

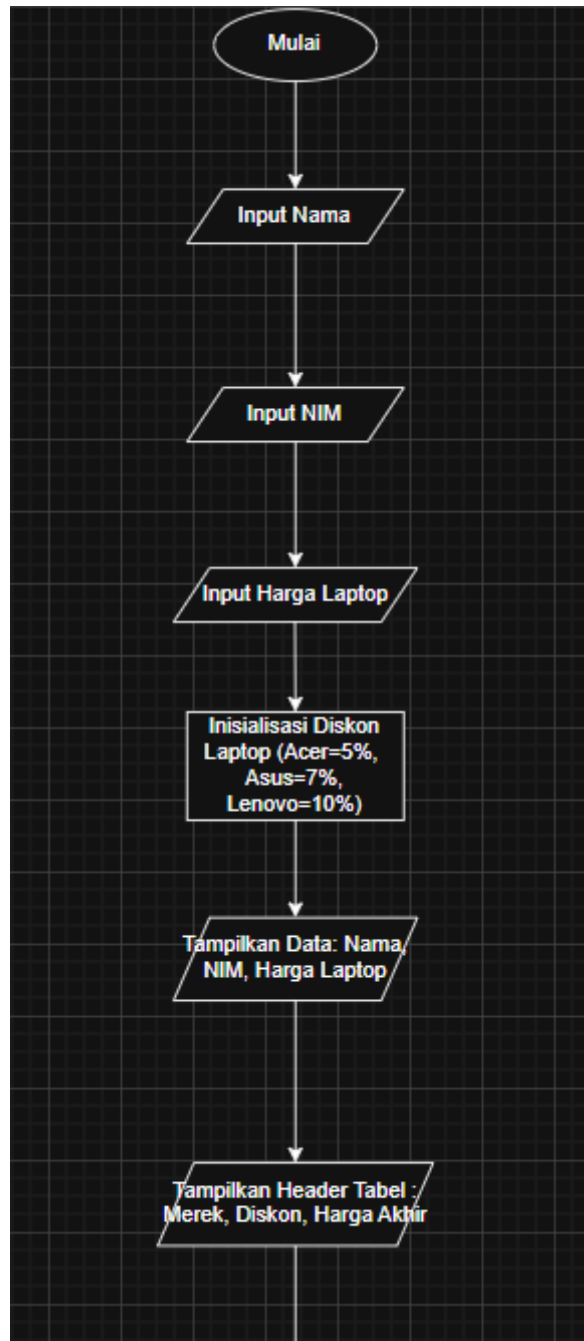
**LAPORAN PRAKTIKUM
POSTTEST (2)
ALGORITMA PEMROGRAMAN DASAR**



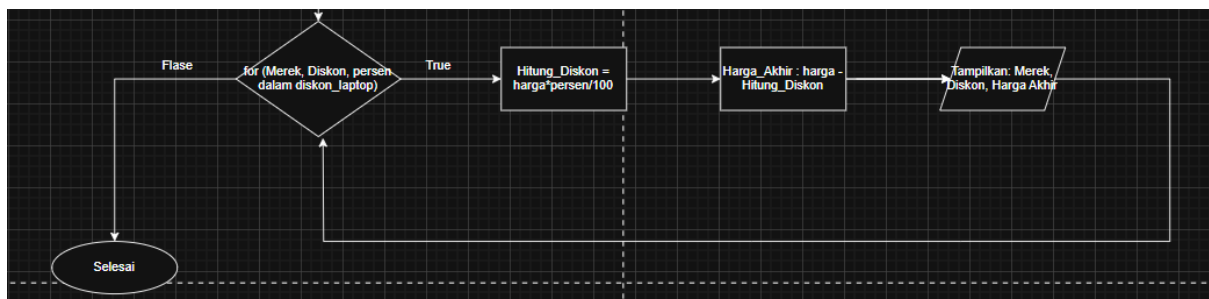
**Disusun oleh:
AWALIYAH RAHMAN (2509106107)
Kelas (C1-25)**

**PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025**

1. Flowchart



Gambar 1.1 Flowchart



Gambar 1.2 Flowchart

1. Mulai
2. Input Nama
3. Input NIM
4. Input Harga Laptop
5. Inisialisas 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_laptop
9. $\text{Harga_Diskon} = \text{harga} * \text{persen}/100$
10. $\text{Harga_Akhir} = \text{harga} - \text{Harga_Diskon}$
11. Tampilkan Harga Akhir
12. Kembali ke nomor 8, sampai merek terakhir
13. Selesai

2. Deskripsi Singkat Program

Program ini dibuat untuk membantu menghitung harga akhir laptop setelah mendapat potongan diskon berdasarkan merek yang dipilih. Pengguna cukup memasukkan nama, NIM, dan harga laptop. Program kemudian menampilkan daftar merek laptop beserta persentase diskon yang tersedia serta harga akhir setelah potongan. Dengan demikian, program bermanfaat untuk memberikan informasi perbandingan harga laptop dari berbagai merek secara cepat dan mudah dipahami.

