

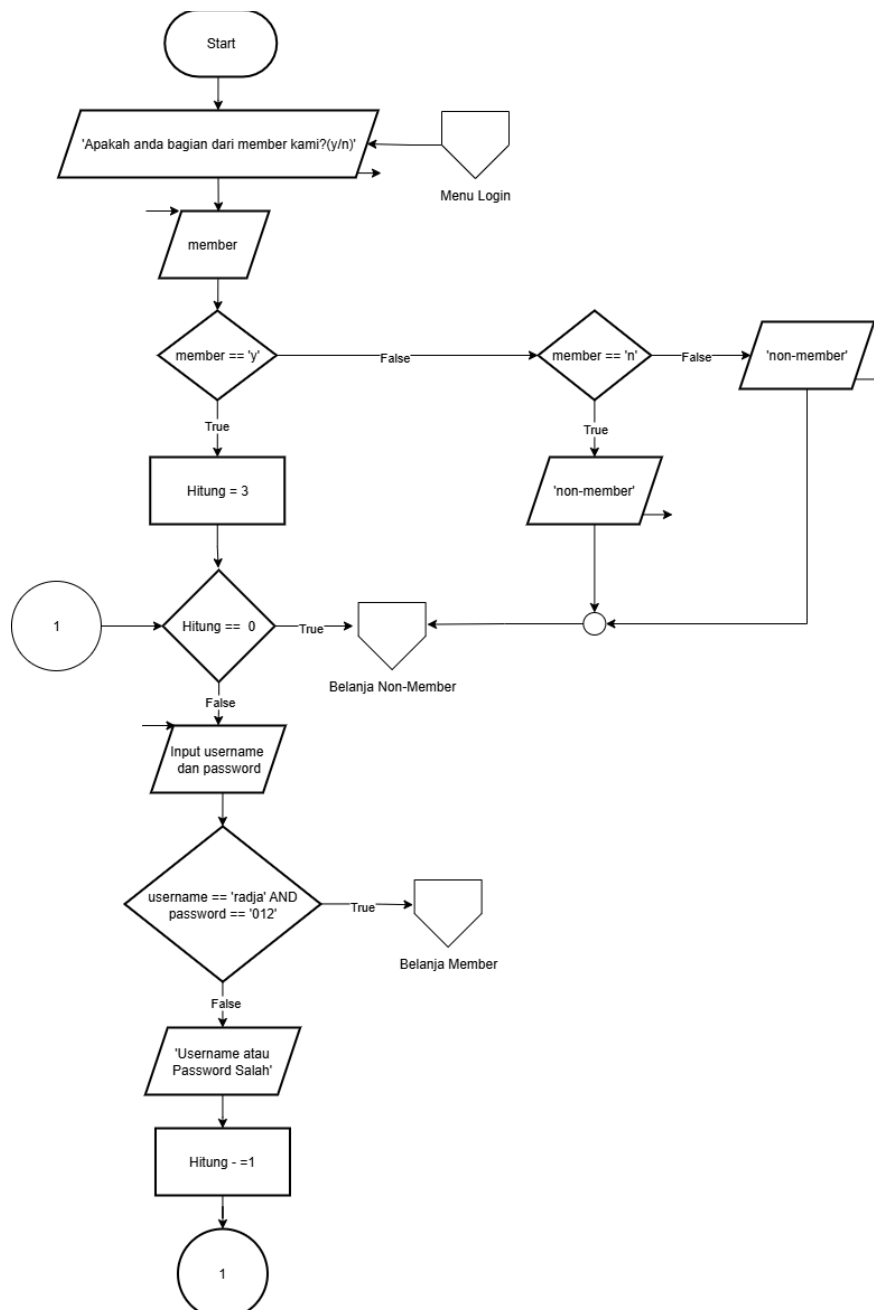
**LAPORAN PRAKTIKUM**  
**POSTTEST 4**  
**ALGORITMA PEMROGRAMAN DASAR**



**Disusun oleh:**  
**Muhamad Radja Nur Akbar (2509106012)**  
**Kelas (A1'25)**

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

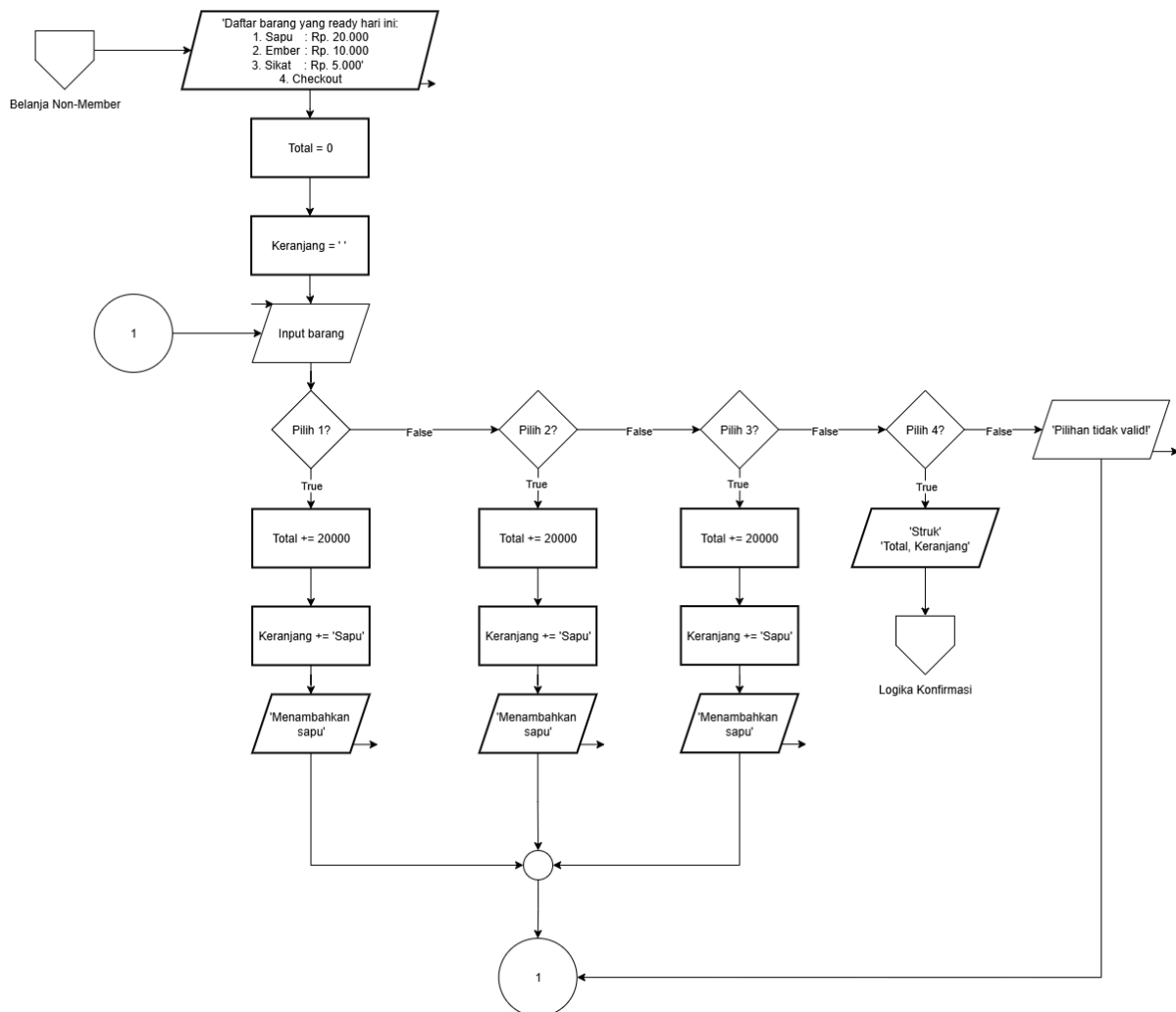
## 1. Flowchart



Gambar 1.1 Flowchart Menu Login

Pada Gambar 1.1 program menanyakan apakah mereka bagian dari *membership* atau bukan, jika pengguna mengetikkan 'y' maka akan lanjut ke fitur *login*, sebelum itu variabel 'Hitung' akan disimpan yang berjumlahkan 3, ketika *username* dan *password* yang dimasukkan **benar** maka program berlanjut ke logika belanja sebagai *member*, namun jika *username* dan *password* yang dimasukkan **salah** jumlah 'Hitung' berkurang satu dan kembali ke *login*. Saat 'Hitung' mencapai 0 fitur *login* akan berakhir dan pengguna dianggap sebagai *non-member*.

