

**LAPORAN PRAKTIKUM**  
**POSTTEST 7**  
**ALGORITMA PEMROGRAMAN DASAR**



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

**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**  
**2025**

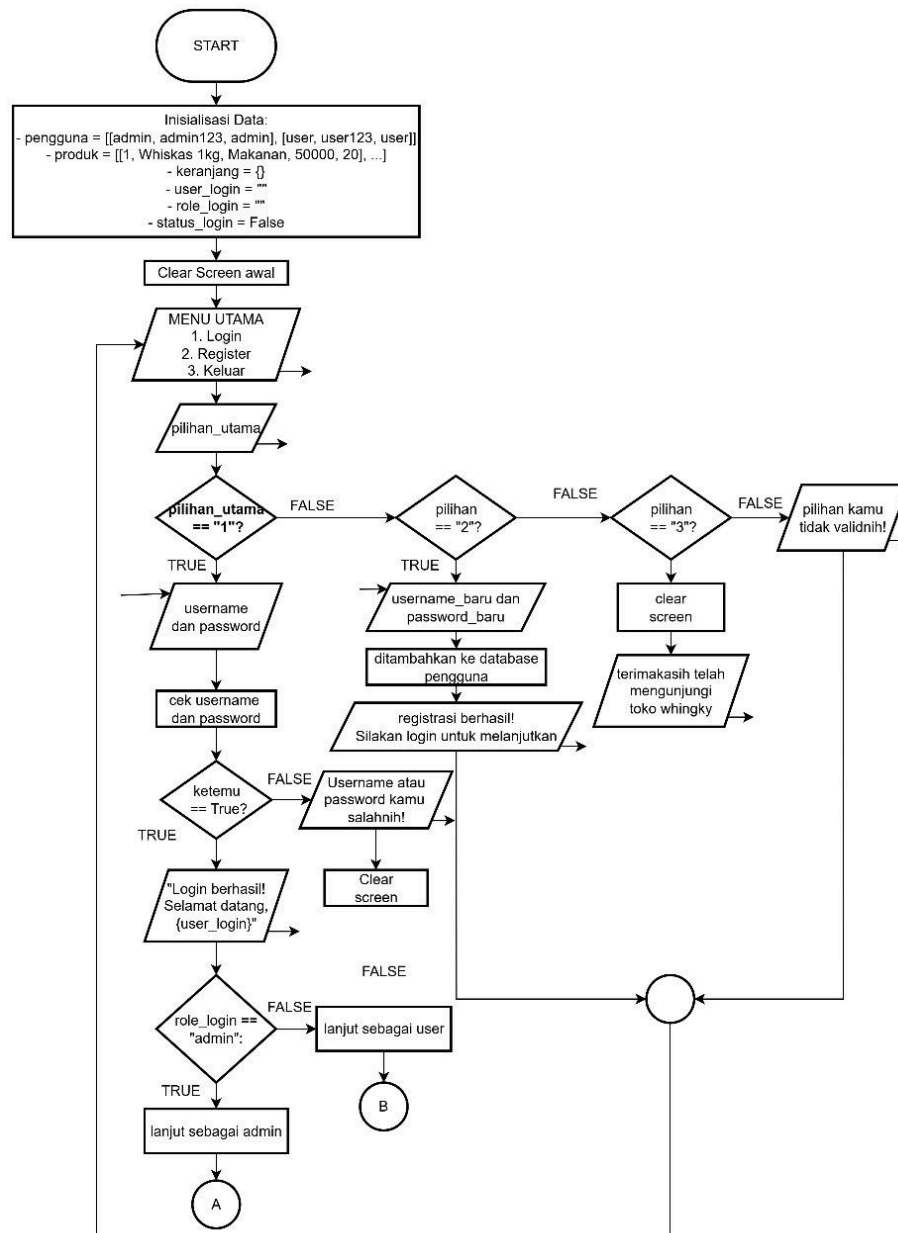
# DAFTAR ISI

<b>DAFTAR ISI.....</b>	<b>1</b>
<b>1. Flowchart.....</b>	<b>3</b>
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
<b>2. Deskripsi Singkat Program.....</b>	<b>14</b>
<b>3. Source Code.....</b>	<b>15</b>
<b>4. Hasil Output.....</b>	<b>27</b>
<b>4.1 Sebagai Admin.....</b>	<b>27</b>
Gambar 4.1.1 Ketika login sebagai admin di pilihan 1 (login).....	27
Gambar 4.1.2 tampilan layar bila jadi admin.....	27
Gambar 4.1.3 output pilihan 1 (melihat semua produk).....	28
Gambar 4.1.4 output pilihan 2 (menambahkan produk baru pada etalase toko).....	28
Gambar 4.1.5 output pilihan 1 (udah ditambahin si shampoo kutu).....	28
Gambar 4.1.6 output pilihan 3 (jadi bisa memperbarui nama/ kategori/ harga/ stok).	29
Gambar 4.1.7 output pilihan 4 (admin dapat menghapus apapun yang ada di etalase)	30
Gambar 4.1.8 output pilihan 5 (admin bisa mencari produk lebih cepat dengan fitur ini).....	30
<b>4.2 sebagai user.....</b>	<b>31</b>
Gambar 4.2.1 output pilihan 2 sedari awal dengan tujuan daftar akun baru.....	31
Gambar 4.2.1.a output pilihan 2 sedari awal dengan tujuan daftar akun baru.....	31
Gambar 4.2.2 Ketika login sebagai user (kalau registrasi itu akunnya nanti jadi user) di pilihan 1 (login).....	31
Gambar 4.2.3 output pilihan 1 (melihat semua produk).....	32
Gambar 4.2.4 output pilihan 2 (fitur mencari produk).....	32
Gambar 4.2.5 output pilihan 3 (jadi user bisa nambahin barang kekeranjang sepuasnya *asalkan stoknya cukup yaa).....	32
Gambar 4.2.6 output pilihan 4 (lihat isi keranjang dan dihitung subtotal belanjanya berapa).....	33
Gambar 4.2.7 output pilihan 5 (fitur menghapus produk pilihan yang ada di keranjang).....	33
Gambar 4.2.8 output pilihan 6 (fitur checkout).....	33
Gambar 4.2.9 jika pengguna memilih pilihan 3 sejak awal.....	33
<b>5. Langkah-langkah GIT.....</b>	<b>34</b>

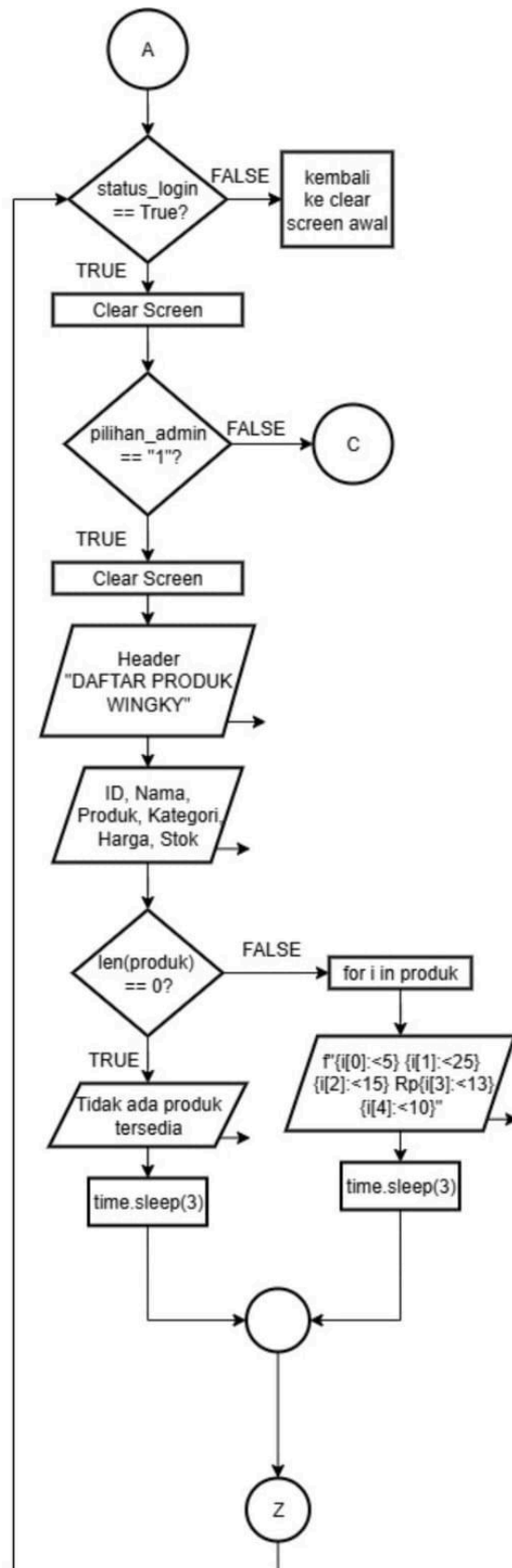
5.1 GIT Add.....	34
Gambar 5.1.1 Penggunaan Git Add.....	34
5.2 GIT Commit.....	34
Gambar 5.2.1 Penggunaan Git Commit.....	34
5.3 GIT Push.....	35
Gambar 5.3.1 penggunaan Git Push untuk menghubungkan repository lokal ke github.....	35

