LAPORAN TUGAS BESAR

“ALGORITMA DAN PEMROGRAMAN”

“Aplikasi Sederhana Toko Online”

Disusun Oleh:

Azaila Dwi Putri Fajawati	**102092400032**
Siska Yulianti	**102092400041**
Monica Khirani Triastary	**102092400131**

PROGRAM STUDI S1 SISTEM INFORMASI

FAKULTAS REKAYASA INDUSTRI

TELKOM UNIVERSITY PURWOKERTO

2024

PEMBAHASAN

- **File Halaman main.py**

```
import halaman_admin
import halaman_pembeli
def login(username, password):
    sukses = False
    role = None
    with open("logindatabase.txt", "r") as file:
        for i in file:
            i = i.strip()
            if "," in i:
                upr = i.split(",", 2)
                if len(upr) == 3:
                    u, p, r = upr
                    p = p.strip()
                    if u == username and p == password:
                        sukses = True
                        role = r.strip()
                        break

    if sukses:
        print(f"Login Berhasil. Selamat Datang {username} dengan
role {role}")
        if role == "admin":
            halaman_admin.halaman_admin(username)
        elif role == "pembeli":
            halaman_pembeli.halaman_pembeli(username)
    else:
        print("Login gagal, username atau password salah atau
belum mempunyai akun")

def cek_username(username):
    with open("logindatabase.txt", "r") as file:
        for i in file :
            i = i.strip()
            if  i:
                upr = i.split(",", 2)
                if len(upr) == 3:
                    u, _, _ = upr
                    if u == username:
                        return True
        return False

def register(username, password):
    if cek_username(username):
```

```
        akses("register")
        return False
    else:
        role = input("Masukkan role Anda (admin atau
pembeli):").lower()
        with open("logindatabase.txt", "a") as file:
            file.write( "\n" + username + "," + password + "," +
role)
        print("Register berhasil, silahkan masuk")
        return True

def akses(option):
    if option == "login":
        username = input("Masukkan Username: ")
        password = input("Masukkan Password: ")
        login(username, password)

    else:
        while True:
            username = input("Masukkan username: ")
            if cek_username(username):
                print("Username sudah terdaftar, pilih username
yang lain!")
            else:
                break

        password = input("Masukkan Password: ")
        register(username, password)

        pilihanlogin = input("Apakah ingin login sekarang
(ya/tidak)? ")
        if pilihanlogin.lower() == "ya":
            akses("login")

def awal():
    print("Selamat Datang Di Toko Online!")
    print("Ketik 'login' jika sudah mempunyai akun")
    print("Ketik 'reg' jika belum mempunyai akun")
    option = input("Silahkan pilih (login/reg) :").lower()
    if option != "login" and option != "reg":
        print("Pilihan tidak valid, silahkan coba lagi!")
        awal()
    else:
        akses(option)
awal()
```

✓ **Output**

Register :

```
Selamat Datang Di Toko Online!
Ketik 'login' jika sudah mempunyai akun
Ketik 'reg' jika belum mempunyai akun
Silahkan pilih (login/reg) :reg
Masukkan username: aza
Username sudah terdaftar, pilih username yang lain!
Masukkan username: azailaa
Masukkan Password: 12345
Masukkan role Anda (admin atau pembeli):admin
Register berhasil, silahkan masuk
Apakah ingin login sekarang (ya/tidak)?
```

Login :

```
Apakah ingin login sekarang (ya/tidak)? ya
Masukkan Username: azailaa
Masukkan Password: 12345
Login Berhasil. Selamat Datang azailaa dengan role admin
Halo azailaa, Selamat Datang Di Halaman Admin!
```

```
Selamat Datang Di Toko Online!
Ketik 'login' jika sudah mempunyai akun
Ketik 'reg' jika belum mempunyai akun
Silahkan pilih (login/reg) :login
Masukkan Username: putri02
Masukkan Password: 12345
Login gagal, username atau password salah atau belum mempunyai akun
```

❖ **Penjelasan Code :**

- Import halaman admin dan halaman pembeli berfungsi untuk mengimport modul python yang bernama halaman admin dan halaman pembeli. Berisi fungsi yang menangani logika halaman admin dan juga halaman pembeli.
- Fungsi login dengan parameter username dan password berisi pengecekan data yang disimpan dalam file logindatabase.txt. Memeriksa apakah username dan password cocok dengan data yang berada di file logindatabase.txt. Jika cocok, login berhasil dan menentukan role sebagai admin atau pembeli.
- sukses = False untuk menandai apakah login berhasil (True) atau tidak (False) karena belum melakukan login maka ditulis sukses = False.
- role = None untuk menyimpan role pengguna sebagai admin atau pembeli.
- Code with open("logindatabase.txt", "r") as file berfungsi untuk membuka file logindatabase.txt dengan mode r (read) atau baca. File ini berisi username, password, role.
- Code perulangan for i in file berfungsi untuk membaca baris di dalam file satu per satu, i = i.strip() digunakan untuk menghapus spasi atau karakter yang tidak diperlukan.

