

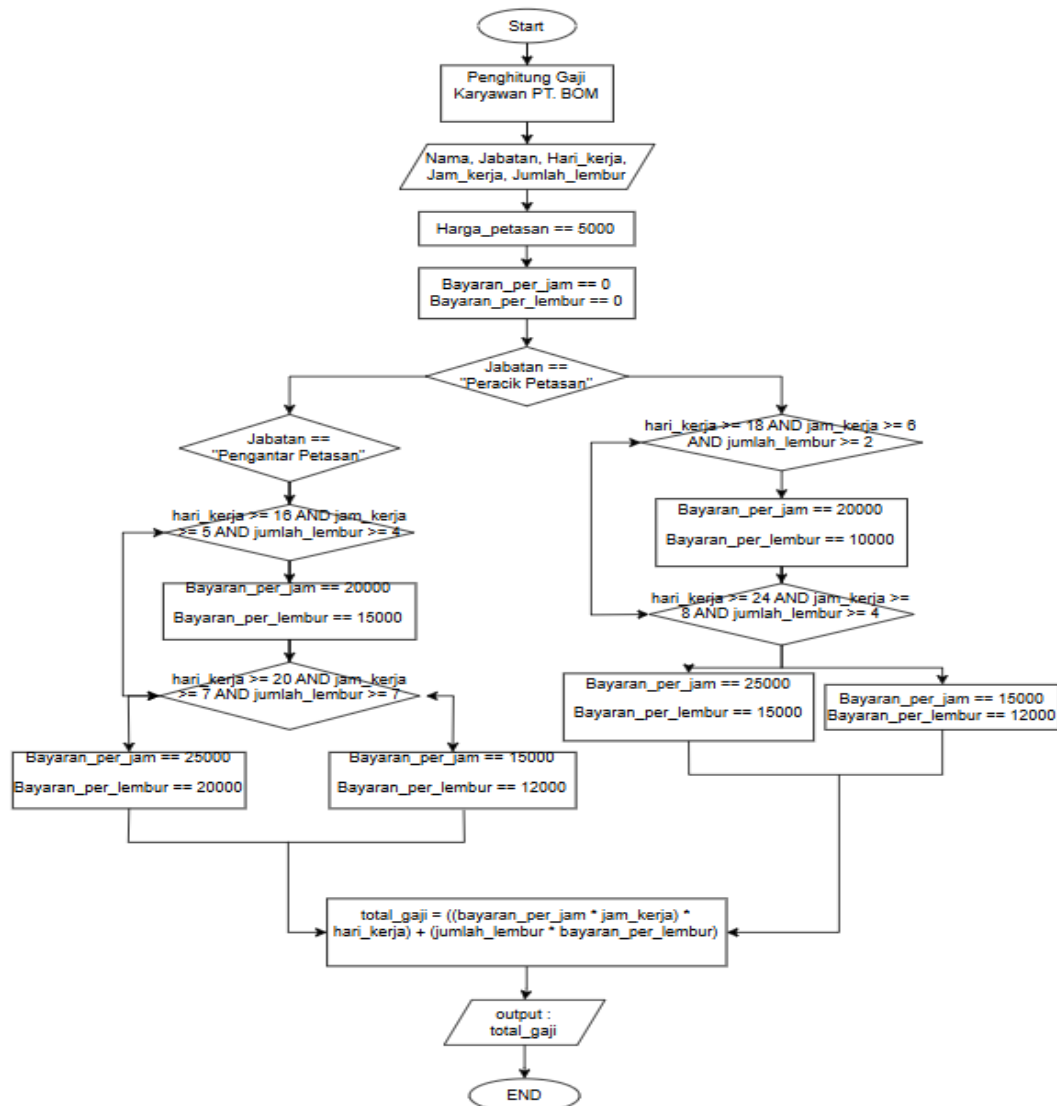
LAPORAN PRAKTIKUM
POSTTEST 3
ALGORITMA PEMROGRAMAN DASAR



Disusun oleh:
Radhika Adhitya Arifin (2509106130)
INFORMATIKA C2 '25

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



pertama-tama kita input dulu nama, jabatan, Hari kerja, Jam kerja, Jumlah lembur agar dia bisa dibaca oleh system terus kita masukan Variable awal harga_petasan=5000, bayaran_perjam=0, bayaran_perlembur=0 lalu kita bikin decision masukan sesuai yang kita inginkan lalu kita masuk ke proses, proses disini saya masukan ($total_gaji = ((bayaran_per_jam * jam_kerja) * hari_kerja) + (jumlah_lembur * bayaran_per_lembur)$) dan mengeluarkan hasil dari program tersebut.