## 1. Flowchart

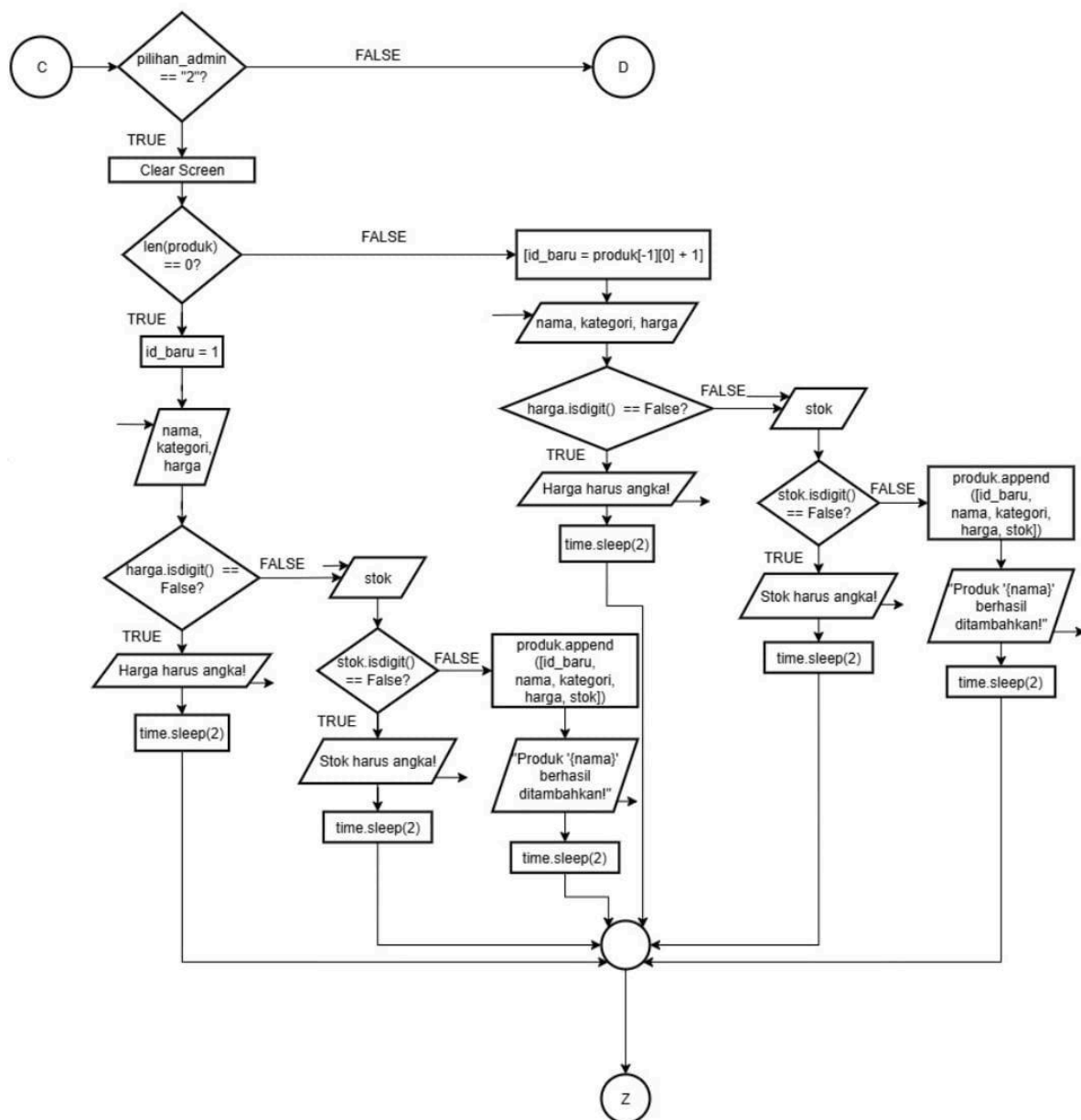
Flowchart merupakan diagram yang biasa digunakan dalam dunia informatika dengan tujuan agar dapat menggambarkan dengan lebih jelas dari suatu langkah-langkah, keputusan, dan alur kerja dari suatu proses dalam sistem. Flowchart terdiri dari berbagai simbol-simbol dengan makna tersendiri yang digunakan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



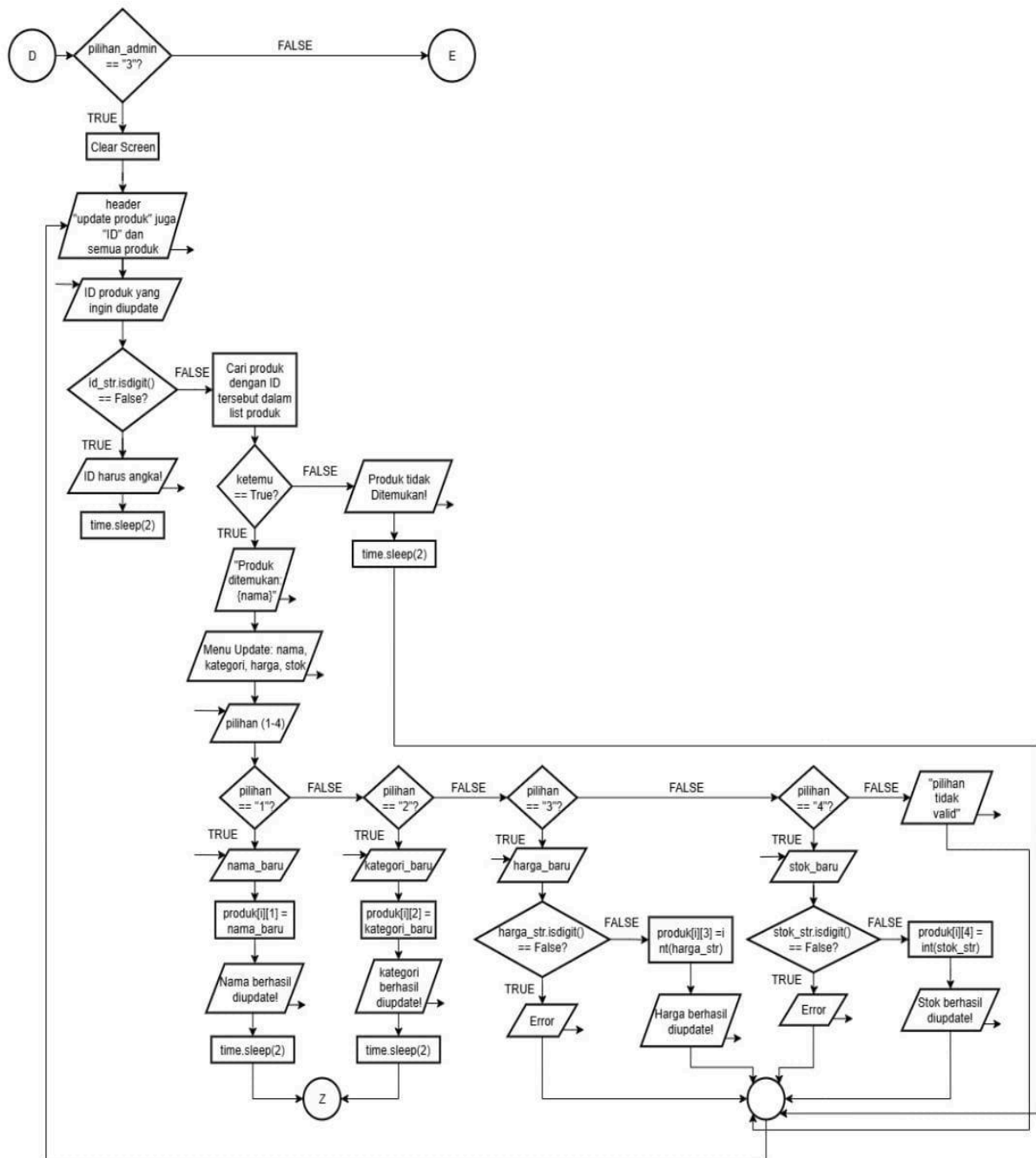
Gambar 1.1 Flowchart bagian awal program toko makanan kucing whinky