- Code if "," in i: dan upr = i.split(",", 2) berfungsi untuk melakukan pengecekan apakah baris mengandung koma (,), jika iya baris dipisahkan menjadi tiga bagian dengan split. “u” = username, “p” = password, “r” = role.
- Code if len(upr) == 3 berfungsi untuk memastikan baris hanya berisi tiga elemen, yaitu username, password, dan role.
- Pada code selanjutnya memeriksa u = username dan p = password jika cocok maka login berhasil dan variabel sukses menjadi True, serta role diisi dengan role pengguna. Selanjutnya keluar dari perulangan dengan break.
- Code if sukses berfungsi untuk memeriksa apakah login berhasil, jika login berhasil maka akan muncul print yang menyatakan login telah berhasil.
- Melakukan pengecekan role apakah login sebagai admin atau sebagai pembeli, jika login sebagai admin maka diarahkan ke halaman admin dan memanggil fungsi halaman_admin.halaman_admin(username) dari modul halaman_admin, fungsi halaman_admin akan menampilkan halaman admin dan fitur yang hanya bisa diakses oleh admin dan jika login sebagai pembeli diarahkan ke halaman pembeli dan memanggil fungsi halaman_pembeli.halaman_pembeli(username) dari modul halaman_pembeli, fungsi halaman_pembeli akan menampilkan halaman_pembeli dan fitur yang hanya bisa diakses oleh halaman_pembeli. Jika login gagal (False) akan menampilkan print bahwa login gagal.
- Fungsi cek_username digunakan untuk memeriksa apakah username sudah terdaftar dalam database login yang disimpan dalam file logindatabase.txt. Fungsi ini akan mengembalikan True jika username sudah ada dalam database, dan False jika username belum terdaftar.
- Fungsi register digunakan untuk mendaftarkan pengguna baru dengan menyimpan data (username, password, dan role) ke dalam file logindatabase.txt. Kemudian memeriksa apakah username sudah terdaftar sebelumnya, code if cek_username(username) digunakan untuk memeriksa apakah username sudah terdaftar di dalam database, jika sudah terdaftar pengguna dapat memasukkan username lain melalui fungsi akses("register") dan jika belum terdaftar, pengguna dapat melanjutkan dengan memilih role (admin atau pembeli).
- Jika username belum terdaftar dalam database, data pengguna (username, password, dan role) ditambahkan dalam file logindatabase.txt menggunakan mode append ("a") dan menampilkan print berhasil melakukan registrasi.
- Fungsi akses(option) berfungsi untuk mengelola login dan pendaftaran pengguna, yaitu opsi login dan register. Jika opsi adalah login maka pengguna akan diminta untuk memasukkan username dan password dengan input.
- Jika opsi selain login, maka dapat diartikan bawa pengguna ingin melakukan registrasi, akan dimita memasukkan username kemudian melakukan pengecekan username melalui fungsi cek_username, jika sudah terdaftar pengguna diminta untuk memasukkan username lain.

Setelah mendapatkan username yang valid, pengguna dapat memasukkan password dan memilih role. Data pengguna baru akan disimpan di file logindatabase.txt

- Setelah selesai melakukan registrasi akan muncul pertanyaan apakah ingin login sekarang atau nanti, jika "ya" maka akan memanggil fungsi akses("login).
- Fungsi awal() digunakan untuk memberikan tampilan awal saat menjalanlan program dengan memberikan pilihan untuk melakukan login atau registrasi. Melakukan input antara login atau reg.
- Jika memasukkan selain login atau reg maka akan menampilkan pilihan tidak valid. Setelah itu memanggil fungsi awal() untuk meminta pengguna memasukkan pilihan yang benar.
- Jika input pengguna valid, yaitu antara login atau reg, maka fungsi akses(option) akan dipanggil untuk memproses login atau registrasi.

