

**LAPORAN PROYEK AKHIR PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN DASAR
GO-SIMPY: SIMULASI OJEK ONLINE BERBASIS PYHTON**



**Anugerah Fakhriza Reswara
Luvita Khairana Salwa
Anah Wiyana**

**2509106025
2509106041
2509106027**

**PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025**

1. LATAR BELAKANG

Perkembangan teknologi informasi yang sangat pesat saat ini telah memberikan dampak besar terhadap berbagai aspek kehidupan manusia, termasuk dalam bidang transportasi dan layanan publik. Namun, di balik kemudahan tersebut, diperlukan sistem yang mampu mengatur berbagai jenis pengguna dengan hak akses yang berbeda-beda.

Untuk mendukung kebutuhan tersebut, dikembangkanlah sebuah program simulasi sistem manajemen akun berbasis Python yang mengimplementasikan konsep autentikasi, validasi input, serta pembagian peran (role-based access). Program ini juga dilengkapi dengan antarmuka berbasis terminal yang interaktif dan menampilkan menu berbeda sesuai dengan peran pengguna yang sedang login.

A. Fitur Utama Sistem

Sistem GO-SIMPY (Go Simulation Python) dirancang dengan fitur-fitur utama yang menggambarkan proses autentikasi dan manajemen pengguna dalam aplikasi berbasis peran (role-based system). Setiap pengguna memiliki akses menu yang berbeda sesuai dengan perannya di dalam sistem. Adapun fitur-fitur yang tersedia dalam bentuk modular adalah sebagai berikut:

1. Fitur Autentikasi

Fitur ini bertanggung jawab untuk memastikan keamanan sistem dengan cara melakukan verifikasi data pengguna.

- **Login:** Pengguna memasukkan username dan password untuk mengakses sistem sesuai dengan perannya.
- **Registrasi:** Pengguna baru dapat membuat akun dengan mengisi data seperti nama lengkap, umur, nomor telepon, dan alamat email.
- **Logout:** Mengakhiri sesi pengguna yang sedang aktif dan kembali ke menu login.

Bagian autentikasi program ini bertugas mengatur proses login dan logout pengguna. Fungsi login meminta pengguna memasukkan username dan password,

kemudian memeriksa apakah username tersebut terdaftar di data akun dan apakah password yang dimasukkan sesuai. Jika cocok, status login diubah menjadi benar dan username pengguna disimpan di variabel global, sehingga bagian program lain dapat mengetahui siapa yang sedang aktif. Jika password atau username salah, pengguna diberi kesempatan maksimal lima kali untuk mencoba login, dan jika gagal setelah percobaan tersebut, program akan berhenti secara otomatis. Fungsi logout mengatur keluar dari sesi dengan mereset status login menjadi salah dan menghapus informasi username aktif, kemudian menampilkan pesan konfirmasi. Fungsi tambahan apakah_sudah_login digunakan untuk mengecek apakah ada pengguna yang sedang login, sementara apa_akun_sekarang mengembalikan username pengguna yang aktif. Dengan mekanisme ini, program memastikan hanya pengguna yang terdaftar dengan kredensial benar yang bisa mengakses fitur tertentu, sekaligus menjaga informasi status login agar konsisten di seluruh bagian program.

B. Fitur Manajemen Akun (CRUD)

Program ini merupakan modul untuk manajemen akun dalam sistem, yang mencakup pembuatan akun, pengeditan profil, penghapusan akun, pengecekan saldo, dan menampilkan daftar akun. Fungsi `buat_akun_customer` dan `buat_akun_driver` memungkinkan pendaftaran akun baru, dengan melakukan validasi untuk username, password, umur, nomor telepon, dan email menggunakan fungsi validasi terpisah. Setiap akun yang berhasil dibuat akan disimpan dalam dictionary akun dengan informasi lengkap seperti nama, umur, kontak, email, password, role, dan saldo (untuk driver).

Fungsi `edit_profil` memungkinkan pengguna mengubah data profil mereka, dengan kemampuan melewati field yang tidak ingin diubah dan melakukan validasi pada input umur dan email. Fungsi `hapus_akun` digunakan untuk menghapus akun berdasarkan username, sementara fungsi `tampilkan_daftar_akun` menampilkan semua akun yang terdaftar menggunakan tabel dari library `PrettyTable`. Fungsi pendukung seperti `dapatkan_data_akun` dan `akun_ada` digunakan untuk mengambil data akun tertentu dan memeriksa keberadaan username, sedangkan `cek_saldo` menampilkan saldo akun pengguna yang aktif. Secara keseluruhan, modul ini mengatur semua interaksi CRUD (Create, Read, Update, Delete) terkait akun dengan validasi input agar data yang tersimpan konsisten dan pengguna bisa melakukan registrasi serta pengelolaan akun dengan aman.

C. Struktur akun

Struktur akun adalah sebuah dictionary Python yang berfungsi sebagai penyimpanan data pengguna dalam sistem. Setiap akun direpresentasikan sebagai sebuah key berupa username, dan di dalamnya terdapat dictionary lain yang menyimpan informasi lengkap pengguna tersebut. Misalnya, akun dengan username “admin” memiliki data berupa password, nama lengkap, umur, nomor telepon, alamat email, role sebagai admin, dan saldo awal bernilai nol. Begitu juga dengan akun “driver-01” yang menyimpan data pengemudi seperti password, identitas lengkap, kontak, email, role sebagai driver, serta saldo awal.

Format penyimpanan seperti ini memudahkan program untuk mengakses, memodifikasi, atau memvalidasi data pengguna berdasarkan username, sekaligus menjadi dasar bagi proses login, registrasi, pengeditan profil, pengecekan saldo, dan fitur-fitur lainnya di dalam aplikasi. Struktur dictionary bertingkat ini juga membuat data pengguna tetap terorganisasi dengan baik dan mudah diolah.

D. Tampilan sistem

Program ini mendefinisikan tiga konstanta (LINE, DOUBLE_LINE, dan BULLET) sebagai dekorasi tampilan menu. Fungsi bersihkan_layar() digunakan untuk membersihkan terminal agar tampilan tidak menumpuk. Fungsi banner(text) menampilkan judul menu dengan garis DOUBLE_LINE dan teks yang dipusatkan.

Menu pertama yang muncul adalah tampilkan_menu_login(), yang membersihkan layar, menampilkan banner “MENU LOGIN”, lalu menampilkan pilihan login, registrasi customer, registrasi driver, dan keluar, menggunakan BULLET sebagai penanda.

Fungsi tampilkan_menu_utama(role) menampilkan menu sesuai peran pengguna. Admin mendapat menu manajemen sistem, customer mendapat menu layanan seperti memesan ojek atau cek saldo, dan driver mendapat menu seperti

cek saldo atau tarik pendapatan. Semua menu selalu diawali banner dan diakhiri garis LINE.

E. Validasi [input.py](#)

Fungsi `validasi_input_umur` memastikan umur yang dimasukkan valid. Program mencoba mengubah input menjadi integer; jika berhasil dan nilainya berada pada rentang 13–100, umur dikembalikan. Jika di luar rentang atau tidak bisa dikonversi karena input bukan angka, fungsi mengembalikan `None` sebagai tanda input tidak sah.