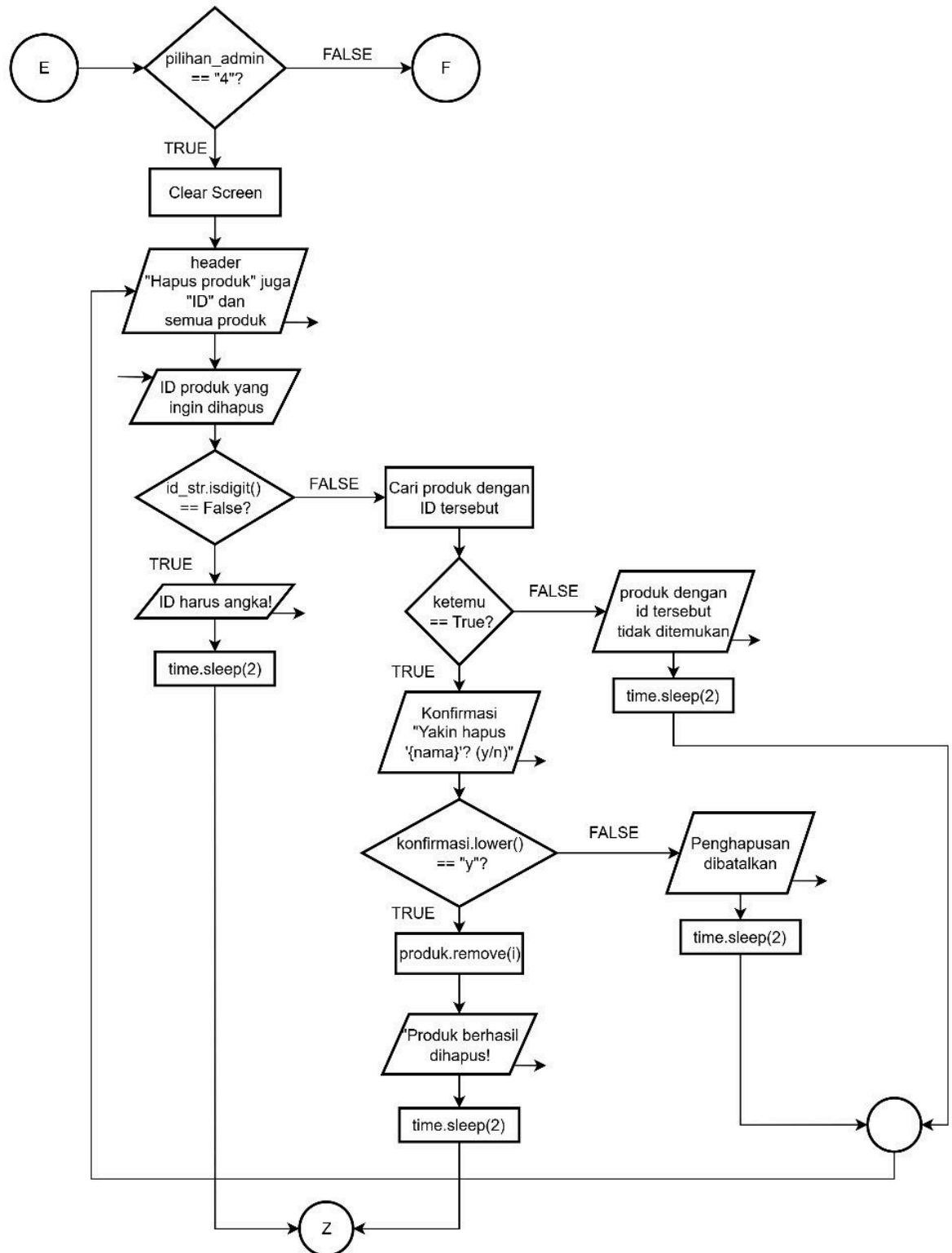
Gambar 1.2 Flowchart bagian on page A untuk program toko makanan kucing whinky



