LAPORAN PRAKTIKUM

POSTTEST 4

ALGORITMA PEMROGRAMAN DASAR

Disusun oleh:

Muhammad Zidane Abdul Kadir (2509106021)

Kelas (A1 ‘25)

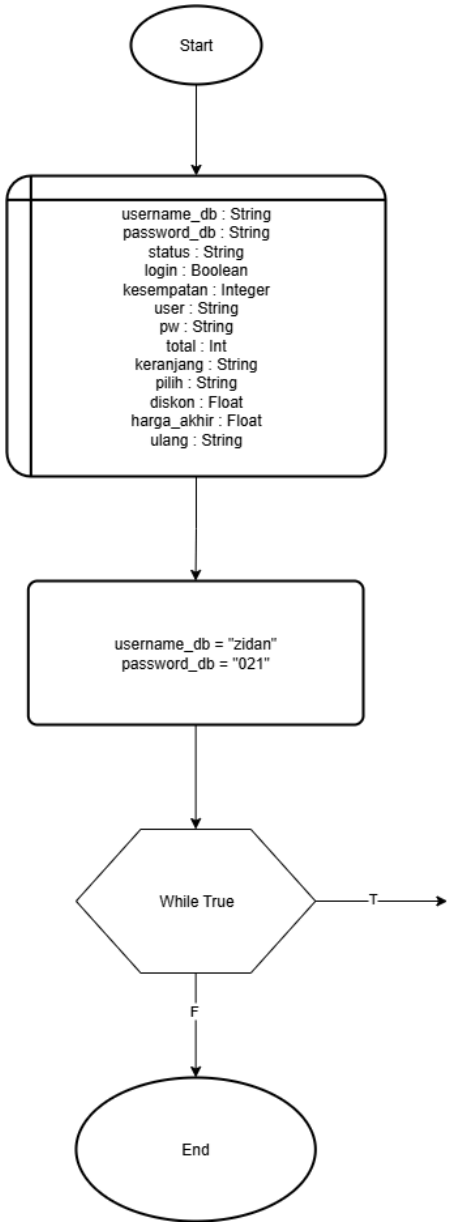
PROGRAM STUDI INFORMATIKA

UNIVERSITAS MULAWARMAN

SAMARINDA

2025

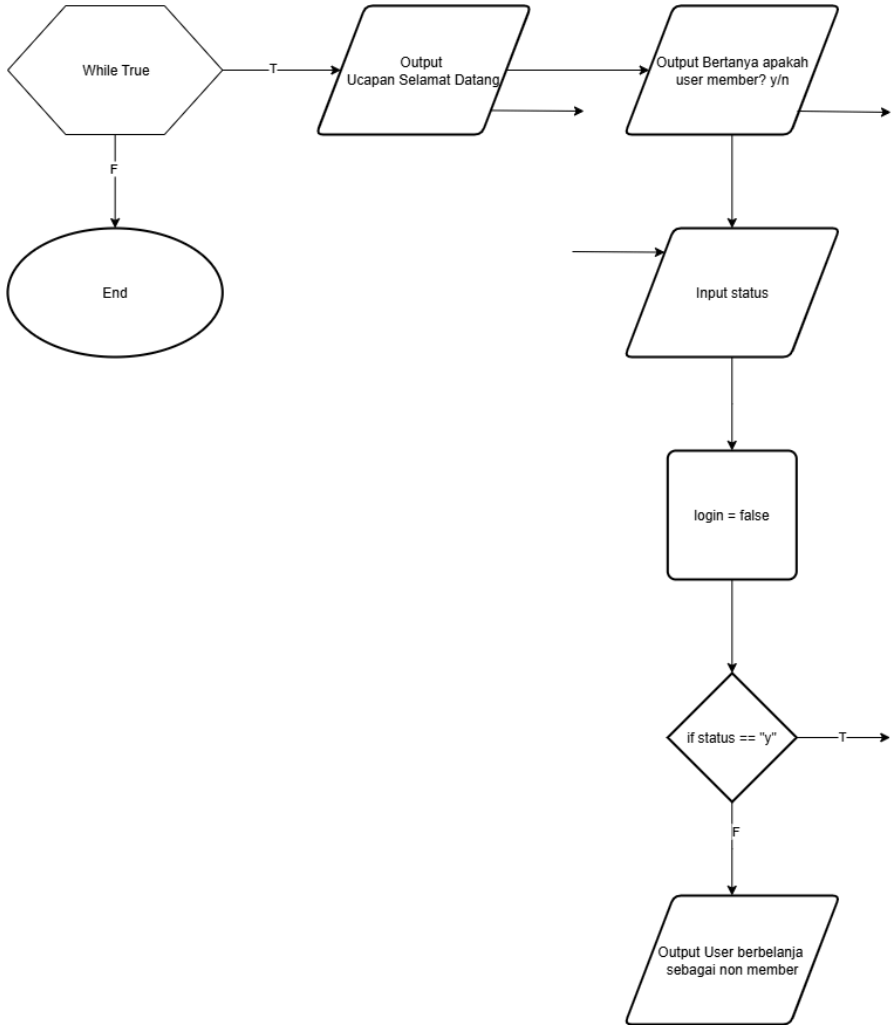
1. Flowchart



gambar 1.1 flowchart deklarasi, inisialisasi, dan looping

Flowchart pada Gambar 1.1 menggambarkan alur dasar dari sebuah program sederhana yang dimulai dengan proses *Start*. Setelah itu, program mendeklarasikan sejumlah variabel yang akan digunakan selama eksekusi, seperti username_db dan password_db untuk menyimpan kredensial sistem, login sebagai status autentikasi, kesempatan untuk menghitung percobaan login, serta variabel lainnya seperti user, pw, total, keranjang, pilih, diskon, harga_akhir, dan ulang yang kemungkinan besar digunakan untuk fitur-fitur tambahan seperti transaksi atau pengulangan proses. Selanjutnya, program

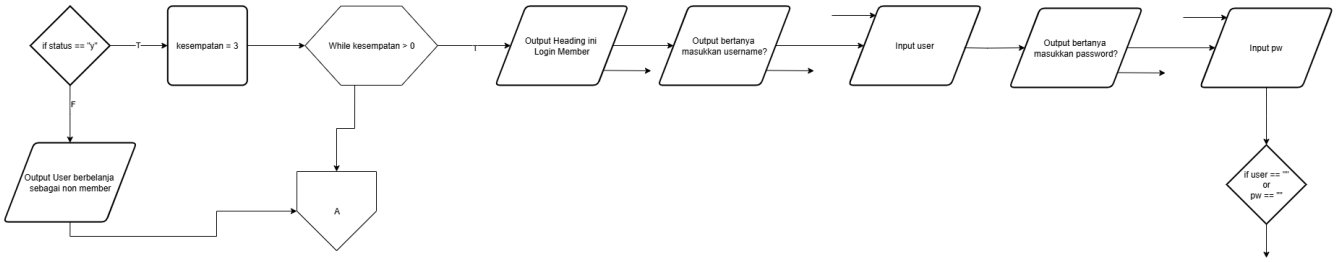
melakukan inisialisasi nilai awal untuk variabel username_db dengan nilai “zidan” dan password_db dengan nilai “021”, yang berfungsi sebagai data user default sistem. Setelah tahap inisialisasi, program memasuki struktur perulangan tak hingga (While True) yang akan terus berjalan selama kondisi bernilai benar (True). Dalam konteks ini, loop tersebut mungkin akan dihentikan secara manual atau melalui mekanisme lain di luar flowchart ini (misalnya dengan perintah break saat kondisi tertentu terpenuhi).



