

## **UAS PEMROGRAMAN DASAR**

**LAPORAN PENJELASAN PROGRAM Dosen Pengampu Bapak Saiful Nur  
Budiman, S.Kom., M.Kom**



**Disusun Oleh :**

1. Haska Destama : 25104410076
2. M. Sakti Maulana A. : 25104410056
3. Dani irawanto : 25104410057
4. Cusunul Fitriawati : 25104413072
5. Heppy Indah : 21104413004

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI UNIVERSITAS ISLAM BALITAR  
BLITAR  
2025**

## LAPORAN PENJELASAN PROGRAM C++

**Judul:**

### **Program Menghitung Gaji Karyawan**

```
// library
#include <iostream>
#include <string>
using namespace std;

// aplikasi penghitung gaji karyawan
int main() {
    // bagian 1
    // variable dan cin/cout
    string nama;
    int golongan, jam_kerja;
    double gaji_pokok, uang_lembur;
    long int gaji_total;
    cout << "Masukkan nama anda: ";
    cin >> nama;
```

#### **1. Penjelasan Deklarasi Fungsi main()**

Fungsi main() merupakan bagian utama dari program C++ yang berfungsi sebagai titik awal eksekusi. Semua perintah yang ditulis di dalam fungsi ini akan dijalankan pertama kali ketika program dijalankan.

Deklarasi int main() berarti bahwa fungsi utama akan mengembalikan nilai bertipe integer, di mana nilai 0 menandakan bahwa program berjalan dengan sukses tanpa kesalahan. Di dalam fungsi main(), terdapat beberapa variabel yang digunakan untuk menyimpan data, seperti:

string nama untuk menyimpan nama karyawan.

int golongan, jam\_kerja untuk menampung golongan kerja dan jumlah jam kerja.

double gaji\_pokok, uang\_lembur untuk menyimpan nilai gaji pokok dan uang lembur.

long int gaji\_total untuk menampung hasil akhir total gaji.

## PENJELASAN PROGRAM C++

```
#include <iostream>
#include <string>
using namespace std;

//menghitung pendapatan parkir
int main() {
```

`#include <iostream>` digunakan untuk library input dan output terminal, pada program ini

`#include <string>` digunakan untuk library variable type data string.

`using namespace std;` dalam C++ adalah deklarasi yang memberitahu kompiler untuk menggunakan semua nama (seperti `cin`, `cout`, `endl`, `vector`) dari namespace standar (`std`) secara langsung tanpa perlu menuliskan `std::` di depannya. Ini menyederhanakan kode dengan menghilangkan kebutuhan untuk mengulang prefiks `std::`, membuatnya lebih ringkas.

Deklarasi `int main()` berarti bahwa fungsi utama akan mengembalikan nilai bertipe integer, di mana nilai 0 menandakan bahwa program berjalan dengan sukses tanpa kesalahan

**2. Pada program ini kita menggunakan percabangan dengan tipe data `if` , `else if` dan juga `if` untuk menentukan suatu golongan gaji**

```
// bagian 2
// output dan input golongan
// percabangan golongan gaji
cout << "Masukkan golongan anda (1/2/3): ";
cin >> golongan;
if (golongan == 1) {
    gaji_pokok = 25000;
}
else if (golongan == 2) {
    gaji_pokok = 35000;
}
else if (golongan == 3) {
    gaji_pokok = 50000;
}
else {
    cout << "Golongan tidak valid!" << "\n";
    return 1;
}
```

- If ini kita isikan variable golongan 1 dengan gaji pokok Rp25.000
- Else If ini kita isikan variable golongan 2 dengan gaji pokok Rp35.000
- Else ini kita isikan variable golongan 1 dengan gaji pokok Rp50.000

```

// Variable untuk inputan
int motor, mobil, jamMasuk, jamKeluar;

// variable untuk total pendapatan
double totalPendapatanParkir;

//masukkan data inputan
cout << "Masukkan jumlah motor : ";
cin >> motor;
cout << "Masukkan jumlah mobil : ";
cin >> mobil;
cout << "Input jam masuk(format 24 jam) : ";
cin >> jamMasuk;
cout << "Input jam keluar (format 24 jam): ";
cin >> jamKeluar;

```

Pada potongan kode ini, program menyiapkan beberapa variabel yang digunakan untuk menyimpan data terkait jumlah kendaraan dan waktu parkir. Setelah itu, program meminta pengguna memasukkan nilai-nilai tersebut.

