LAPORAN PRAKTIKUM

POSTTEST 3

ALGORITMA PEMROGRAMAN DASAR

Disusun oleh:

Muhammad Geo Gilang Saputra

(2509106121)

INFORMATIKA C2 ‘25

PROGRAM STUDI INFORMATIKA

UNIVERSITAS MULAWARMAN

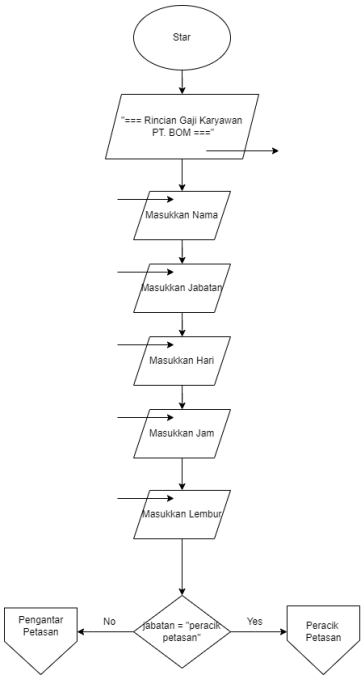
SAMARINDA

2025

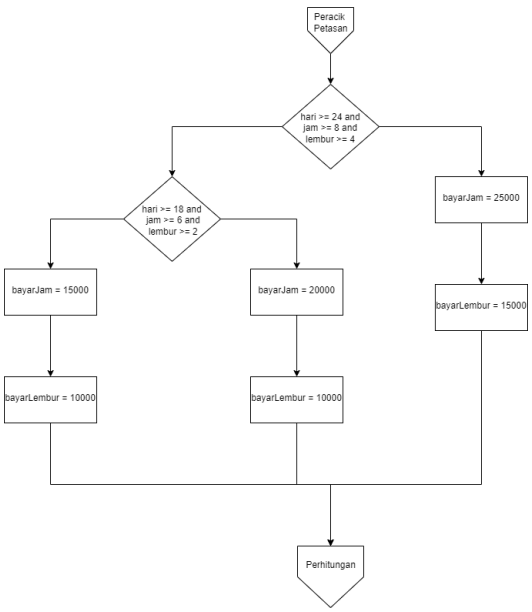
1. Flowchart

Pada flowchart Menu Utama, alur dimulai dengan tampilan judul “Rincian Gaji Karyawan PT. BOM”. Setelah itu pengguna diminta untuk melakukan input data berupa nama karyawan, jabatan, jumlah hari kerja, jam kerja per hari, serta jumlah jam lembur. Data tersebut menjadi dasar untuk proses perhitungan gaji. Setelah semua input dimasukkan, sistem akan memeriksa jabatan karyawan apakah termasuk “peracik petasan” atau “pengantar petasan”, kemudian diarahkan ke cabang flowchart yang sesuai untuk menentukan aturan gaji dan lemburnya.

