LAPORAN PRAKTIKUM

POSTTEST 2

ALGORITMA PEMROGRAMAN DASAR

Disusun oleh:

Muhammad Fajar (2509106117)

Kelas (C2 ‘25)

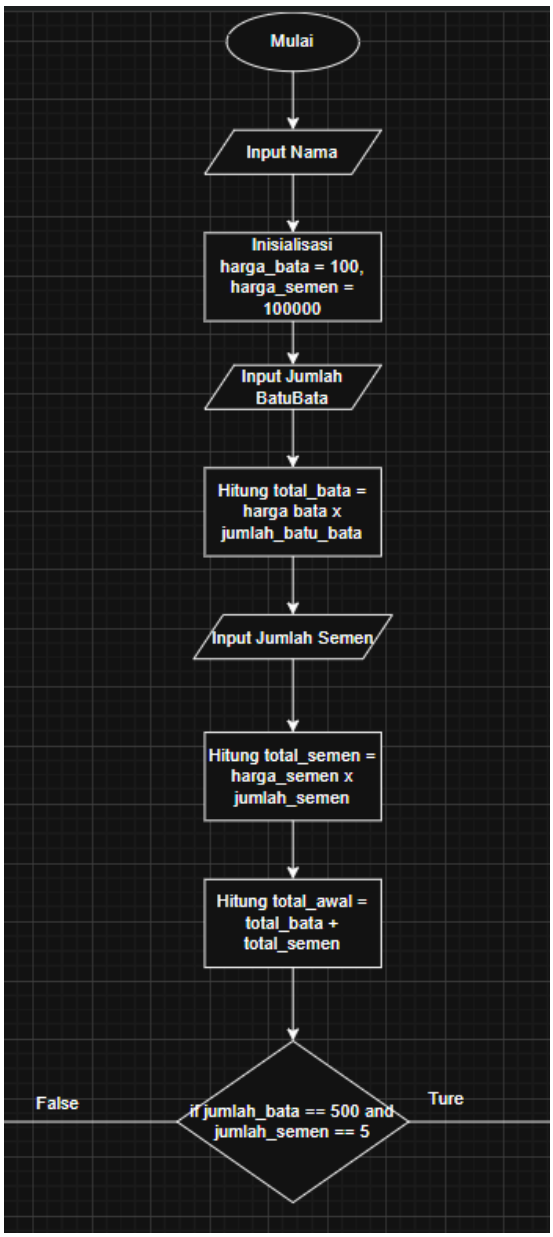
PROGRAM STUDI INFORMATIKA

UNIVERSITAS MULAWARMAN

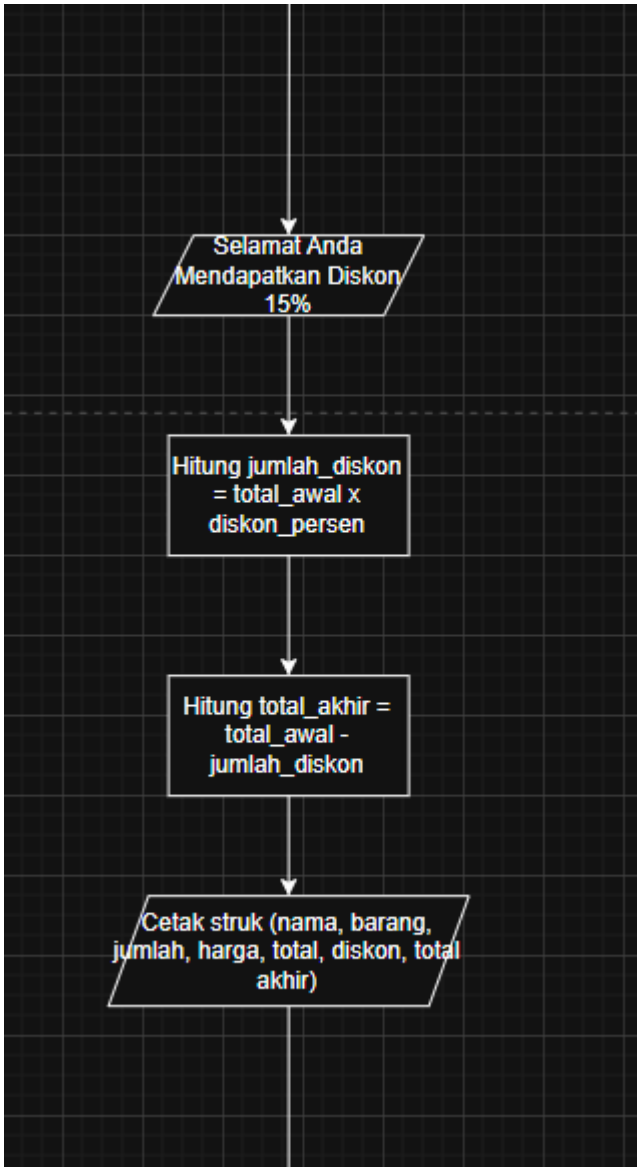
SAMARINDA

2025

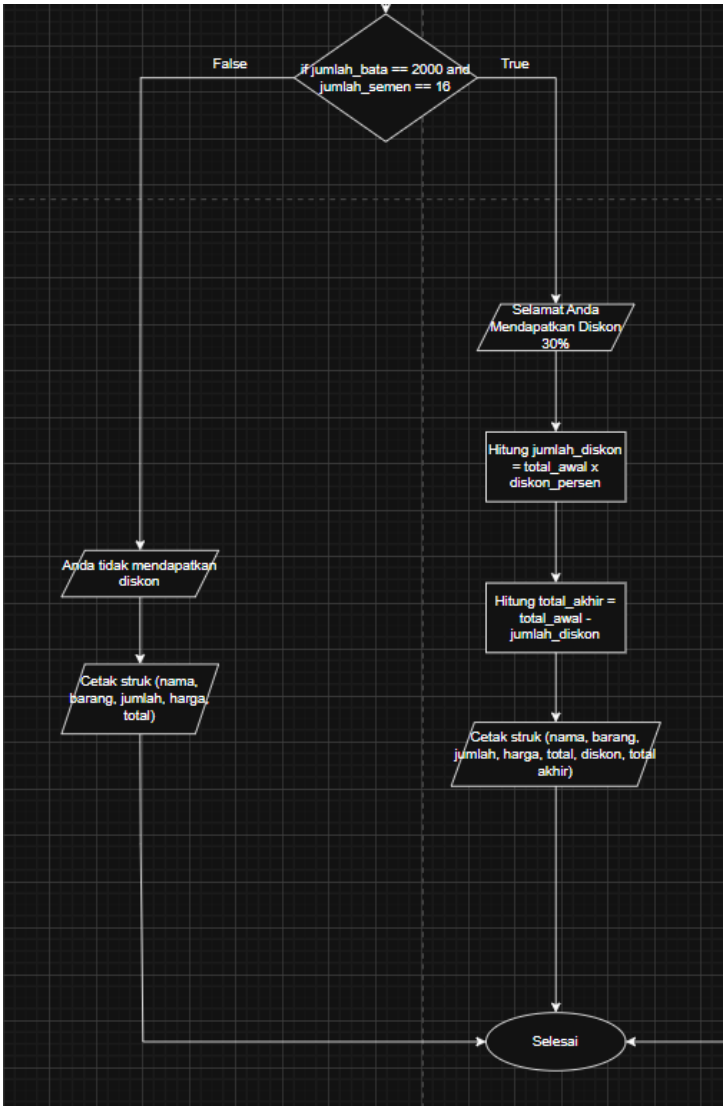
1. Flowchart



Gambar 1.1 Flowchart



Gambar 1.2 Flowchart



Gambar 1.3 Flowchart

Penjelasan Flowchart :

1. Mulai
2. Masukkan input Nama
3. Inisialisasi harga_bata = 100, harga_semen = 100000
4. Input Jumlah_BatuBata
5. Hitung total_bata = harga bata x jumlah_batu_bata
6. Input Jumlah Semen
7. Hitung total_semen = harga_semen x jumlah_semen
8. Hitung total_awal = total_bata + total_semen
9. Jika jumlah_bata sama dengan 500 dan jumlah_semen sama dengan 5, maka dapat diskon 15%
10. Hitung jumlah_diskon = total_awal x diskon_persen
11. Hitung total_akhir = total_awal - jumlah_diskon
12. Cetak struk (nama, barang, jumlah, harga, total, diskon, total akhir)
13. Selesai
14. Mulai
15. Masukkan input Nama
16. Inisialisasi harga_bata = 100, harga_semen = 100000
17. Input Jumlah_BatuBata
18. Hitung total_bata = harga bata x jumlah_batu_bata
