

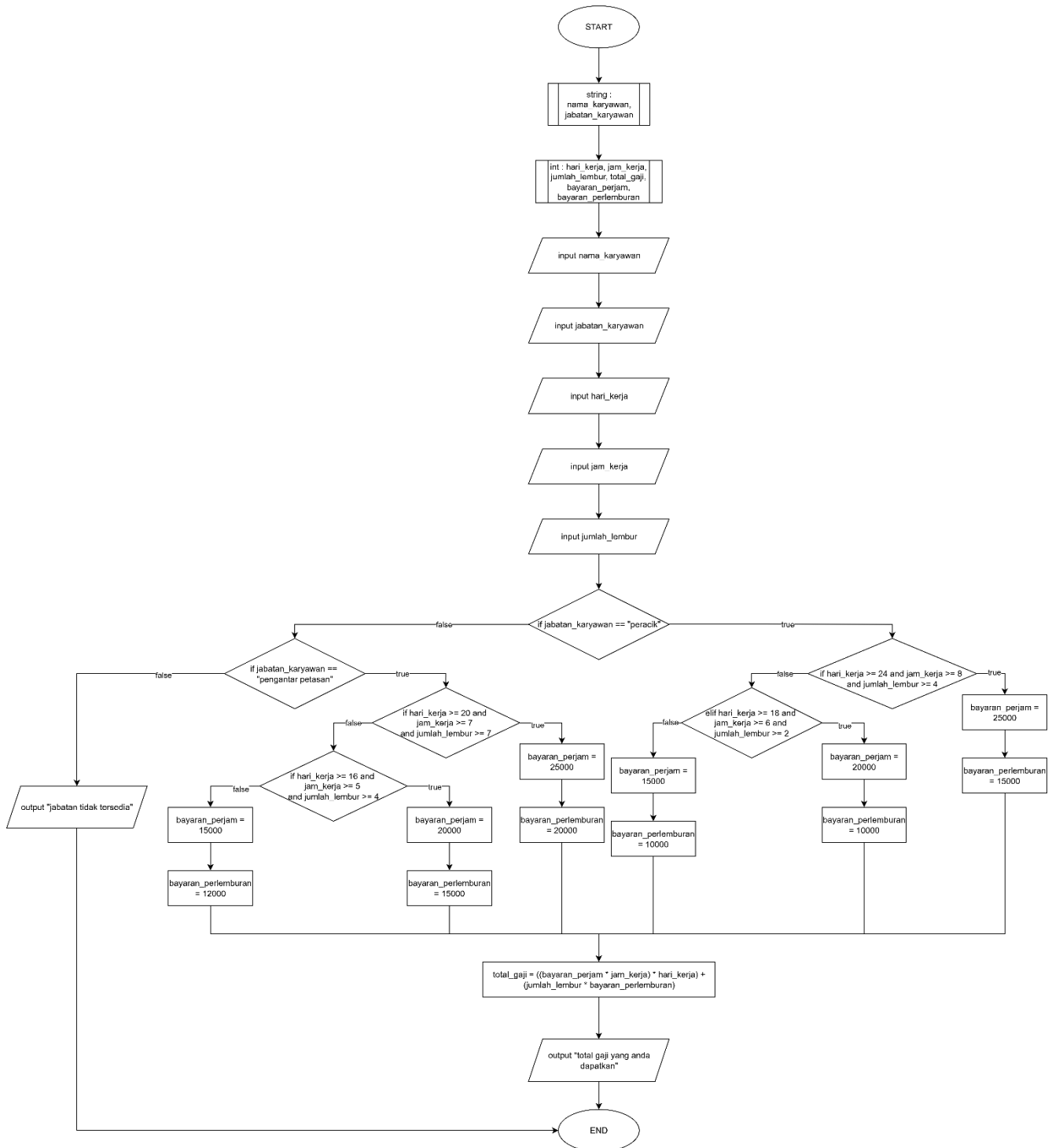
LAPORAN PRAKTIKUM
POSTTEST 3
ALGORITMA PEMROGRAMAN DASAR



Disusun oleh:
Muhammad Hajar Aswad (2509106118)
Kelas (C2 '25)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