Kembali ke *input* bagian *member* atau bukan, jika pengguna mengetik 'n' atau selain itu maka program akan lanjut ke fitur belanja *non-member*.

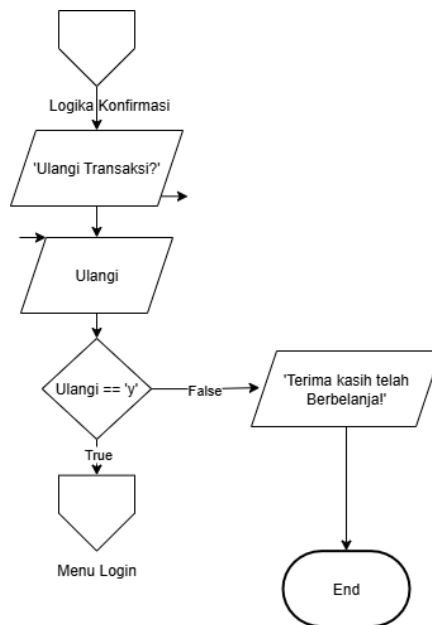


Gambar 1.2 Flowchart Belanja Non-member

Gambar 1.2 fitur belanja *non-member* menggunakan *off-page reference* yaitu ketika sebuah *flowchart* terlalu panjang atau lebar untuk satu halaman lalu dibagi menjadi beberapa *flowchart* dan diletakkan di halaman yang berbeda-beda, ini berfungsi untuk membuat *flowchart* menjadi lebih rapi dan nyaman dibaca bagi semua kalangan. Pengguna diberi 4 pilihan, 3 diantaranya adalah jenis barang yang bisa dibeli. Sebelum ke percabangan ada variabel 'Total' yang isinya 0 atau harga barang nantinya dan 'keranjang' yaitu variabel yang menambahkan setiap yang dibeli ke keranjang.

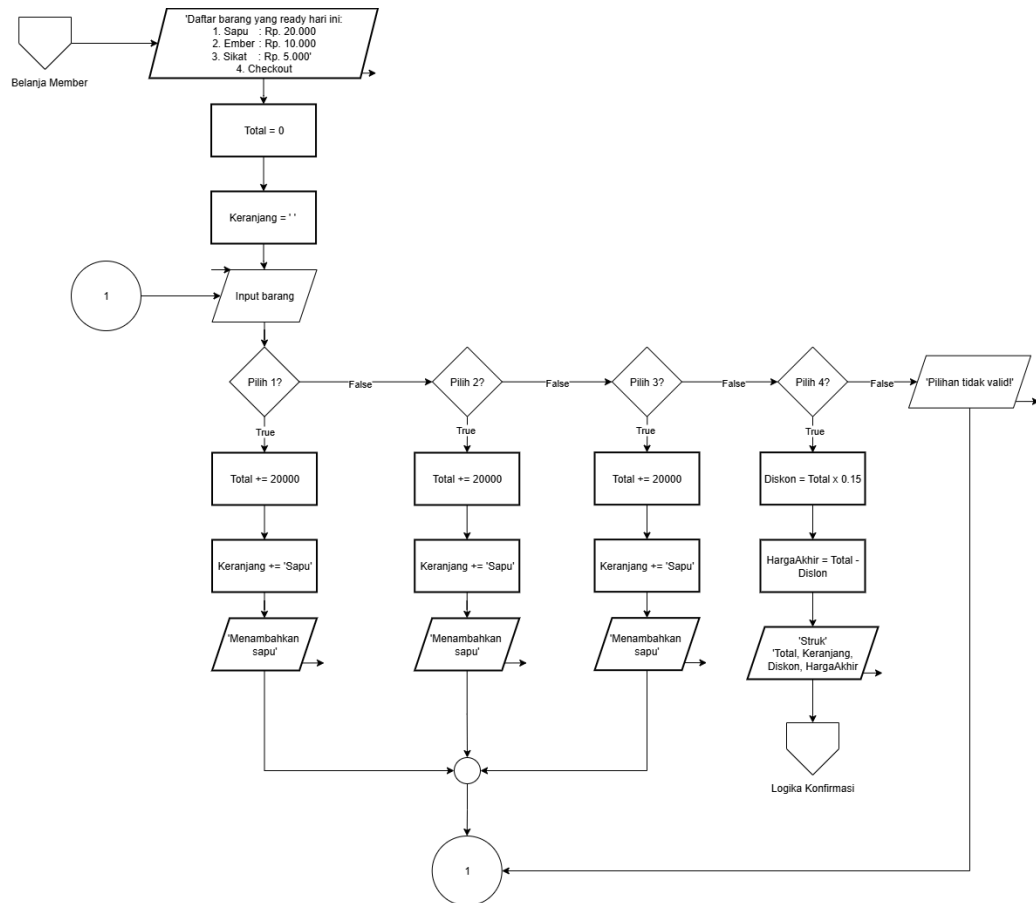
Masuk ke percabangan setiap barang memiliki harga yang berbeda saat dimasukkan ke keranjang lalu kembali ke *input* barang, fitur ini menggunakan *on-page reference* yang di mana memiliki dua fungsi, ibarat kata simbol ini sebagai portal untuk menghubungkan fitur

