

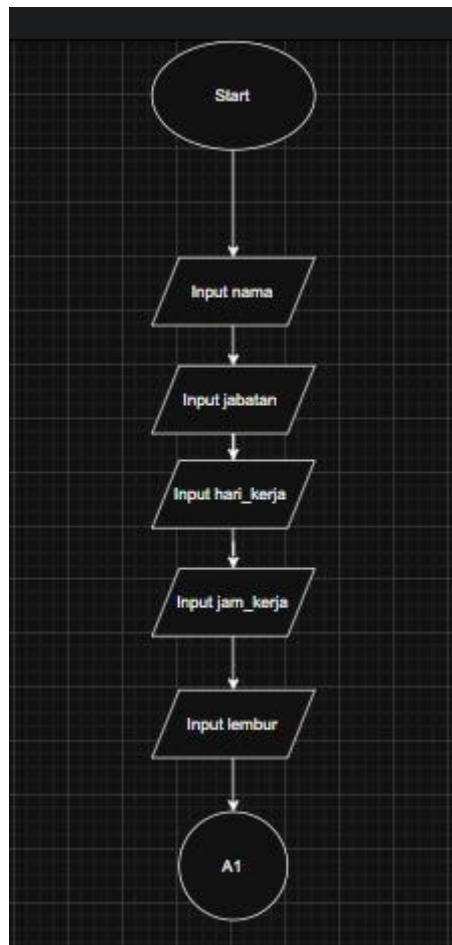
LAPORAN PRAKTIKUM
POSTTEST 3
ALGORITMA PEMROGRAMAN DASAR



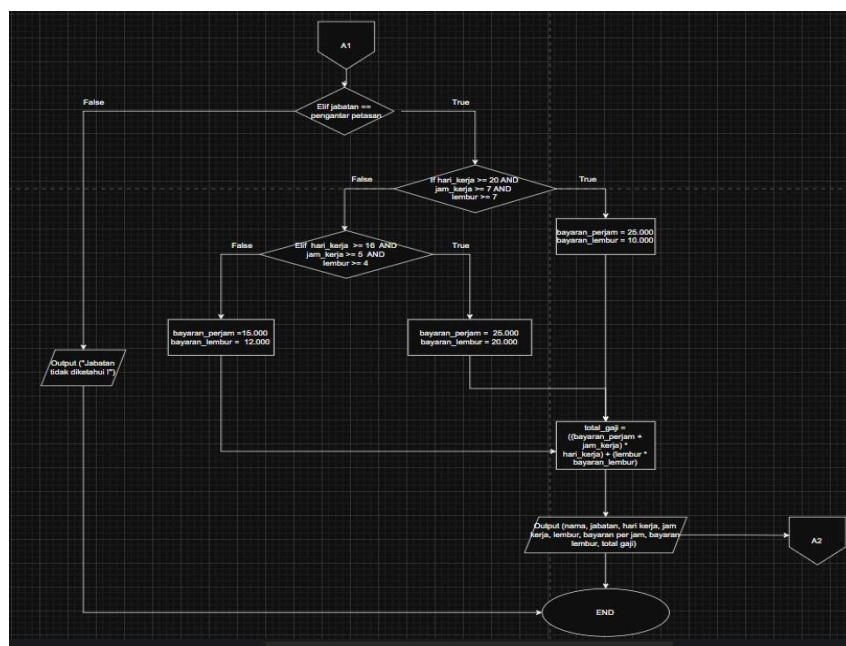
Disusun oleh:
Ahmad Aril Fadillah.B (2509106119)
Kelas (C '25)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