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

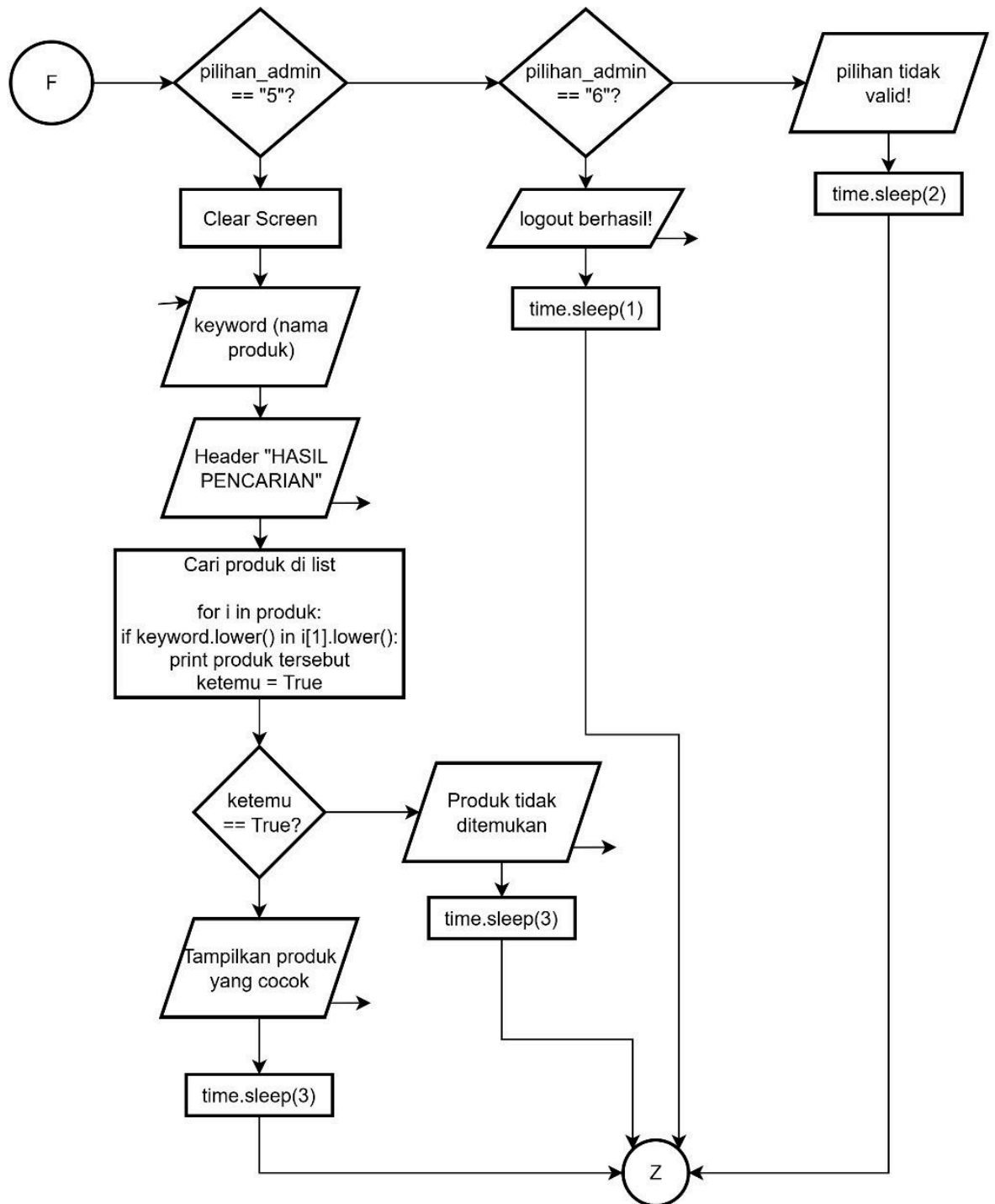


**Gambar 1.4 Flowchart bagian on page D untuk program toko makanan kucing whinky**

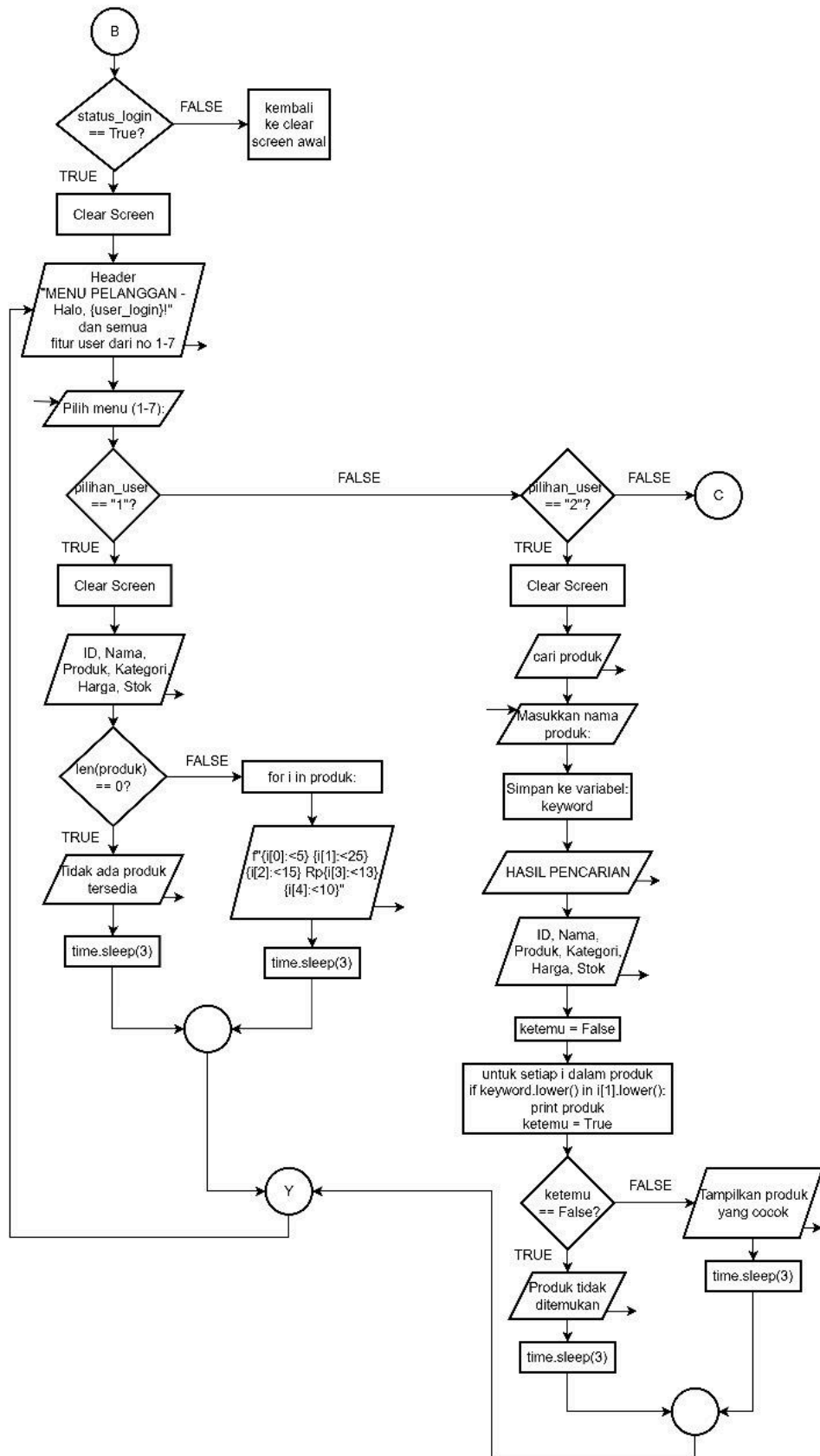


**Gambar 1.5 Flowchart bagian on page E untuk program toko makanan kucing