Fungsi `validasi_input_email` memeriksa format dasar email dengan memastikan adanya karakter "@" dan ".". Jika keduanya ada, fungsi mengembalikan `True`; jika tidak, `False`. Validasi sederhana ini membantu menyaring kesalahan penulisan email sebelum diproses lebih lanjut.

F. Fitur Role-based Menu

Setiap peran pengguna memiliki tampilan menu dan fungsi yang berbeda sesuai kebutuhan.

Admin:

1. Edit profil
2. Tambah akun customer
3. Tambah akun driver
4. Hapus akun
5. Lihat semua akun
6. Lihat semua transaksi
7. Logout
8. Keluar

Customer:

1. Edit profil
2. Pesan Ojek

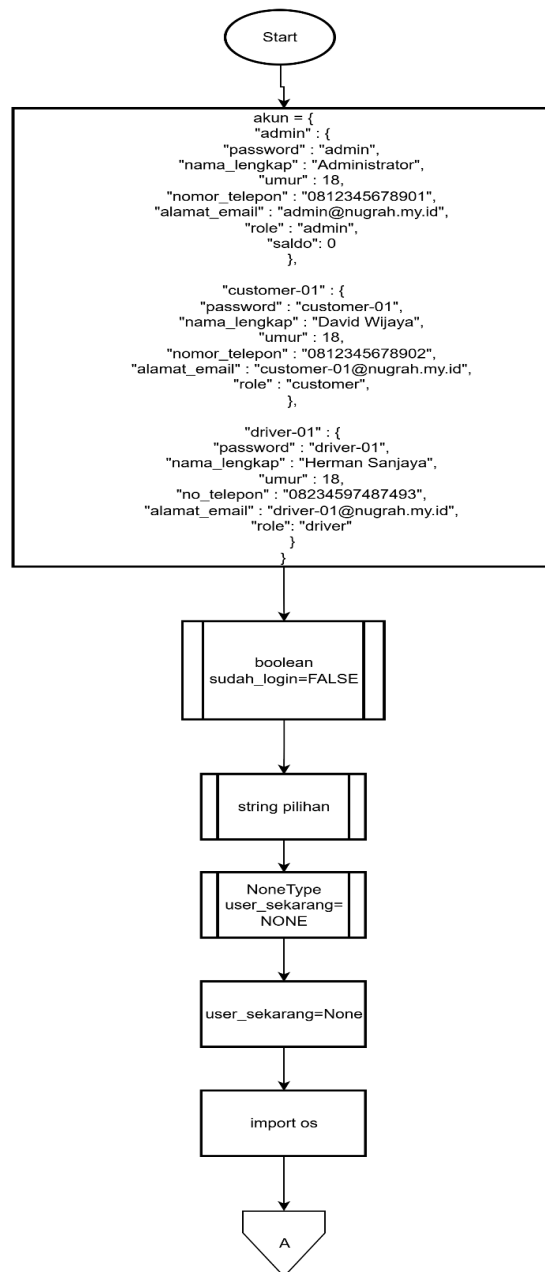
3. Lihat Histori Pemesanan
4. Logout
5. Keluar

Driver

1. Edit Profil
2. Lihat Pendapatan
3. Logout
4. Keluar

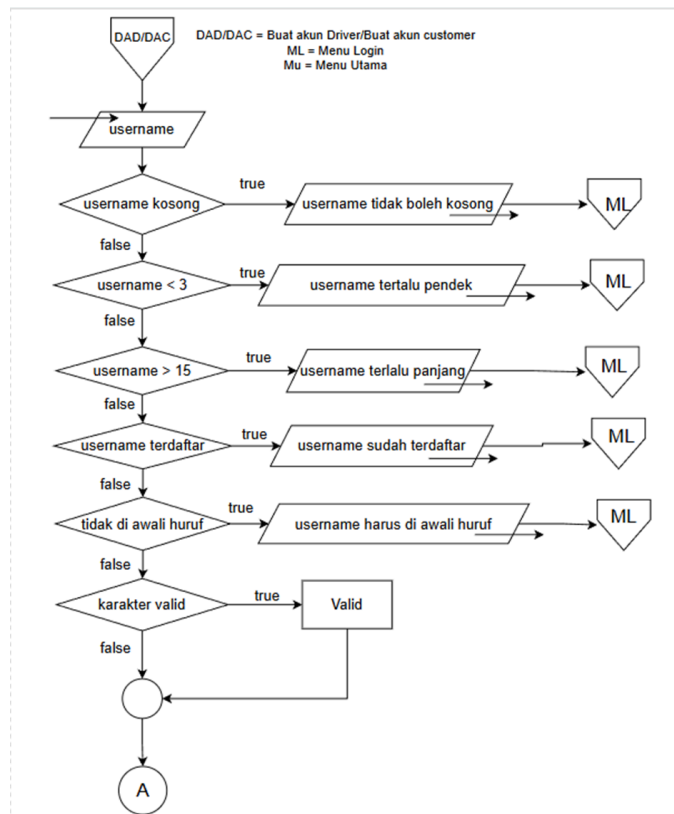
2. FLOWCHART

1. [Akun.py](#)