perulangan seperti pada gambar di atas sehingga *flowchart* menjadi lebih rapi tanpa harus menarik garis yang merusak estetika. Fungsi kedua yaitu menghubungkan dua *flowchart* berbeda yang masih dalam satu halaman. Kembali ke percabangan, jika pengguna mengetikkan '4' alias menu *checkout* maka struk dicetak, menampilkan barang-barang yang dibeli dan total harga. Namun jika pengguna memasukkan pilihan selain empat tadi akan dihitung sebagai pilihan tidak valid dan otomatis kembali ke *input* barang.



Gambar 1.3 Flowchart Logika Konfirmasi

Setelah struk dicetak, program tidak langsung berakhir melainkan ada opsi seterusnya, program akan bertanya ingin mengulang transaksi atau tidak, jika tidak maka program berakhir jika iya program kembali *Gambar 1.1* apakah bagian *membership* atau bukan menggunakan *off-page reference*.



Gambar 1.4 Flowchart Belanja Member

Mari kita kembali ke fitur *login*, pengguna adalah bagian dari *member* dan memasukkan *username* dan *password* yang benar dan program berlanjut ke fitur belanja sebagai *member*.

Tidak ada banyak perbedaan dengan *non-member*, pembeda hanya di fitur diskon, setelah *checkout* (opsi 4) 'Total' di diskon 15% dan 'hargaAkhir' diperoleh mengurangi 'Total' dengan 'Diskon'. Terakhir pencetakan struk dengan tambahan diskon dan harga akhir dan lanjut ke logika konfirmasi.

## 2. Deskripsi Singkat Program

Tujuan dari program ini untuk membuat fitur login bagi para *membership* sehingga tidak bergabung dengan mereka yang adalah pelanggan biasa. Program ini juga memberi kelebihan bagi para *membership* yaitu diskon untuk setiap jenis barang yang dibeli.

### 3. Source Code

#### 3.1 Fitur Hapus (cls)

Ketika sedang bekerja terminal Python, tampilan sering kali terlihat penuh, entah karena hasil *output* sebelumnya, terjadi *error*, dan sebagainya. Untuk membersihkan itu semua, kita bisa mengetikkan 'cls' di terminal selama kode belum berjalan atau yang lebih cepat lagi dengan menekan *ctrl + l*, ini akan langsung menghapus semua *output* yang ada di terminal. Namun, jika kita ingin membersihkan terminal saat kode berjalan, kita bisa mengetikkan ini di kode.

```
import os
os.system('cls')
```

Gambar 3.1 Fitur Hapus

Fitur ini sangat berguna terutama di perulangan karena dalam perulangan terminal menjadi sangat penuh akibat kode yang terus berjalan, untuk mencegah hal itu diperlukanlah kode di atas.

```
import os

usr = 'radja'
pw = '012'

print(''''---TOKO CERIA---
Selamat datang di Toko Ceria''')
print('Anda akan mendapat diskon 15% jika bagian dari member kami')
```

Gambar 3.1 Kode Pembuka

Seperti pada kode sebelumnya kita harus meng-*import* os agar bisa menggunakan kode 'cls'. Lanjut ke deklarasi variabel 'usr' dan 'pw' yang nantinya akan dipake di fitur *login*. Ditambah dengan sedikit kalimat pembuka dan keterangan diskon 15% untuk *member*.