whinky**

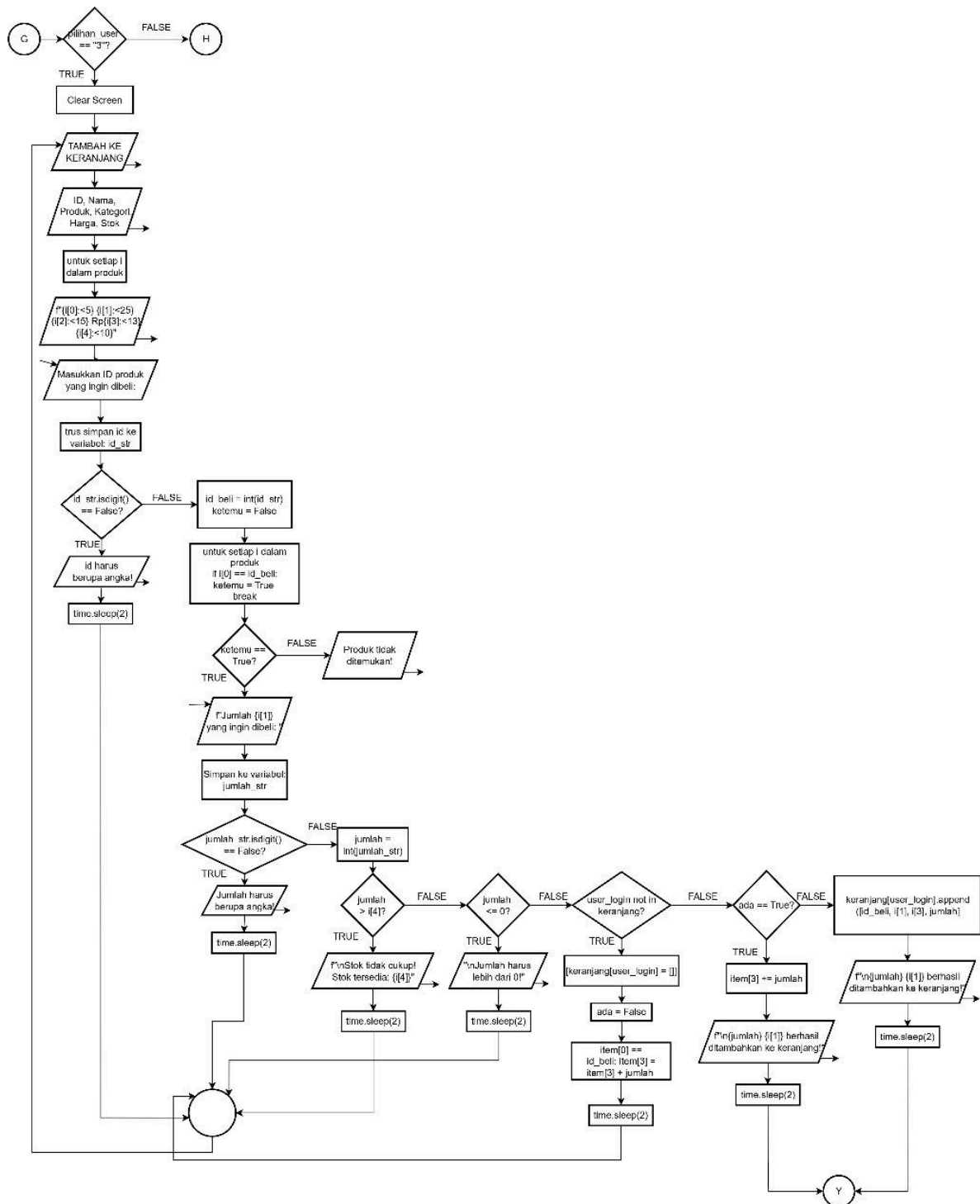




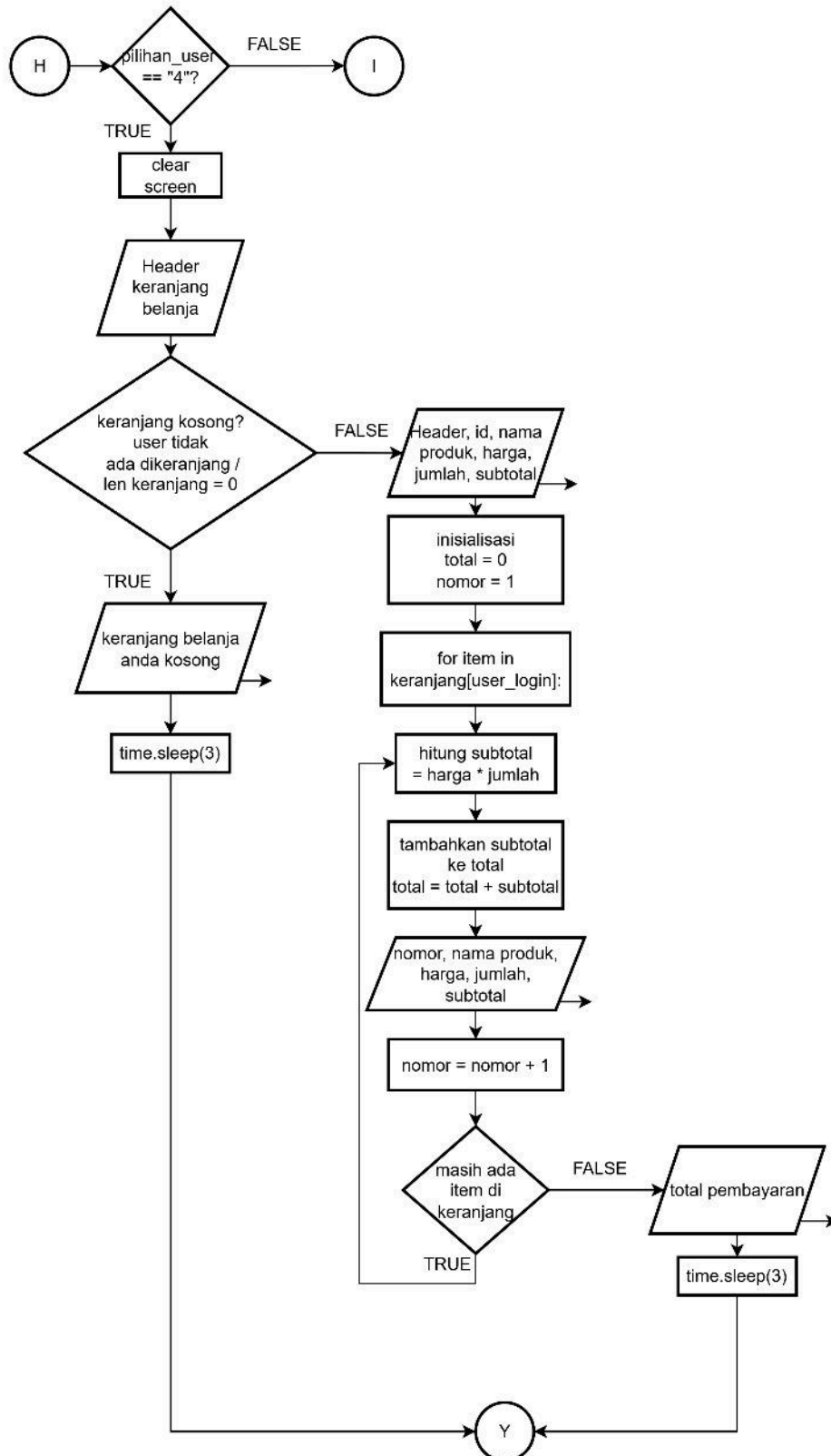
**Gambar 1.6 Flowchart bagian on page F untuk program toko makanan kucing whinky**



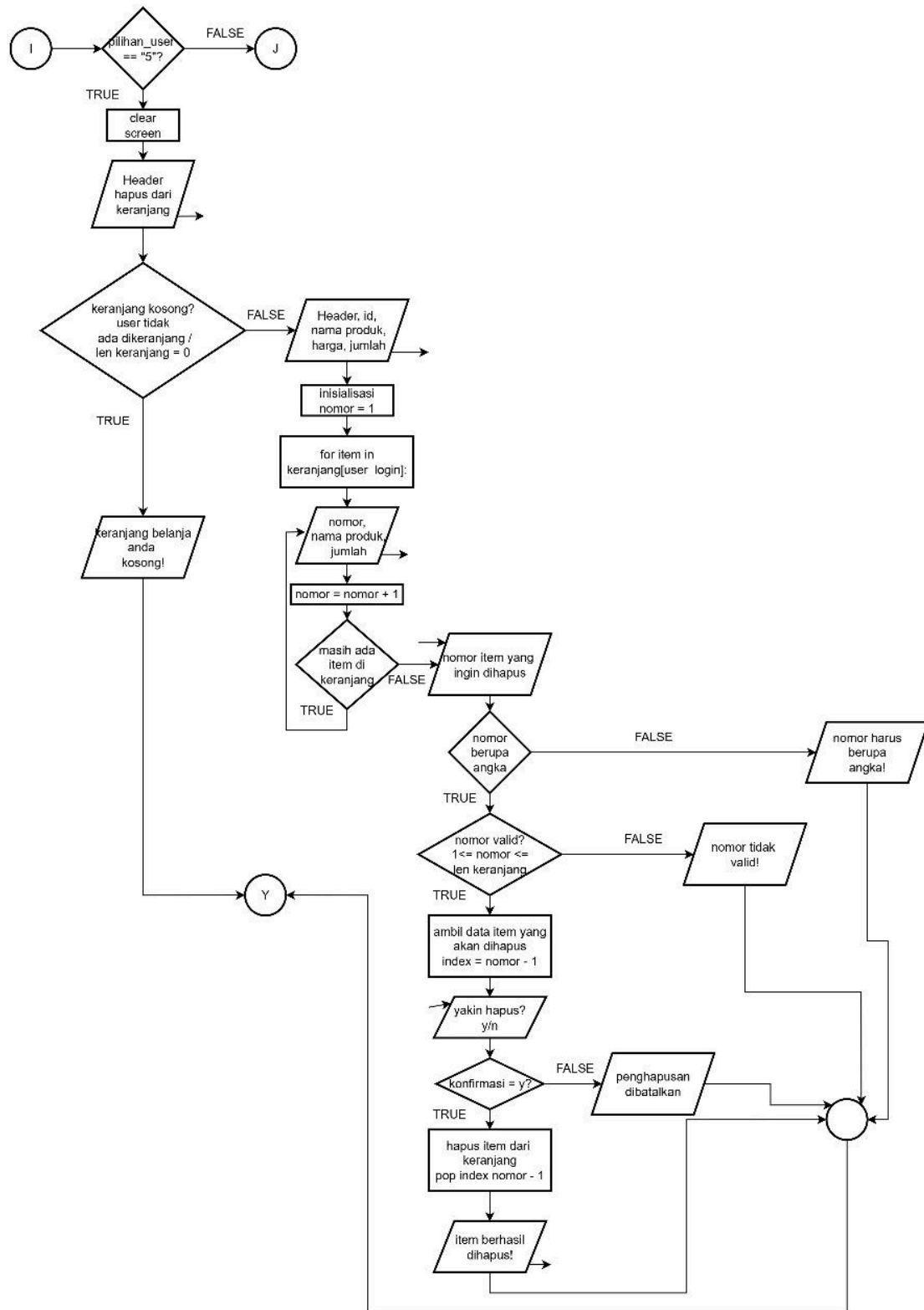
Gambar 1.7 Flowchart bagian on page B untuk program toko makanan kucing whinky