Gambar 1.2 pertanyaan apakah user member?

Flowchart pada Gambar 1.2 melanjutkan alur program dari Gambar 1.1 setelah masuk ke dalam perulangan While True. Proses dimulai dengan menampilkan pesan sambutan “Ucapan Selamat Datang” kepada pengguna sebagai output awal. Selanjutnya, sistem akan menanyakan status keanggotaan pengguna melalui prompt: *“Bertanya apakah user member? y/n”*. Jawaban dari pengguna kemudian diambil melalui proses input variabel status. Sebelum

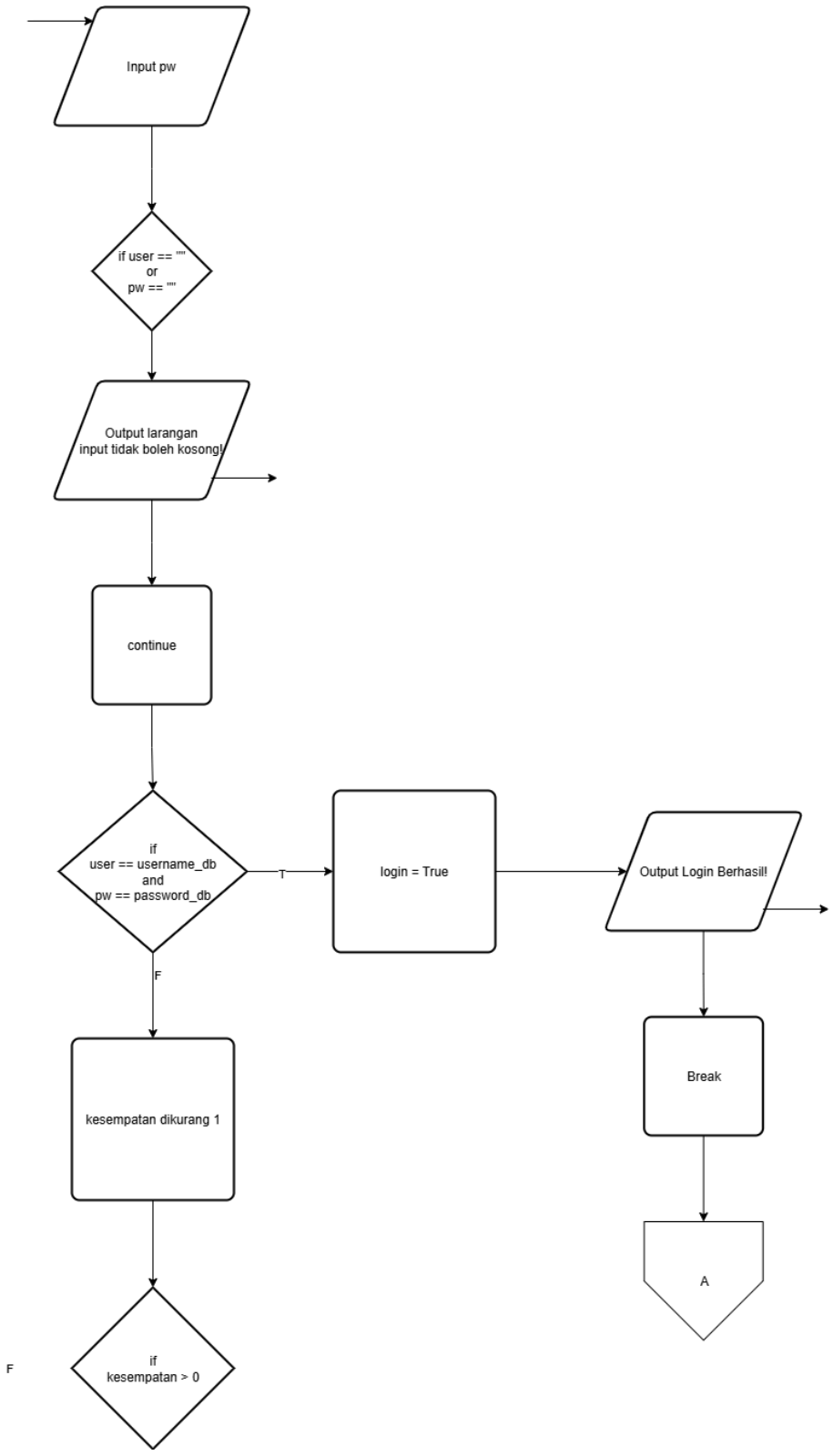
memproses jawaban, variabel login diinisialisasi dengan nilai false, yang menandakan bahwa pengguna belum berhasil masuk atau diverifikasi sebagai member. Kemudian, program melakukan pengecekan kondisi: jika nilai status sama dengan "y" (artinya pengguna memilih ya sebagai member), maka alur akan berlanjut ke cabang benar (True) meskipun bagian ini tidak ditampilkan secara lengkap pada gambar, dapat diasumsikan bahwa program akan meminta autentikasi lebih lanjut (misalnya username dan password). Jika kondisi bernilai salah (False), artinya pengguna memilih "n" atau input lainnya, maka program akan menampilkan output: *"User berbelanja sebagai non member"*, yang mengindikasikan bahwa pengguna akan melanjutkan transaksi tanpa hak istimewa member. Alur ini menunjukkan mekanisme dasar dalam membedakan antara pengguna member dan non-member sebelum proses selanjutnya, seperti login atau pembelian.



Gambar 1.2 proses mempertanyakan username dan password

Setelah pengguna menyatakan bahwa ia adalah member (melalui input status == "y"), program akan menginisialisasi variabel kesempatan dengan nilai 3, yang berarti pengguna diberikan tiga kali kesempatan untuk memasukkan kredensial login yang benar. Selanjutnya, program memasuki perulangan While kesempatan > 0, yang akan terus berjalan selama jumlah kesempatan masih tersedia. Di dalam loop ini, sistem menampilkan pesan "Heading ini Login Member" sebagai pemberitahuan bahwa pengguna sedang berada di tahap autentikasi member. Kemudian, program meminta pengguna untuk memasukkan *username* melalui prompt "masukkan username?", diikuti oleh permintaan *password* melalui prompt "masukkan password?". Setelah kedua input tersebut diterima, program melakukan validasi awal: apakah *username* atau *password* kosong (dengan kondisi if user == "" or pw == ""). Jika salah satu atau keduanya kosong, maka sistem akan menganggap input tidak valid dan kemungkinan besar akan mengurangi jumlah kesempatan atau meminta ulang input meskipun bagian ini belum ditampilkan secara lengkap pada gambar. Jika

