
        raise KeyError("Produk dengan ID tersebut tidak ditemukan!")

    konfirmasi = input(f"\nYakin ingin menghapus '{produk[id_hapus]['nama']}'? (y/n): ")
    if konfirmasi.lower() == "y":
        del produk[id_hapus]
        print("\nProduk berhasil dihapus!")
    else:
        print("\nPenghapusan dibatalkan.")
    time.sleep(4)

except (ValueError, KeyError) as e:
    print(f"\nError: {e}")
    time.sleep(4)

# fitur buat mempermudah cari produk
elif pilihan_admin == "5":
    os.system('cls || clear')
    print("=" * 59)
    print("|          CARI PRODUK          |")
    print("=" * 59)
    keyword = input("\nMasukkan nama produk: ")

    cari_produk_dan_tampilkan(keyword)
    time.sleep(4)

# fitur buat log out
elif pilihan_admin == "6":
    print("\nLogout berhasil!")
    time.sleep(4)
    status_login = False

else:
    print("\nPilihan tidak valid!")

```

```

time.sleep(4)

# menu pengguna
else:
    while status_login == True:
        os.system('cls || clear')
        print("=" * 50)
        print(f'|      MENU PELANGGAN - Halo, {user_login}!      |")
        print("=" * 50)
        print("\n1. Lihat Semua Produk")
        print("2. Cari Produk")
        print("3. Tambah ke Keranjang")
        print("4. Lihat Keranjang")
        print("5. Hapus dari Keranjang")
        print("6. Checkout")
        print("7. Logout")

    pilihan_user = input("\nPilih menu (1-7): ")

    # fitur buat lihat produk
    if pilihan_user == "1":
        os.system('cls || clear')
        tampilan_daftar_produk() #ini pakai prosedur 1
        time.sleep(7)

    # fitur buat mempermudah cari produk
    elif pilihan_user == "2":
        os.system('cls || clear')
        print("=" * 50)
        print("|      CARI PRODUK      |")
        print("=" * 50)
        keyword = input("\nMasukkan nama produk: ")

        cari_produk_dan_tampilkan(keyword)
        time.sleep(5)

    # fitur tambahkan produk ke keranjang
    elif pilihan_user == "3":
        os.system('cls || clear')
        print("=" * 59)
        print("|      TAMBAH KE KERANJANG      |")
        print("=" * 59)

    # Menggunakan PrettyTable untuk menampilkan list produk
    table = PrettyTable()
    table.field_names = ["ID", "Nama Produk", "Kategori", "Harga", "Stok"]
    table.align["ID"] = "c"
    table.align["Nama Produk"] = "l"
    table.align["Kategori"] = "l"
    table.align["Harga"] = "r"
    table.align["Stok"] = "c"

```

```

for id_produk, data in produk.items():
    table.add_row([
        id_produk,
        data['nama'],
        data['kategori'],
        f"Rp {data['harga']:,}",
        data['stok']
    ])
print(table)

try:
    global keranjang

    id_beli = validasi_input_angka("\nMasukkan ID produk yang ingin dibeli: ", "ID harus berupa angka!")
    if id_beli is None:
        raise ValueError("Input dibatalkan")

    if id_beli not in produk:
        raise KeyError("Produk dengan ID tersebut tidak ditemukan!")

    jumlah_beli = validasi_input_angka("Jumlah: ", "Jumlah harus berupa angka!")
    if jumlah_beli is None or jumlah_beli <= 0:
        raise ValueError("Jumlah harus lebih dari 0")

    if jumlah_beli > produk[id_beli]["stok"]:
        raise ValueError(f"Stok tidak cukup! Stok tersedia: {produk[id_beli]['stok']}")

    # Inisialisasi keranjang user jika belum ada
    if user_login not in keranjang:
        keranjang[user_login] = {}

    # Tambah ke keranjang
    if id_beli in keranjang[user_login]:
        keranjang[user_login][id_beli]["jumlah"] += jumlah_beli
    else:
        keranjang[user_login][id_beli] = {
            "nama": produk[id_beli]["nama"],
            "harga": produk[id_beli]["harga"],
            "jumlah": jumlah_beli
        }

    print(f"\n{produk[id_beli]['nama']} ({jumlah_beli}) berhasil ditambahkan ke keranjang!")
    time.sleep(4)
    print("=" * 59)

except (ValueError, KeyError) as e:
    print(f"\nError: {e}")

```

```

        print("=" * 59)
        time.sleep(4)