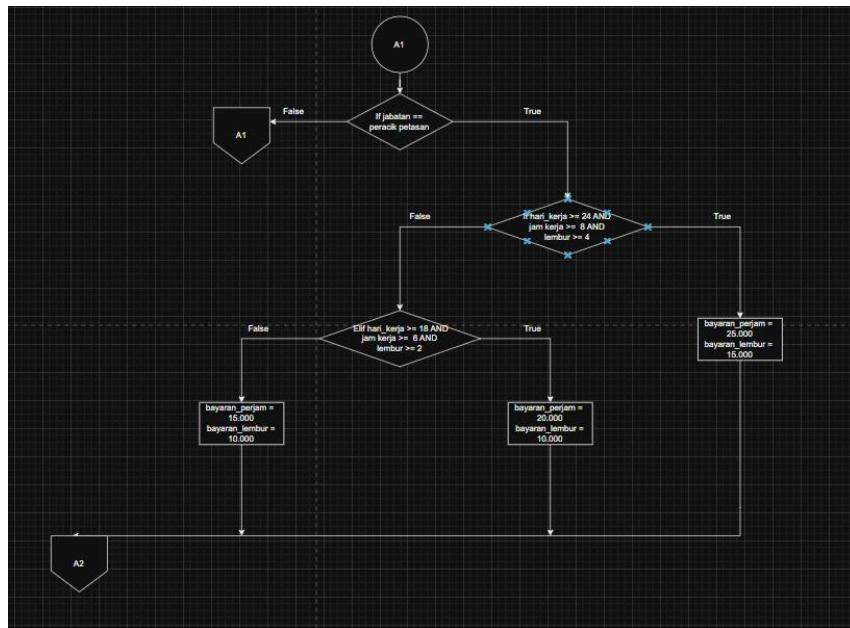
1.Flowchart



Gambar 1.2 Flowchart



Gambar 1.2 Flowchart



Gambar 1.3 Flowchart

Penjelasan :

1. Start :Adalah proses dimulai/awalan.

2. Input Data :

- Nama Karyawan : Masukkan nama karyawan.
- Jabatan (peracik petasan / pengantar petasan) : Masukkan Jabatanmu
- Jumlah Hari Kerja : Masukkan jumlah hari kerjamu
- Jam Kerja per Hari : Masukkan jumlah jam kerja per harimu
- Jumlah Lembur : Masukkan jumlah lemburmu

3. Keputusan(Decision) Apakah : jabatan = "peracik petasan"?

A. Benar (true) Masuk ke percabangan syarat hari kerja, jam kerja, dan lembur:

- If hari ≥ 24 dan jam ≥ 8 dan lembur ≥ 4 Bayaran per jam = 25.000, lembur = 15.000
- Else If hari ≥ 18 dan jam ≥ 6 dan lembur ≥ 2 Bayaran per jam = 20.000, lembur = 10.000
- Else Bayaran per jam = 15.000, lembur = 10.000

B. Salah (False) Cek lagi : Apakah jabatan == "pengantar petasan"?

- Jika hari ≥ 20 dan jam ≥ 7 dan lembur ≥ 7 Bayaran per jam = 25.000, lembur = 20.000
- Else If hari ≥ 16 dan jam ≥ 5 dan lembur ≥ 4 Bayaran per jam = 20.000, lembur = 15.000
- Else Bayaran per jam = 15.000, lembur = 12.000

C. Jika tidak keduanya maka : Output: *"Jabatan tidak dikenali !"*

4. Process (Perhitungan) Hitung total gaji:

$$\text{total gaji} = ((\text{bayaran per jam} * \text{jam kerja}) * \text{hari kerja}) + (\text{lembur} * \text{bayaran lembur})$$

5. Output :

- Nama Karyawan
- Jabatan
- Hari Kerja
- Jam Kerja / Hari
- Jumlah Lembur
- Bayaran per Jam
- Bayaran per Lembur
- Total Gaji

6. End : Program selesai.

2. Deskripsi Singkat Program

Program ini dipakai untuk menghitung gaji karyawan PT. BOM berdasarkan jabatan, jumlah hari kerja, jam kerja per hari, dan lembur. Setelah data dimasukkan, program menentukan upah gaji per jam dan upah gaji lembur sesuai aturan yang ada, lalu menghitung total gaji dan menampilkannya sebagai laporan.

Fungsi/Manfaat Utama:

1. Menghitung gaji karyawan PT. BOM secara otomatis berdasarkan jabatan, hari kerja, jam kerja, dan lembur.
2. Memudahkan perusahaan dalam menghitung gaji karyawan dengan cepat, mengurangi kesalahan dalam perhitungan manual, dan membuat laporan gaji lebih rapi.