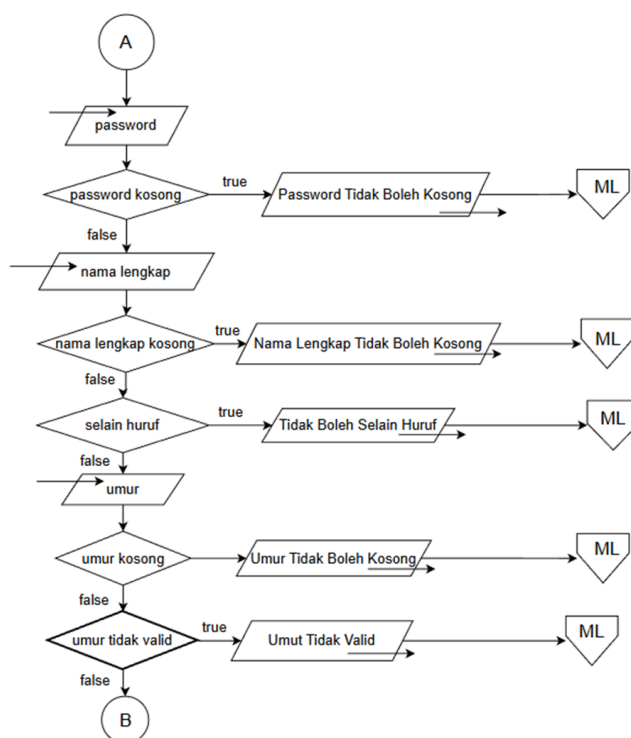


2.1 Flowchart akun.py

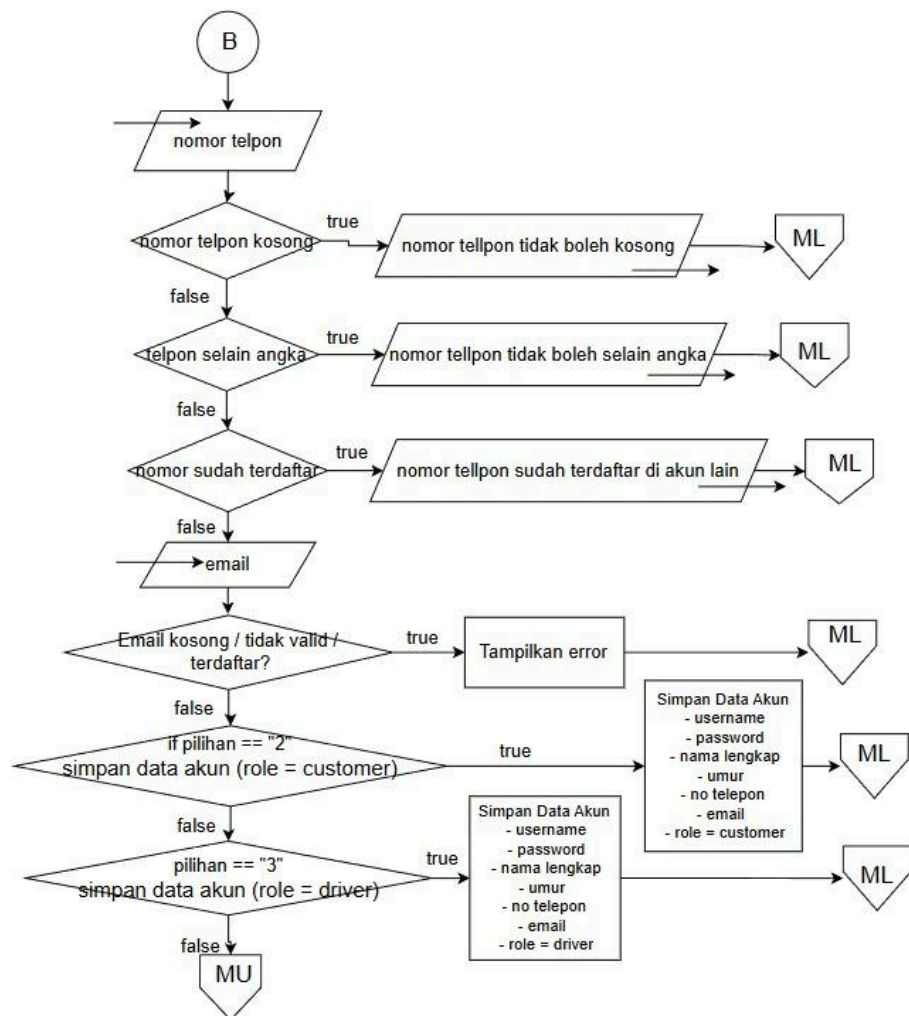
2. Manajemen akun
 - A. Menu Registrasi (Menu login dan utama)