➢ **File Halaman admin.py**

```
def halaman_admin(username):
    print(f"Halo {username}, Selamat Datang Di Halaman Admin!")
    while True:
        print("\nMenu Admin")
        print("1. Lihat Produk")
        print("2. Tambah Produk")
        print("3. Edit Produk")
        print("4. Hapus Produk")
        print("5. Keluar")

        pilih_menu = int(input("Pilih Menu gunakan angka: "))
        if pilih_menu == 1:
            data_produk, daftar_harga = baca_produk()
            if not data_produk:
                print("Tidak Ada Produk Yang Tersedia")
            else:
                print("List Produk")
                for produk in data_produk:
                    print(f"ID: {produk} Nama:
{data_produk[produk]}, Harga: {daftar_harga[produk]}")
        elif pilih_menu == 2:
            tambah_produk()
        elif pilih_menu == 3:
             edit_produk()
        elif pilih_menu == 4:
             hapus_produk()
        elif pilih_menu == 5:
            print("Keluar Dari Halaman Admin")
```

```
            break
        else:
            print("Pilihan tidak valid. Silakan coba lagi!")

def baca_produk():
    data_produk = {}
    daftar_harga = {}
    with open("produk.txt", "r") as file:
        for line in file:
            proha = line.strip().split(",")
            if len(proha) == 3:
                id_produk = int(proha[0])
                nama_produk = proha[1]
                harga_produk = int(proha[2])
                data_produk[id_produk] = nama_produk
                daftar_harga[id_produk] = harga_produk
    return data_produk, daftar_harga

def tambah_produk():
    id_produk = int(input("Masukkan ID Produk: "))
    nama_produk = input("Masukkan Nama Produk: ")
    harga = int(input("Masukkan Harga Produk: "))

    with open("produk.txt", "a") as file:
        file.write(f"{id_produk},{nama_produk},{harga}\n")
    print("Produk berhasil ditambahkan")

def edit_produk():
    data_produk, daftar_harga = baca_produk()

    print("Daftar Produk:")
    for id_produk in data_produk:
        print(f"ID: {id_produk}, Nama: {data_produk[id_produk]},
Harga: {daftar_harga[id_produk]}")

    edit = int(input("Masukkan ID produk yang ingin diedit: "))

    if edit in data_produk:

        print(f"Produk ditemukan: ID: {edit}, Nama:
{data_produk[edit]}, Harga: {daftar_harga[edit]}")

        nama_baru = input("Masukkan Nama Produk Baru: ")
        harga_baru = int(input("Masukkan Harga Produk Baru: "))

        data_produk[edit] = nama_baru
        daftar_harga[edit] = harga_baru
```

```
        with open("produk.txt", "w") as file:
            for id_produk in data_produk:
                file.write(f"{id_produk},{data_produk[id_produk]},
{daftar_harga[id_produk]}\n")

        print("Produk Berhasil Diedit!")
    else:
        print("Produk Tidak Ditemukan")

def hapus_produk():
    data_produk, daftar_harga = baca_produk()

    print("\nDaftar Produk:")
    for id_produk in data_produk:
        print(f"ID: {id_produk}, Nama: {data_produk[id_produk]},
Harga: {daftar_harga[id_produk]}")

    hapus = int(input("Masukkan ID Produk Yang Ingin Dihapus: "))

    if hapus in data_produk:
        print(f"Produk Yang Ingin Dihapus: ID: {hapus}, Nama:
{data_produk[hapus]}, Harga: {daftar_harga[hapus]}")

        del data_produk[hapus]

        with open("produk.txt", "w") as file:
            for id_produk in data_produk:
                file.write(f"{id_produk},{data_produk[id_produk]},
{daftar_harga[id_produk]}\n")

        print("Produk berhasil dihapus!")
    else:
        print("Produk tidak ditemukan")
```

✓ **Output**

Lihat Produk :

```
Halo azaila, Selamat Datang Di Halaman Admin!

Menu Admin
1. Lihat Produk
2. Tambah Produk
3. Edit Produk
4. Hapus Produk
5. Keluar
Pilih Menu gunakan angka: 1
List Produk
ID: 1 Nama: Shampoo, Harga: 10000
ID: 2 Nama: Dress, Harga: 150000
ID: 3 Nama: Topi, Harga: 50000
ID: 4 Nama: Earphone, Harga: 100000
ID: 5 Nama: Sepatu, Harga: 300000
```

Tambah Produk :

```
Menu Admin
1. Lihat Produk
2. Tambah Produk
3. Edit Produk
4. Hapus Produk
5. Keluar
Pilih Menu gunakan angka: 2
Masukkan ID Produk: 6
Masukkan Nama Produk: Sabun
Masukkan Harga Produk: 5000
Produk berhasil ditambahkan
```

Edit Produk :

```
Menu Admin
1. Lihat Produk
2. Tambah Produk
3. Edit Produk
4. Hapus Produk
5. Keluar
Pilih Menu gunakan angka: 3
Daftar Produk:
ID: 1, Nama: Shampoo, Harga: 10000
ID: 2, Nama: Dress, Harga: 150000
ID: 3, Nama: Topi, Harga: 50000
ID: 4, Nama: Earphone, Harga: 100000
ID: 5, Nama: Sepatu, Harga: 300000
ID: 6, Nama: Sabun, Harga: 5000
Masukkan ID produk yang ingin diedit: 6
Produk ditemukan: ID: 6, Nama: Sabun, Harga: 5000
Masukkan Nama Produk Baru: Parfum
Masukkan Harga Produk Baru: 80000
Produk Berhasil Diedit!
```