3. Source Code

```
for merek, persen in diskon_laptop.items() :  
    diskon = harga * (persen / 100)  
    harga_akhir = harga - diskon  
    print(f"{merek:<10} {persen:<10}% {harga_akhir:>20,.0f}")
```

4. Hasil Output

```
PS C:\partikum-apd\pratikum-apd\post-test\post-test-apd-1> & C:/Users/HP/AppData/Local/Programs/Python/Python313/python.exe c:/partikum-apd/pratikum-apd/post-test-apd-1/2509106107-AWALIYH_RAHMANT-PT-1.py
Masukkan nama lengkap :Awaliyah Rahman
Masukkan NIM: 2509106107
Masukkan Harga Laptop (Rp): 15000

=====
Awaliyah Rahman dengan NIM 2509106107 ingin membeli laptop seharga Rp 15,000
=====
Merek      Diskon      Harga Akhir (Rp)
-----
Acer       5           %           14,250
Asus       7           %           13,950
Lenovo     10          %           13,500
PS C:\partikum-apd\pratikum-apd\post-test\post-test-apd-1> 
```

Gambar 4.1 Output

5. Langkah-langkah GIT

5.1 GIT Init

```
PS C:\partikum-apd\pratikum-apd> git init
Initialized empty Git repository in C:/partikum-apd/pratikum-apd/.git/
```

Gambar 5.1.1 Git Init

Git init adalah perintah untuk membuat repositori Git baru pada sebuah folder atau proyek. Saat perintah ini dijalankan, Git akan menambahkan folder tersembunyi bernama .git di dalam direktori tersebut. Folder inilah yang digunakan Git untuk menyimpan seluruh riwayat perubahan, konfigurasi, serta informasi versi dari proyek.

Dengan kata lain, git init berfungsi mengubah sebuah folder biasa menjadi folder yang bisa dikelola oleh Git, sehingga setiap perubahan file di dalamnya dapat dilacak, disimpan, dan dikelola versinya.

5.2 GIT Add

```
PS C:\partikum-apd\pratikum-apd> git add .
```

Gambar 5.2.1 Git Add

Git add . adalah perintah untuk menambahkan semua perubahan file di dalam folder proyek ke dalam staging area Git.

Staging area ini ibarat tempat sementara sebelum perubahan benar-benar disimpan ke dalam riwayat proyek menggunakan git commit.

Titik (.) setelah kata add berarti semua file di direktori saat ini, jadi bukan hanya satu file tertentu. Dengan begitu, setiap perubahan—baik file baru, file yang diubah, maupun file yang dihapus—akan dicatat untuk siap dikirim pada proses commit berikutnya.

5.3 GIT Commit

```
PS C:\partikum-apd\pratikum-apd> git commit -m "Memasukkan Ke Github"
[main (root-commit) 62ed65f] Memasukkan Ke Github
2 files changed, 22 insertions(+)
create mode 100644 kelas/pertemuan-1.py
create mode 100644 post-test/post-test-apd-1/2509106107-AWALIYH_RAHMAN-PT-1.py
```

Gambar 5.3.1 Git Commit

Git commit adalah perintah untuk menyimpan perubahan yang sudah dicatat (staging) ke dalam riwayat Git. Setiap commit dapat dianggap sebagai "titik simpan" atau "versi" dari proyek pada saat tertentu. Di dalam commit, Git menyimpan informasi perubahan file beserta pesan yang ditulis oleh pengguna untuk menjelaskan maksud dari perubahan tersebut.

Dengan menggunakan git commit, kita bisa menelusuri kembali riwayat pengembangan proyek, melihat siapa yang melakukan perubahan, kapan dilakukan, dan apa saja yang diubah. Hal ini memudahkan kolaborasi sekaligus menjaga agar setiap tahap perkembangan proyek terdokumentasi dengan baik.

5.4 GIT Remote

```
PS C:\partikum-apd\pratikum-apd> git remote add origin https://github.com/awalyahbima-dotcom/pratikum-apd.git
```

Gambar 5.4.1 Git Remote

Git remote adalah perintah yang digunakan untuk menghubungkan repositori lokal dengan repositori yang berada di server atau layanan penyimpanan Git, seperti GitHub, GitLab, atau Bitbucket. Dengan adanya remote, kita bisa mengirimkan (push) atau mengambil (pull/fetch) perubahan kode dari repositori jarak jauh.

Sederhananya, git remote berfungsi sebagai "jembatan" antara proyek yang ada di komputer lokal dengan repositori yang ada di internet. Tanpa remote, proyek hanya tersimpan di komputer sendiri dan tidak bisa dibagikan atau dikerjakan bersama orang lain.

5.5 GIT Push

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

Gambar 5.5.1 Git Push

Git push adalah perintah untuk mengirimkan perubahan dari repositori lokal ke repositori jarak jauh (remote), misalnya di GitHub atau GitLab. Setelah melakukan commit di lokal, perubahan tersebut belum otomatis tersimpan di remote. Dengan menjalankan git push, barulah commit-commit yang sudah dibuat dikirim dan disimpan di repositori online.

Sederhananya, git push berfungsi untuk membagikan hasil kerja dari komputer kita ke repositori yang bisa diakses orang lain, sehingga proyek dapat dikelola dan dikembangkan secara bersama-sama.