2.2 Flowchart Menu Login untuk akun d/c



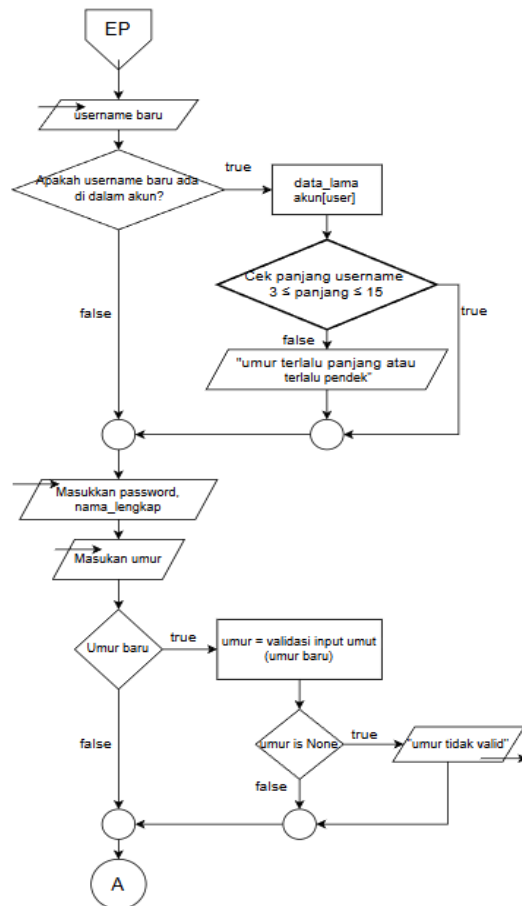
2.3 Flowchart



2.4 Flowchart Menu Login untuk akun d/c

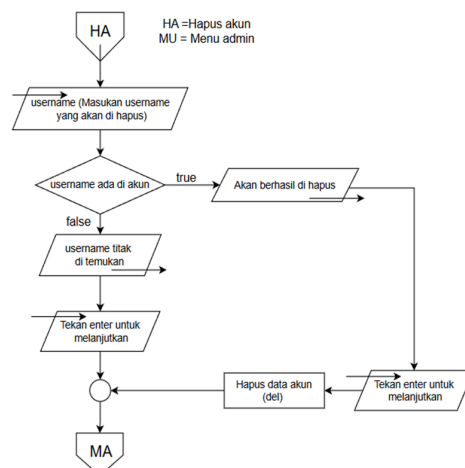
b. Menu Admin

a.edit profil



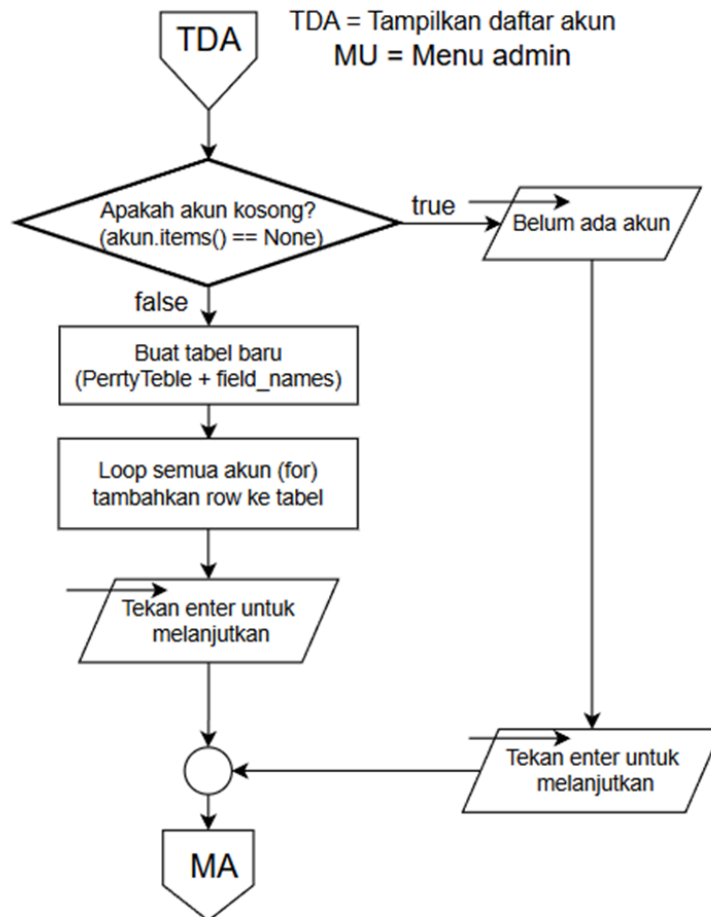
2.5 Edit Profil

b. Hapus akun



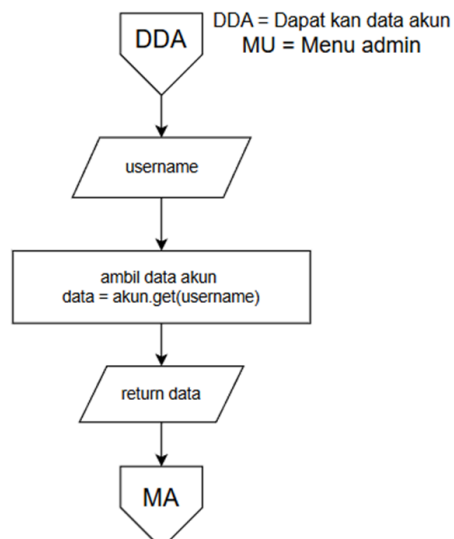
2.6 Hapus Akun

c. Tampilkan daftar akun



2.7 Tampilkan daftar akun

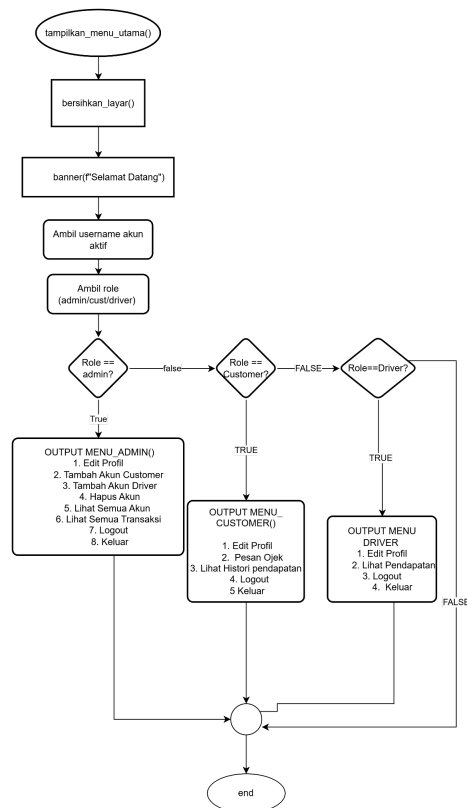
d. Dapatkan data akun



2.7 Dapatkan data akun

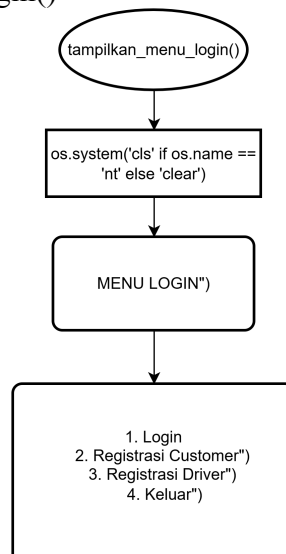
3. [Tampilan.py](#)

a. Tampilan menu_utama()



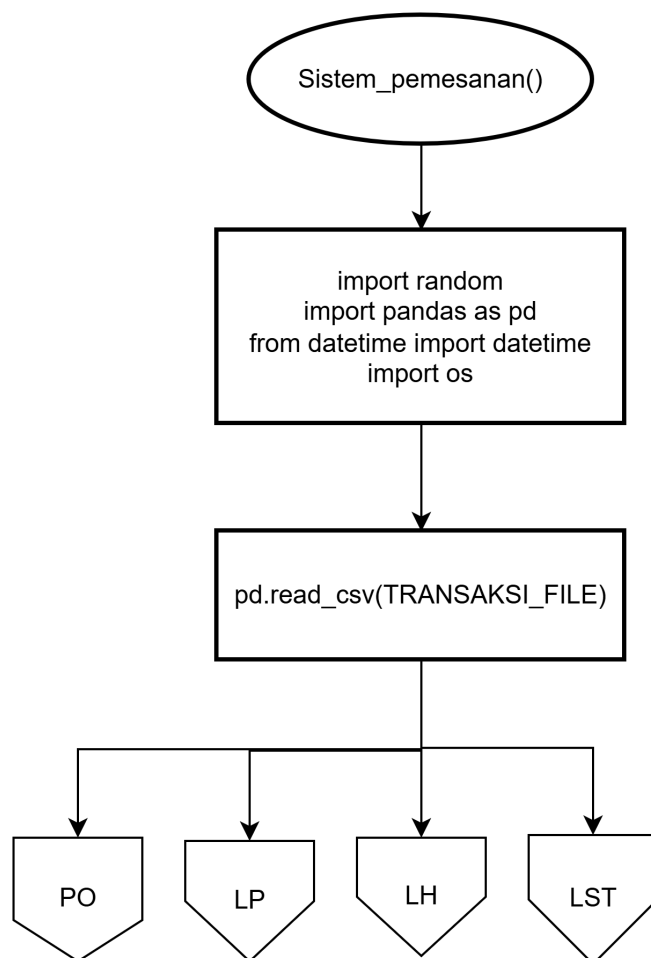
2.8 Tampilan Menu Utama

b. Tampilan menu_login()



