

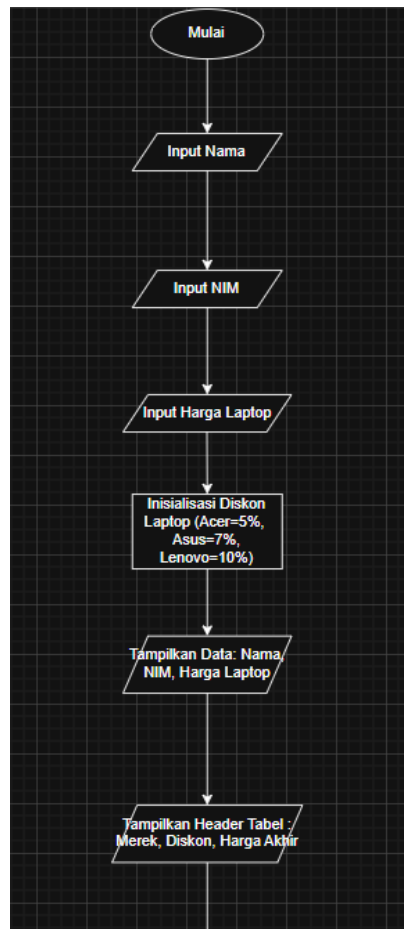
LAPORAN PRAKTIKUM
POSTTEST 2
ALGORITMA PEMROGRAMAN DASAR



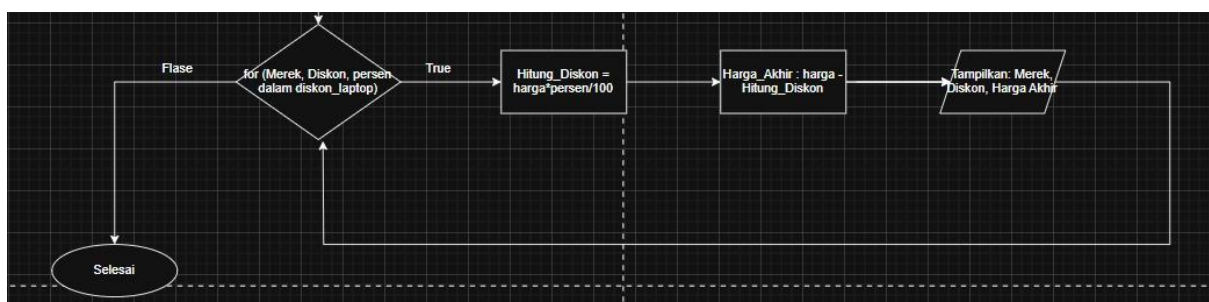
Disusun oleh:
KIRANA CINTA MENTARI (2509106106)
Kelas (C1 '25)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



Gambar 1.1 Flowchart



Gambar 1.2 Flowchart

Penjelasan Flowchart :

1. Mulai
2. input Nama
3. Input NIM
4. Input Harga Laptop
5. Inisialisasi Diskon Laptop (Acer=5%, Asus =7%, Lenovo=10%)
6. Tampilkan Data: Nama, NIM, Harga Laptop
7. Tampilkan Header Tabel: Merek, Diskon, Harga Akhir
8. For (Merek, Diskon, Persen Dalam Diskon _Laptop
9. Hitung_Diskon = Hitung Persen/100
10. Harga_Akhir : Harga -Hitung _Diskon
11. Tampilkan: Merek, Diskon, Harga Akhir
12. Kembali ke nomor delapan, sampai merek akhir
13. Selesai

2. Deskripsi Singkat Program

Program ini dibuat untuk menghitung harga akhir laptop setelah diberikan potongan harga berdasarkan merek yang dipilih. Pengguna diminta untuk memasukkan nama lengkap, NIM, serta harga laptop. Selanjutnya, program menampilkan daftar merek laptop beserta besar diskon yang berlaku, yaitu Acer dengan diskon 5%, Asus dengan diskon 7%, dan Lenovo dengan diskon 10%. Tujuan utama dari program ini adalah membantu pengguna mengetahui perbandingan harga akhir laptop setelah diskon sesuai dengan mereknya. Dengan demikian, program ini bermanfaat sebagai alat sederhana untuk simulasi perhitungan harga laptop dengan berbagai pilihan diskon.