Hapus Produk :

```
Menu Admin
1. Lihat Produk
2. Tambah Produk
3. Edit Produk
4. Hapus Produk
5. Keluar
Pilih Menu gunakan angka: 4

Daftar Produk:
ID: 1, Nama: Shampoo, Harga: 10000
ID: 2, Nama: Dress, Harga: 150000
ID: 3, Nama: Topi, Harga: 50000
ID: 4, Nama: Earphone, Harga: 100000
ID: 5, Nama: Sepatu, Harga: 300000
ID: 6, Nama: Parfum, Harga: 80000
Masukkan ID Produk Yang Ingin Dihapus: 6
Produk Yang Ingin Dihapus: ID: 6, Nama: Parfum, Harga: 80000
Produk berhasil dihapus!

Menu Admin
1. Lihat Produk
2. Tambah Produk
3. Edit Produk
4. Hapus Produk
5. Keluar
Pilih Menu gunakan angka: 1
List Produk
ID: 1 Nama: Shampoo, Harga: 10000
ID: 2 Nama: Dress, Harga: 150000
ID: 3 Nama: Topi, Harga: 50000
ID: 4 Nama: Earphone, Harga: 100000
ID: 5 Nama: Sepatu, Harga: 300000
```

Keluar :

```
Menu Admin
1. Lihat Produk
2. Tambah Produk
3. Edit Produk
4. Hapus Produk
5. Keluar
Pilih Menu gunakan angka: 5
Keluar Dari Halaman Admin
```

❖ **Penjelasan Code:**

- def halaman_admin(username): Fungsi ini menerima parameter yang berfungsi untuk menyapa pengguna, yaitu , dan menampilkan pesan.

- print(f"Halo {username}, Selamat Datang Di Halaman Admin!") untuk mencatak pesan yang menyertakan nama pengguna.
- While True: untuk memulai loop tak terbatas untuk menampilkan menu admin hingga pengguna memilih keluar.
- Pada menu admin ini berfungsi untuk mencetak opsi bagi admin untuk melihat, menambahkan, mengedit, menghapus produk, dan keluar dari program. Pengguna (admin) diminta untuk memilih salah satu opsi dengan memasukkan angka yang sesuai.
- `pilih_menu = int(input("Pilih Menu gunakan angka: "))` berfungsi untuk Mengambil input dai pengguna dan mengubahnya menjadi integer untuk menentukan pilihan menu yang diinginkan.
- `if pilih_menu == 1:` untuk memeriksa apakah pengguna memilih menu 1
- `data_produk, daftar_harga = baca_produk()` untuk memanggil fungsi untuk mendapatkan daftar produk dan harga, yang disimpan dalam dua variable yakni *data_produk* dan *daftar_harga* .
- `if not data_produk:`
  `print("Tidak Ada Produk Yang Tersedia")` untuk memeriksa apakah kosong, jika ya, maka akan mencetak pesan bahwa “tidak ada produk yang tersedia”.
- `else:`
  `print("List Produk")` jika produk tersedia maka akan mencetak “list produk” sebagai pengantar untuk menampilkan produk.
- `for produk in data_produk:`
  `print(f"ID: {produk} Nama: {data_produk[produk]}, Harga: {daftar_harga[produk]}")`
  menggunakan loop untuk mencetak setiap produk dalam format ID, nama, dan harga.
- Selanjutnya program akan menampilkan pilihan menu untuk memilih menu yang dimasukkan oleh pengguna dan memanggil fungsi yang tepat sesuai pilihan tersebut:

  Menu 1 untuk menampilkan daftar produk

  Menu 2 untuk menambahkan produk baru

  Menu 3 untuk mengedit produk yang sudah ada

  Menu 4 untuk menghapus produk

  Menu 5 untuk keluar dari program

  Jika pilihan yang dimansukkan tidak valid (angka selain 1-5), maka pengguna akan diminta untuk mecoba lagi dengan angka yang sesuai.
- Break berfungsi untuk menghantikan loop.
- Fungsi `def baca_produk():`
  `data_produk = {}`
  `daftar_harga = {}`
  digunakan untuk membaca data produk dari file *produk.txt* .

- `with open("produk.txt", "r") as file:` digunakan untuk membuka file *produk.txt* memastikan file akan ditutup secara otomatis setelah selesai digunakan.
- `for line in file:` menggunakan loop untuk membaca setiap baris dalam file satu persatu.
- `proha = line.strip().split(",")` setiap baris dalam file dipishkan oleh tanda koma.
- `if len(proha) == 3:` memeriksa apakah jumlah elemen dalam list 3 yang memuat ID, nama, dan harga.
- `data_produk[id_produk] = nama_produk`
- `daftar_harga[id_produk] = harga_produk` untuk menyimpan dua dictionary : data_produk untuk menyimpan nama produk. Dan daftar_produk untuk meyimpan harga produk.
- `return data_produk, daftar_harga` untuk mengembalikan fungsi kedua yang berisi data produk dan harga untuk memanggil fungsi.