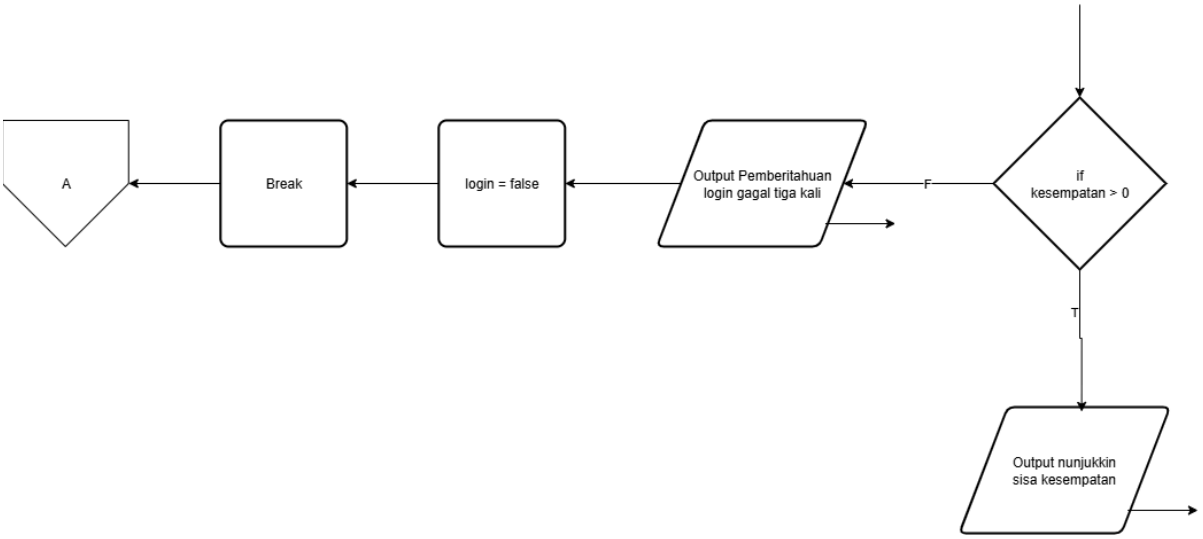
input tidak kosong, maka proses akan dilanjutkan ke tahap verifikasi lebih lanjut (misalnya membandingkan dengan username_db dan password_db yang telah diinisialisasi sebelumnya). Jika pengguna gagal login setelah tiga kali percobaan, maka perulangan akan berhenti dan program akan beralih ke jalur alternatif, yaitu menampilkan pesan “User berbelanja sebagai non member”, yang berarti pengguna tetap dapat melanjutkan aktivitas tanpa status member.



Gambar 1.3 proses pemeriksaan

Gambar 1.3 melanjutkan alur dari proses login member yang dimulai pada Gambar 1.2. Setelah pengguna memasukkan *username* dan *password*, program melakukan pemeriksaan awal, apakah salah satu atau kedua input

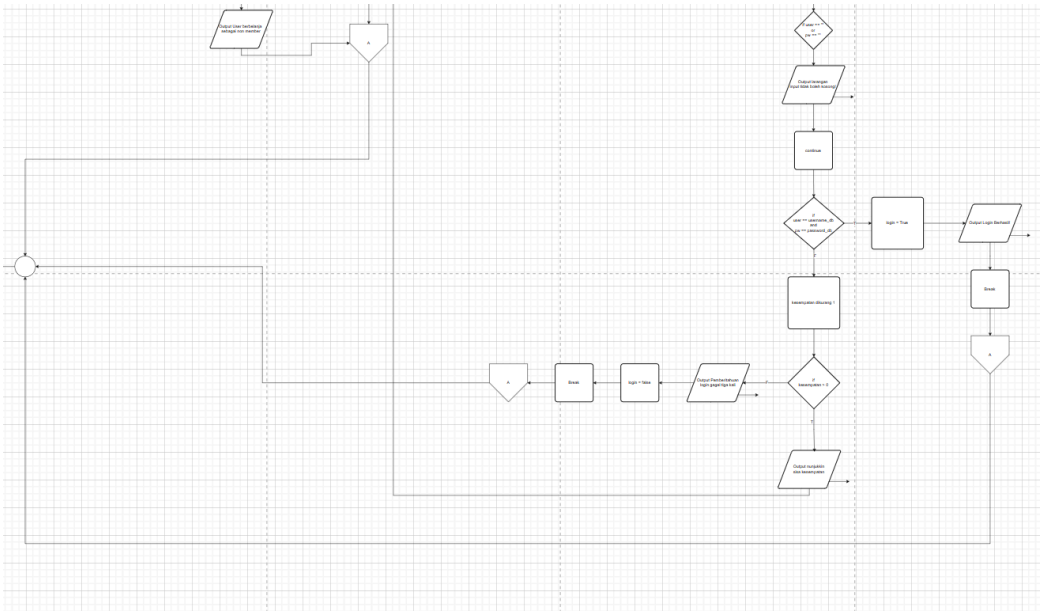
tersebut kosong? (dengan kondisi if user == "" or pw == ""). Jika kondisi ini terpenuhi, maka sistem akan menampilkan pesan peringatan: *"Output larangan: input tidak boleh kosong!"* sebagai validasi dasar, kemudian melanjutkan eksekusi dengan perintah continue yang berarti program akan kembali ke awal perulangan While kesempatan > 0 tanpa mengurangi jumlah kesempatan, sehingga pengguna diberi kesempatan untuk mengisi ulang data dengan benar. Jika input tidak kosong, program akan memverifikasi kebenaran kredensial dengan membandingkan nilai variabel user dan pw terhadap nilai yang telah disimpan di username_db dan password_db. Jika kedua nilai cocok (if user == username_db and pw == password_db), maka variabel login diubah menjadi True, dan sistem menampilkan pesan *"Output Login Berhasil!"*. Setelah itu, program langsung keluar dari perulangan login menggunakan perintah Break, lalu menuju ke simbol *connector* **A**, yang menandakan bahwa alur program akan dilanjutkan ke tahap berikutnya (misalnya list belanja). Namun, jika verifikasi gagal (kondisi False), maka jumlah kesempatan login akan dikurangi satu (kesempatan dikurang 1). Kemudian, program memeriksa apakah masih ada kesempatan tersisa (if kesempatan > 0).