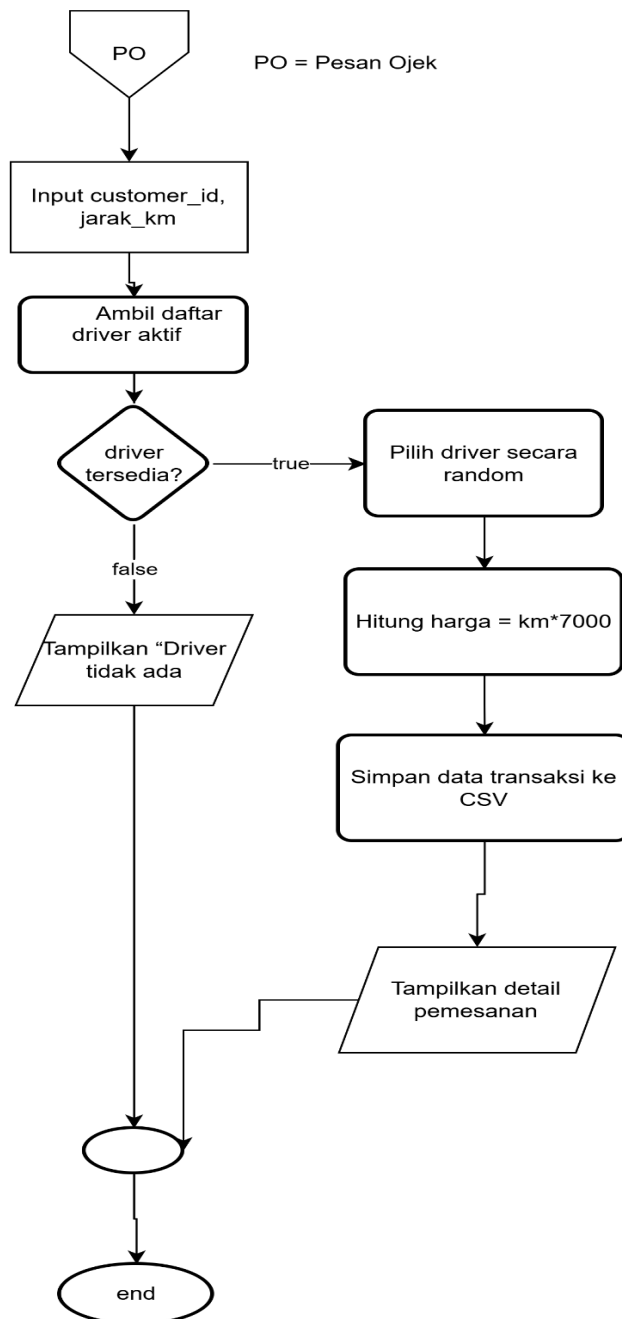
2.9 Tampilan Menu Login

4. Sistem_pemesanan.py



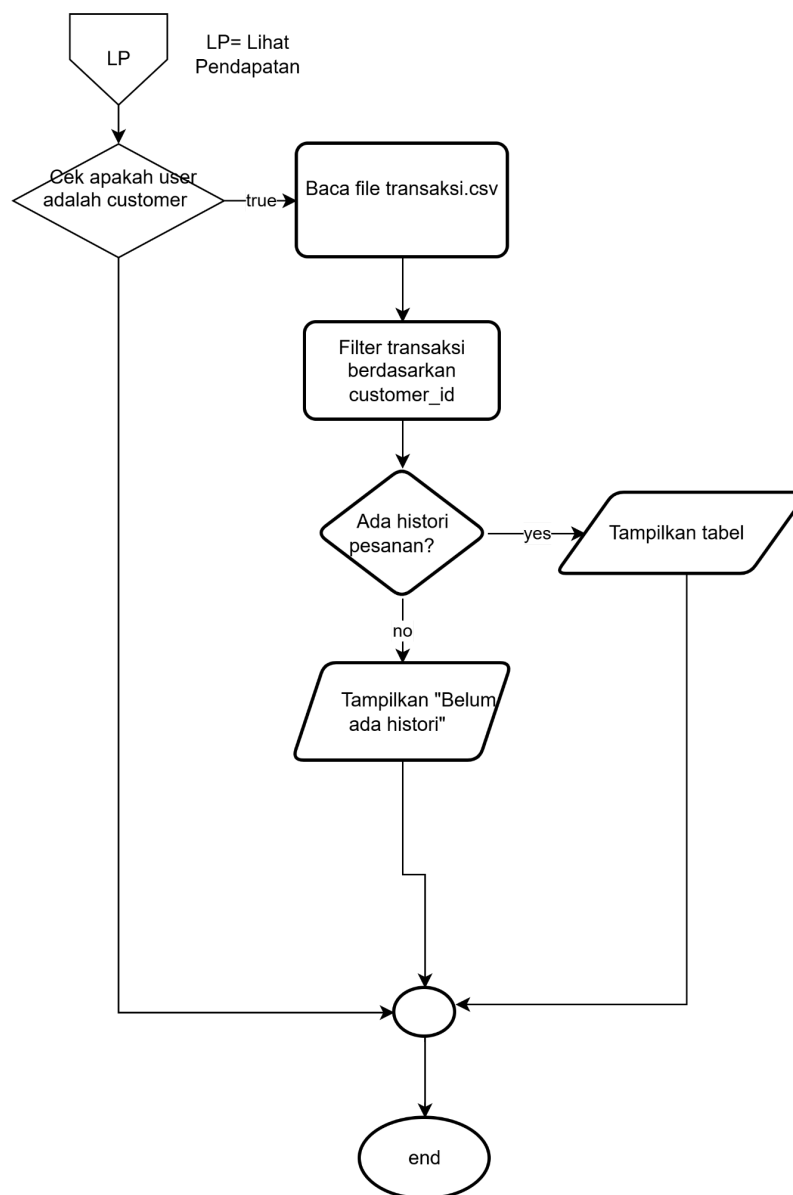
2.10 Flowchart Sistem_pemesanan.py

a. PO (pesan_ojek())



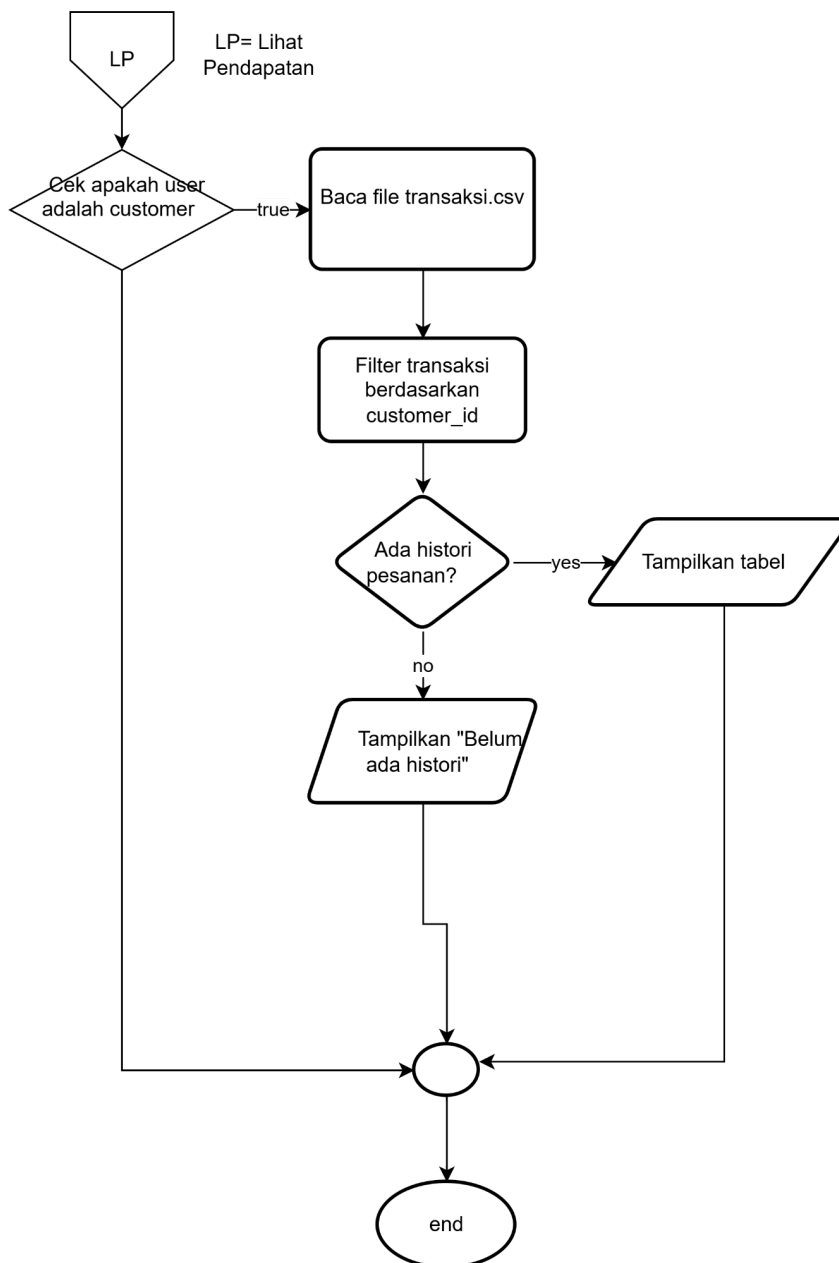
2.11 Pesan_ojek.py

b. LP (Lihat_pendapatan())