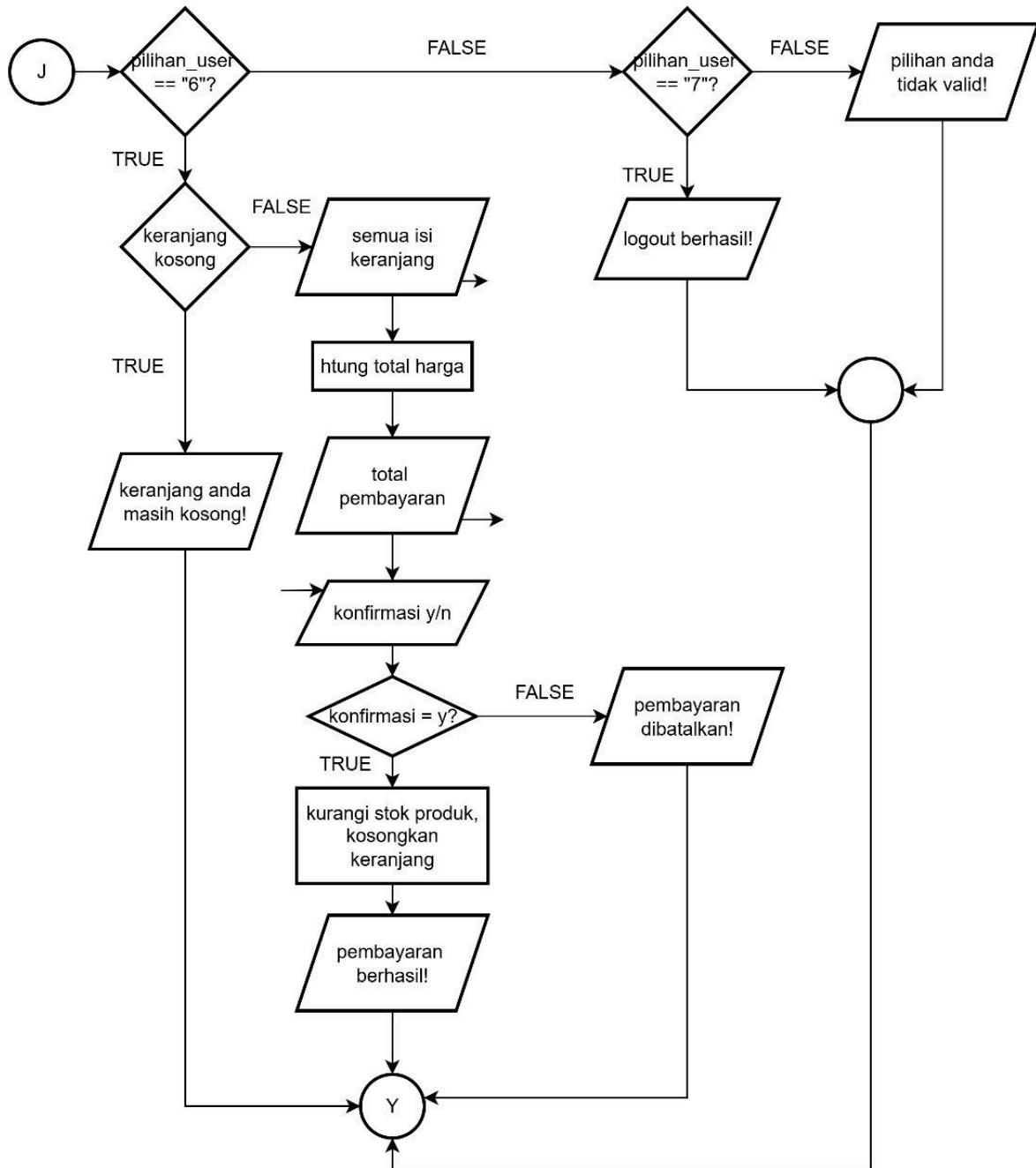
**Gambar 1.8 Flowchart bagian on page G untuk program toko makanan kucing whinky**



**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 fitur 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 sekaligus menghitung subtotal sementara dari belanjanya user), pilihan 5 (hapus dari keranjang disini maksudnya apabila user tidak jadi membeli produk di toko whinky 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 pesannya 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 genggam 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 posttest ke 7 ini saya hanya menyesuaikan beberapa bagian agar sesuai dengan modul terbaru juga perintah yang diminta.

```
import os
import time

# cara bacanya {username: {"password": password, "role": role}}
pengguna = { #ini variabel global 1
    "gea": {
        "password": "pacar seokjin",
        "role": "admin"
    },
    "user": {
        "password": "user123",
        "role": "user"
    }
}

# cara bacanya: {id: {"nama": nama, "kategori": kategori, "harga": harga, "stok": stok}}
produk = { #variabel global 2
    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}
}

keranjang = {} #variabel global 3

# 3 Variabel dibawah ini dipakai untuk ngecek user yang login
user_login = "" #variabel global 4
role_login = "" #variabel global 5
status_login = False #variabel global 6

# 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
            hasil = int(input_str) #ini variabel lokal
            return hasil
        except ValueError:
            print(f"\n{pesan_error}")
            time.sleep(1)
            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("=" * 80)
    print("|                                DAFTAR PRODUK WINGKY                                |")
    print("=" * 80)
    print(f'|{"ID":<5}| {"Nama Produk":<25}| {"Kategori":<15}| {"Harga":<15}| {"Stok":<10}|'|)
    print("-" * 80)

    if len(produk) == 0:
        print("Tidak ada produk tersedia.")
    else:
        for id_produk, data in produk.items(): #ini variabel lokal yang ada dalam loop
            print(f'|{"id_produk":<5}| {data["nama"]:<25}| {data["kategori"]:<15}| Rp{data["harga"]:<13}| {data["stok"]:<10}|'|)

    print("=" * 80)

def tampilkan_isi_keranjang(): #prosedur ke 2
    print("=" * 80)
    print("|                                KERANJANG BELANJA                                |")
    print("=" * 80)

    if user_login not in keranjang or len(keranjang[user_login]) == 0:
        print("\nKeranjang belanja kosong.")
        print("=" * 80)
    else:
        print(f'| {"No":<5}| {"Nama Produk":<25}| {"Harga":<15}| {"Jumlah":<10}| {"Subtotal":<12}| '|)

```

```

print("-" * 80)

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

for id_produk, item in keranjang[user_login].items(): #ini ada pakai variabel lokal
    subtotal = item["harga"] * item["jumlah"] #ini ada pakai variabel lokal
    list_items.append(item)
    print(f'| {nomor:<5}| {item["nama"]:<25}| Rp {item["harga"]:<13}| {item["jumlah"]:<10}|
Rp {subtotal:<12}|')
    nomor += 1

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

print("-" * 80)
print(f'| {'TOTAL':<56} Rp {total:<14} |')
print("=" * 80)

while True: #program utamanya
    tampilkan_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":
        tampilkan_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 = input("Password: ") #ini variabel lokal
            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(f"|          TAMBAH PRODUK BARU          |")
        print("=" * 50)

        try: # Error handling untuk bagian tambah 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 berupa 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(2)

except ValueError as e:
    print(f"\nError: {e}")
    time.sleep(2)

# fitur update produknya
elif pilihan_admin == "3":
    os.system('cls || clear')
    print("=" * 50)
    print("|          UPDATE PRODUK          |")
    print("=" * 50)
    print(f"|{'ID':<5}| {'Nama Produk':<41}|")
    print("=" * 50)
    for id_produk, data in produk.items():
        print(f"|{'id_produk':<5}| {'data['nama']':<41}|")
        print("=" * 50)

    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 diatas merupakan 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(3)

except (ValueError, KeyError) as e:
    print(f"\nError: {e}")
    time.sleep(3)

# fitur hapus produk
elif pilihan_admin == "4":
    os.system('cls || clear')
    print("=" * 50)
    print("|          HAPUS PRODUK          |")
    print("=" * 50)
    print(f"| {ID':<5}| | {Nama Produk':<37}|")
    print("-" * 50)
    for id_produk, data in produk.items():
        print(f"| {id_produk:<5}| | {data['nama']:<37}|")
        print("-" * 50)

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("=" * 50)
    print("|          CARI PRODUK          |")
    print("=" * 50)
    keyword = input("\nMasukkan nama produk: ")

    print("\n" + "=" * 80)
    print("|          HASIL PENCARIAN          |")
    print("=" * 80)
    print(f"| { 'ID':<2} | { 'Nama Produk':<25} | { 'Kategori':<10} | { 'Harga':<15} | { 'Stok':<10} |")
    print(f"| { 'Rp {data['harga']:<13} | {data['stok']:<10} |")

    print("-" * 80)

    ketemu = False
    for id_produk, data in produk.items():
        if keyword.lower() in data['nama'].lower():
            print(f"| {id_produk:<2} | {data['nama']:<25} | {data['kategori']:<10} |")
            print(f"| { 'Rp {data['harga']:<13} | {data['stok']:<10} |")
            ketemu = True

    if ketemu == False:
        print("Produk tidak ditemukan.")

    print("=" * 80)
    time.sleep(4)

# fitur buat log out
elif pilihan_admin == "6":
    print("\nLogout berhasil!")
    time.sleep(1)
    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')
    tampilkan_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: ")

    print("\n" + "=" * 80)
    print("|          HASIL PENCARIAN          |")
    print("=" * 80)
    print(f'{"ID":<5} {"Nama Produk":<25} {"Kategori":<15} {"Harga":<15} {"Stok":<10}"')
    print("-" * 80)

    ketemu = False
    for id_produk, data in produk.items():
        if keyword.lower() in data['nama'].lower():
            print(f'| {"id_produk":<2} | {"data["nama"]:<25} | {"data["kategori"]:<10} | '
                  f'Rp{"data["harga"]:<13} | {"data["stok"]:<10} |')
            ketemu = True

    if ketemu == False:
        print("Produk tidak ditemukan.")

    print("=" * 80)
    time.sleep(4)

# fitur tambahkan produk ke keranjang
elif pilihan_user == "3":
    os.system('cls || clear')
    print("=" * 80)
    print("|          TAMBAH KE KERANJANG          |")
    print("=" * 80)
    print(f'{"ID":<1} | {"Nama Produk":<26} | {"Kategori":<15} | {"Harga":<15} | {"Stok":<10}|')
    print("-" * 80)