- Fungsi `def tambah_produk():` ini memungkinkan pengguna untuk menambahkan produk baru dengan meminta pengguna memasukkan ID, nama, dan harga produk yang diinginkan.
- `with open("produk.txt", "a") as file:` untuk membuka file dalam mode append( ) agar data baru ditambahkan ke akhir file tanpa menghapus isi yang sudah ada
- `file.write(f"{id_produk},{nama_produk},{harga}\n")` untuk menulis informasi produk baru kedalam file produk.txt dengan format yang sesuai (ID, nama, harga).

- Fungsi edit produk menampilkan daftar produk dan meminta pengguna untuk memasukkan ID produk yang akan diedit. Jika ID ditemukan dan sesuai, maka pengguna dapat menedit dengan memasukkan nama dan harga produk baru tersebut. Perubahan akan kembali tersimpan dalam file.
- `def edit_produk():`
  `data_produk, daftar_harga = baca_produk()` berguna memanggil fungsi untuk mendapatkan daftar produk dan harga produk saat ini.
- ```
  print("Daftar Produk:")
  for id_produk in data_produk:
      print(f"ID: {id_produk}, Nama: {data_produk[id_produk]}, Harga: {daftar_harga[id_produk]}")
  ```
  untuk mencetak semua produk yang ada dengan format ID, nama, dan harga menggunakan loop.
- `edit = int(input("Masukkan ID produk yang ingin diedit: "))` berfungsi untuk meminta pengguna memasukkan ID dari produk yang akan diedit.
- `if edit in data_produk`: untuk memeriksa apakah ID yang dimasukkan ada dalam dictionary. Jika ditemukan maka program akan mencetak informasi tentang produk tersebut.
- Selanjutnya program akan meminta pengguna untuk memasukkan nama dan harga baru untuk produk tersebut. Program akanmemperbarui dictionary dengan nama dan harga baru berdasarkan ID yang diedit.
  ```

- `with open("produk.txt", "w") as file:` program akan membuka file produk.txt dalam mode tulis (write), sehingga isi file akan dihapus sebelum menulis ulang semua data.
- Selanjutnya program akan menulis Kembali semua data yang telah diperbarui kedalam file dengan format yang sesuai (ID, nama, harga). Jika ID ditemukan maka produk dapat diedit dan akan melakukan perubahan, tetapi jika ID tidak ditemukan maka akan mencetak "produk tidak ditemukan".
- Fungsi hapus pada program ini mirip dengan fungsi edit, tetapi fungsi ini lebih spesifik untuk mengapus produk dari daftar. Setelah pengguna memasukkan ID produk yang akan dihapus, fungsi ini akan memperbarui file dengan manghapus entri tersebut dari struktur data yang sudah ada, sehingga produk yang telah dihapus tidak akan muncul dalam daftar produk yang tersedia di *produk.txt*.
- `def hapus_produk():`
  `data_produk, daftar_harga = baca_produk()` memanggil fungsi baca_produk untuk mendapatkan daftar produk saat ini.
- Selanjutnya program akan mencetak daftar semua produk yang ada menggunakan loop denga format ID, nama, dan harga.
- `hapus = int(input("Masukkan ID Produk Yang Ingin Dihapus: "))` Meminta pengguna untuk menginput ID dari produk yang ingin dihapus. Setelah itu program akan memeriksa apakah ID tersebut ada dalaam dictionary. Jika ditemukan maka akan mencetak informasi tentang produk yang akan dihapus.
- `del data_produk[hapus]` berfungsi untuk menghapus data dari dictionary berdasarkan ID yang dimasukkan oleh pengguna.
- `with open("produk.txt", "w") as file:` untuk membuka file produk.txt dalam mode tulis(write), sehingga isi sebelumnya akan dihapus sebelum menulis ulang semua data yang tersisa.
- Selanjutnya program akan menulis Kembali semua data yang tersisa ke dalam file setelah menghapus entri tertentu dengan format (ID, nama, harga).
- Program akan mencetak pesan konfirmasi bahwa penghapusan berhasil dilakukan. Dan jika ID tidak ditemukan maka program akan mencetak "produk tidak ditemukan".

➢ **File Halaman pembeli.py**