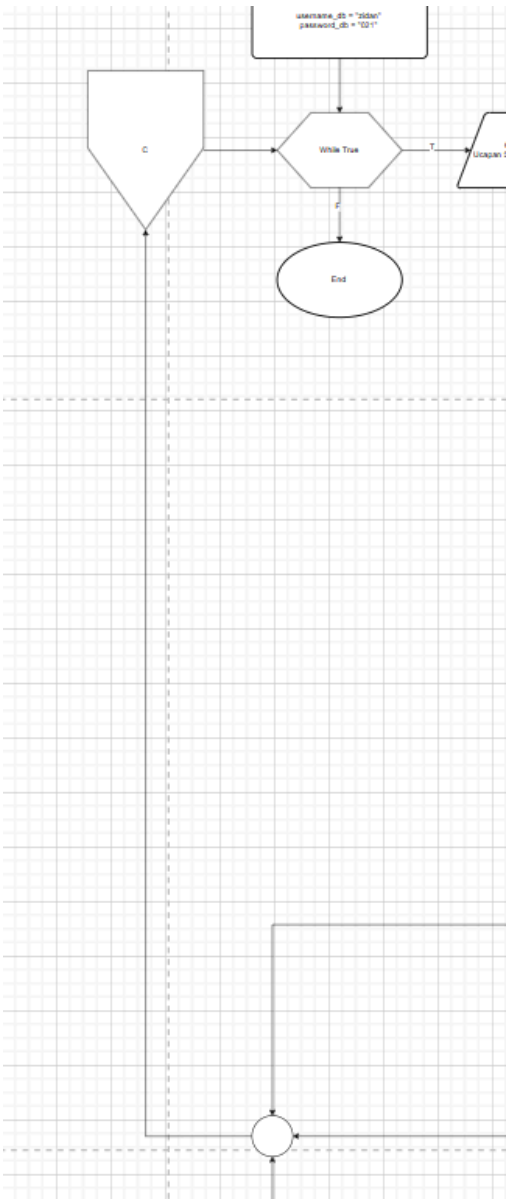
Gambar 1.4 setelah pemeriksaan sisa kesempatan

Gambar 1.4 menunjukkan alur program yang dijalankan ketika pengguna telah mencapai batas maksimal percobaan login (yaitu 3 kali) dan masih gagal memasukkan kredensial yang benar. Setelah jumlah kesempatan habis (kesempatan <= 0), sistem akan mengecek kondisi if kesempatan > 0. Karena kondisi ini bernilai **False**, maka program akan menampilkan pesan pemberitahuan: *"Output Pemberitahuan: login gagal tiga kali"* sebagai notifikasi kepada pengguna bahwa akses member telah dibatasi. Selanjutnya,

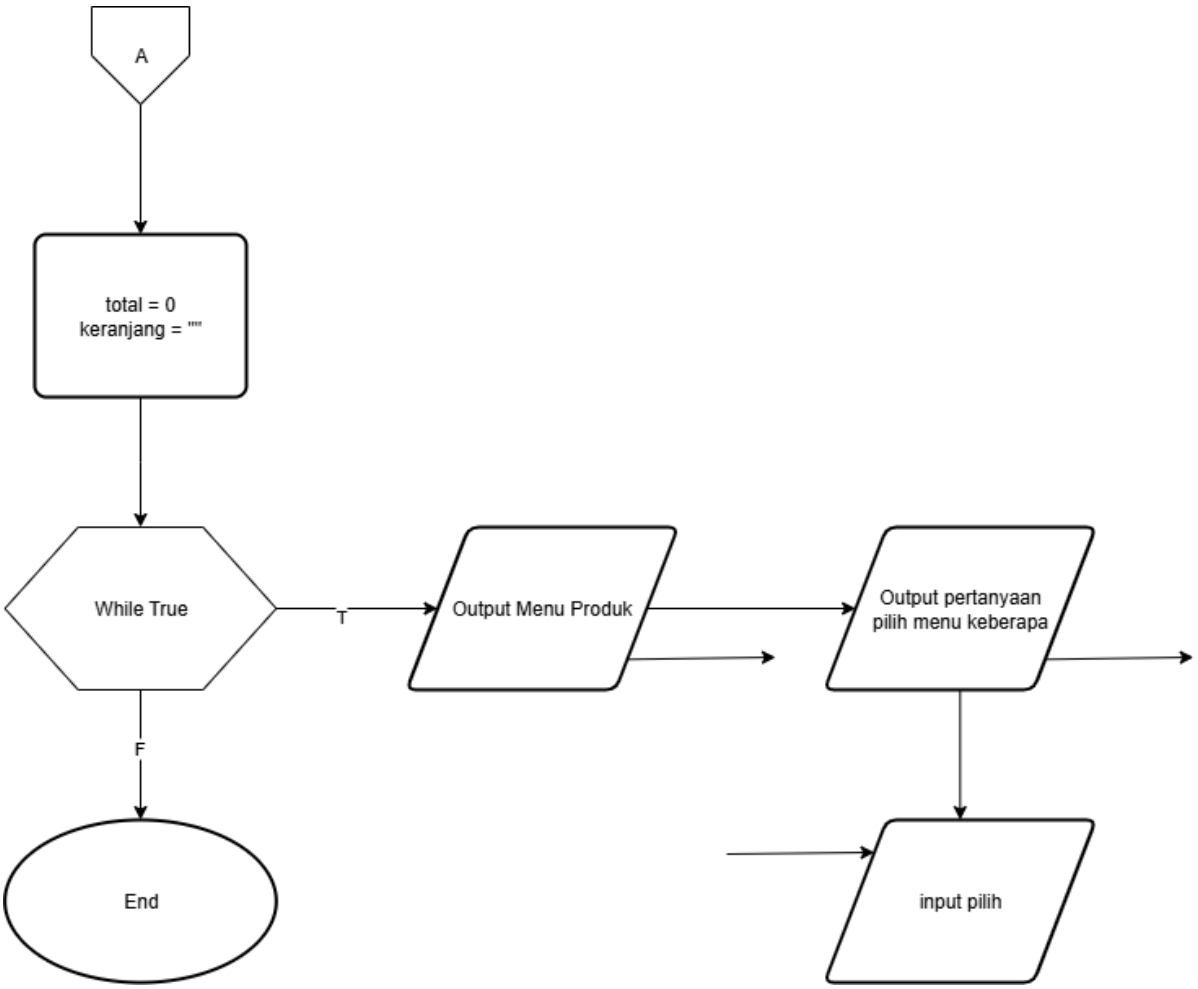
variabel login diatur kembali ke nilai false, menandakan bahwa pengguna tidak berhasil masuk sebagai member. Program kemudian menggunakan perintah Break untuk keluar dari perulangan login secara permanen, dan menuju ke simbol *connector* **A**, yang mengarahkan alur ke bagian selanjutnya dari sistem biasanya berupa fitur belanja sebagai *non-member*, atau kembali ke menu awal. Namun, jika pada kondisi if kesempatan > 0 ternyata masih ada sisa kesempatan (misalnya karena sistem belum mencapai limit), maka program akan menampilkan pesan yang membagikan sisa kesempatan, maksudnya adalah memberi informasi kepada pengguna tentang berapa kali lagi ia dapat mencoba login. Setelah itu, program akan kembali ke dalam perulangan login untuk memberikan kesempatan tambahan.