```

```

        for id_produk, data in produk.items():
            print(f"| {id_produk:<1}| {data['nama']:<26}| {data['kategori']:<15}|
Rp {data['harga']:<13}| {data['stok']:<10}| |")
            print("=" * 80)

        try: # Error handling untuk fitur bagian tambah produk ke keranjang
            id_beli = validasi_input_angka("\nMasukkan ID produk yang ingin dibeli: ", "ID harus
berupa angka!")

            #line diatas ini menggunakan fungsi 1 dengan parameter dan merupakan variabel lokal
            if id_beli is None:
                raise ValueError("Input dibatalkan")

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

            jumlah = validasi_input_angka(f"Jumlah {produk[id_beli]['nama']} yang ingin dibeli: ",
"Jumlah harus berupa angka!")
            #line diatas merupakan variabel lokal dan menggunakan fungsi 1 dengan parameter
            if jumlah is None:
                raise ValueError("Input dibatalkan")

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

            if jumlah <= 0:
                raise ValueError("Jumlah harus lebih dari 0!")

            # Inisialisasi keranjang
            if user_login not in keranjang:
                keranjang[user_login] = {}

            # disini bakal di cek apakah produk yang di add user sudah ada di keranjang
            if id_beli in keranjang[user_login]:
                keranjang[user_login][id_beli]["jumlah"] += jumlah
            else:
                keranjang[user_login][id_beli] = {
                    "nama": produk[id_beli]["nama"],
                    "harga": produk[id_beli]["harga"],
                    "jumlah": jumlah
                }

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

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

        # fitur lihat isi keranjang
        elif pilihan_user == "4":
            os.system('cls || clear')

```



```

        tampilkan_isi_keranjang() #ini pakai prosedur 2 yang ada rekursifnya tadi
        time.sleep(7)

# fitur hapus produk dari keranjang
elif pilihan_user == "5":
    os.system('cls || clear')
    print("=" * 50)
    print("|          HAPUS DARI KERANJANG          |")
    print("=" * 50)

    if user_login not in keranjang or len(keranjang[user_login]) == 0:
        print("\nKeranjang belanja kosong.")
        time.sleep(4)
    else:
        print(f" {'No':<5} {'Nama Produk':<25} {'Jumlah':<10}")
        print("-" * 50)

        nomor = 1
        list_id = list(keranjang[user_login].keys())
        for id_produk in list_id:
            item = keranjang[user_login][id_produk]
            print(f" {'nomor':<5} {'item['nama']':<25} {'item['jumlah']':<10}")
            nomor += 1

        try: # Error handling untuk fitur hapus produk dari keranjang
            no_hapus = validasi_input_angka("\nMasukkan nomor item yang ingin dihapus: ",
            "Nomor harus berupa angka!")
            # line diatas ini pakai fungsi 1 dengan parameter
            if no_hapus is None:
                raise ValueError("Input dibatalkan")

            if no_hapus < 1 or no_hapus > len(keranjang[user_login]):
                raise ValueError("Nomor tidak valid!")

            id_hapus = list_id[no_hapus - 1]
            nama_item = keranjang[user_login][id_hapus]["nama"]
            konfirmasi = input(f"\nYakin ingin menghapus '{nama_item}' dari keranjang? (y/n): ")

            if konfirmasi.lower() == "y":
                del keranjang[user_login][id_hapus]
                print("\nItem berhasil dihapus dari keranjang!")
            else:
                print("\nPenghapusan dibatalkan.")
            time.sleep(4)

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

# fitur checkout keranjang
elif pilihan_user == "6":

```

```

os.system('cls || clear')
print("=" * 80)
print("|                                CHECKOUT                                |")
print("=" * 80)

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:
    print(f"| {'No':<2} | {'Nama Produk':<25} | {'Harga':<15} | {'Jumlah':<3} | {'Subtotal':<15} |")
    print("-" * 80)

    nomor = 1
    list_items = []

    for id_produk, item in keranjang[user_login].items():
        subtotal = item["harga"] * item["jumlah"]
        list_items.append(item) #ini list untuk fungsi rekursif
        print(f"| {'nomor':<2} | {'item['nama']':<25} | Rp {item['harga']:<13} | {item['jumlah']:<5} | Rp {subtotal:<15} |")
        nomor += 1

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

    print("-" * 80)
    print(f"| {'TOTAL PEMBAYARAN':<56} Rp {total:<15} |")
    print("=" * 80)

    konfirmasi = input("\nLanjutkan pembayaran? (y/n): ")

    if konfirmasi.lower() == "y":
        # Kurangi stok produk setelah di co
        for id_produk, item in keranjang[user_login].items():
            produk[id_produk]["stok"] -= item["jumlah"]

        # Kosongkan keranjang setelah di co
        keranjang[user_login] = {}

        print("\n" + "=" * 80)
        print("                                PEMBAYARAN BERHASIL!                                ")
        print("                                Terima kasih telah berbelanja di Wingky!                                ")
        print("=" * 80)
        time.sleep(8)
    else:
        print("\nPembayaran dibatalkan.")
        time.sleep(4)

# fitur log out
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 user baru
elif pilihan_utama == "2":
    tampilkan_header_utama()
    print("\n----- REGIST SECTION -----")

    try: # Error handling untuk register user baru
        username_baru = input("Username: ") #ini variabel lokal
        if not username_baru:
            raise ValueError("Username tidak boleh kosong")
        if username_baru in pengguna:
            raise ValueError("Username sudah digunakan")

        password_baru = input("Password: ") #ini variabel lokal
        if not password_baru:
            raise ValueError("Password tidak boleh kosong")

        pengguna[username_baru] = {
            "password": password_baru,
            "role": "user"
        }
        print(f"\nRegistrasi berhasil! Akun '{username_baru}' telah dibuat.")
        print("Silakan login untuk melanjutkan.")
        time.sleep(4)

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

# bagian keluar dari menu utama
elif pilihan_utama == "3":
    os.system('cls || clear')
    print("=" * 60)
    print("|      Terima kasih telah mengunjungi Toko Wingky!      |")
    print("=" * 60)
    break

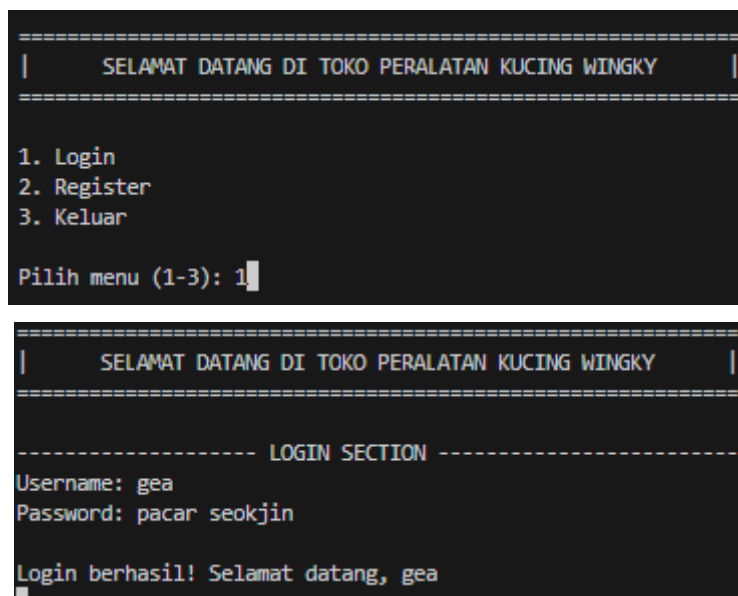
```

```
else:  
    print("\nPilihan kamu tidak valid nih!")  
    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

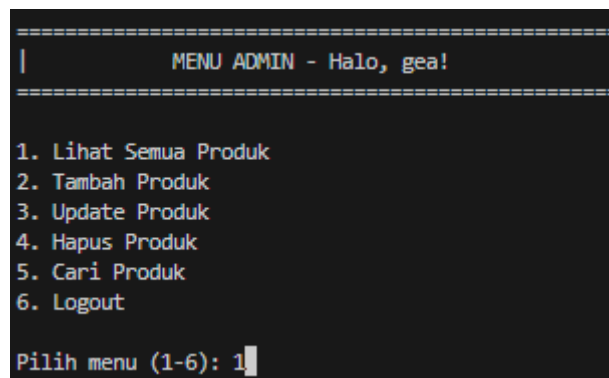
Pilih menu (1-3): 1

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

----- LOGIN SECTION -----
Username: gea
Password: pacar seokjin

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	Rp50000	20
2	Royal Canin 2kg	Makanan	Rp150000	15
3	Pasir Gumpal 5kg	Kebersihan	Rp45000	30
4	Sisir Kucing	Grooming	Rp25000	50
5	Mainan Bola	Mainan	Rp15000	40

Gambar 4.1.3 output pilihan 1 (melihat semua produk)