Penjelasan Flowchart

1. Di mulai start.
2. Input nama karyawan, jabatan karyawan, hari kerja, jam kerja, dan jumlah lembur.
3. Program akan mengecek kondisi jika jabatan karyawan sebagai “*peracik*” maka akan mengecek kondisi untuk hari kerja, jam kerja, dan jumlah lembur :
 - a. Jika hari kerja ≥ 24 , hari kerja ≥ 8 , dan jumlah lembur ≥ 4 maka akan mendapatkan bayaran perjam = 25.000 dan bayaran per lembur = 15.000 kemudian program akan menghitung total gaji yang akan di dapatkan oleh karyawan dengan rumus = $((\text{bayaran perjam} * \text{jam kerja}) * \text{hari kerja}) + (\text{jumlah lemburan} * \text{bayaran per lemburan})$ dan program akan mengeluarkan output total gaji.
 - b. Jika kondisi pertama tidak terpenuhi lanjut kondisi ke dua yaitu, jika hari kerja ≥ 18 , hari kerja ≥ 6 , dan jumlah lembur ≥ 2 maka akan mendapatkan bayaran perjam = 20.000 dan bayaran per lembur = 10.000 kemudian program akan menghitung total gaji yang akan di dapatkan oleh karyawan dengan rumus = $((\text{bayaran perjam} * \text{jam kerja}) * \text{hari kerja}) + (\text{jumlah lemburan} * \text{bayaran per lemburan})$ dan program akan mengeluarkan output total gaji.
 - c. Jika kedua kondisi tersebut juga tidak terpenuhi maka akan mendapatkan biaya per jam = 15.000 dan bayaran per lembur = 10.000 kemudian program akan menghitung total gaji yang akan di dapatkan oleh karyawan dengan rumus = $((\text{bayaran perjam} * \text{jam kerja}) * \text{hari kerja}) + (\text{jumlah lemburan} * \text{bayaran per lemburan})$ dan program akan mengeluarkan output total gaji.
4. Namun jika kondisi jabatannya bukan sebagai “*peracik*” maka program akan mengecek kondisi selanjutnya yaitu karyawan sebagai “*pengantar petasan*” dan akan mengecek kondisi lagi untuk hari kerja, jam kerja, dan jumlah lembur :
 - a. Jika hari kerja ≥ 20 , hari kerja ≥ 7 , dan jumlah lembur ≥ 7 maka akan mendapatkan bayaran perjam = 25.000 dan bayaran per lembur = 20.000 kemudian program akan menghitung total gaji yang akan di dapatkan oleh karyawan dengan rumus = $((\text{bayaran perjam} * \text{jam kerja}) * \text{hari kerja}) + (\text{jumlah lemburan} * \text{bayaran per lemburan})$ dan program akan mengeluarkan output total gaji.
 - b. Jika kondisi pertama tidak terpenuhi lanjut kondisi ke dua yaitu, jika hari kerja ≥ 16 , hari kerja ≥ 5 , dan jumlah lembur ≥ 4 maka akan mendapatkan bayaran perjam = 20.000 dan bayaran per lembur = 15.000 kemudian program akan menghitung total gaji yang akan di dapatkan oleh karyawan dengan rumus = $((\text{bayaran perjam} * \text{jam kerja}) * \text{hari kerja}) + (\text{jumlah lemburan} * \text{bayaran per lemburan})$ dan program akan mengeluarkan output total gaji.
 - c. Jika kedua kondisi tersebut juga tidak terpenuhi maka akan mendapatkan biaya per jam = 15.000 dan bayaran per lembur = 12.000 kemudian program akan menghitung total gaji yang akan di dapatkan oleh karyawan dengan rumus = $((\text{bayaran perjam} * \text{jam kerja}) * \text{hari kerja}) + (\text{jumlah lemburan} * \text{bayaran per lemburan})$ dan program akan mengeluarkan output total gaji.

5. Namun jika kondisi karyawan sebagai “*pengantar petasan*” juga tidak terpenuhi maka program akan langsung mengeluarkan output “*jabatan karyawan tidak tersedia*”.
6. End.

2. Deskripsi Singkat Program

A. Tujuan

Tujuan di buat nya program ini adalah untuk menghitung total gaji karyawan dari PT.BOM yang gaji nya di hitung berdasarkan dari jumlah hari kerja, jam kerja, serta total lemburnya

B. Manfaat

- Memudahkan untuk proses perhitungan gaji karyawan berdasarkan jabatan karyawan.
- Menyesuaikan gaji karyawan karena perhitungan berdasarkan hari kerja, jam kerja, dan total lembur karyawan.

3. Source Code

A. Fitur Kondisi (if == peracik)

Fitur ini untuk mengecek kondisi apakah jabatan karyawan itu sebagai peracik jika ya maka akan lanjut untuk mengecek apakah hari kerja,jam kerja,dan jumlah lembur nya.