Gambar 1.5 Connector A Yang Terhubung dengan bulatan agar rapi



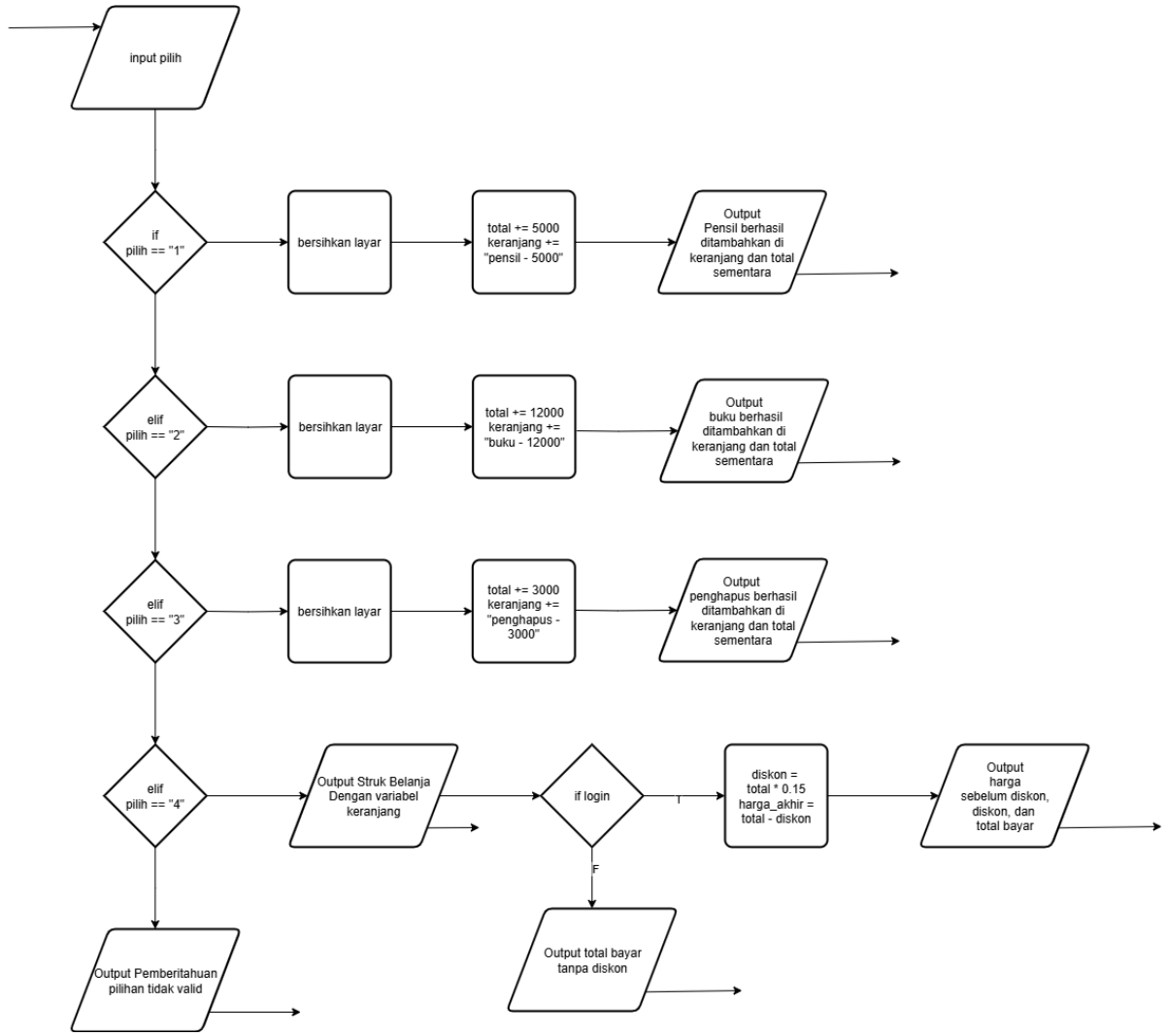
Gambar 1.5 Dan Berakhir di Connector C Yang akan Mengembalikan User kembali mengulang perulangan



Gambar 1.6

Gambar 1.6 menggambarkan awal dari modul utama aplikasi setelah pengguna berhasil atau gagal login, di mana program memasuki fase pemilihan produk. Proses dimulai dari *connector* **A**, yang menghubungkan alur dari tahap sebelumnya (baik login berhasil maupun gagal) ke bagian ini. Sebagai langkah awal, program melakukan inisialisasi variabel: total = 0 (untuk menyimpan jumlah total harga belanjaan) dan keranjang = "" (untuk menyimpan daftar item yang dipilih pengguna). Selanjutnya, program memasuki perulangan tak hingga (While True) yang akan terus berjalan selama pengguna belum memilih opsi keluar. Di dalam loop ini, sistem pertama-tama menampilkan *"Output Menu Produk"* yaitu daftar produk yang tersedia untuk dibeli. Setelah itu, program menanyakan kepada pengguna: *"Output pertanyaan: pilih menu beberapa?"*, yang merupakan prompt untuk meminta input nomor atau kode produk yang ingin dipilih. Kemudian, program membaca input dari pengguna melalui proses *"Input pilih"*. Input ini nantinya akan diproses lebih lanjut (misalnya untuk menambahkan produk ke keranjang atau menghitung total harga) pada bagian flowchart berikutnya meskipun bagian tersebut tidak ditampilkan dalam

Gambar 1.6. Perulangan While True akan terus berjalan sampai ada kondisi tertentu yang menghentikannya (misalnya pengguna memilih opsi “keluar” atau “selesai belanja”), yang kemungkinan besar akan dijelaskan pada gambar selanjutnya. Jika kondisi tersebut terpenuhi, maka program akan keluar dari loop dan menuju proses End.



Gambar 1.7

Gambar 1.7 melanjutkan alur dari Gambar 1.6 setelah pengguna memasukkan pilihan produk melalui input pilih. Program kemudian melakukan pengecekan kondisi berantai menggunakan struktur if-elif untuk menentukan produk apa yang dipilih oleh pengguna.

- Jika pengguna memilih "1", maka sistem akan menambahkan item “pensil” ke keranjang dengan harga Rp5.000. Variabel total diperbarui dengan menambahkan nilai tersebut, dan variabel keranjang juga diperbarui dengan mencatat item yang ditambahkan. Setelah itu, program