```
def halaman_pembeli(username):
    print(f"Halo {username}, Selamat Berbelanja!")
    while True:
        print("Menu Pembelian: \n1. List Produk\n2. Pembelian\n3.
Keluar")
        pilihan = input("Pilih Menu: ")
        if pilihan == "1":
            data_produk, daftar_harga = baca_produk()
            if not data_produk:
                print("Tidak Ada Produk Yang Tersedia")
            else:
                print("List Produk")
                for produk in data_produk:
                    print(f"ID: {produk} Nama:
{data_produk[produk]}, Harga: {daftar_harga[produk]}")
        elif pilihan == "2":
            proses_pembelian()
        elif pilihan == "3":
            print("Keluar Dari Halaman Pembelian")
            break

def baca_produk():
    data_produk = {}
    daftar_harga = {}
    with open("produk.txt", "r") as file:
        for proha in file:
            proha = proha.strip().split(",")
            if len(proha) == 3:
                id_produk = int(proha[0])
                nama_produk = proha[1]
                harga_produk = int(proha[2])
                data_produk[id_produk] = nama_produk
                daftar_harga[id_produk] = harga_produk
    return data_produk, daftar_harga

def proses_pembelian():
    data_produk, daftar_harga = baca_produk()

    daftar_belanja = []
    total_harga = 0

    while True:
        print("List Produk")
        for produk in data_produk:
            print(f"ID: {produk} Nama: {data_produk[produk]},
Harga: {daftar_harga[produk]}")
```

```
        id_produk = int(input("Pilih ID Produk (0 Untuk Selesai):
"))
        if id_produk == 0:
            break
        jumlah = int(input(f"Jumlah Untuk
{data_produk[id_produk]}: "))
        daftar_belanja.append((id_produk, jumlah))
        total_harga += daftar_harga[id_produk] * jumlah

        print(f"{data_produk[id_produk]} Sebanyak {jumlah} Telah
Ditambahkan")

    if daftar_belanja:
        print("Daftar Belanjaan:")
        for id_produk, jumlah in daftar_belanja:
            total = daftar_harga[id_produk] * jumlah
            print(f"Produk: {data_produk[id_produk]}, Jumlah:
{jumlah}, Total Harga: {total}")

        print(f"Total Harga Semua Produk: {total_harga}")

    print("Metode Pembayaran:")
    print("1. Kredit")
    print("2. Virtual Account")
    print("3. Transfer Bank")
    print("4. Pembayaran digagalkan")
    metode_pembayaran = input("Pilih metode pembayaran (1/2/3/4):
")

    if metode_pembayaran == "1":
        print("Pembayaran berhasil menggunakan Kredit.")
    elif metode_pembayaran == "2":
        print("Pembayaran berhasil menggunakan Virtual Account.")
    elif metode_pembayaran == "3":
        print("Pembayaran berhasil menggunakan Transfer Bank.")
    else:
        print("Pilihan tidak valid. Pembayaran batal.")
```

✓ **Output**

List Produk :

```
Login Berhasil. Selamat Datang putri dengan role pembeli
Halo putri, Selamat Berbelanja!
Menu Pembelian:
1. List Produk
2. Pembelian
3. Keluar
Pilih Menu: 1
List Produk
ID: 1 Nama: Shampoo, Harga: 10000
ID: 2 Nama: Dress, Harga: 150000
ID: 3 Nama: Topi, Harga: 50000
ID: 4 Nama: Earphone, Harga: 100000
ID: 5 Nama: Sepatu, Harga: 300000
```

Pembelian :

```
Menu Pembelian:
1. List Produk
2. Pembelian
3. Keluar
Pilih Menu: 2
List Produk
ID: 1 Nama: Shampoo, Harga: 10000
ID: 2 Nama: Dress, Harga: 150000
ID: 3 Nama: Topi, Harga: 50000
ID: 4 Nama: Earphone, Harga: 100000
ID: 5 Nama: Sepatu, Harga: 300000
Pilih ID Produk (0 Untuk Selesai): 1
Jumlah Untuk Shampoo: 3
Shampoo Sebanyak 3 Telah Ditambahkan
List Produk
ID: 1 Nama: Shampoo, Harga: 10000
ID: 2 Nama: Dress, Harga: 150000
ID: 3 Nama: Topi, Harga: 50000
ID: 4 Nama: Earphone, Harga: 100000
ID: 5 Nama: Sepatu, Harga: 300000
Pilih ID Produk (0 Untuk Selesai): 2
Jumlah Untuk Dress: 4
Dress Sebanyak 4 Telah Ditambahkan
List Produk
ID: 1 Nama: Shampoo, Harga: 10000
ID: 2 Nama: Dress, Harga: 150000
ID: 3 Nama: Topi, Harga: 50000
ID: 4 Nama: Earphone, Harga: 100000
ID: 5 Nama: Sepatu, Harga: 300000
Pilih ID Produk (0 Untuk Selesai): 0
Daftar Belanjaan:
Produk: Shampoo, Jumlah: 3, Total Harga: 30000
Produk: Dress, Jumlah: 4, Total Harga: 600000
Total Harga Semua Produk: 630000
Metode Pembayaran:
1. Kredit
2. Virtual Account
3. Transfer Bank
4. Pembayaran digagalkan
Pilih metode pembayaran (1/2/3/4): 1
Pembayaran berhasil menggunakan Kredit.
```

Keluar :