#### 3.2 Fitur Login

```
while True:
    keranjang = ''
    total = 0
    loginn = False
    member = input('Apakah anda bagian dari membership?(y/n):')

    if member == 'y':
```

```

hitung = 3
while hitung > 0:
    username = input('Username\t:')
    password = input('Password\t:')
    if username == '' or password == '':
        print('Username dan Password tidak boleh kosong!')
        continue
    if username == usr and password == pw:
        loginn = True
        break
    else:
        print('Username atau Password salah!')
        hitung -= 1
        print(f'Sisa kesempatan login: {hitung}')
    if hitung == 0:
        print('Kesempatan login telah habis! Anda berbelanja sebagai
pelanggan biasa')

elif member == 'n':
    print('Anda berbelanja sebagai pelanggan biasa')

else:
    print('Anda berbelanja sebagai pelanggan biasa')

```

Gambar 3.2 Logika Login dengan Menggunakan Loop While

Kita mulai dengan *while True* yang menjadi pembungkus semua kode-kode berikutnya, *while* memiliki kondisi dan akan terus berulang selama kondisi terpenuhi. Di sini *while True* pembungkus akan berhenti setelah ada *break* di Fitur Konfirmasi nanti, jadi selama belum ke fitur tersebut program akan terus jalan.

lalu kemudian ada tiga variabel baru, masing-masing memiliki nilai sendiri seperti pada kode di atas. Ada 'loginn' dengan *value False*, 'keranjang' yang memiliki *value* ' ', kenapa kosong? Ini berfungsi menampung barang yang dibeli nantinya lalu di cetak di Logika *Checkout*. Lalu 'total' dengan *value* '0' akan menampung harga barang-barang tersebut nantinya.

Setelah itu ada permintaan *input* bagian dari *membership* atau tidak. Mengetik 'y' akan lanjut ke logika *login*, jika diperhatikan ada variabel 'hitung' dengan *value* 3, anggapannya ini kesempatan login kamu sebanyak tiga kali. Lanjut, selama 'hitung' lebih dari nol, logika *login membership* masih berjalan, meminta pengguna memasukkan *username* dan *password* yang tepat, **catatan:**

```
if username == '' or password == '':  
    print('Username dan Password tidak boleh kosong!')  
    continue
```

ini menandakan *username* dan *password* tidak boleh kosong, jika kosong akan ada pesan seperti di atas dan kode berlanjut, di sini fungsi *continue* ialah melewati iterasi saat ini dan lanjut ke iterasi berikutnya.

Kembali ke *login*, jika *username* dan *password* benar, logika *login* akan berakhir dengan *break* yaitu perintah khusus yang dipakai untuk memaksa sebuah perulangan berhenti sebelum waktunya. Jika diperhatikan ada

```
if username == usr and password == pw:  
    loginn = True  
    break
```

‘loginn = True’ berfungsi untuk melanjutkan ke Logika Belanja, sebelumnya *value* ini berisi *False*. Sebelum ke Logika Belanja, mari kita bahas apa yang terjadi jika *login* salah dan kesempatan habis, dan jika pengguna bukan member.

```
else:  
    print('Username atau Password salah!')  
    hitung -= 1  
    print(f'Sisa kesempatan login: {hitung}')  
    if hitung == 0:  
        print('Kesempatan login telah habis! Anda berbelanja sebagai  
pelanggan biasa')  
  
    elif member == 'n':  
        print('Anda berbelanja sebagai pelanggan biasa')  
  
else:  
    print('Anda berbelanja sebagai pelanggan biasa')
```

Jika pengguna salah memasukkan *username* atau *password*, kesempatan *login* berkurang satu, ini menggunakan ‘hitung -=1’ dan program menampilkan sisa kesempatan, ‘-=’ digunakan untuk mengurangi *value* di dalam sebuah variabel dengan *value* lain. Jika ‘hitung’ habis atau ‘0’, pengguna dinyatakan sebagai pelanggan biasa dan menuju Logika Belanja tanpa diskon. Ini juga berlaku jika pengguna sebelumnya *menginput* ‘n’ atau selain kedua itu, maka pengguna berbelanja sebagai pelanggan biasa.