        # fitur lihat keranjang
        elif pilihan_user == "4":
            os.system('cls || clear')
            tampilkan_isi_keranjang(user_login) #ini pakai prosedur 2
            time.sleep(7)

        # fitur hapus dari keranjang
        elif pilihan_user == "5":
            os.system('cls || clear')
            print("=" * 38)
            print("|      HAPUS DARI KERANJANG      |")
            print("=" * 38)

        if user_login not in keranjang or len(keranjang[user_login]) == 0:
            print("\nKeranjang belanja kosong.")
            time.sleep(4)
        else:
            # Menggunakan PrettyTable untuk menampilkan isi keranjang
            table = PrettyTable()
            table.field_names = ["No", "ID", "Nama Produk", "Jumlah"]
            table.align["No"] = "c"
            table.align["ID"] = "c"
            table.align["Nama Produk"] = "l"
            table.align["Jumlah"] = "c"

            nomor = 1
            id_list = []
            for id_produk, item in keranjang[user_login].items():
                table.add_row([
                    nomor,
                    id_produk,
                    item['nama'],
                    item['jumlah']
                ])
                id_list.append(id_produk)
                nomor += 1

            print(table)

        try:
            id_hapus = validasi_input_angka("\nMasukkan ID produk yang ingin dihapus: ",
            "ID harus berupa angka!")
            if id_hapus is None:
                raise ValueError("Input dibatalkan")

            if id_hapus not in keranjang[user_login]:
                raise KeyError("Produk tidak ada di keranjang!")

```

```

        konfirmasi = input(f"\nYakin ingin menghapus
'{keranjang[user_login][id_hapus]['nama']}' dari keranjang? (y/n): ")
        if konfirmasi.lower() == "y":
            del keranjang[user_login][id_hapus]
            print("\nProduk berhasil dihapus dari keranjang!")
        else:
            print("\nPenghapusan dibatalkan.")
            time.sleep(4)

    except (ValueError, KeyError) as e:
        print(f"\nError: {e}")
        time.sleep(4)

# fitur checkout
elif pilihan_user == "6":
    os.system('cls || clear')
    print("=" * 59)
    print("|           CHECKOUT           |")
    print("=" * 59)

    if user_login not in keranjang or len(keranjang[user_login]) == 0:
        print("\nKeranjang belanja kosong. Tidak ada yang bisa dicheckout.")
        time.sleep(4)
    else:
        tampilkan_isi_keranjang(user_login)

        konfirmasi = input("\nLanjutkan checkout? (y/n): ")
        if konfirmasi.lower() == "y":
            # Kurangi stok
            for id_produk, item in keranjang[user_login].items():
                produk[id_produk]["stok"] -= item["jumlah"]

            # Kosongkan keranjang
            keranjang[user_login] = {}

            print("\n" + "=" * 59)
            print("|   Checkout berhasil! Terima kasih telah berbelanja!   |")
            print("=" * 59)
            time.sleep(5)
        else:
            print("\nCheckout dibatalkan.")
            time.sleep(4)

# fitur logout
elif pilihan_user == "7":
    print("\nLogout berhasil!")
    time.sleep(4)
    status_login = False

else:
    print("\nPilihan tidak valid!")

```

```

        time.sleep(4)
    else:
        print("\nUsername atau password salah!")
        time.sleep(4)

except ValueError as e:
    print(f"\nError: {e}")
    time.sleep(4)

# Menu Register
elif pilihan_utama == "2":
    tampilan_header_utama()
    print("\n----- REGISTER SECTION -----")

try:
    username_baru = input("Username baru: ")
    if not username_baru:
        raise ValueError("Username tidak boleh kosong")

    if username_baru in pengguna:
        raise ValueError("Username sudah terdaftar!")

    password_baru = pwininput.pwininput(prompt="Silahkan input password baru: ", mask="*") #

Menggunakan passwordnya
    if not password_baru:
        raise ValueError("Password tidak boleh kosong")

    pengguna[username_baru] = {
        "password": password_baru,
        "role": "user"
    }

    print(f"\nRegistrasi berhasil! Silakan login dengan username '{username_baru}'")
    time.sleep(4)

except ValueError as e:
    print(f"\nError: {e}")
    time.sleep(4)