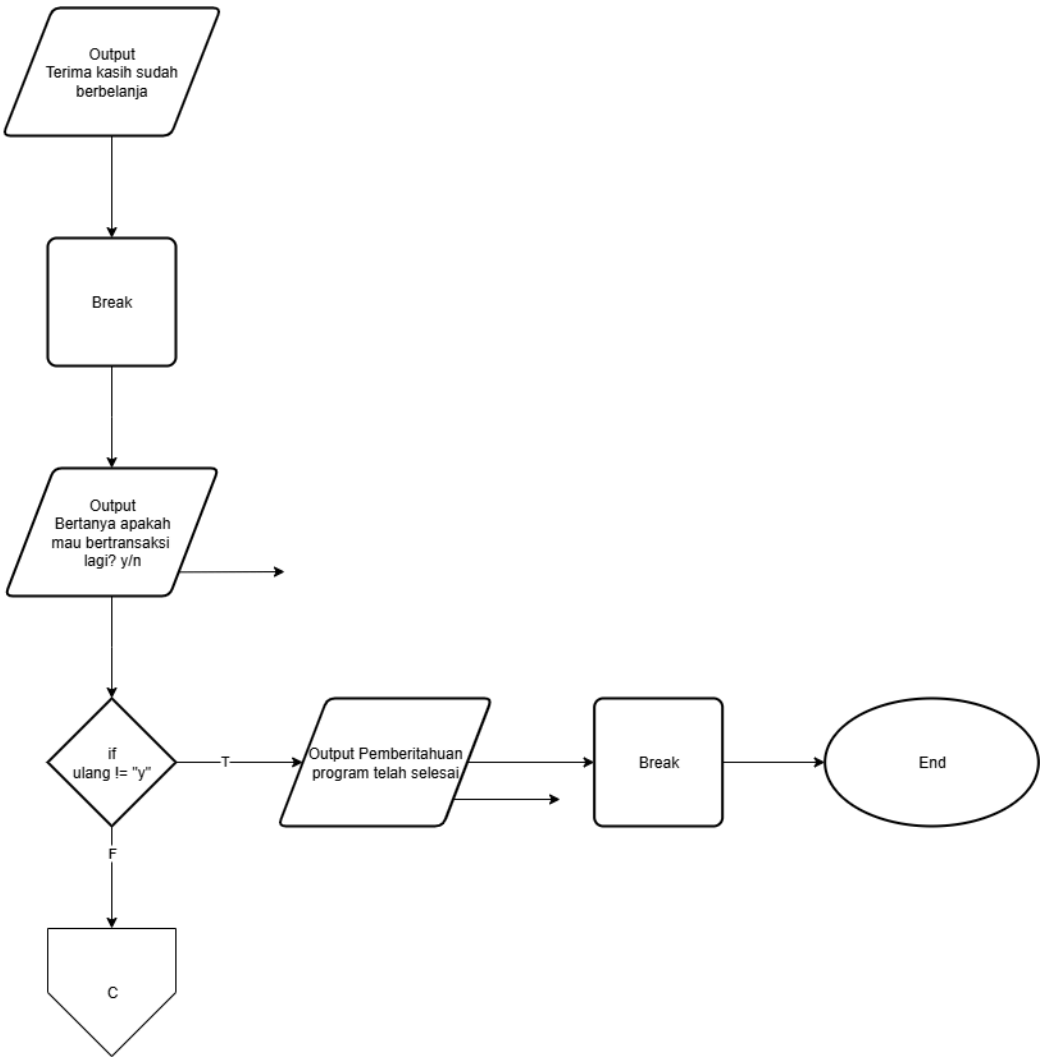
menampilkan pesan: *“Output Pensil berhasil ditambahkan di keranjang dan total sementara”* sebagai konfirmasi kepada pengguna.

- Jika pengguna memilih "2", maka item “buku” seharga Rp12.000 akan ditambahkan ke keranjang, dan total serta keranjang diperbarui sesuai. Pesan konfirmasi yang muncul adalah: *“Output buku berhasil ditambahkan di keranjang dan total sementara”*.
- Jika pengguna memilih "3", maka item “penghapus” seharga Rp3.000 akan ditambahkan, disertai pesan konfirmasi serupa: *“Output penghapus berhasil ditambahkan di keranjang dan total sementara”*.

Jika pengguna memilih "4", maka program menganggap bahwa pengguna ingin melihat ringkasan belanjaan. Sistem akan menampilkan *“Output Struk Belanja Dengan variabel keranjang”*, yang menunjukkan daftar item yang telah dipilih. Setelah itu, program memeriksa status login dengan kondisi if login:

- Jika nilai login bernilai **True** (artinya pengguna adalah member), maka sistem akan menghitung diskon sebesar 15% dari total belanja (diskon = total * 0.15) dan menghitung harga akhir setelah diskon (harga_akhir = total - diskon). Lalu, program menampilkan output: *“Output harga sebelum diskon, diskon, dan total bayar”*.
- Jika nilai login bernilai **False** (pengguna non-member), maka tidak ada diskon yang diberikan, dan sistem hanya menampilkan *“Output total bayar tanpa diskon”*.

Namun, jika pengguna memasukkan pilihan yang tidak valid (selain “1”, “2”, “3”, atau “4”), maka program akan menampilkan pesan: *Output Pemberitahuan: pilihan tidak valid*, dan kembali ke menu utama untuk meminta input ulang meskipun bagian ini tidak ditampilkan secara eksplisit pada gambar, logika umumnya akan kembali ke awal loop While True di Gambar 1.6.



Gambar 1.8

Gambar 1.8 menggambarkan alur akhir dari program setelah pengguna menyelesaikan proses pemilihan dan pembayaran produk. Proses dimulai dengan menampilkan pesan *"Output Terima kasih sudah berbelanja"* sebagai ucapan penutup kepada pengguna atas transaksinya. Setelah itu, program menggunakan perintah Break untuk keluar dari loop utama (misalnya loop While True pada Gambar 1.6) dan melanjutkan ke tahap konfirmasi akhir.

Selanjutnya, sistem menanyakan kepada pengguna: *"Output Bertanya apakah mau bertransaksi lagi? y/n"*, memberikan opsi untuk melanjutkan atau mengakhiri sesi. Jawaban pengguna akan disimpan dalam variabel ulang, lalu dicek dengan kondisi if ulang != "y":

- Jika pengguna memasukkan nilai selain "y" (misalnya "n" atau input lainnya), maka kondisi bernilai **True**, dan program akan menampilkan pesan *"Output Pemberitahuan: program telah selesai"*, diikuti perintah

Break untuk menghentikan eksekusi, dan akhirnya menuju proses End artinya program benar-benar berhenti.

- Jika pengguna memasukkan "y", maka kondisi bernilai **False**, dan program akan melanjutkan ke simbol *connector* **C**, yang menandakan bahwa alur akan kembali ke awal proses belanja (misalnya ke menu login), sehingga pengguna dapat melakukan transaksi baru.

Dengan demikian, Gambar 1.8 berfungsi sebagai modul penutup yang memberikan fleksibilitas kepada pengguna: bisa langsung keluar setelah belanja, atau memilih untuk berbelanja lagi dalam satu sesi yang sama meningkatkan pengalaman pengguna dan efisiensi program.

2. Deskripsi Singkat Program

Program ini merupakan simulasi sistem kasir toko sederhana yang membedakan pengguna berdasarkan status keanggotaan (*member* dan *non-member*). Pengguna yang memilih sebagai *member* harus login dengan username "zidan" dan password "021", dengan maksimal 3 kali percobaan. Jika gagal, sistem otomatis menganggap pengguna sebagai *non-member*. Setelah itu, pengguna dapat memilih produk dari menu (pensil, buku, penghapus), menambahkannya ke keranjang, dan melakukan *checkout*. Jika pengguna adalah *member*, ia mendapatkan diskon 15% dari total belanja. Setelah transaksi selesai, program menanyakan apakah pengguna ingin berbelanja lagi. Jika tidak, program berhenti dengan pesan penutup. Program ini dirancang untuk berjalan dalam satu sesi interaktif di terminal/console.

3. Source Code

A. Penyiapan Data Login

```
username_db = "zidan"
password_db = "021"
```

Pada bagian ini, program mendeklarasikan dua variabel global username_db dan password_db yang digunakan sebagai kredensial default sistem. Variabel username_db diisi dengan nilai "zidan", dan password_db diisi dengan nilai "021". Kedua variabel ini berfungsi sebagai *database sederhana* untuk autentikasi pengguna member — artinya, hanya pengguna yang memasukkan

kombinasi username dan password ini yang akan dianggap sebagai member dan berhak mendapatkan diskon.