19. Input Jumlah Semen
20. Hitung total_semen = harga_semen x jumlah_semen
21. Hitung total_awal = total_bata + total_semen
22. Jika jumlah_bata sama dengan 2000 dan jumlah_semen sama dengan 16, maka dapat diskon 30%
23. Hitung jumlah_diskon = total_awal x diskon_persen
24. Hitung total_akhir = total_awal - jumlah_diskon
25. Cetak struk (nama, barang, jumlah, harga, total, diskon, total akhir)
26. Selesai
27. Mulai
28. Masukkan input Nama
29. Inisialisasi harga_bata = 100, harga_semen = 100000
30. Input Jumlah_BatuBata
31. Hitung total_bata = harga bata x jumlah_batu_bata
32. Input Jumlah Semen
33. Hitung total_semen = harga_semen x jumlah_semen
34. Hitung total_awal = total_bata + total_semen
35. Cetak struk (nama, barang, jumlah, harga, total)
36. Selesai

2. Deskripsi Singkat Program

Program ini dibuat untuk menghitung estimasi biaya pembelian bahan bangunan berupa batu bata dan semen. Program akan meminta pengguna untuk memasukkan nama pelanggan, jumlah batu bata, serta jumlah karung semen yang ingin dibeli. Setiap barang memiliki harga satuan yang telah ditentukan, yaitu Rp100 untuk satu buah batu bata dan Rp100.000 untuk satu karung semen.