```
Menu Pembelian:
1. List Produk
2. Pembelian
3. Keluar
Pilih Menu: 3
Keluar Dari Halaman Pembelian
```

❖ **Penjelasan Code:**

- `def halaman_pembeli(username):` Berfungsi untuk mendefinisikan fungsi halaman pembeli yang menerima parameter username.
- `print(f"Halo {username}, Selamat Berbelanja!"):` Berfungsi untuk mencetak pesan penyambut bagi *user*.
- `while True:` Berfungsi sebagai *loop* tak terbatas, guna menampilkan menu secara terus menerus sampai *user* memilih untuk keluar.
- `print("Menu Pembelian: \n1. List Produk\n2. Pembelian\n3. Keluar"):` Berfungsi untuk mencetak menu pilihan kepada *user*.
- `pilihan = input("Pilih Menu: "):` Berguna untuk mengambil input dari *user* untuk memilih menu.
- `if pilihan == "1"` dan `data_produk, daftar_harga = baca_produk():` Berfungsi untuk memeriksa pilihan user. Jika pilihan adalah "1", maka fungsi `baca_produk()` dipanggil untuk mendapatkan data produk dan harga.
- `if not data_produk` dan `print("Tidak Ada Produk Yang Tersedia"):` Untuk memastikan apabila produk tidak tersedia, maka akan mencetak pesan bahwa poduk tidak tersedia.
- `else, print("List Produk"), for produk in data_produk,` dan `print(f"ID: {produk} Nama: {data_produk[produk]}, Harga: {daftar_harga[produk]}"):` Bertugas untuk menampilkan produk berdasarkan dengan ID Produk, Nama Produk, dan Harga Produk.
- `elif pilihan == "2"` dan `proses_pembelian():` Berfungsi untu memilih pembelian. Jika memilih "2", maka akan memanggil fungsi `proses_pembelian()` untuk memproses pembeliannya.
- `elif pilihan == "3", print("Keluar Dari Halaman Pembelian"),` dan `break:` Berfungsi untuk keluar dari proses pembelian. Jika meilih "3", maka akan menampilkan pesan keluar dan akan menghentikan *loop* dengan `break`.

- `def baca_produk():` Untuk mendefinisikan fungsi `baca_produk()` yang tidak menerima parameter.
- `data_produk = {}` dan `daftar_harga = {}:` Berfungsi untuk membuat dua *dictionary* kosong untuk menyimpan data produk dan harga.
- `with open("produk.txt", "r") as file, for proha in file,` dan `proha = proha.strip().split(","):` Berfungsi untuk membuka file *produk.txt* dengan akses baca `"r"` atau *read*. Setiap baris dalam file dipecah menjadi *list* berdasarkan komanya.
- `if len(proha) == 3:` Mendefinisikan "produk harga" sebagai `proha`. Berfungsi untuk memastikan setiap baris memiliki `3` komponen, yaitu format ID, Nama Produk, dan Harga.
- `id_produk = int(proha[0])` : Berguna untuk menyimpan dan menempatkan elemen pertama dari *list* `proha`, yaitu ID produk. Serta mengonversi elemen pertama menjadi bilangan bulat.
- `nama_produk = proha[1]` : Berguna untuk mengambil elemen kedua dari `proha`, dan menyimpan nama produk sebagai *string*.
- `harga_produk = int(proha[2])` : Berguna untuk menympan dan menempatkan elemen ketiga dari *list* `proha`, yaitu harga produk. Serta mengonversi elemen ketiga menjadi bilangan bulat.
- `data_produk[id_produk] = nama_produk:` Merupakan *dictionary* yang menyimpan data produk dengan ID produk sebagai kunci dan nama produk sebagai nilainya.
- `daftar_harga[id_produk] = harga_produk:` Dictionary yang menyimpan data harga produk dengan ID produk sebagai kunci dan harga produk sebagai nilainya.
- `return data_produk, daftar_harga:` Berfungsi untuk mengembalikan dua *dictionary* yang berisi data produk dan harga produk.
- `def proses_pembelian()` dan `data_produk, daftar_harga = baca_produk():` Berfungsi untuk mendefinisikan fungsi `proses_pembelian()` dan memanggil `baca_produk()` untuk mendapatkan data produk dan harga produk.
- `daftar_belanja = []` dan `total_harga = 0:` Berguna untuk membuat *list* kosong dan menyimpan daftar belanja, serta *variable* untuk total harga produk.
- `while True:` Berfungsi untuk memulai *loop* daftar produk berulang kali sampai *user* memutuskan untk keluar.
- `print("List Produk"):` Berfungsi untuk mecetak *header* pada daftar produk yang akan ditampilkan.
- `for produk in data_produk:` Berfungsi untuk memberikan nilai produk pada setap iterasinya yang merupakan ID produk.