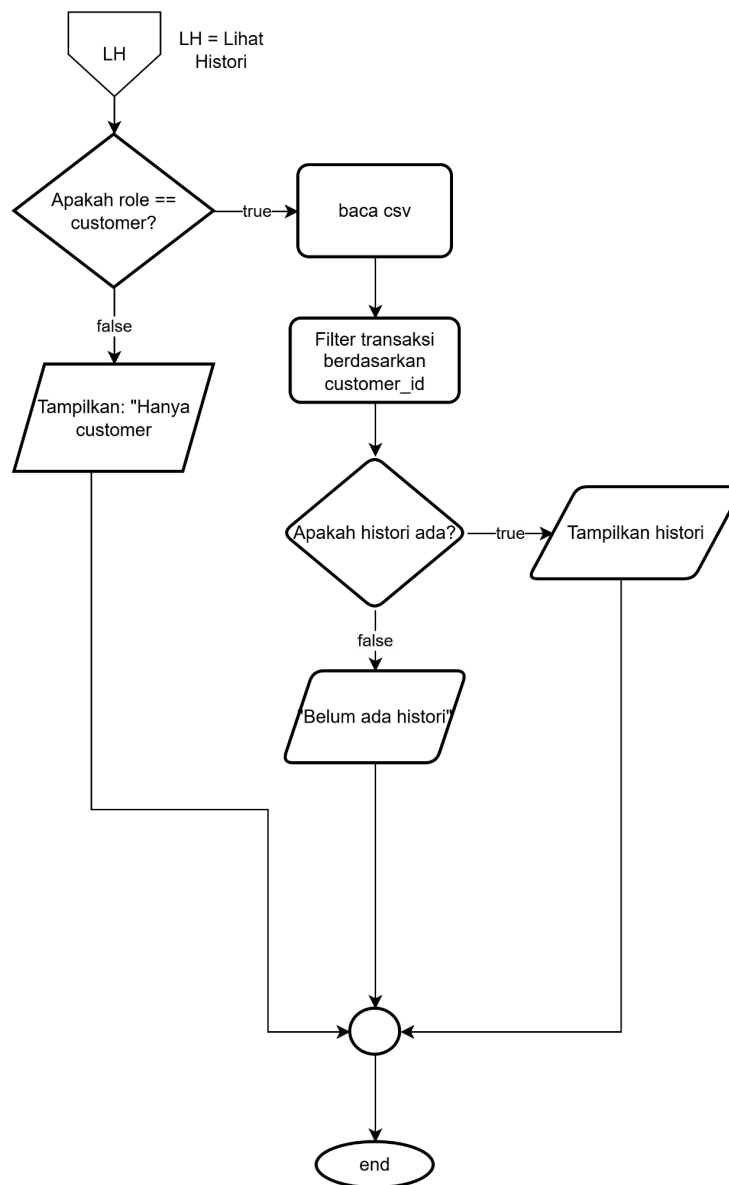
2.12 Lihat_pendapatan-py(driver)

c. LP (Lihat_pendapatan())



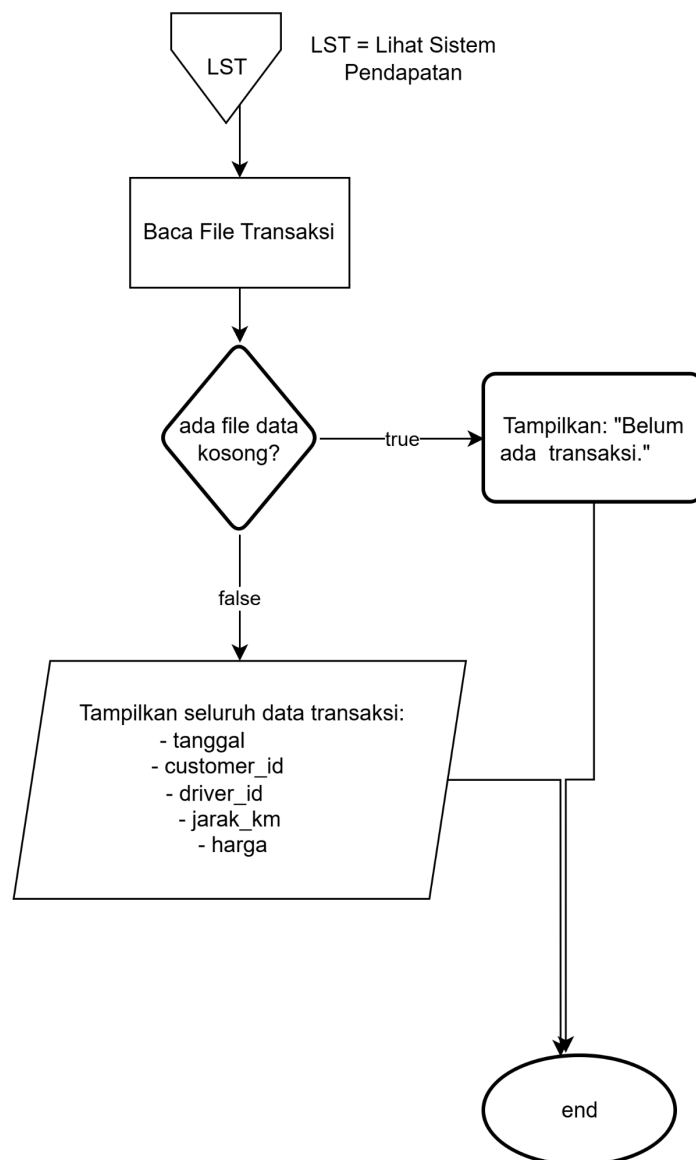
2.12 Lihat_pendapatan-py(driver)

d. LH (Lihat_Histori())



2.13 Lihat Histori (customer)

e. LST (Lihat_sistem_pendapatan())



2.14 Lihat_sistem_pendapatan

3. OUTPUT PROGRAM

a. Menu Login dan Register

```
=====
                        MENU LOGIN
=====
> 1. Login
> 2. Registrasi Customer
> 3. Registrasi Driver
> 4. Keluar

Pilih menu: █
```

3.1 Menu Login dan Register Customer dan Driver

1) Registrasi Customer

```
Pilih menu: 2
=== Proses pembuatan akun ===
Masukkan username: hai
Masukkan password: miaw
Masukkan nama lengkap: miaw miaw
Masukkan umur: 21
Masukkan nomor telepon: 08768787865
Masukkan alamat email: miawmiaw@gmail.com
Akun customer berhasil dibuat!, Silahkan login
Tekan Enter untuk melanjutkan█
```

3.2 Menu Login Pilihan 2 (Registrasi Customer)

2) Registrasi Driver

```
Pilih menu: 3
=== Proses pembuatan akun ===
Masukkan username: umaa
Masukkan password: umaa
Masukkan nama lengkap: umaa
Masukkan umur: 21
Masukkan nomor telepon: 081280290292
Masukkan alamat email: uma@gmail.com
Akun driver berhasil dibuat!, Silahkan login
Tekan Enter untuk melanjutkan█
```

Menu Login Pilihan 3 (driver)

b. Menu Admin

```
=====
                        Selamat Datang
=====
MENU ADMIN
-----
> 1. Edit Profil
> 2. Tambah Akun Customer
> 3. Tambah Akun Driver
> 4. Hapus Akun
> 5. Lihat Semua Akun
> 6. Lihat Semua Transaksi
> 7. Logout
> 8. Keluar
-----
Pilih menu: █
```

3.4 Menu Admin

1. Menu Admin bagian edit profil

```
Pilih menu: 1
===Edit Profil===
===Kosongkan jika tidak ingin mengubah!===
Masukkan username baru (admin): █
```