# Menu Keluar
elif pilihan_utama == "3":
    print("\nTerima kasih telah mengunjungi Toko Peralatan Kucing Wingky!")
    break

else:
    print("\nPilihan tidak valid!")
    time.sleep(4)

```

4. Hasil Output

Berikut ini merupakan hasil output dari program toko makanan kucing whinky secara sederhana yang sudah saya buat.

4.1 Sebagai Admin

```
=====
|      SELAMAT DATANG DI TOKO PERALATAN KUCING WINGKY      |
=====

1. Login
2. Register
3. Keluar

Pilih menu (1-3): 1|
```



```
=====
|      SELAMAT DATANG DI TOKO PERALATAN KUCING WINGKY      |
=====

----- LOGIN SECTION -----
Username: gea
Password: *****

Login berhasil! Selamat datang, gea|
```

Gambar 4.1.1 Ketika login sebagai admin di pilihan 1 (login)

```
=====
|          MENU ADMIN - Halo, gea!          |
=====

1. Lihat Semua Produk
2. Tambah Produk
3. Update Produk
4. Hapus Produk
5. Cari Produk
6. Logout

Pilih menu (1-6): 1|
```

Gambar 4.1.2 tampilan layar bila jadi admin

DAFTAR PRODUK WINGKY				
ID	Nama Produk	Kategori	Harga	Stok
1	Whiskas 1kg	Makanan	Rp 50,000	20
2	Royal Canin 2kg	Makanan	Rp 150,000	15
3	Pasir Gumpal 5kg	Kebersihan	Rp 45,000	30
4	Sisir Kucing	Grooming	Rp 25,000	50
5	Mainan Bola	Mainan	Rp 15,000	40

Gambar 4.1.3 output pilihan 1 (melihat semua produk)

```
=====
|          TAMBAH PRODUK BARU          |
=====

Nama Produk: bedak kutu
Kategori (Makanan/Kebersihan/Grooming/Mainan): grooming
Harga: 20000
Stok: 11

Produk 'bedak kutu' berhasil ditambahkan!
```

Gambar 4.1.4 output pilihan 2 (menambahkan produk baru pada etalase toko)

DAFTAR PRODUK WINGKY				
ID	Nama Produk	Kategori	Harga	Stok
1	Whiskas 1kg	Makanan	Rp 50,000	20
2	Royal Canin 2kg	Makanan	Rp 150,000	15
3	Pasir Gumpal 5kg	Kebersihan	Rp 45,000	30
4	Sisir Kucing	Grooming	Rp 25,000	50
5	Mainan Bola	Mainan	Rp 15,000	40
6	bedak kutu	grooming	Rp 20,000	11

Gambar 4.1.5 output pilihan 1 (udah ditambahin si bedak kutu)

```
=====|          UPDATE PRODUK |=====
+---+-----+
| ID | Nama Produk      |
+---+-----+
| 1  | Whiskas 1kg       |
| 2  | Royal Canin 2kg    |
| 3  | Pasir Gumpal 5kg   |
| 4  | Sisir Kucing        |
| 5  | Mainan Bola         |
| 6  | bedak kutu          |
+---+-----+
Masukkan ID produk yang ingin diupdate: 6
Produk ditemukan: bedak kutu
Pilih yang ingin diupdate:
1. Nama
2. Kategori
3. Harga
4. Stok
Pilihan: 4
Stok baru: 17
Stok produk berhasil diupdate!
```

Gambar 4.1.6 output pilihan 3 (jadi bisa memperbarui nama/ kategori/ harga/ stok)

```
=====|          HAPUS PRODUK |=====
+---+-----+
| ID | Nama Produk      |
+---+-----+
| 1  | Whiskas 1kg       |
| 2  | Royal Canin 2kg    |
| 3  | Pasir Gumpal 5kg   |
| 4  | Sisir Kucing        |
| 5  | Mainan Bola         |
| 6  | bedak kutu          |
+---+-----+
Masukkan ID produk yang ingin dihapus: 6
Yakin ingin menghapus 'bedak kutu'? (y/n): y
Produk berhasil dihapus!
```

Gambar 4.1.7 output pilihan 4 (admin dapat menghapus apapun yang ada di etalase)