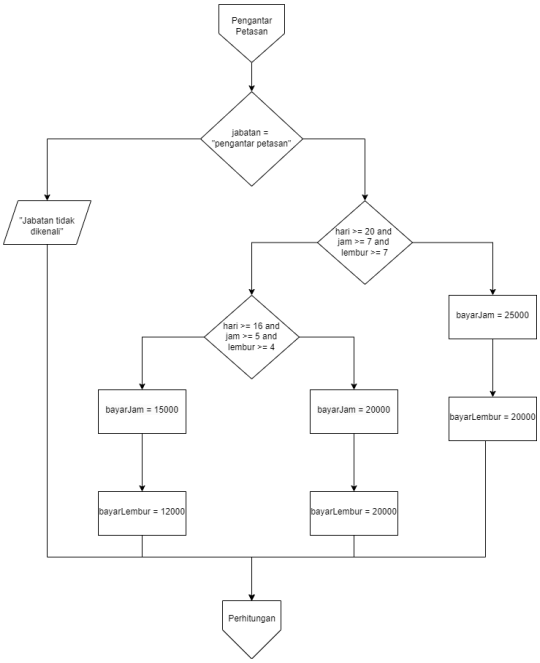
a. Menu Utama



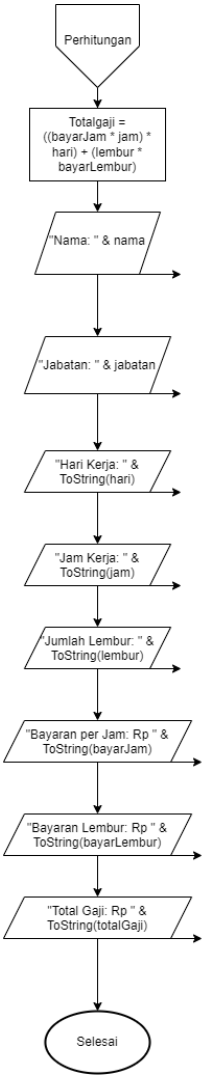
b. Menu Peracik Petasan



a. Menu Pengantar Petasan



b. Menu Perhitungan



2. Deskripsi Singkat Program

Program **Penghitung Gaji Karyawan PT. BOM** dibuat untuk membantu menghitung gaji karyawan berdasarkan jabatan, hari kerja, jam kerja, dan jumlah lembur. Pada awalnya, program meminta input berupa nama karyawan, jabatan karyawan (peracik petasan atau pengantar petasan), jumlah hari kerja, jam kerja per hari, dan jumlah lembur. Setelah itu, program menggunakan percabangan `if-elif-else` untuk menentukan besar bayaran per jam dan bayaran lembur sesuai dengan jabatan serta syarat-syarat tertentu.

Apabila jabatan yang dimasukkan adalah *peracik petasan*, terdapat tiga kondisi yang diperiksa. Jika hari kerja minimal 24, jam kerja minimal 8, dan lembur minimal 4, maka bayaran per jam adalah Rp25.000 dengan bayaran lembur Rp15.000. Jika hari kerja minimal 18, jam kerja minimal 6, dan lembur minimal 2, maka bayaran per jam adalah Rp20.000 dengan bayaran lembur Rp10.000. Selain dua kondisi tersebut, bayaran per jam ditetapkan Rp15.000 dan bayaran lembur Rp10.000. Sementara itu, jika jabatan adalah *pengantar petasan*, maka juga terdapat tiga kondisi. Jika hari kerja minimal 20, jam kerja minimal 7, dan lembur minimal 7, maka bayaran per jam Rp25.000 dengan bayaran lembur Rp20.000. Jika hari kerja minimal 16, jam kerja minimal 5, dan lembur minimal 4, maka bayaran per jam Rp20.000 dengan bayaran lembur Rp15.000. Jika selain dari dua kondisi itu, bayaran per jam ditetapkan Rp15.000 dan bayaran lembur Rp12.000.

Apabila jabatan yang dimasukkan tidak sesuai dengan dua pilihan yang tersedia, maka program akan menghentikan proses dan menampilkan pesan bahwa jabatan tidak dikenali. Setelah nilai bayaran per jam dan bayaran lembur ditentukan, program menghitung gaji karyawan dengan rumus: **total gaji = ((bayaran per jam × jam kerja) × hari kerja) + (jumlah lembur × bayaran lembur)**.

Hasil akhirnya, program menampilkan rincian gaji berupa nama karyawan, jabatan, jumlah hari kerja, jumlah jam kerja per hari, jumlah lembur, bayaran per jam, bayaran lembur, serta total gaji yang diterima. Dengan demikian, program ini memudahkan pengguna untuk mengetahui total gaji karyawan secara cepat dan akurat berdasarkan aturan yang telah ditetapkan.

3. Source Code *:*

```
print("=== Penghitung Gaji Karyawan PT. BOM ===")
nama = input("Masukkan Nama Karyawan: ")
jabatan = input("Masukkan Jabatan Karyawan (peracik petasan/pengantar
petasan): ").lower()
hari_kerja = int(input("Masukkan Jumlah Hari Kerja: "))
jam_kerja = int(input("Masukkan Jam Kerja per Hari: "))
lembur = int(input("Masukkan Jumlah Lembur: "))
bayaran_perjam = 0
bayaran_lembur = 0

if jabatan == "peracik petasan":
    if hari_kerja >= 24 and jam_kerja >= 8 and lembur >= 4:
        bayaran_perjam = 25000
        bayaran_lembur = 15000
    elif hari_kerja >= 18 and jam_kerja >= 6 and lembur >= 2:
        bayaran_perjam = 20000
        bayaran_lembur = 10000
    else:
        bayaran_perjam = 15000
        bayaran_lembur = 10000

elif jabatan == "pengantar petasan":
    if hari_kerja >= 20 and jam_kerja >= 7 and lembur >= 7:
        bayaran_perjam = 25000
        bayaran_lembur = 20000
    elif hari_kerja >= 16 and jam_kerja >= 5 and lembur >= 4:
        bayaran_perjam = 20000
        bayaran_lembur = 15000
    else:
        bayaran_perjam = 15000
        bayaran_lembur = 12000

else:
    print("Jabatan tidak dikenali. Program dihentikan.")
    exit()
total_gaji = ((bayaran_perjam * jam_kerja) * hari_kerja) + (lembur *
bayaran_lembur)
print("\n=== Rincian Gaji Karyawan PT. BOM ===")
print("Nama Karyawan   :", nama)
print("Jabatan         :", jabatan)
print("Hari Kerja      :", hari_kerja)
print("Jam Kerja       :", jam_kerja, "jam")
print("Jumlah Lembur   :", lembur, "jam")
print("Bayaran per Jam : Rp", bayaran_perjam)
print("Bayaran Lembur  : Rp", bayaran_lembur)
print("Total Gaji      : Rp", total_gaji)
```