```

=====
|                                |
=====
Nama Produk: shampoo kutu
Kategori (Makanan/Kebersihan/Grooming/Mainan): Grooming
Harga: 39900
Stok: 10

Produk 'shampoo 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	Rp50000	20
2	Royal Canin 2kg	Makanan	Rp150000	15
3	Pasir Gumpal 5kg	Kebersihan	Rp45000	30
4	Sisir Kucing	Grooming	Rp25000	50
5	Mainan Bola	Mainan	Rp15000	40
6	shampoo kutu	grooming	Rp39900	10

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

```
=====
|          UPDATE PRODUK          |
=====
|ID  | Nama Produk |
=====
|1   | Whiskas 1kg |
=====
|2   | Royal Canin 2kg |
=====
|3   | Pasir Gumpal 5kg |
=====
|4   | Sisir Kucing |
=====
|5   | Mainan Bola |
=====
|6   | shampoo kutu |
=====

Masukkan ID produk yang ingin diupdate: 6

Produk ditemukan: shampoo kutu

Pilih yang ingin diupdate:
1. Nama
2. Kategori
3. Harga
4. Stok
Pilihan: 4
Stok baru: 15

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  | shampoo kutu |
-----

Masukkan ID produk yang ingin dihapus: 6

Yakin ingin menghapus 'shampoo 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 | Rp50000 | 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          |
=====

----- REGIST SECTION -----
Username: wwh
Password: seokjin

Registrasi berhasil! Akun 'wwh' telah dibuat.
Silakan login untuk melanjutkan.
```

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

```
=====
|          MENU PELANGGAN - Halo, wwh!          |
=====
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):
```

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	Rp50000	20
2	Royal Canin 2kg	Makanan	Rp150000	15
3	Pasir Gumpal 5kg	Kebersihan	Rp45000	30
4	Sisir Kucing	Grooming	Rp25000	50
5	Mainan Bola	Mainan	Rp15000	40

Gambar 4.2.3 output pilihan 1 (melihat semua produk)

CARI PRODUK				
Masukkan nama produk: pasir				
HASIL PENCARIAN				
ID	Nama Produk	Kategori	Harga	Stok
3	Pasir Gumpal 5kg	Kebersihan	Rp45000	30

Gambar 4.2.4 output pilihan 2 (fitur mencari produk)

TAMBAH KE KERANJANG				
ID	Nama Produk	Kategori	Harga	Stok
1	Whiskas 1kg	Makanan	Rp50000	20
2	Royal Canin 2kg	Makanan	Rp150000	15
3	Pasir Gumpal 5kg	Kebersihan	Rp45000	30
4	Sisir Kucing	Grooming	Rp25000	50
5	Mainan Bola	Mainan	Rp15000	40

Masukkan ID produk yang ingin dibeli: 2  
Jumlah Royal Canin 2kg yang ingin dibeli: 2  
2 Royal Canin 2kg berhasil ditambahkan ke keranjang!

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



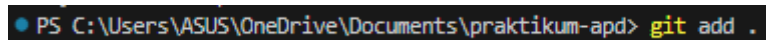
Jadi apabila sedari awal pengguna memilih registrasi pada pilihan 2 itu sudah pasti jadi user karna adminnya hanya 1 orang di program ini (bisa ditambah sih kalau mau), lalu kalau pengguna selalu memilih yang tidak ada di pilihan pasti akan keluar output eror yang beragam (seperti outputan eror yang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

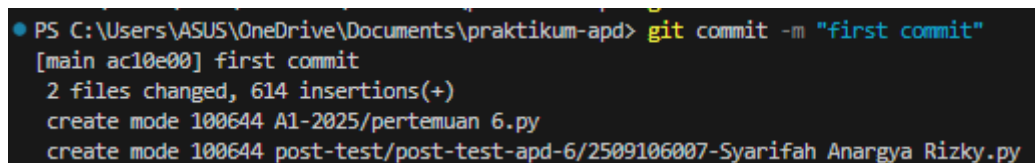


```
PS C:\Users\ASUS\OneDrive\Documents\praktikum-apd> git add .
```

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).

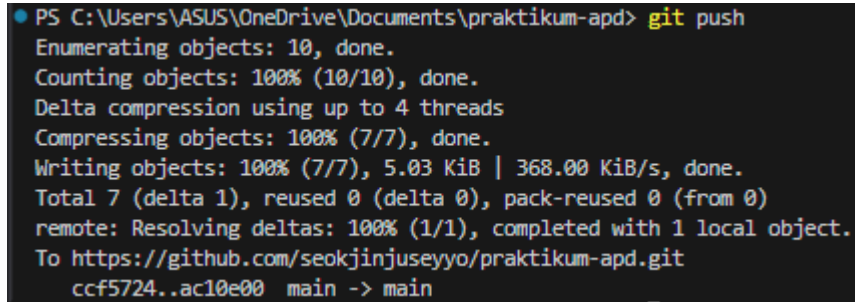


```
PS C:\Users\ASUS\OneDrive\Documents\praktikum-apd> git commit -m "first commit"
[main ac10e00] first commit
2 files changed, 614 insertions(+)
create mode 100644 A1-2025/pertemuan 6.py
create mode 100644 post-test/post-test-apd-6/2509106007-Syarifah Anargya Rizky.py
```

Gambar 5.2.1 Penggunaan Git Commit

### 5.3 GIT Push

Git push digunakan untuk mengirim commit kita ke github nanti, jadi disambungkan gitu repository lokal kita ke github.



```
PS C:\Users\ASUS\OneDrive\Documents\praktikum-apd> git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 5.03 KiB | 368.00 KiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/seokjinjuseyyo/praktikum-apd.git
ccf5724..ac10e00 main -> main
```

**Gambar 5.3.1** penggunaan Git Push untuk menghubungkan repository lokal ke github