### 3.3 Logika Belanja

```
while True:
    print('=' * 39)
    print('|No          |Barang          |Harga(Rp)  |')
    print('-' * 39)
    print('|1          |Sapu            |20.000     |')
    print('|2          |Ember           |10.000     |')
    print('|3          |Sikat           |5.000      |')
    print('-' * 39)
    print('|4          |Checkout       |            |')
    print('=' * 39)
    pilihan = int(input('Pilih (1-4):'))

    if pilihan == 1:
        keranjang += 'Sapu\t:20.000\n'
        total += 20000
        os.system('cls')
        print('Sapu dimasukkan ke keranjang')
        print(f'Total sementara: {total}')
    elif pilihan == 2:
        keranjang += 'Ember\t:10.000\n'
        total += 10000
        os.system('cls')
        print('Ember dimasukkan ke keranjang')
        print(f'Total sementara: {total}')
    elif pilihan == 3:
        keranjang += 'Sikat\t:5.000\n'
        total += 5000
        os.system('cls')
        print('Sikat dimasukkan ke keranjang')
        print(f'Total sementara: {total}')
    elif pilihan == 4:
        os.system('cls')
        print('=' * 19)
        print('---STRUK BELANJA---')
        print('=' * 19)
        print(f'{keranjang}')

        if loginn: #Jika Member
            potongan_harga = total * 0.15
            harga_akhir = total - potongan_harga
            print(f'Harga Asli\t: Rp. {total:,}')
            print(f'Diskon Member\t: (15%): Rp. -{potongan_harga:,}')
            print(f'Total\t          : Rp. {harga_akhir:,}')
            break
```

```

        else: #Non-Member
            print(f'Total\t\t\t\t\t: Rp. {total:,}')
            print('---TERIMA KASIH TELAH BERBELANJA---')
            break
    else:
        os.system('cls')
        print('Input tidak valid!')

```

Gambar 3.3 Logika Belanja

*while True* di atas adalah *while* yang membungkus logika belanja, jadi akan terus berjalan sampai ada kondisi yang memicu untuk berhenti. Di bawahnya terdapat susunan *print* mirip seperti tabel, nantinya di *output* daftar barang akan tampil di dalam tabel beserta harga dan pilihannya, disini ada 4 pilihan berbeda.

Ambil contoh di pilihan 1 atau sapu maka sapu akan ditambahkan ke ‘keranjang’ menggunakan ‘+=’, menggunakan ‘\t’ agar jarak antar barang dengan titik dua ‘:’ sama dengan barang di bawahnya, untuk ‘\n’ agar barang yang telah ditambahkan berada di bawah barang sebelumnya saat di *print*. Selesai dengan ‘keranjang’ kita pinda ke ‘total’ harga sapu adalah 20.000, maka nilai ‘total’ akan bertambah sesuai harga barang. Nah di sini fitur hapus terpakai, setelah memasukkan ‘1’, seluruh *output* di terminal akan terhapus tapi tidak dengan tabel barang karena *while True* di logika belanja tadi. Setelah itu akan menampilkan barang apa yang ditambahkan ke ‘keranjang’ karena ini sapu maka sapu akan ditampilkan dan ‘total’ belanja saat ini ditambahkan.

Pilihan 2 dan 3 juga kurang lebih sama, hanya nama barang dan harga yang berbeda. Pilihan 1,2, dan 3 memiliki satu kesamaan lagi, setelah memilih salah satu pilihan, kita bisa memilih lagi barang yang diinginkan tidak peduli berapapun itu, ini disebabkan tidak adanya kondisi diantara tiga pilihan itu yang dapat membuat *while True* dari logika belanja berhenti, disinilah fungsi *checkout* atau pilihan ke-4.

Pilihan ke-4 struk akan dicetak menyusun ‘keranjang’ dengan rapi, kemudian di sini logika *member* dan *non-member*, sebelumnya, ketika *login* lalu *username* dan *password* benar, maka *value* dari ‘loginn’ menjadi *True*, maka ‘if loginn ==:’ menjadi terpenuhi. Pengguna mendapat diskon sebesar 15% dan saat mencetak struk ada tambahan harga sebelum diskon, berapa diskonnya, dan harga setelah diskon, ada perintah *break* setelah itu maka logika belanja akan berakhir. Lalu jika sebelumnya pengguna gagal *login* tiga kali atau sedari awal bukan bagian dari *membership*, maka bagian diskon tidak akan terpenuhi dan menuju ‘else’ yang hanya menampilkan struk berisi jumlah barang dan harga.

Bagaimana jika pengguna *menginput* angka selain pilihan di atas? Ada solusinya, di kumpulan *line* paling bawah ada kondisi *else* yang di mana akan menampilkan pesan ‘Input tidak valid’ lalu kembali ke menu *input* pilihan.