- `print(f"ID: {produk} Nama: {data_produk[produk]}, Harga: {daftar_harga[produk]}")` : Berfungsi untuk mencetak produk berdasarkan dengan ID Produk, Nama Produk, dan Harga Produk.
- `id_produk = int(input("Pilih ID Produk (0 Untuk Selesai): ")). if id_produk == 0, break`: Berfungsi untuk mengambil input ID Produk dari *user*. Apabila *user* memasukkan angka "0", maka fungsi `break` akan mengeluarkan *user* dari *loop*.
- `jumlah = int(input(f"Jumlah Untuk {data_produk[id_produk]}: "))`: Berfungsi untuk menentukan banyak jumlah produk yang ingin dibeli.
- `daftar_belanja.append((id_produk, jumlah))`: Berguna untuk mencatat produk yang dibeli *user* beserta jumlahnya ke dalam daftar belanja.
- `total_harga += daftar_harga[id_produk] * jumlah`: Berguna untuk menghitung total biaya untuk produk yang baru ditambahkan dan meng-*update* total harga belanja secara keseluruhan.
- `print(f"{data_produk[id_produk]} Sebanyak {jumlah} Telah Ditambahkan")`: Untuk mencetak pesan kepada pelanggan, bahwa poduk tertentu telah berhasil ditambahan ke daftar belanja sesuai dengan jumlah yang ditentukan.
- `if daftar_belanja`: Berfungsi untuk memastikan bahwa proses pencetakan daftar belanjaan hanya dilakukan jika ada produk yang telah dibeli.
- `print("Daftar Belanjaan:")` : Berfungsi untuk mencetak pesan untuk *user*, bahwa daftar belanjaan mereka akan ditampilkan.
- `for id_produk, jumlah in daftar_belanja`: Untuk mengakses masing-masing produk dalam daftar belanja untuk diproses lebih lanjut.
- `total = daftar_harga[id_produk] * jumlah`: Berfungsi untuk menghitung total harga untuk setiap produk berdasarkan jumlah yang dibeli.
- `print(f"Produk: {data_produk[id_produk]}, Jumlah: {jumlah}, Total Harga: {total}")`: Untuk menampilkan informasi lengkap tentang setiap produk dalam daftar belanja, termasuk nama, jumlah yang dibeli, dan total harganya.
- `print(f"Total Harga Semua Produk: {total_harga}")`: Digunakan untuk memberikan ringkasan total biaya yang harus dibayar *user* untuk semua produk dalam daftar belanja.
- `print("Metode Pembayaran:")`: Untuk mencetak bagian program yang berkaitan dengan metode pembayaran.
- `print("1. Kredit")`: Untuk menampilkan pilihan metode pembayaran pertama kepada *user*.

- `print("2. Virtual Account")`: Untuk menampilkan pilihan metode pembayaran kedua kepada *user*.
- `print("3. Transfer Bank")`: Untuk menampilkan pilihan metode pembayaran ketiga kepada pengguna.
- `print("4. Pembayaran digagalkan")`: Berfungsi mencetak opsi bagi *user* untuk membatalkan pembayaran jika mereka berubah pikiran atau tidak ingin melanjutkan transaksi.
- `metode_pembayaran = input("Pilih metode pembayaran (1/2/3/4) : ")`: Memungkinkan *user* memilih metode pembayaran dengan memasukkan angka yang sesuai dengan pilihan mereka.
- `if metode_pembayaran == "1"`: Untuk memeriksa apakah *user* memilih opsi metode pembayaran pertama, yaitu `Kredit`.
- `print("Pembayaran berhasil menggunakan Kredit.")`: Untuk memberikan pesan kepada *user* bahwa pembayaran telah berhasil dilakukan menggunakan metode `Kredit`.
- `elif metode_pembayaran == "2"`: Berfungsi untuk memeriksa apakah *user* memilih opsi metode pembayaran kedua, yaitu `Virtual Account`.
- `print("Pembayaran berhasil menggunakan Virtual Account.")`: Untuk memberi pesan kepada *user* bahwa pembayaran telah berhasil dilakukan menggunakan metode `Virtual Account`.
- `elif metode_pembayaran == "3"`: Untuk memeriksa apakah *user* memilih opsi metode pembayaran ketiga, yaitu `Transfer Bank`.
- `print("Pembayaran berhasil menggunakan Transfer Bank.")`: Berfungsi mencetak pesan kepada *user* bahwa pembayaran telah berhasil dilakukan menggunakan metode `Transfer Bank`.
- `else`: Untuk menangani situasi di mana *user* memasukkan pilihan metode pembayaran yang tidak valid.
- `print("Pilihan tidak valid. Pembayaran batal.")`: Berfungsi untuk mencetak pesan kepada *user* bahwa input mereka tidak valid, dan pembayaran dibatalkan.