3.5 Menu Admin pilihan 1

2. Menu Admin bagian Tambah Akun Customer

```
Pilih menu: 2
=== Proses pembuatan akun ===
Masukkan username: hemlo
Masukkan password: hai█
```

3.6 Menu Admin pilihan 2

3. Menu Admin bagian Tambah Akun Driver

```
Pilih menu: 3
=== Proses pembuatan akun ===
Masukkan username: hai
Masukkan password: miaw
Masukkan nama lengkap: hai miaw
Nama lengkap hanya boleh diisi dengan huruf!
Tekan Enter untuk kembali...
```

3.7 Menu Admin pilihan 3

4. Menu Admin bagian Hapus Akun

```
Pilih menu: 4
Masukkan username akun yang akan dihapus: hai miaw
Username tidak ditemukan!
Tekan Enter untuk melanjutkan...
```

3.8 Menu Admin pilihan 4

5. Menu Admin bagian Lihat Semua Akun

```
Pilih menu: 5

=== DAFTAR AKUN ===
+-----+-----+-----+-----+-----+-----+
| Username | Nama Lengkap | Umur | Nomor Telepon | Alamat Email | Role |
+-----+-----+-----+-----+-----+-----+
| admin    | Administrator | 18   | 085349095718  | admin@nugrah.my.id | admin |
| customer-01 | David Wijaya | 18   | 0812345678902 | customer-01@nugrah.my.id | customer |
| driver-01 | Herman Sanjaya | 18   | 08234597487493 | driver-01@nugrah.my.id | driver |
+-----+-----+-----+-----+-----+-----+
Tekan Enter untuk melanjutkan...
```

3.9 Menu Admin pilihan 5

6. Menu Admin bagian Lihat Semua Transaksi

```
Pilih menu: 6

Semua transaksi Go-Simpy
+-----+-----+-----+-----+-----+
| Waktu | Customer | Driver | Jarak (km) | Harga (Rp) |
+-----+-----+-----+-----+-----+
Tekan Enter untuk kembali...
```

4.0 Menu Admin pilihan 6

7. Menu Admin bagian Logout

```
Pilih menu: 7
Anda telah logout, silahkan login kembali
Tekan Enter untuk kembali...
```

4.1 Menu Admin pilihan 7

c. Menu Customer

```
=====
                        Selamat Datang
=====
                        MENU CUSTOMER
=====
> 1. Edit Profil
> 2. Pesan Ojek
> 3. Lihat histori pemesanan
> 4. Logout
> 5. Keluar
=====
Pilih menu: 
```

4.2 Menu Customer

1. Edit Profil

```
Pilih menu: 1
===Edit Profil===
===Kosongkan jika tidak ingin mengubah!===
Masukkan username baru (kelompok2): miaw
Masukkan password baru: miaw
Masukkan nama lengkap baru (Kelompok): miaw
Masukkan umur (21): 21
Masukkan nomor telepon baru (08772277272):
Masukkan alamat email baru (kelompok2@gmail.com):
Profil akun berhasil diubah!
Tekan Enter untuk kembali...
```

2. Pesan ojek

```
Pesanan bernas11:  
Jarak: 3.0 km  
Harga: Rp21,000  
Driver: driver-01  
Pesan Ojek Online selesai... selamat menikmati perjalananmu!
```

4.4 Menu Customer pilihan 2

3. Lihat Histori Pemesanan

```
Pilih menu: 3  
  
Histori Pemesanan untuk kelompok2:  
+-----+-----+-----+-----+  
|          Waktu          | Driver | Jarak (km) | Harga (Rp) |  
+-----+-----+-----+-----+  
| Tanggal: 27-11-2025 Jam: 08:54:07 | driver-01 | 3.0      | 21000      |  
+-----+-----+-----+-----+  
Tekan Enter untuk kembali...
```

4.5 Menu Customer pilihan 3

4. Logout

```
Pilih menu: 4  
Anda telah logout, silahkan login kembali  
Tekan Enter untuk kembali...
```

4.6 Menu Customer Pilihan 4

D. Menu Driver

```
=====
                        Selamat Datang
=====
MENU DRIVER
-----
> 1. Edit Profil
> 2. Lihat Pendapatan
```


4.7 Menu Driver

1. Edit Profil

```
Pilih menu: 1
===Edit Profil===
===Kosongkan jika tidak ingin mengubah!===
Masukkan username baru (driver-03):
Masukkan password baru:
Masukkan nama lengkap baru (Heiza R): www
Masukkan umur (21): 19
Masukkan nomor telepon baru (02929299292):
Masukkan alamat email baru (mwah@gmail.com):
Profil akun berhasil diubah!
Tekan Enter untuk kembali...
```

4.8 Menu Driver pilihan 1

2. Lihat Pendapatan

```
Pilih menu: 2
Belum ada pendapatan untuk driver-03.
Tekan Enter untuk kembali...
```

4.9 Menu Driver pilihan 2

4. LANGKAH-LANGKAH GIT

```
nugrah@nugrah-debian:~/PA$ git add .
nugrah@nugrah-debian:~/PA$ git commit -m "Sudah bisa login sebagai user Administrator (admin)"
[main 637c580] Sudah bisa login sebagai user Administrator (admin)
 7 files changed, 100 insertions(+), 62 deletions(-)
 create mode 100644 Go-Simpy/__pycache__/akun.cpython-311.pyc
 create mode 100644 Go-Simpy/__pycache__/autentikasi.cpython-311.pyc
 create mode 100644 Go-Simpy/__pycache__/tampilan.cpython-311.pyc
nugrah@nugrah-debian:~/PA$ git push -u origin main
Enter passphrase for key '/home/nugrah/github-keys/nugrahreswara':
Enumerating objects: 18, done.
Counting objects: 100% (18/18), done.
Delta compression using up to 4 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (11/11), 4.85 KiB | 1.62 MiB/s, done.
Total 11 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:nugrahreswara/Go-Simpy.git
 6e44c93..637c580  main -> main
branch 'main' set up to track 'origin/main'.
nugrah@nugrah-debian:~/PA$ |
```

4.1 Screenshot Github

1. inisialisasi Git di lokal

Masuk ke folder proyek dan jalankan:

```
git init
```

- Perintah ini membuat folder `.git` tersembunyi di proyek, yang berfungsi menyimpan semua riwayat versi file.
- Setelah ini, folder proyek resmi menjadi repositori Git lokal.

2. Konfigurasi identitas Git

Git perlu tahu siapa yang membuat perubahan. Atur identitas dengan:

```
git config --global user.name "Nama Anda"
```

```
git config --global user.email "email@domain.com"
```

- `--global` artinya untuk semua repositori di komputer.

- Bisa juga diatur khusus proyek tanpa --global:

```
git config user.name "Nama Proyek"
```

3. Menambahkan file ke staging area

Git memiliki 3 area:

1. Working directory → tempat kita edit file.
2. Staging area → file siap di-commit.
3. Repository → snapshot permanen.

Langkah:

```
git status      # lihat file yang belum dilacak
```

```
git add nama_file # menambahkan file tertentu
```

```
git add .      # menambahkan semua file
```

git add menandai file yang siap disimpan di commit berikutnya.

4. Commit perubahan

Setelah file di staging, simpan snapshot:

```
git commit -m "Pesan commit menjelaskan perubahan"
```

- commit = menyimpan versi proyek.
- -m = pesan singkat yang menjelaskan perubahan.