```
=====  
|          CARI PRODUK          |  
=====  
  
Masukkan nama produk: whiskas  
  
=====  
|          HASIL PENCARIAN          |  
=====  
+---+-----+-----+-----+  
| ID | Nama Produk | Kategori |     Harga | Stok |  
+---+-----+-----+-----+  
| 1 | Whiskas 1kg | Makanan | Rp 50,000 |  20 |  
+---+-----+-----+-----+  
=====
```

Gambar 4.1.8 output pilihan 5 (admin bisa mencari produk lebih cepat dengan fitur ini)

4.2 sebagai user

```
=====  
|      SELAMAT DATANG DI TOKO PERALATAN KUCING WINGKY      |  
=====  
  
1. Login  
2. Register  
3. Keluar  
  
Pilih menu (1-3): 2
```

Gambar 4.2.1 output pilihan 2 sedari awal dengan tujuan daftar akun baru

```
=====  
|      SELAMAT DATANG DI TOKO PERALATAN KUCING WINGKY      |  
=====  
  
----- REGISTER SECTION -----  
Username baru: seokjin  
Silahkan input password baru: *****  
  
Registrasi berhasil! Silakan login dengan username 'seokjin'
```

Gambar 4.2.1.a output pilihan 2 sedari awal dengan tujuan daftar akun baru

```
=====
|           MENU PELANGGAN - Halo, seokjin!           |
=====

1. Lihat Semua Produk
2. Cari Produk
3. Tambah ke Keranjang
4. Lihat Keranjang
5. Hapus dari Keranjang
6. Checkout
7. Logout

Pilih menu (1-7): 1
```

Gambar 4.2.2 Ketika login sebagai user (kalau registrasi itu akunnya nanti jadi user) di pilihan 1 (login)

DAFTAR PRODUK WINGKY				
ID	Nama Produk	Kategori	Harga	Stok
1	Whiskas 1kg	Makanan	Rp 50,000	20
2	Royal Canin 2kg	Makanan	Rp 150,000	15
3	Pasir Gumpal 5kg	Kebersihan	Rp 45,000	30
4	Sisir Kucing	Grooming	Rp 25,000	50
5	Mainan Bola	Mainan	Rp 15,000	40

Gambar 4.2.3 output pilihan 1 (melihat semua produk)

CARI PRODUK				
Masukkan nama produk: royal				
HASIL PENCARIAN				
ID	Nama Produk	Kategori	Harga	Stok
2	Royal Canin 2kg	Makanan	Rp 150,000	15

Gambar 4.2.4 output pilihan 2 (fitur mencari produk)

```

=====
|          TAMBAH KE KERANJANG          |
=====
+---+-----+-----+-----+
| ID | Nama Produk      | Kategori   | Harga | Stok |
+---+-----+-----+-----+
| 1  | Whiskas 1kg       | Makanan    | Rp 50,000 | 20  |
| 2  | Royal Canin 2kg    | Makanan    | Rp 150,000 | 15  |
| 3  | Pasir Gumpal 5kg   | Kebersihan  | Rp 45,000 | 30  |
| 4  | Sisir Kucing        | Grooming   | Rp 25,000 | 50  |
| 5  | Mainan Bola         | Mainan     | Rp 15,000 | 40  |
+---+-----+-----+-----+
Masukkan ID produk yang ingin dibeli: 3
Jumlah: 1
Pasir Gumpal 5kg (x1) berhasil ditambahkan ke keranjang!

```

Gambar 4.2.5 output pilihan 3 (jadi user bisa nambahin barang kekeranjang sepuasnya
*asalkan stoknya cukup yaa)

```

=====
|          KERANJANG BELANJA          |
=====
+---+-----+-----+-----+
| No | Nama Produk      | Harga | Jumlah | Subtotal |
+---+-----+-----+-----+
| 1  | Pasir Gumpal 5kg   | Rp 45,000 | 1      | Rp 45,000 |
| 2  | Royal Canin 2kg    | Rp 150,000 | 2      | Rp 300,000 |
| 3  | Mainan Bola         | Rp 15,000  | 1      | Rp 15,000 |
+---+-----+-----+-----+
TOTAL Rp 360,000
=====
```

Gambar 4.2.6 output pilihan 4 (lihat isi keranjang dan dihitungkan subtotal belanjanya berapa)