Setelah data dimasukkan, program akan menghitung total harga masing-masing barang, kemudian menjumlahkannya sebagai total biaya awal. Jika jumlah pembelian memenuhi syarat paket tertentu, yaitu 500 batu bata dan 5 karung semen atau 2000 batu bata dan 16 karung semen, maka pelanggan akan mendapatkan potongan harga (diskon) sebesar 15% atau 30%. Apabila tidak memenuhi syarat, maka pelanggan tidak memperoleh diskon.

Program kemudian menampilkan rincian pembelian dalam bentuk tabel yang berisi nama pelanggan, jumlah dan harga barang, total biaya awal, besaran diskon yang diperoleh, serta total biaya akhir setelah diskon. Di akhir proses, pengguna diberikan pilihan apakah ingin melanjutkan program untuk melakukan perhitungan baru atau menghentikan program.

3. Source Code

```
    total_awal = total_bata + total_semen
    print(f"Total Biaya Awal Sementara: Rp {total_awal:,}\n")

    if jumlah_batu_bata == 500 and jumlah_semen == 5:
        diskon_persen = 0.15
        keterangan_diskon = "Paket Hemat (15%)"
        print("Selamat Anda Mendapatkan Diskon 15%\n")
    elif jumlah_batu_bata == 2000 and jumlah_semen == 16:
        diskon_persen = 0.30
        keterangan_diskon = "Paket Ultra Mantap (30%)"
    else:
        diskon_persen = 0
        keterangan_diskon = "Tidak Ada Diskon"

    jumlah_diskon = total_awal * diskon_persen
    total_akhir = float(total_awal - jumlah_diskon)
```

4. Hasil Output

```
Masukkan Nama Pelanggan: Muhammad Fajar


Masukkan jumlah batu bata yang akan dibeli: 500
Total harga batu bata: Rp 50,000

Masukkan jumlah karung semen yang akan dibeli: 5
Total harga semen: Rp 500,000

Total Biaya Awal Sementara: Rp 550,000

Selamat Anda Mendapatkan Diskon 15%

========================================
        ESTIMASI BIAYA BAHAN BANGUNAN
========================================
Nama Pelanggan: Muhammad Fajar
-----------------------------------------------------------
| Barang     | Jumlah | Harga Satuan      |
-----------------------------------------------------------
| Batu Bata  | 500    | Rp100             |
| Semen      | 5      | Rp100000          |
-----------------------------------------------------------
Total Biaya Awal              : Rp 550,000
Diskon yang Didapat           : Paket Hemat (15%)
Jumlah Diskon                 : Rp 82,500
-----------------------------------------------------------
TOTAL BIAYA AKHIR : Rp 467,500
========================================

Apakah Anda ingin melanjutkan program? (y/n): n
PS D:\Muhammad Fajar\Kuliah\APD\Pratikum\Posstest-2\post-test\post-test-apd-2>
```

Gambar 4.1 Output

5. Langkah-langkah GIT

5.1 GIT Init

```
PS D:\Muhammad Fajar\Kuliah\APD\Pratikum\Posstest-2> git init
Initialized empty Git repository in D:/Muhammad Fajar/Kuliah/APD/Pratikum/Posstest-2/.git/
```

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. Setelah itu, folder tersebut resmi menjadi sebuah repository Git sehingga setiap perubahan file yang ada di dalamnya bisa dilacak dan dikelola dengan perintah-perintah Git lainnya, seperti git add, git commit, atau git status.

Perlu diperhatikan bahwa git init hanya dijalankan sekali pada saat awal membuat repository di sebuah folder. Jika sudah menjadi repository Git, tidak perlu mengulanginya lagi.

5.2 GIT Add

Gambar 5.5.2 Langkah Git

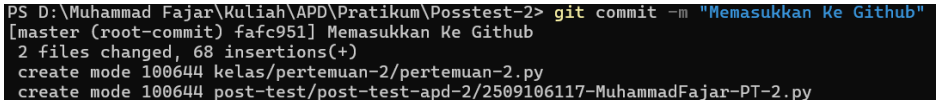
Perintah git add . digunakan untuk menambahkan semua perubahan file yang ada di dalam folder proyek ke dalam staging area Git. Staging area adalah tempat sementara di mana perubahan file disiapkan sebelum benar-benar disimpan ke dalam riwayat repository melalui perintah git commit.

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.

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.

Namun, karena git add . menambahkan seluruh perubahan, kita perlu berhati-hati agar tidak secara tidak sengaja menyertakan file yang sebenarnya tidak ingin dimasukkan ke dalam repository. Untuk menghindari hal ini, biasanya digunakan file .gitignore agar Git mengabaikan file tertentu.

5.3 GIT Commit



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 D:\Muhammad Fajar\Kuliah\APD\Pratikum\Posstest-2> git remote add origin https://github.com/Celrik08/praktikum-apd.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 D:\Muhammad Fajar\Kuliah\APD\Pratikum\Posstest-2> git push -u origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (8/8), 1.26 KiB | 184.00 KiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Celrik08/praktikum-apd.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

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.