Source Code:

```
if jabatan_karyawan == "peracik":
    if hari_kerja >= 24 and jam_kerja >= 8 and jumlah_lembur >= 4:
        bayaran_perjam = 25000
        bayaran_perlemburan = 15000

    elif hari_kerja >= 18 and jam_kerja >= 6 and jumlah_lembur >= 2:
        bayaran_perjam = 20000
        bayaran_perlemburan = 10000

    else :
        bayaran_perjam = 15000
        bayaran_perlemburan = 10000
```

B. Fitur Kondisi (elif == pengantar petasan)

Fitur ini untuk mengecek kondisi jika kondisi sebagai peracik tadi false maka akan lanjut mengecek kondisi jabatan karyawan itu sebagai pengantar petasan jika ya maka akan lanjut untuk mengecek apakah hari kerja, jam kerja, dan jumlah lembur nya. Dan jika juga masih tidak sesuai program akan otomatis selesai karena data karyawan tidak valid

Source Code:

```
elif jabatan_karyawan == "pengantar petasan":  
    if hari_kerja >= 20 and jam_kerja >= 7 and jumlah_lembur >= 7:  
        bayaran_perjam = 25000  
        bayaran_perlemburan = 20000  
  
    elif hari_kerja >= 16 and jam_kerja >= 5 and jumlah_lembur >= 4:  
        bayaran_perjam = 20000  
        bayaran_perlemburan = 15000  
  
    else :  
        bayaran_perjam = 15000  
        bayaran_perlemburan = 12000  
  
else:  
    print("Jabatan tidak tersedia")  
    exit()
```

C. Fitur Menghitung Total Gaji

Fitur ini di gunakan untuk menghitung total gaji karyawan

Source Code:

```
total_gaji = ((bayaran_perjam * jam_kerja) * hari_kerja) + (jumlah_lembur *  
bayaran_perlemburan)
```

4. Hasil Output

A. Sebagai Peracik

```
PS D:\praktikum-apd> & C:/Users/Luzkyyy/AppData/Local/Programs/Python/Python38-32/Python.exe C:/Users/Luzkyyy/AppData/Local/Programs/Python/Python38-32/Python.exe MuhammadHajarAswad-PT-3.py
=====Input Data Perhitungan Gaji Karyawan PT.BOM=====
Masukkan nama karyawan :udin
Masukkan jabatan karyawan :peracik
Masukkan hari kerja anda :25
Masukkan jam kerja anda :9
Masukkan jumlah lembur anda :4
=====
                Penghitung Gaji Karyawan PT. BOM
=====
Nama Karyawan       : udin
Jabatan             : peracik
Hari Kerja          : 25 hari
Jam Kerja           : 9 jam
Jumlah Lembur       : 4
-----
Bayaran per jam     : Rp 25,000
Bayaran per lemburan : Rp 15,000
-----
Total gaji anda    : Rp 5,685,000
-----
PS D:\praktikum-apd> 
```

Gambar 4.1 Output program

B. Sebagai Pengantar Petasan

```
PS D:\praktikum-apd> & C:/Users/Luzkyyy/AppData/Local/Programs/Python/Py
MuhammadHajarAswad-PT-3.py
=====Input Data Perhitungan Gaji Karyawan PT.BOM=====
Masukkan nama karyawan :udin
Masukkan jabatan karyawan :pengantar petasan
Masukkan hari kerja anda :17
Masukkan jam kerja anda :6
Masukkan jumlah lembur anda :6
=====
                Penghitung Gaji Karyawan PT. BOM
=====
Nama Karyawan      : udin
Jabatan            : pengantar petasan
Hari Kerja         : 17 hari
Jam Kerja          : 6 jam
Jumlah Lembur      : 6
-----
Bayaran per jam    : Rp 20,000
Bayaran per lemburan : Rp 15,000
-----
Total gaji anda   : Rp 2,130,000
-----
PS D:\praktikum-apd> █
```

Gambar 4.2 Output program

5. Langkah-langkah GIT

5.1 GIT Add

Membuat file kita menjadi ke staging area sebelum benar-benar di simpan ke dalam repository kita.

```
PS D:\praktikum-apd> git add .
PS D:\praktikum-apd> █
```

Gambar 5.1 Git Add

5.2 GIT Commit

Git Commit berfungsi untuk menyimpan perubahan yang telah kita lakukan pada file kita yang dari staging area dan biasanya berisi pesan singkat sebagai catatan perubahan apa yang kita lakukan.

```
PS D:\praktikum-apd> git commit -m "revisi code print jabatan"
[main 65462c7] revisi code print jabatan
 1 file changed, 11 insertions(+), 11 deletions(-)
PS D:\praktikum-apd> █
```

Gambar 5.2 Git commit

5.3 GIT Push

Git Push berfungsi untuk mengupload perubahan kita dari repository lokal kita agar tersimpan online di akun github.

```
PS D:\praktikum-apd> git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 566 bytes | 283.00 KiB/s, done.
Total 5 (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/Aswad129/praktikum-apd.git
   d910d81..65462c7  main -> main
PS D:\praktikum-apd> █
```

Gambar 5.3 Git push