### 3.4 Logika Konfirmasi

```
while True:
    ulang = input('Ingin memulai transaksi baru?(y/n):')
    if ulang == 'y':
        break
    elif ulang == 'n':
        print('Terima kasih telah berbelanja!')
        break
    else:
        print('Input tidak valid')

if ulang == 'n':
    break
```

Gambar 3.4 Logika Konfirmasi

Jika kita berpikir setelah mencetak struk program berakhir maka salah, program akan kembali ke menu pertanyaan *membership* atau tidak dan akan terus berulang, itu karena tidak ada yang menghentikan *while True* pembungkus. Maka diperlukanlah logika konfirmasi, pengguna akan ditanya apakah ingin memulai transaksi baru atau tidak. Jika ‘y’ maka *while* konfirmasi berakhir dan kembali ke *while* utama, jika ‘n’ maka mengakhiri *while* logika konfirmasi. Lalu jika *input* selain ‘y’ atau ‘n’, maka tidak valid. Terakhir ada *if* yang mengulang *if* sebelumnya untuk mengakhiri *while* utama.

#### 4. Hasil Output

```
---TOKO CERIA---
Selamat datang di Toko Ceria
Anda akan mendapat diskon 15% jika bagian dari member kami
Apakah anda bagian dari membership?(y/n):y
Username      :radja
Password      :012
=====
|No           |Barang       |Harga(Rp)   |
|-----|
|1           |Sapu         |20.000      |
|2           |Ember        |10.000      |
|3           |Sikat        |5.000       |
|-----|
|4           |Checkout     |             |
|-----|
Pilih (1-4):
```

Gambar 4.1 Login Membership Berhasil dan Lanjut ke Logika Belanja

```
=====
---STRUK BELANJA---
=====
Sapu      :20.000
Ember     :10.000
Sikat     :5.000

Harga Asli      : Rp. 35,000
Diskon Member   : (15%): Rp. -5,250.0
Total           : Rp. 29,750.0
Ingin memulai transaksi baru?(y/n):
```

Gambar 4.2 Struk Membership

```

=====
---STRUK BELANJA---
=====
Sapu      :20.000
Ember     :10.000
Sikat     :5.000

Harga Asli      : Rp. 35,000
Diskon Member   : (15%): Rp. -5,250.0
Total           : Rp. 29,750.0
Ingin memulai transaksi baru?(y/n):y
Apakah anda bagian dari membership?(y/n):y
Username        :radja
Password        :012
=====
|No          |Barang      |Harga(Rp)  |
|-----|
|1           |Sapu        |20.000     |
|2           |Ember       |10.000     |
|3           |Sikat       |5.000      |
|-----|
|4           |Checkout    |           |
|-----|
Pilih (1-4):

```

*Gambar 4.3 Transaksi Ulang*

```

=====
---STRUK BELANJA---
=====
Sapu      :20.000
Ember     :10.000
Ember     :10.000

Harga Asli      : Rp. 40,000
Diskon Member   : (15%): Rp. -6,000.0
Total           : Rp. 34,000.0
Ingin memulai transaksi baru?(y/n):n
Terima kasih telah berbelanja!
PS D:\Praktikum APD>

```

*Gambar 4.4 Akhiri Transaksi*

```

=====
---STRUK BELANJA---
=====
Sapu      :20.000
Ember     :10.000
Sikat    :5.000

Total           : Rp. 35,000
---TERIMA KASIH TELAH BERBELANJA---
Ingin memulai transaksi baru?(y/n):n
Terima kasih telah berbelanja!
PS D:\Praktikum APD>

```

Gambar 4.5 Struk Non-member

```

---TOKO CERIA---
Selamat datang di Toko Ceria
Anda akan mendapat diskon 15% jika bagian dari member kami
Apakah anda bagian dari membership?(y/n):y
Username      :ra
Password      :1
Username atau Password salah!
Sisa kesempatan login: 2
Username      :ad
Password      :12
Username atau Password salah!
Sisa kesempatan login: 1
Username      :1
Password      :2
Username atau Password salah!
Sisa kesempatan login: 0
Kesempatan login telah habis! Anda berbelanja sebagai pelanggan biasa
=====
|No      |Barang    |Harga(Rp) |
|-----|-----|-----|
|1       |Sapu      |20.000    |
|2       |Ember     |10.000    |
|3       |Sikat     |5.000     |
|-----|-----|-----|
|4       |Checkout  |           |
|-----|-----|-----|
Pilih (1-4):