#### 1. Deklarasi Variabel

- motor, mobil

Digunakan untuk menyimpan jumlah kendaraan roda dua dan roda empat yang masuk ke area parkir.

- jamMasuk, jamKeluar

Menyimpan informasi waktu masuk dan keluar kendaraan dalam format 24 jam.

- totalPendapatanParkir

Tipe double karena pendapatan dapat berupa nilai desimal. Variabel ini akan digunakan untuk menghitung total uang yang diperoleh dari parkir.

#### 2. Input Data dari Pengguna

Program menggunakan cout untuk menampilkan perintah, lalu cin untuk menerima input dari pengguna:

- Masukkan jumlah motor

Pengguna memasukkan jumlah motor disimpan ke variabel motor.

- Masukkan jumlah mobil

Pengguna memasukkan jumlah mobil disimpan ke variabel mobil.

- Input jam masuk (format 24 jam)

Pengguna mengisi jam masuk parkir, misalnya 8, 13, atau 20 dst.

### 3. Penjelasan memasukkan jam kerja :

```
//masukkan jam kerja
cout << "Masukkan jam kerja anda per minggu : ";
cin >> jam_kerja;
```

#### Penjelasan :

- cout merupakan console output, digunakan untuk menampilkan teks ke dalam layar, di situ ada perintah agar user memasukkan jumlah jam kerjanya.
- cin >> jam\_kerja → cin berfungsi untuk membaca inputan user sedangkan variable jam\_kerja di gunakan untuk menyimpan inputan user berupa jumlah jam kerja.

### Coding untuk menghitung uang lembur :

```
//menghitung uang_lembur
if (jam_kerja > 48) {
    int jam_lembur = jam_kerja - 48;
    uang_lembur = jam_lembur * 10000;
} else {
    uang_lembur = 0;
}
```

#### Penjelasan :

- Pada soal no 1, pemberian uang lembur di dasarkan pada 2 kondisi, yaitu :
  - a. Uang lembur akan diberikan jika karyawan bekerja lebih dari 48 jam per minggu
  - b. Jika jam kerja karyawan kurang dari atau sama dengan 48 jam per minggu, maka karyawan tidak berhak mendapatkan uang lembur.
- Berdasarkan kondisi di atas maka kita gunakan if-else
- Dikarenakan di atas belum di deklarasikan variable jam\_lembur.. maka kita deklarasikan dulu variable jam\_lembur dengan type data integer yang nantinya akan di gunakan untuk menyimpan jumlah jam lembur (selisih antara jam kerja di kurangi dengan 48)
- Pada kondisi yang pertama, program akan mengecek apakah jam kerja > 48, jika iya :
  - a. Maka dilakukan perhitungan jumlah jam lemburnya

b. Jika jam lembur sudah ketemu, maka akan dilakukan perhitungan uang lembur dengan mengalikan jumlah jam lebur dengan 10.000 dan hasilnya akan di simpan pada variable `uang_lembur`.

- Jika inputan user kurang dari 48, maka akan masuk ke kondisi else, yaitu `uang_lembur = 0`

### Coding untuk mengecek inputan user

```
// Cek input*
if (jamMasuk < 0 || jamMasuk > 23 || jamKeluar < 0 || jamKeluar > 23) {
    cout << "Jam masuk atau keluar tidak valid!" << endl;
    return 1;
}
```

#### Penjelasan :

- Kondisi di atas digunakan untuk memastikan bawa inputan user pada untuk jam masuk dan jam keluar benar, yaitu dengan menggunakan format 24 jam (benar hanya jika inputan angka 0 sampai 23)
- Operator II (atau) di gunakan untuk mengecek inputan user, jika salah satu inputan user salah (yaitu nilai kurang dari 0 atau lebih dari 23) maka program akan menampilkan pesan "*Jam masuk atau keluar tidak valid*" lalu program akan berhenti yang di tandai dengan perintah `return 1;`

## Coding untuk menghitung lamanya parkir

```
//hitung lamanya parkir
int jamParkir;
if (jamKeluar >= jamMasuk) {
    jamParkir = jamKeluar - jamMasuk;
} else {
    jamParkir = (24 - jamMasuk) + jamKeluar;
}
if (jamParkir == 0) {
    jamParkir = 1;
}
```