B. Proses Verifikasi Status Member dan Login

```
while True:
    print("=== Selamat Datang di Toko ===")
    status = input("Apakah Anda member? (y/n): ")
    login = False
    if status == "y":
        kesempatan = 3
        while kesempatan > 0:
            print("\n=== Login Member ===")
            user = input("Masukkan Username: ")
            pw = input("Masukkan Password: ")

            if user == "" or pw == "":
                print("Input tidak boleh kosong!")
                continue

            if user == username_db and pw == password_db:
                login = True
                print("Login berhasil, silakan
belanja!\n")
                break
            else:
                kesempatan -= 1
                if kesempatan > 0:
                    print(f"Login gagal! Sisa percobaan:
{kesempatan}")
                else:
                    print("Login gagal 3 kali! Anda
dianggap Non-Member.\n")
                    login = False
                    break
    else:
        print("\nAnda berbelanja sebagai Non-Member.\n")
```

Bagian ini merupakan awal dari alur utama program, di mana pengguna ditanyakan status keanggotaannya melalui input "y" (ya) atau "n" (tidak). Jika pengguna memilih **bukan member** (status == "n"), sistem langsung menampilkan pesan bahwa pengguna akan berbelanja sebagai *non-member* dan melanjutkan ke proses pemilihan produk. Namun, jika pengguna memilih **sebagai member** (status == "y"), program akan memulai proses autentikasi login. Pengguna diberikan **maksimal 3 kali kesempatan** untuk memasukkan *username* dan *password*. Sebelum verifikasi, program memastikan bahwa input tidak kosong — jika salah satu atau keduanya kosong, sistem akan menampilkan pesan peringatan dan meminta input ulang tanpa mengurangi kesempatan. Jika kredensial yang dimasukkan sesuai dengan data yang tersimpan (username_db = "zidan" dan password_db = "021"), maka variabel login diubah menjadi True, dan pengguna dinyatakan berhasil login. Namun, jika setelah 3 kali percobaan tetap gagal, sistem secara otomatis menganggap pengguna sebagai *non-member* dan melanjutkan proses belanja tanpa hak diskon. Mekanisme ini menunjukkan implementasi dasar dari **sistem keamanan login** dengan batasan percobaan dan validasi input, yang penting untuk mencegah penyalahgunaan serta memberikan pengalaman pengguna yang jelas dan terstruktur.

C. Fitur Pemilihan Produk dan Checkout

```
total = 0
keranjang = ""

while True:
    print("=== Menu Produk ===")
    print("1. Pensil    - Rp5.000")
    print("2. Buku      - Rp12.000")
    print("3. Penghapus - Rp3.000")
    print("4. Checkout")

    pilih = input("\nPilih menu (1-4): ")

    if pilih == "1":
```

```
            os.system("cls")
            total += 5000
            keranjang += "Pensil - Rp5.000\n"
            print("Pensil berhasil ditambahkan ke
keranjang.")
            print(f"Total sementara: Rp{total:,}\n")

        elif pilih == "2":
            os.system("cls")
            total += 12000
            keranjang += "Buku - Rp12.000\n"
            print("Buku berhasil ditambahkan ke
keranjang.")
            print(f"Total sementara: Rp{total:,}\n")

        elif pilih == "3":
            os.system("cls")
            total += 3000
            keranjang += "Penghapus - Rp3.000\n"
            print("Penghapus berhasil ditambahkan ke
keranjang.")
            print(f"Total sementara: Rp{total:,}\n")

        elif pilih == "4":
            print("\n=== Struk Belanja ===")
            print(keranjang if keranjang != "" else
"Tidak ada barang dibeli.")
            if login:
                diskon = total * 0.15
                harga_akhir = total - diskon
                print(f"Harga sebelum diskon :
Rp{total:,}")
                print(f"Diskon (15%)         :
Rp{diskon:,}")
                print(f"Total bayar          :
Rp{harga_akhir:,}")
```

```
            else:
                print(f"Total bayar          :
Rp{total:,}")
            print("\nTerima kasih sudah berbelanja!\n")
            break
        else:
            print("Pilihan tidak valid!\n")

    ulang = input("Apakah ingin memulai transaksi baru?
(y/n): ")
    if ulang != "y":
        print("Program selesai. Sampai jumpa!")
        break
```

Setelah status pengguna (member atau non-member) ditentukan, program memasuki modul utama belanja. Variabel total diinisialisasi dengan nilai 0 untuk menyimpan akumulasi harga, sedangkan variabel keranjang (bertipe string) digunakan untuk mencatat daftar barang yang dipilih. Program kemudian menampilkan menu produk yang terdiri dari tiga item: pensil (Rp5.000), buku (Rp12.000), dan penghapus (Rp3.000), serta opsi ke-4 untuk *checkout*.

Pengguna dapat memilih nomor menu berulang kali. Setiap kali memilih produk (1–3), program akan:

- Membersihkan layar menggunakan os.system("cls") agar tampilan lebih rapi,
- Menambahkan harga produk ke variabel total,
- Mencatat nama dan harga produk ke dalam keranjang,
- Menampilkan konfirmasi penambahan dan total sementara.

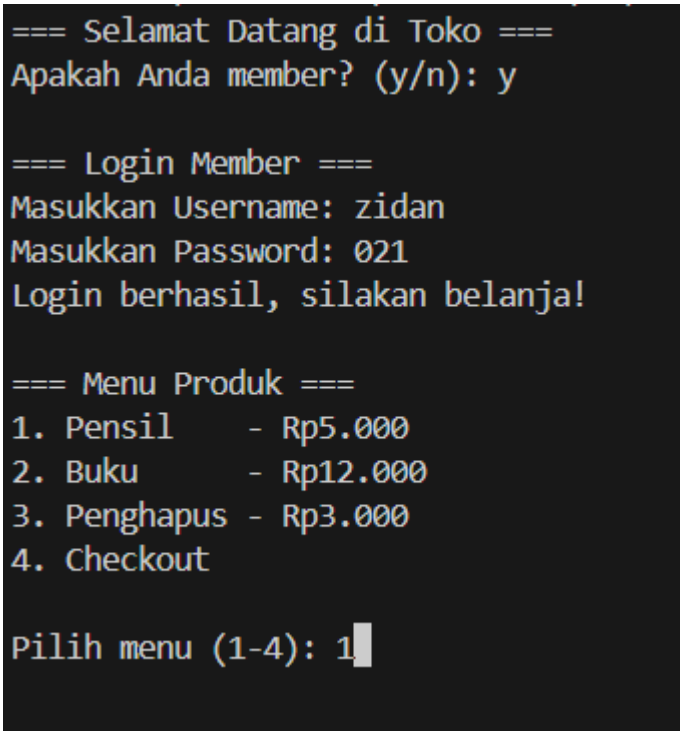
Jika pengguna memilih opsi **4 (Checkout)**, sistem akan menampilkan struk belanja berisi daftar item yang dibeli. Selanjutnya, program memeriksa nilai variabel login:

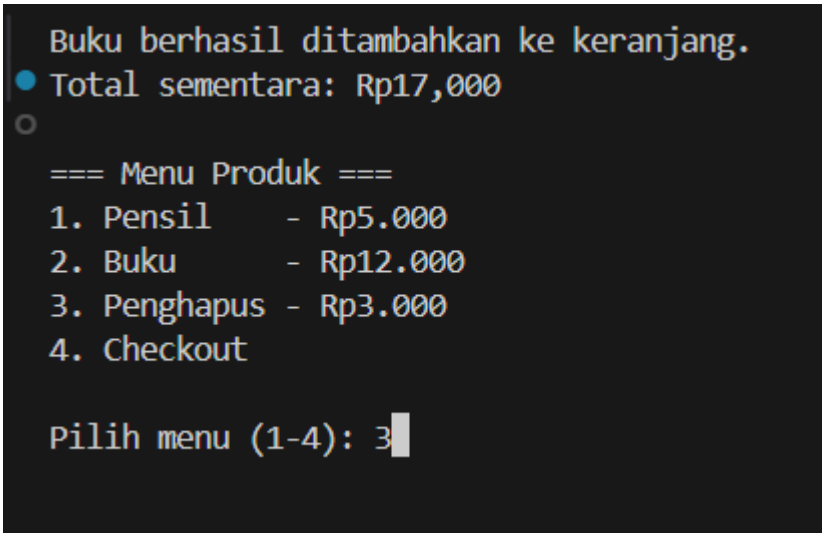
- Jika login == True (pengguna adalah member), maka diskon sebesar **15%** diterapkan pada total belanja. Program menampilkan rincian harga sebelum diskon, nilai diskon, dan total akhir yang harus dibayar.

- Jika login == False (non-member), hanya total akhir tanpa diskon yang ditampilkan.

Setelah struk ditampilkan, loop belanja dihentikan dengan perintah break. Kemudian, pengguna ditanyakan apakah ingin melakukan transaksi baru. Jika memilih "y", program akan mengulang dari awal (dengan mengatur ulang total = 0 dan keranjang = ""). Jika tidak, program menampilkan pesan penutup dan berhenti sepenuhnya

4. Output





```
Penghapus berhasil ditambahkan ke keranjang.
Total sementara: Rp20,000

=== Menu Produk ===
1. Pensil     - Rp5.000
2. Buku       - Rp12.000
3. Penghapus  - Rp3.000
4. Checkout

Pilih menu (1-4): 4

=== Struk Belanja ===
Pensil - Rp5.000
Buku - Rp12.000
Penghapus - Rp3.000

Harga sebelum diskon : Rp20,000
Diskon (15%)         : Rp3,000.0
Total bayar          : Rp17,000.0

Terima kasih sudah berbelanja!

Apakah ingin memulai transaksi baru? (y/n):
```

```
Apakah ingin memulai transaksi baru? (y/n): n
Program selesai. Sampai jumpa!
```

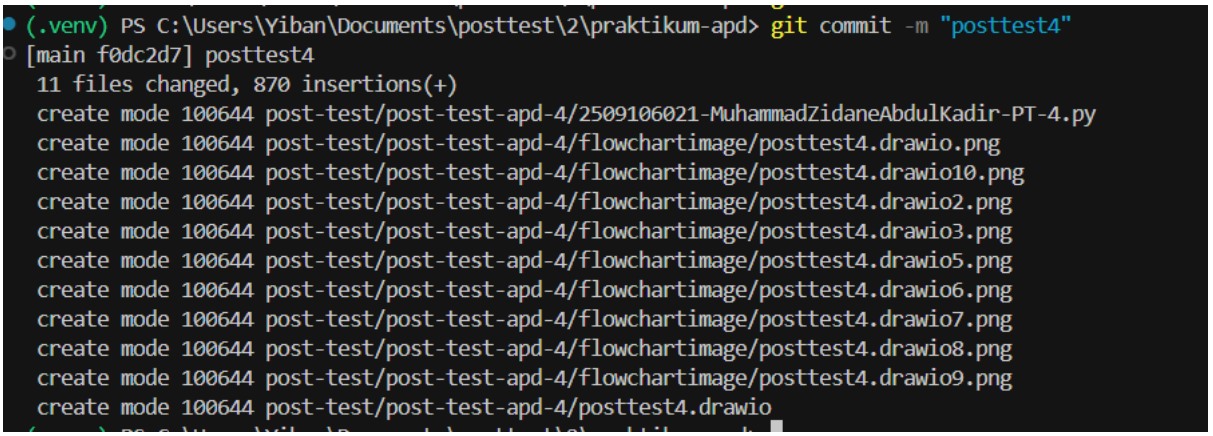
5. Git

5.1 Git Add

```
(.venv) PS C:\Users\Yiban\Documents\posttest\2\praktikum-apd> git add .
```

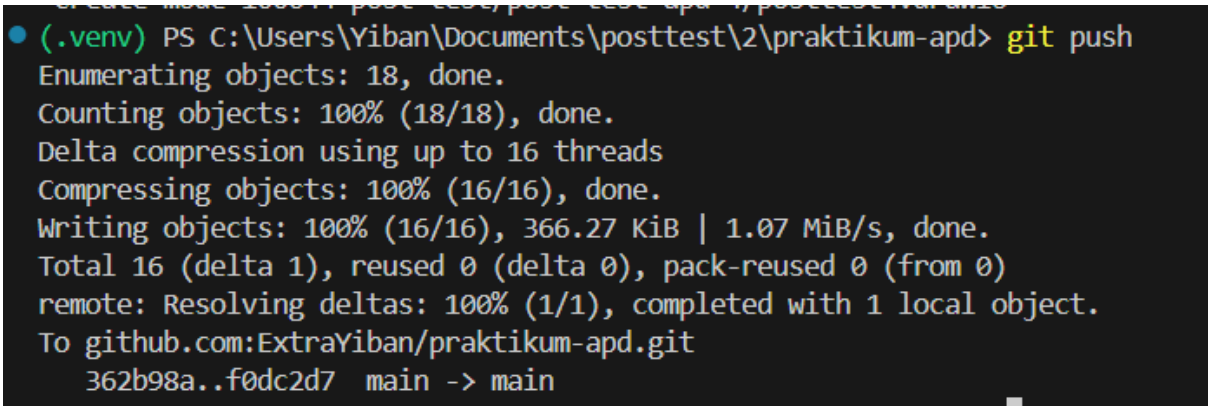
git add berfungsi untuk menambahkan file ke staging area, menandai file mana saja yang akan disimpan pada commit berikutnya.

5.2 Git Commit



git commit digunakan untuk menyimpan perubahan yang sudah di-add ke repository lokal, biasanya disertai pesan singkat yang menjelaskan perubahan tersebut.

5.3 Git Push



git push digunakan untuk mengirim commit dari repository lokal ke repository remote di GitHub, agar perubahan bisa terlihat online dan diakses oleh orang lain.