```

Gambar 4.6 Gagal Login Tiga Kali

```

=====
---STRUK BELANJA---
=====
Ember     :10.000
Sapu      :20.000

Total           : Rp. 30,000
---TERIMA KASIH TELAH BERBELANJA---
Ingin memulai transaksi baru?(y/n):j
Input tidak valid
Ingin memulai transaksi baru?(y/n):

```

Gambar 4.7 Input Selain y dan n

## 5. Langkah-Langkah GIT

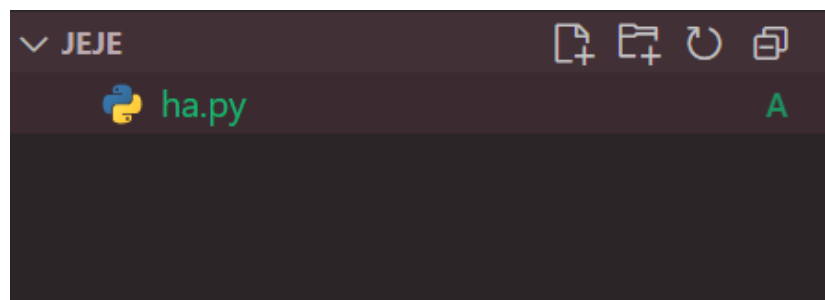
Jika dalam suatu folder kita sudah pernah melakukan *git init* dan *git remote* maka kita tidak perlu lagi melakukannya dan langsung ke *git add*.

### 5.1 GIT Add

```
PS D:\jeje> git add ha.py
PS D:\jeje> git add .
PS D:\jeje> 
```

Gambar 5.1 Proses Git Add

*Git add* berfungsi untuk memindah semua perubahan di area kerja ke indeks, ada dua cara untuk melakukan *git add*, yang pertama dengan mengetikkan nama *file* yang ingin di *add*. Jika kalian ingin menambahkan banyak *file* sekaligus, ketik “*git add .*”



Gambar 5.2 Sebuah File yang Telah di Add

*File* yang telah di *add* akan menampilkan huruf ‘A’ tepat di sebelah nama *file*, ini menandakan *file* sudah ditambahkan.

### 5.2 GIT Commit

```
PS D:\jeje> git commit -m "hellothere"
[main (root-commit) 55b666b] hellothere
1 file changed, 1 insertion(+)
create mode 100644 ha.py
PS D:\jeje> 
```

Gambar 5.3 Proses Git Commit

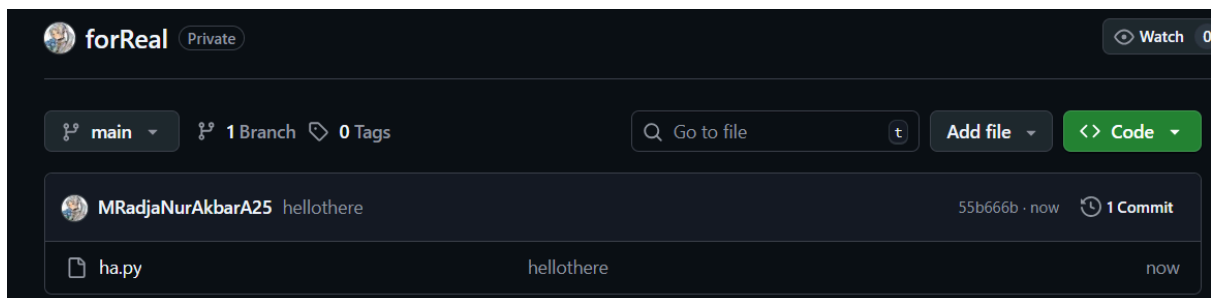
*Git commit* berfungsi untuk mengkonfirmasi setiap perubahan pada *repository* kalian dengan mengetikkan “`git commit -m ‘pesan yang ingin ditulis’`”

### 5.3 GIT Push

```
PS D:\jeje> git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 235 bytes | 235.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/MRadjaNurAkbarA25/forReal.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS D:\jeje> █
```

*Gambar 5.4 Repository Lokal diunggah ke Github*

*Git push* berfungsi untuk mengunggah *file* lokal tadi ke GitHub kalian dengan mengetikkan “`git push -u origin main`”.



*Gambar 5.5 File sudah terunggah di Akun Github*