3. Source Code

```
for merk, persen in diskon_dict.items():
    diskon = harga * (persen / 100)
    harga_akhir = harga - diskon
    print("{:<10} {:<15} Rp{:, .0f}".format(merk, f"{persen}%",
    harga_akhir))
```

4. Hasil Output

```
Masukkan Nama Lengkap: kirana cinta mentari
Masukkan NIM: 2509106106
Masukkan harga laptop: 12000000

=====
kirana cinta mentari dengan NIM 2509106106 ingin membeli laptop seharga Rp12,000,000
=====
Merk      Diskon      Harga Akhir
=====
Acer      5%          Rp11,400,000
Asus      7%          Rp11,160,000
Lenovo    10%         Rp10,800,000
=====
PS D:\Blabla> 
```

Gambar 4.1 Output

5. Langkah-langkah GIT

5.1 GIT Init

```
PS D:\Blabla> & C:/Users/sarid/AppData/Local/Programs/Python/Python313/python.exe "d:/Blabla/2509106106-KiranaCintaMen
tari -PT-2.py"
```

Gambar 5.5.1 Langkah Git

Perintah git init digunakan untuk membuat sebuah repository Git baru di dalam folder yang dipilih. Dengan kata lain, perintah ini mengubah sebuah folder biasa menjadi folder yang dapat dikelola oleh Git. Setelah menjalankan git init, Git akan membuat sebuah sub-folder tersembunyi bernama .git. Folder inilah yang menyimpan semua informasi penting terkait version control, seperti riwayat perubahan, konfigurasi, dan metadata repository.

Contohnya, ketika kita sedang membuat sebuah proyek baru, langkah pertama yang biasanya dilakukan adalah masuk ke folder proyek tersebut melalui terminal, lalu menjalankan perintah git init.

5.2 GIT Add

```
PS C:\partikum-apd\pratikum-apd> git push -u origin main
```

Gambar 5.5.2 Langkah Git

Perintah git add . digunakan untuk menambahkan semua perubahan file yang ada di dalam folder proyek ke dalam staging area Git. Staging area adalah tempat sementara di mana perubahan file disiapkan sebelum benar-benar disimpan ke dalam riwayat repository melalui perintah git commit.

Tanda titik (.) setelah git add berarti semua file yang ada di direktori saat ini, termasuk subfolder di dalamnya, akan ikut ditambahkan. Jadi, jika ada file baru, file

yang sudah diubah, atau file yang dihapus, semuanya akan masuk ke dalam staging area sekaligus.

Tanda titik (.) setelah git add berarti semua file yang ada di direktori saat ini, termasuk subfolder di dalamnya, akan ikut ditambahkan. Jadi, jika ada file baru, file yang sudah diubah, atau file yang dihapus, semuanya akan masuk ke dalam staging area sekaligus.

Contohnya, ketika kita selesai mengedit beberapa file sekaligus, daripada menambahkan file satu per satu dengan git add namafile, kita bisa langsung mengetik git add . agar semua perubahan tercatat dalam satu langkah. Setelah itu, barulah perubahan tersebut bisa disimpan secara permanen menggunakan git commit.

5.3 GIT Commit

```
PS C:\Users\user\Documents\pratikum-apd> git config --global user.email "kiranacintamentari818@gmail.com"
PS C:\Users\user\Documents\pratikum-apd> git commit -m "tugas"
[master (root-commit) b520f85] tugas
 2 files changed, 23 insertions(+)
 create mode 100644 kelas/cinta.py
 create mode 100644 post-test/2509106106-KiranaCintaMentari -PT-2.py
```

Gambar 5.5.3 Langkah Git

Commit dalam Git dapat diibaratkan seperti menyimpan catatan atau rekaman atas perubahan yang telah dilakukan pada proyek. Setiap kali kita melakukan commit, Git akan menyimpan kondisi file yang sudah dimasukkan ke staging area sebagai satu titik riwayat baru. Titik riwayat ini nantinya bisa dilihat, dilacak, bahkan dikembalikan lagi jika dibutuhkan.