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: "))


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!")
```

```
total_gaji = ((bayaran_perjam * jam_kerja) * hari_kerja) + (lembur * bayaran_lembur)
```

```
print("\n=== Laporan Gaji Karyawan ===")
```

```
print(f>Nama Karyawan    : {nama}")
```

```
print(f>Jabatan        : {jabatan}")
```

```
print(f>Hari Kerja      : {hari_kerja}")
```

```
print(f>Jam Kerja / Hari : {jam_kerja}")
```

```
print(f>Jumlah Lembur    : {lembur}")
```

```
print(f>Bayaran per Jam   : Rp {bayaran_perjam}")
```

```
print(f>Bayaran per Lembur: Rp {bayaran_lembur}")
```

```
print(f>Total Gaji      : Rp {total_gaji}")
```

4. Hasil Output

```
PS C:\Users\MyBook Hype AMD\OneDrive\Dokumen\P1\post-test\post-test-apd-3> & "C:/Users/MyBook Hype AMD/AppData/Local/Programs/Python/Python313/python.exe" "c:/Users/MyBook Hype AMD/OneDrive/Dokumen/P1/post-test/post-test-apd-3/2509106119-AHMAD ARILFADILLAH.B-PT-3.py.py"
=== Penghitung Gaji Karyawan PT. BOM ===
Masukkan Nama Karyawan: Agus Bijer
Masukkan Jabatan Karyawan (peracik petasan/pengantar petasan): Peracik petasan
Masukkan Jumlah Hari Kerja: 360
Masukkan Jam Kerja per Hari: 24
Masukkan Jumlah Lembur: 200

=== Laporan Gaji Karyawan ===
Nama Karyawan      : Agus Bijer
Jabatan            : peracik petasan
Hari Kerja         : 360
Jam Kerja / Hari   : 24
Jumlah Lembur      : 200
Bayaran per Jam    : Rp 25000
Bayaran per Lembur: Rp 15000
Total Gaji         : Rp 219000000
PS C:\Users\MyBook Hype AMD\OneDrive\Dokumen\P1\post-test\post-test-apd-3>
```

5. Langkah-langkah GIT

5.1 GIT Init

Tidak diperlukan lagi dalam menambahkan file baru bisa langsung **git add .** saja

Git init : adalah git yang digunakan pada pembuatan repository git baru di folder yang sedang kamu gunakan.

5.2 GIT Add

```
Reinitialized existing Git repository in C:/Users/MyBook Hype AMD/OneDrive/Dokumen/P1/.git/
PS C:\Users\MyBook Hype AMD\OneDrive\Dokumen\P1> git add .
PS C:\Users\MyBook Hype AMD\OneDrive\Dokumen\P1> git commit -m "masukkan commit"
[main 915af23] masukkan commit
```

Git add : adalah git untuk menambahkan perubahan file ke area persiapan sebelum disimpan secara permanen didalam commit.

5.3 GIT Commit

```
Reinitialized existing Git repository in C:/Users/MyBook Hype AMD/OneDrive/Dokumen/P1/.git/
PS C:\Users\MyBook Hype AMD\OneDrive\Dokumen\P1> git add .
PS C:\Users\MyBook Hype AMD\OneDrive\Dokumen\P1> git commit -m "masukkan commit"
[main 915af23] masukkan commit
```

Git commit : adalah git untuk menyimpan rekaman ke adaan dari perubahan yang sudah kamu siapkan di area persiapan ke dalam riwayat repository

5.4 GIT Remote

Ga perlu lagi menggunakan git remote karena dipakai hanya sekali di awal pembuatan repository untuk menghubungkan repository lokal dengan repository GitHubmu .

Git remote : adalah git yang digunakan untuk mengelola koneksi ke repository

5.5 GIT push -u origin main

```
PS C:\Users\MyBook Hype AMD\OneDrive\Dokumen\P2> git push -u origin main
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 8 threads
Compressing objects: 100% (13/13), done.
Writing objects: 100% (16/16), 593.59 KiB | 9.57 MiB/s, done.
Total 16 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/ahmadarilfadilah1222-beep/praktikum-apd.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

1. git push : ngirim perubahan dari lokal ke remote.
2. origin : nama remote (alamat repo GitHub yang sudah kamu hubungkan).
3. main : nama branch yang dipush.