```

=====
|          HAPUS DARI KERANJANG          |
=====
+---+-----+-----+
| No | ID | Nama Produk      | Jumlah |
+---+-----+-----+
| 1  | 3  | Pasir Gumpal 5kg   | 1      |
| 2  | 2  | Royal Canin 2kg    | 2      |
| 3  | 5  | Mainan Bola         | 1      |
+---+-----+-----+
Masukkan ID produk yang ingin dihapus: 5
Yakin ingin menghapus 'Mainan Bola' dari keranjang? (y/n): y
Produk berhasil dihapus dari keranjang!

```

Gambar 4.2.7 output pilihan 5 (fitur menghapus produk pilihan yang ada di keranjang)

```

=====
|          CHECKOUT          |
=====

=====
|          KERANJANG BELANJA          |
=====

+---+-----+-----+-----+
| No | Nama Produk      | Harga | Jumlah | Subtotal |
+---+-----+-----+-----+
| 1  | Pasir Gumpal 5kg | Rp 45,000 | 1     | Rp 45,000 |
| 2  | Royal Canin 2kg  | Rp 150,000 | 2     | Rp 300,000 |
+---+-----+-----+-----+
      TOTAL Rp 345,000
=====

Lanjutkan checkout? (y/n): y

=====
|      Checkout berhasil! Terima kasih telah berbelanja!      |
=====
```

Gambar 4.2.8 output pilihan 6 (fitur checkout)

```

=====
|      SELAMAT DATANG DI TOKO PERALATAN KUCING WINGKY      |
=====

1. Login
2. Register
3. Keluar

Pilih menu (1-3): 3

Terima kasih telah mengunjungi Toko Peralatan Kucing Wingky!
```

Gambar 4.2.9 jika pengguna memilih pilihan 3 sejak awal

Jadi apabila sedari awal pengguna memilih registrasi pada pilihan 2 itu sudah pasti jadi user karna adminnya hanya 1 orang di program ini (bisa ditambah sih kalau mau), lalu kalau pengguna selalu memilih yang tidak ada di pilihan pasti akan keluar output eror yang beragam (seperti outputan eror yang ada di program toko peralatan kucing whinky ini) sebelum kembali ke looping awal lagi.

5. Langkah-langkah GIT

Git adalah sistem kontrol versi yang membantu kita melacak perubahan pada kode atau proyek kita, jadi dengan adanya git ini kita itu bisa ngeliat semua perubahan yang kita buat dari awal gitu, bahkan kita bisa kembali ke versi sebelumnya tanpa kehilangan jejak dari projek yang sudah kita buat sebelumnya.

5.1 GIT Add

Git add digunakan untuk menambahkan file dan memasukan file ke daftar perubahan. Jadi kita bisa memindahkan file dari tempat asalnya ke tempat dimana file siap di commit dengan menggunakan Git add ini.

```
● PS C:\Users\ASUS\OneDrive\Documents\praktikum-apd> git add .
```

Gambar 5.1.1 Penggunaan Git Add

5.2 GIT Commit

Git commit ini digunakan untuk menyimpan perubahan secara permanen, jadi kita bisa kembali lagi kesini kapan aja (udah seperti titik check point di game gitu dia).

```
● PS C:\Users\ASUS\OneDrive\Documents\praktikum-apd> git commit -m "second commit"
[main 137bdb4] second commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 post-test/post-test-apd-8/2509106007-Syarifah Anargya Rizky-PT-8.pdf
○ PS C:\Users\ASUS\OneDrive\Documents\praktikum-apd>
```

Gambar 5.2.1 Penggunaan Git Commit

5.3 GIT Push

Git push digunakan untuk mengirim commit kita ke github nanti, jadi disambungin gitu repositori lokal kita ke github.

```
● PS C:\Users\ASUS\OneDrive\Documents\praktikum-apd> git push
Enumerating objects: 23, done.
Counting objects: 100% (23/23), done.
Delta compression using up to 4 threads
Compressing objects: 100% (19/19), done.
Writing objects: 100% (19/19), 1.34 MiB | 154.00 KiB/s, done.
Total 19 (delta 8), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (8/8), completed with 2 local objects.
To https://github.com/seokjinjuseyyo/praktikum-apd.git
  66c64cd..137bdb4 main -> main
○ PS C:\Users\ASUS\OneDrive\Documents\praktikum-apd>
```

Gambar 5.3.1 penggunaan Git Push untuk menghubungkan repository lokal ke github