2. Deskripsi Singkat Program

untuk dapat memudahkan penghitungan gaji karyawan, agar tidak repot repot menghitung gaji satu persatu. dengan program ini kita dapat mengetahui seberapa besar gaji karyawan tersebut.

3. Source Code

Source Code:

```
print("=== Penghitung Gaji Karyawan PT. BOM ===")

# Input data karyawan

nama = input("Masukkan nama karyawan: ")

jabatan = input(f"Masukkan jabatan karyawan (peracik/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: "))

# Harga petasan per pcs

harga_petasan = 5000

# Inisialisasi bayaran

bayaran_per_jam = 0

bayaran_lembur = 0

# Percabangan utama berdasarkan jabatan
```

```

if jabatan == "peracik":

    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":

    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. Silakan masukkan 'peracik' atau 'pengantar'.")

```

```

# Perhitungan total gaji

total_gaji = ((bayaran_per_jam * jam_kerja) * hari_kerja) + (lembur *
bayaran_lembur)

# Output hasil

print("\n<=== Rincian Gaji Karyawan ==>")

print("Nama Karyawan      :", nama)

print("Jabatan            :", jabatan.capitalize())

print("Harga Petasan/pcs    : Rp{:, .0f}".format(harga_petasan))

print("Hari Kerja           :", hari_kerja)

print("Jam Kerja per Hari    :", jam_kerja)

print("Jumlah Lembur         :", lembur)

print("Bayaran per Jam       : Rp{:, .0f}".format(bayaran_per_jam))

print("Bayaran Lembur        : Rp{:, .0f}".format(bayaran_lembur))

print("Total Gaji            : Rp{:, .0f}".format(total_gaji))

print("\n b Terima kasih telah menggunakan sistem PT. BOM")

```

4. Hasil Output

```
=== Penghitung Gaji Karyawan PT. BOM ===
Masukkan nama karyawan: Radhika
Masukkan jabatan karyawan (peracik/pengantar): peracik
Masukkan jumlah hari kerja: 26
Masukkan jumlah jam kerja per hari: 6
Masukkan jumlah lembur: 1

<=== Rincian Gaji Karyawan ===>
Nama Karyawan      : Radhika
Jabatan            : Peracik
Harga Petasan/pcs  : Rp5,000
Hari Kerja         : 26
Jam Kerja per Hari : 6
Jumlah Lembur      : 1
Bayaran per Jam    : Rp15,000
Bayaran Lembur     : Rp10,000
Total Gaji         : Rp2,350,000
```

5. Langkah-langkah GIT

5.1 GIT Add

```
PS C:\Users\USER\Documents\Praktikum1\Praktikum_APD\Posttest\post-test-apd-3> git add .
```

jangan lupa pakai spasi dan dot (.)

5.2 GIT Commit

```
PS C:\Users\USER\Documents\Praktikum1\Praktikum_APD\Posttest\post-test-apd-3> git commit -m "post-test-3"
[main ca9ece2] post-test-3
1 file changed, 59 insertions(+)
create mode 100644 post-test-apd-3/2509106130-RadhikaAdhityaArifin-PT-3.py
```

5.3 GIT Push

```
PS C:\Users\USER\Documents\Praktikum1\Praktikum_APD\Posttest\post-test-apd-3> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 1001 bytes | 333.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/adhityaradhika3/praktikum-APD.git
32d5684..ca9ece2 main -> main
PS C:\Users\USER\Documents\Praktikum1\Praktikum_APD\Posttest\post-test-apd-3>
```