LAPORAN PRAKTIKUM

POSTTEST 3

ALGORITMA PEMROGRAMAN DASAR

Disusun oleh:

Cecilia Marsya Pua (2509106125)

INFORMATIKA C2 ‘25

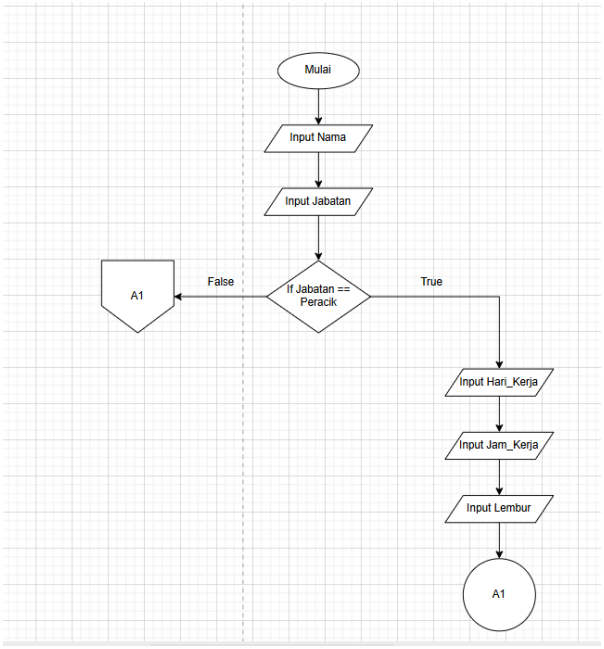
PROGRAM STUDI INFORMATIKA

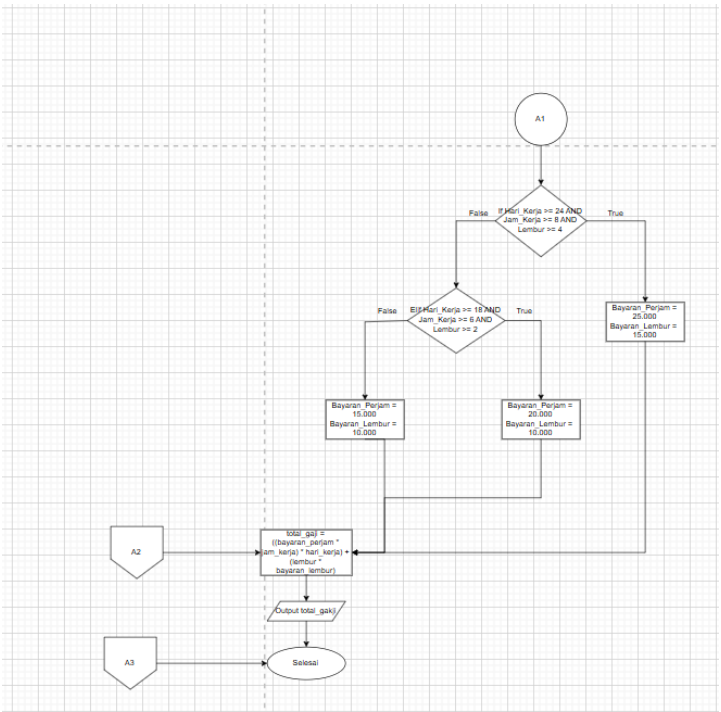
UNIVERSITAS MULAWARMAN

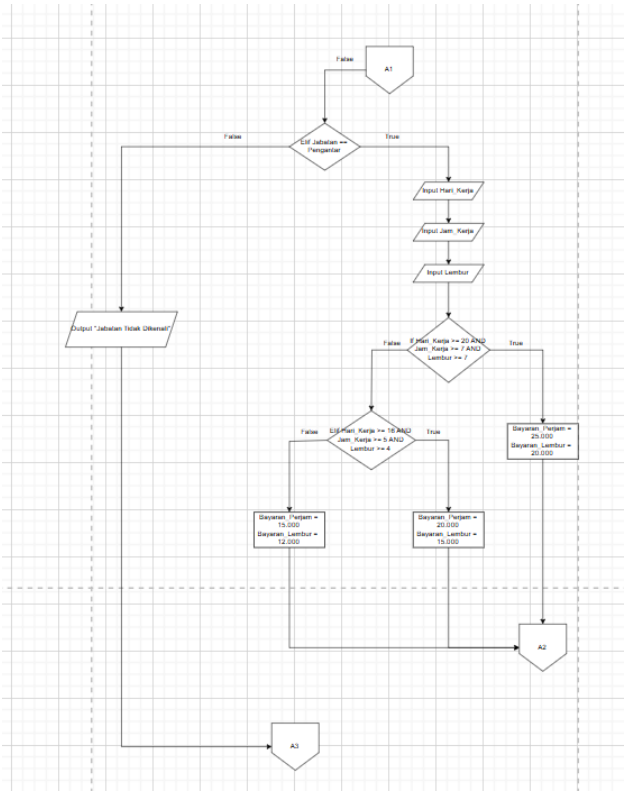
SAMARINDA

2025

1. Flowchart


Mulai
Input Nama
Input Jabatan
If Jabatan == Peracik
False
True
A1
Input Hari_Kerja
Input Jam_Kerja
Input Lembur
A1