### Penjelasan :

- Karena di atas belum di deklarasikan variable jam parkir, maka kita deklarasikan dulu variable jam\_parkir dengan type data integer yang nantinya akan digunakan untuk menyimpan data jumlah jam parkir.
- Batasan perhitungan jam parkir ini hanya di gunakan untuk kondisi parkir maksimal 2 hari
- Pada kondisi if yang pertama, digunakan untuk menghitung jumlah jam parkir di hari yang sama yaitu jam keluar pasti lebih besar nilainya dari pada jam masuk dengan menggunakan cara perhitungan seperti di atas.
- Jika tidak (else), yang artinya jika inputan jam keluar lebih kecil dari pada jam masuk ( telah melewati tengah malam) maka untuk perhitungan jumlah jam parkirnya seperti coding di atas.
- Pada kondisi if yang kedua, kita gunakan untuk menghitung minimum jumlah jam parkir (misalakan parkir kurang dari 1 jam, maka tetap akan kita hitung sebagai 1 jam) agar tidak ada nilai pembayaran 0.

## 4. Penjelasam Coding untuk menghitung gaji total :

### Penjelasan :

- gaji\_total: Variabel yang menyimpan hasil perhitungan total gaji karyawan.
- (gaji\_pokok \* jam\_kerja):
- gaji\_pokok = tarif gaji per jam
- jam\_kerja = jumlah jam kerja dalam seminggu
- Menghasilkan gaji pokok total dengan mengalikan tarif per jam dengan total jam kerja
- Contoh: Rp 50.000 per jam  $\times$  48 jam = Rp 2.400.000
- uang\_lembur: Tambahan uang dari jam lembur yang sudah dihitung sebelumnya.
- Operator +: Menjumlahkan gaji pokok total dengan uang lembur untuk mendapatkan gaji keseluruhan.

Coding untuk menampilkan hasil :

### Penjelasan :

- "Total Gaji ": Teks label sebagai keterangan awal.
- nama: Menampilkan nama karyawan untuk personalisasi output.
- " per minggu : Rp. ": Teks keterangan periode gaji dan mata uang.
- gaji\_total: Menampilkan hasil perhitungan total gaji.
- "\n": Membuat baris baru setelah output.

#### BAGIAN 4 . 2

Coding untuk Hitung total pendapatan

Penjelasan :

- int totalPendapatan: Deklarasi variabel dengan tipe data integer (bilangan bulat) bernama totalPendapatan yang akan menyimpan hasil perhitungan total pendapatan parkir.
- (motor \* 2000 \* jamParkir):
- motor = jumlah kendaraan motor yang parkir
- 2000 = tarif parkir per jam untuk motor (Rp 2.000)
- jamParkir = durasi parkir dalam jam
- Rumus ini menghitung: jumlah motor  $\times$  tarif motor  $\times$  durasi parkir
- Contoh: 5 motor  $\times$  Rp 2.000  $\times$  3 jam = Rp 30.000
- (mobil \* 5000 \* jamParkir):
- mobil = jumlah kendaraan mobil yang parkir
- 5000 = tarif parkir per jam untuk mobil (Rp 5.000)
- jamParkir = durasi parkir dalam jam
- Rumus ini menghitung: jumlah mobil  $\times$  tarif mobil  $\times$  durasi parkir
- Contoh: 3 mobil  $\times$  Rp 5.000  $\times$  3 jam = Rp 45.000
- Operator +: Menjumlahkan total pendapatan dari motor dan mobil
- Total = Rp 30.000 + Rp 45.000 = Rp 75.000

Coding untuk tampilan hasil akhir

Penjelasan :

- cout: Perintah untuk menampilkan output ke layar.
- 
- <<: Operator untuk mengirimkan data ke layar, dapat digunakan secara berantai.
- 
- "Total pendapatan parkir: Rp ": Teks label yang ditampilkan sebagai keterangan.
- 
- totalPendapatan: Menampilkan nilai hasil perhitungan total pendapatan.
- 
- jamParkir: Menampilkan durasi waktu parkir.
- 
- \n: Membuat baris baru setelah output ditampilkan.
- \*Contoh output:\*
- Total pendapatan parkir: Rp 75000 (3 jam)