Sebuah commit biasanya disertai dengan pesan (commit message) yang menjelaskan apa perubahan yang dilakukan. Misalnya, ketika menambahkan fitur baru, memperbaiki bug, atau mengubah bagian tertentu dari kode. Pesan commit ini sangat penting agar orang lain — atau bahkan diri kita sendiri di kemudian hari — dapat memahami tujuan dari perubahan yang dibuat.

Setiap commit memiliki identitas unik berupa hash (serangkaian angka dan huruf) yang membuat Git bisa membedakan satu commit dengan commit lainnya. Dengan adanya commit, kita bisa menelusuri riwayat proyek dari awal hingga kondisi terbaru, serta berkolaborasi dengan lebih teratur.

5.4 GIT Remote

```
PS C:\Users\user\Documents\pratikum-apd> git remote add origin https://github.com/kiranacintamentari818-glitch/pratikum-apd-2.git
```

Gambar 5.5.4 Langkah Git

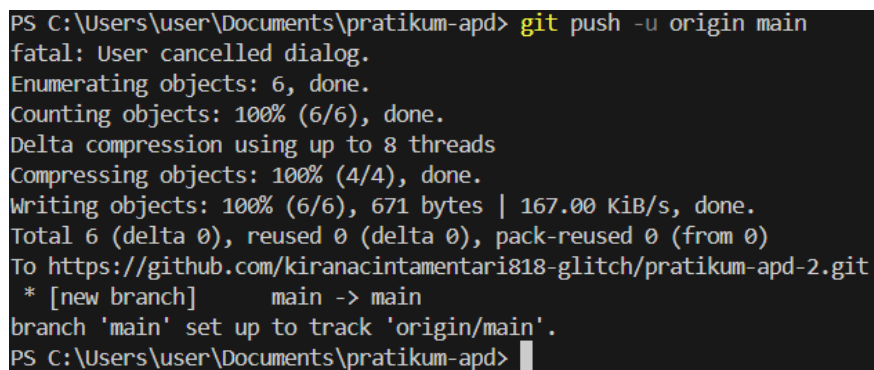
Git remote adalah perintah dalam Git yang digunakan untuk menghubungkan repository lokal dengan repository yang berada di server atau layanan hosting kode, seperti GitHub, GitLab, atau Bitbucket. Dengan adanya remote, kita bisa mengirim (push) perubahan dari komputer lokal ke server, atau mengambil (pull/fetch) perubahan dari server ke komputer lokal.

Di sini, origin adalah nama default yang diberikan untuk repository remote. Setelah remote ditambahkan, kita bisa menjalankan perintah git push untuk mengunggah commit ke repository online, atau git pull untuk mengambil pembaruan dari repository tersebut.

Selain itu, git remote -v dapat digunakan untuk melihat daftar remote yang sudah terhubung beserta URL-nya.

Dengan kata lain, git remote membuat proyek kita tidak hanya tersimpan di komputer lokal, tetapi juga dapat dibagikan, disinkronkan, dan dikerjakan bersama-sama melalui repository yang ada di server.

5.5 GIT Push



```
PS C:\Users\user\Documents\pratikum-apd> git push -u origin main
fatal: User cancelled dialog.
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 671 bytes | 167.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/kiranacintamentari818-glitch/pratikum-apd-2.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\user\Documents\pratikum-apd> █
```

Gambar 5.5.5 Langkah Git

git push adalah perintah yang digunakan untuk mengirimkan atau mengunggah commit dari repository lokal ke repository remote, misalnya ke GitHub, GitLab, atau Bitbucket. Dengan melakukan push, semua perubahan yang sudah kita simpan melalui commit di komputer akan tersalin ke repository yang ada di server sehingga bisa diakses oleh orang lain atau digunakan pada perangkat lain.

Pada contoh di atas, origin adalah nama remote (default ketika kita menambahkan repository GitHub), dan main adalah nama branch yang akan dikirim. Artinya, semua commit yang ada di branch main pada komputer kita akan diunggah ke branch main di repository remote.

Tambahan -u (atau --set-upstream) berfungsi agar pada push berikutnya kita cukup mengetik git push tanpa harus menuliskan nama remote dan branch lagi, karena Git sudah mengingat pengaturan tersebut.

Dengan kata lain, git push adalah cara kita membagikan hasil kerja dari repository lokal ke repository online, sehingga proyek bisa ter-update dan mudah dikelola bersama tim.