A1
If Hari_Kerja >= 24 AND Jam_Kerja >= 8 AND Lembur >= 4
False
True
Else If Hari_Kerja >= 18 AND Jam_Kerja >= 6 AND Lembur >= 2
False
True
Bayaran_Perjam = 25.000
Bayaran_Lembur = 15.000
Bayaran_Perjam = 15.000
Bayaran_Lembur = 10.000
Bayaran_Perjam = 20.000
Bayaran_Lembur = 10.000
A2
total_gaji = ((bayaran_perjam * jam_kerja) * hari_kerja) + (lembur * bayaran_lembur)
Output total_gaji
A3
Selesai



Penjelasan:

1. Mulai
2. Input Nama
3. Input Jabatan
4. Maka program akan memeriksa, jika Jabatan == “Peracik”
5. Input Hari_Kerja
6. Input Jam_Kerja
7. Input Lembur
8. Jika true dan memeriksa program ini true, maka masuk ke eksekusi program ke If Hari_Kerja >= 24 AND Jam_Kerja >= 8 AND Lembur >= 4
9. Maka Bayaran_Perjam = 25.000, Bayaran_Lembur = 15.000
10. Hitung Total_Gaji = ((Bayaran_Perjam * Jam_Kerja) * Hari_Kerja) + (Lembur * Bayaran_Lembur))
11. Output (nama, jabatan, hari kerja, jam kerja, lembur, total gaji)
12. Jika false dan memeriksa program ini, maka masuk ke eksekusi program ke If Hari_Kerja >= 18 AND Jam_Kerja >= 6 AND Lembur >= 2
13. Maka Bayaran_Perjam = 20.000, Bayaran_Lembur = 10.000
14. Hitung Total_Gaji = ((Bayaran_Perjam * Jam_Kerja) * Hari_Kerja) + (Lembur * Bayaran_Lembur))
15. Output (nama, jabatan, hari kerja, jam kerja, lembur, total gaji)
16. Jika false
17. Maka Bayaran_Perjam = 15.000, Bayaran_Lembur = 10.000
18. Hitung Total_Gaji = ((Bayaran_Perjam * Jam_Kerja) * Hari_Kerja) + (Lembur * Bayaran_Lembur))
19. Output (nama, jabatan, hari kerja, jam kerja, lembur, total gaji)

20. Maka program akan memeriksa, jika Jabatan == “Pengantar”
21. Input Hari_Kerja
22. Input Jam_Kerja
23. Input Lembur
24. Jika true dan memeriksa program ini true, maka masuk ke eksekusi program ke If Hari_Kerja >= 20 AND Jam_Kerja >= 7 AND Lembur >= 7
25. Maka Bayaran_Perjam = 25.000, Bayaran_Lembur = 10.000
26. Hitung Total_Gaji = ((Bayaran_Perjam * Jam_Kerja) * Hari_Kerja) + (Lembur * Bayaran_Lembur))
27. Output (nama, jabatan, hari kerja, jam kerja, lembur, total gaji)
28. Jika false dan memeriksa program ini, maka masuk ke eksekusi program ke If Hari_Kerja >= 16 AND Jam_Kerja >= 5 AND Lembur >= 4
29. Maka Bayaran_Perjam = 25.000, Bayaran_Lembur = 20.000
30. Hitung Total_Gaji = ((Bayaran_Perjam * Jam_Kerja) * Hari_Kerja) + (Lembur * Bayaran_Lembur))
31. Output (nama, jabatan, hari kerja, jam kerja, lembur, total gaji)
32. Jika false
33. Maka Bayaran_Perjam = 15.000, Bayaran_Lembur = 12.000
34. Hitung Total_Gaji = ((Bayaran_Perjam * Jam_Kerja) * Hari_Kerja) + (Lembur * Bayaran_Lembur))
35. Output (nama, jabatan, hari kerja, jam kerja, lembur, total gaji)
36. Jika Jabatan tidak sesuai, maka false
37. Output (“Jabatan Tidak Bisa Dikenali”)
38. Selesai

2. Deskripsi Singkat Program

A. Tujuan Program

Tujuan utamanya adalah mempermudah perusahaan maupun karyawan dalam mengetahui rincian gaji secara cepat dan jelas.