4. Hasil Output

```
=== Penghitung Gaji Karyawan PT. BOM ===
Masukkan Nama Karyawan: geo
Masukkan Jabatan Karyawan (peracik petasan/pengantar petasan): peracik petasan
Masukkan Jumlah Hari Kerja: 26
Masukkan Jam Kerja per Hari: 10
Masukkan Jumlah Lembur: 8

=== Rincian Gaji Karyawan PT. BOM ===
Nama Karyawan   : geo
Jabatan         : peracik petasan
Hari Kerja      : 26
Jam Kerja       : 10 jam
Jumlah Lembur   : 8 jam
Bayaran per Jam : Rp 25000
Bayaran Lembur  : Rp 15000
Total Gaji      : Rp 6620000
```

5. Langkah-langkah GIT

5.1 GIT init

Perintah git init digunakan untuk menginisialisasi repository Git baru di dalam folder yang sedang digunakan. Setelah dijalankan, Git akan membuat folder tersembunyi bernama .git yang menyimpan semua konfigurasi dan riwayat versi proyek. Dengan langkah ini, direktori lokal resmi menjadi repository Git yang siap untuk melacak perubahan.

```
PS C:\Users\HYPE AMD\OneDrive\Documents\praktikum-apd> git init
Reinitialized existing Git repository in C:/Users/HYPE AMD/OneDrive/Documents/praktikum-apd/.git/
```

5.2 GIT add .

Perintah git add . digunakan untuk menambahkan semua perubahan file di dalam direktori ke staging area. Staging area adalah tempat sementara sebelum perubahan benar-benar disimpan ke dalam repository. Dengan menggunakan tanda titik (.), artinya semua file baru, yang diubah, maupun yang dihapus akan dimasukkan ke dalam daftar persiapan commit.

```
PS C:\Users\HYPE AMD\OneDrive\Documents\praktikum-apd> git add .
```

5.3 GIT Commit

Perintah git commit -m "Pesan" digunakan untuk menyimpan perubahan yang ada di staging area ke dalam repository lokal dengan sebuah pesan yang menjelaskan perubahan tersebut. Pesan commit sangat penting untuk mendokumentasikan riwayat proyek sehingga setiap perubahan dapat ditelusuri dengan mudah. Pada contoh, pesan commit adalah "Post-Test-3".

```
PS C:\Users\HYPE AMD\OneDrive\Documents\praktikum-apd> git commit -m "Post-Test-3"
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

5.4 GIT Remote

Perintah git remote add origin <url> digunakan untuk menghubungkan repository lokal dengan repository yang ada di GitHub. Dengan cara ini, repository lokal memiliki alamat tujuan (remote) untuk menyimpan atau mengambil update kode. Jika muncul error "remote origin already exists", artinya repository lokal sudah pernah dihubungkan dengan remote sebelumnya.

```
PS C:\Users\HYPE AMD\OneDrive\Documents\praktikum-apd> git remote add origin https://github.com/geogilang14-12/praktikum-apd
error: remote origin already exists.
```

5.4 GIT Push

Perintah git push -u origin main digunakan untuk mengirim commit dari repository lokal ke repository GitHub pada branch main. Opsi -u membuat branch lokal main otomatis terhubung dengan branch remote origin/main, sehingga pada push berikutnya cukup menggunakan git push. Setelah berhasil, perubahan yang dibuat secara lokal akan tampil di GitHub.

```
PS C:\Users\HYPE AMD\OneDrive\Documents\praktikum-apd> git push -u origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 16 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 561.14 KiB | 18.70 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: This repository moved. Please use the new location:
remote:   https://github.com/geogilang14-12/praktikum-apd.git
To https://github.com/geogilang14-12/pertemuan-1.git
   54ba562..3149a4b  main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\HYPE AMD\OneDrive\Documents\praktikum-apd>
```