Contoh pesan bagus: "Menambahkan halaman login" atau "Memperbaiki bug hitung total"

5. Buat repositori di GitHub

1. Login ke GitHub → klik New Repository.
2. Beri nama repositori, misal proyekku.
3. Jangan centang "Initialize with README" jika folder lokal sudah ada.

4. Klik Create Repository.

Ini membuat tempat penyimpanan cloud.

6. Sambungkan Git lokal ke GitHub

Hubungkan repositori lokal dengan GitHub:

```
git remote add origin https://github.com/username/proyekku.git
```

```
git remote -v # cek remote
```

- origin = nama default untuk remote.
- Sekarang Git tahu ke mana harus mengirim (push) file.

7. Push ke GitHub

Kirim commit pertama ke GitHub:

```
git branch -M main # pastikan branch utama bernama 'main'
```

```
git push -u origin main
```

- -u → menetapkan remote default untuk push/pull berikutnya.
- Setelah ini, file di lokal muncul di GitHub.

8. Update / perubahan berikutnya

1. Edit file di lokal.
2. Tambahkan ke staging:

```
git add .
```

3. Commit:

```
git commit -m "Perubahan terbaru"
```

4. Push:

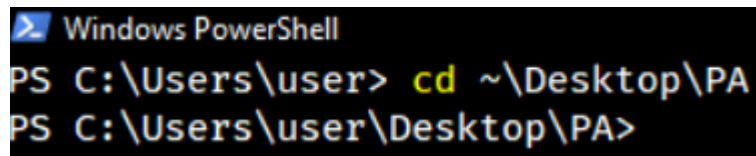
```
git push
```

Setiap perubahan akan tercatat di GitHub, sehingga versi proyek selalu up-to-date.

GIT CLONE

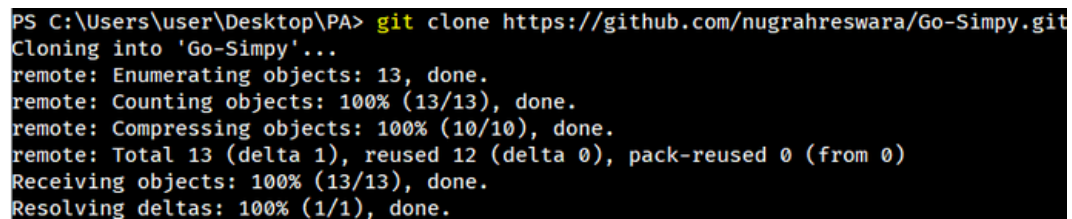
1. Pertama kita buka Windows Powershell lalu untuk pergi ke direktori PA:

Copy Paste: `cd ~\Desktop\PA`



```
Windows PowerShell
PS C:\Users\user> cd ~\Desktop\PA
PS C:\Users\user\Desktop\PA>
```

2. Git clone dulu untuk ngambil semua resource yang ada di girthub repository (CUMAN DILAKUIN SEKALI)

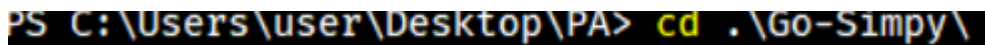


```
PS C:\Users\user\Desktop\PA> git clone https://github.com/nugrahreswara/Go-Simp.py.git
Cloning into 'Go-Simp.py'...
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 13 (delta 1), reused 12 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (13/13), done.
Resolving deltas: 100% (1/1), done.
```

Copy paste: `git clone https://github.com/nugrahreswara/Go-Simp.py.git`

3. Masuk kedalam Folder / Direktori Go-Simp.py,

Copy Paste: `cd Go-Simp.py`



```
PS C:\Users\user\Desktop\PA> cd .\Go-Simp.py\
```

4. Mengatur username dan email untuk Github (CUMAN DILAKUIN SEKALI)

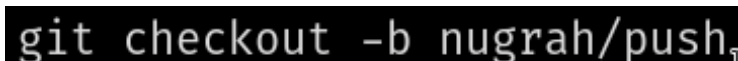
Copy Paste: `git config --global user.name <username github>`

5. Nah sebelum kamu ngubah apapun, git pull dulu supaya ngambil data terbaru dari repositori githubnya,

Copy Paste: `git pull origin main`

6. Setelah itu, buat branch baru (bisa juga dipake buat ganti branch):

Copy Paste: `git checkout -b <nama>/push`



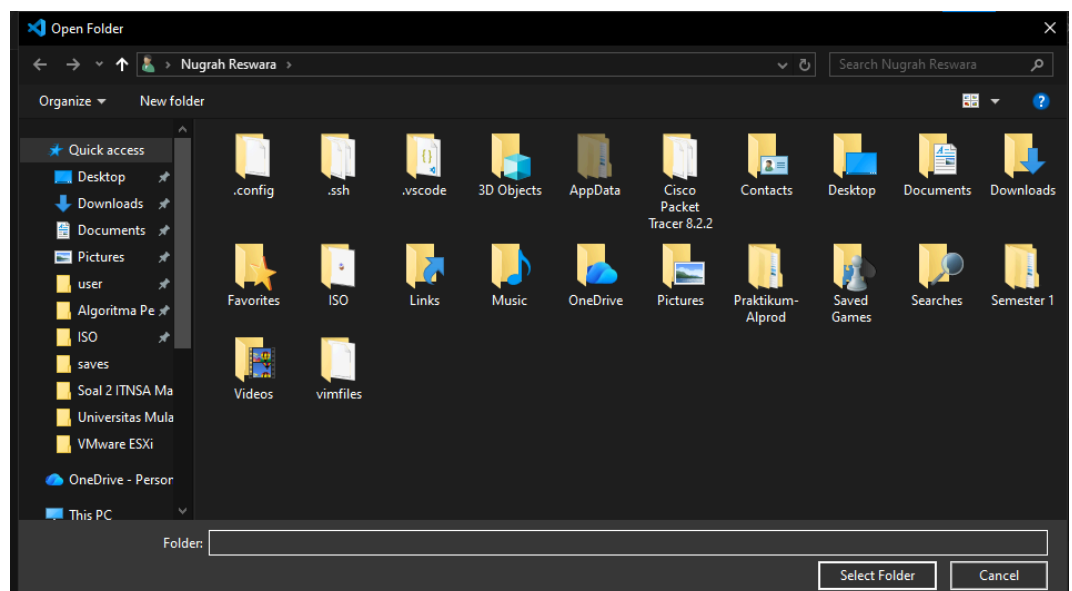
```
git checkout -b nugrah/push
```

7. Kita bisa cek lagi di branch mana (* nunjukin kita di branch mana):

Copy Paste: git branch

```
nugrah@nugrah-debian:~/PA$ git branch
main
* nugrah/push
```

8. Nah untuk mempermudah mending pakai Microsoft Visual Studio Code (VS Code), buka VS Codenya lalu gunakan kombinasi: Ctrl + K lalu Ctrl + O



DESKRIPSI PEMBAGIAN TUGAS

1. Anugrah Fakhri Reswara: debugging, sistem_pemesanan.py
2. Luvita Khairana Salwa: Mengerjakan [tampilan.py](#), [akun.py](#), validasi input.py
3. Anah Wiyana: Mengerjakan bagian manajemen [akun.py](#) bagian edit profil, hapus akun, buat akun, autentikasi, main.py