B. Fungsi Utama

- Menerima data karyawan berupa nama dan jabatan.
- Menentukan tarif gaji per jam dan tarif lembur sesuai dengan jabatan serta kinerja (hari kerja, jam kerja, dan lembur).
- Menghitung total gaji yang diperoleh karyawan.
- Menampilkan rincian gaji secara lengkap agar mudah dipahami.

3. Source Code

```
if jabatan == "peracik":
    hari_kerja = int(input("Masukkan jumlah hari kerja: "))
    jam_kerja = int(input("Masukkan jumlah jam kerja per hari: "))
    lembur = int(input("Masukkan jumlah lembur: "))
    if hari_kerja >= 24 and jam_kerja >= 8 and lembur >= 4:
        bayaran_per_jam = 25000
        bayaran_lembur = 15000
    elif hari_kerja >= 18 and jam_kerja >= 6 and lembur >= 2:
        bayaran_per_jam = 20000
        bayaran_lembur = 10000
    else:
        bayaran_per_jam = 15000
        bayaran_lembur = 10000

elif jabatan == "pengantar":
    hari_kerja = int(input("Masukkan jumlah hari kerja: "))
    jam_kerja = int(input("Masukkan jumlah jam kerja per hari: "))
    lembur = int(input("Masukkan jumlah lembur: "))
    if hari_kerja >= 20 and jam_kerja >= 7 and lembur >= 7:
        bayaran_per_jam = 25000
        bayaran_lembur = 20000
    elif hari_kerja >= 16 and jam_kerja >= 5 and lembur >= 4:
        bayaran_per_jam = 20000
        bayaran_lembur = 15000
    else:
        bayaran_per_jam = 15000
        bayaran_lembur = 12000

else:
    print("Jabatan tidak dikenali")
    exit()
```

4. Hasil Output

```
PS C:\Users\MyASUS\AppData\Local\Programs\Microsoft VS Code> & C:\Users\M
ktikum-apd/post-test/post-test-apd-3/2509106125-CeciliaMarsyaPua-PT-3.py
=== Penghitung Gaji Karyawan PT. BOM ===
Masukkan nama karyawan: Cecilia Marsya
Masukkan jabatan karyawan (peracik/pengantar): peracik
Masukkan jumlah hari kerja: 24
Masukkan jumlah jam kerja per hari: 8
Masukkan jumlah lembur: 4

<=== Rincian Gaji Karyawan ===>
Nama Karyawan       : Cecilia Marsya
Jabatan             : Peracik
Hari Kerja          : 24
Jam Kerja per Hari  : 8
Jumlah Lembur       : 4
Bayaran per Jam     : Rp25,000
Bayaran Lembur      : Rp15,000
Total Gaji          : Rp4,860,000

 b Terima kasih telah menggunakan sistem PT. BOM
PS C:\Users\MyASUS\AppData\Local\Programs\Microsoft VS Code>
```

5. Langkah-langkah GIT

5.1 GIT Add

```
PS C:\Users\MyASUS\OneDrive\文档\praktikum-apd> git add .
```

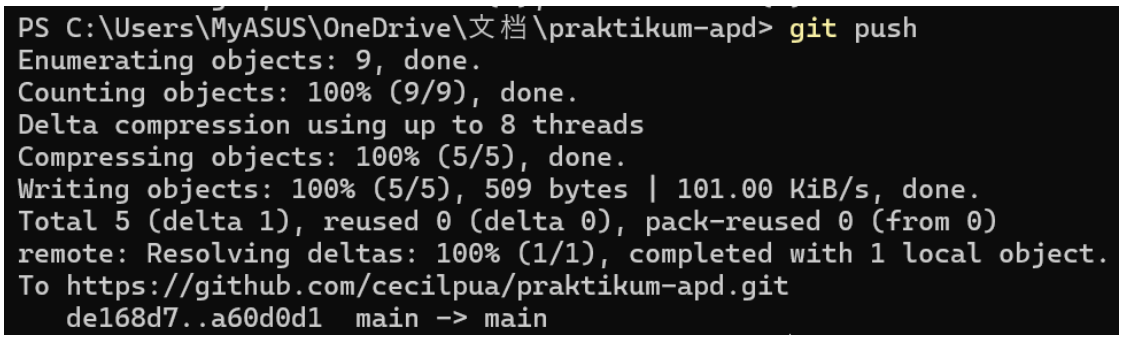
Perintah git add . digunakan untuk menambahkan semua perubahan file yang ada di dalam folder proyek ke dalam staging area Git. Staging area adalah tempat sementara di mana perubahan file disiapkan sebelum benar-benar disimpan ke dalam riwayat repository melalui perintah git commit.

5.3 GIT Commit

```
PS C:\Users\MyASUS\OneDrive\文档\praktikum-apd> git commit -m "Menambahkan File Python"
[main a60d0d1] Menambahkan File Python
 1 file changed, 8 insertions(+), 11 deletions(-)
```

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.

5.5 GIT Push



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.