

**ALGORITMA DAN PEMROGRAMAN II**  
**PROGRAM UNTUK MENGHITUNG GAJI BERSIH PEGAWAI**  
**PADA CV. XXXI**



Nama : Muh. Zhafar Zahara

Nim : 13020190252

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS MUSLIM INDONESIA**  
**MAKASSAR**

**2020**

## **KATA PENGANTAR**

Bismillahi Rahmani Rahim, Alhamdullillahi Rabbil Alamin segala puji kita panjatkan kehadiran Allah SWT atas segala nikmat dan karunianya tercurahkan kepada kita yang tak terhingga ini, sholawat serta salam kita panjatkan kepada junjungan kita Nabi besar Muhammad SAW dan keluarganya, sahabatnya, beserta pengikutnya sampai akhir zaman aminn ya robal alamin.

Makalah ini telah penulis susun dan mendapatkan bantuan dari berbagai pihak sehingga yang dapat memperlancar pembuatan makalah ini. Untuk itu kami menyampaikan banyak terima kasih kepada semua pihak yang telah berkontribusi dalam pembuatan makalah ini.

Terlepas dari semua itu, Penulis menyadari sepenuhnya bahwa masih ada kekurangan baik dari segi susunan kalimat maupun tata bahasanya. Oleh karena itu dengan tangan terbuka kami menerima segala saran dan kritik dari pembaca agar penulis dapat memperbaiki makalah ini.

## DAFTAR ISI

KATA PENGANTAR .....	2
DAFTAR ISI.....	3
BAB I PENDAHULUAN.....	4
1.1 Deskripsi Kasus .....	4
1.2 Tujuan .....	4
BAB II URAIAN KEGIATAN .....	5
2.1 Analisis dan Desain Prosedur atau mekanisme kerja menggunakan algoritma Flowchart.....	5
2.2 Hasil Implementasi .....	6
2.3 Pengujian program.....	9
BAB III PENUTUP .....	10
3.1 Kesimpulan .....	10
3.2 Saran .....	10
DAFTAR PUSTAKA.....	11

## BAB 1

### PENDAHULUAN

#### 1.1 Deskripsi Kasus

- Pengguna memasukkan jumlah pegawai kemudian mengisi data pegawai: nip(nomor induk pegawai), nama pegawai, gaji pokok, golongan dan jam kerja, lembur dimasukkan melalui keyboard
- Gaji bersih pegawai dihitung dari gaji pokok dan tunjangan dipotong pajak yang harus dibayar. Besar tunjangan adalah 15 % dari gaji pokok, dan besar pajak adalah 10 % dari gaji pokok ditambah tunjangan
- Gaji pegawai ditentukan berdasarkan gaji tetap dan juga dari lamanya bekerja. Gaji tetap dan gaji per jam tergantung kepada golongan pegawai sesuai:

Golongan	Gaji tetap(Rp)	Upah lembur perjam(Rp)
1	500.000	5000
2	300.000	3000
3	250.000	2000
4	100.000	1500
5	50.000	1000

- Menghitung total gaji pegawai yang diterima dengan rumus: gaji sesuai golongan + (banyaknya jam lembur \* tarif lembur sesuai golongan karyawan).
- Ketentuan yang tidak disebutkan di atas dapat anda tentukan sendiri dalam laporan, minimal mengerjakan sesuai dengan kasus.
- mengurutkan dan mencari data nilai yang diinginkan sesuai kategori

#### 1.2 Tujuan Praktikum

Tujuan disusunnya laporan ini yaitu untuk menyelesaikan tugas mata pelajaran Algoritma dan Pemrograman II Selain itu, penulis berharap dibuatnya laporan ini bukan hanya untuk tugas yang diberikan semata, akan tetapi bisa membantu kita semua dalam hal mempelajari ilmu pengetahuan khususnya Pemrograman C++ ini. harapan penulis yaitu semoga laporan ini tidak hanya bermanfaat bagi penulis akan tetapi sangat bermanfaat khususnya juga bagi para pembaca atau kita semua.

## **BAB II**

### **PEMBAHASAN**

#### **2.1 Analisis dan Desain Prosedur atau mekanisme kerja menggunakan algoritma Flowchart**

#### **2.2 Hasil Implementasi**

```

1  #include <iostream>
2  using namespace std;
3  int r;
4
5  struct pegawai{
6      char nip[20], nama[30];
7      float gaji, gaji1, gaji2, upah, tgaji;
8      int golongan, jam;
9  };
10 struct pegawai >[10];
11
12 void garis(){
13     cout<<"\n-----\n";
14 }
15
16 int inputar(){
17     for(int i=0; i<r; i++){
18         cout<<"\nPegawai ke-"<<(i+1)<<endl;
19         cout<<"Nama Pegawai      : "; cin >> >[i].nama;
20         cout<<"No.Induk Pegawai : "; cin >> >[i].nip;
21         cout<<"Gaji Pokok       : "; cin >> >[i].gaji;
22         cout<<"Golongan        : "; cin >> >[i].golongan;
23         cout<<"Berapa Jam kerja : "; cin >> >[i].jam;
24     }
25     return 0;
26 }
27
28 void totalgaji(){
29     for(int i=0; i<r; i++){
30         if(>[i].golongan== 1 && >[i].jam >= 5){
31             >[i].gaji1 = 500000;
32             >[i].upah = (5000*>[i].jam-5));
33             >[i].tgaji = >[i].gaji1 + >[i].upah;
34         }
35         else if(>[i].golongan== 2 && >[i].jam >= 5){
36             >[i].gaji1 = 300000;
37             >[i].upah = (3000*>[i].jam-5));
38             >[i].tgaji = >[i].gaji1 + >[i].upah;

```

```

28 void totalgaji(){
29     for(int i=0; i<r; i++){
30         if(>[i].golongar== 1 && >[i].jam >= 1){
31             >[i].gaji1 = 50000;
32             >[i].upah = (5000*(>[i].jam-1));
33             >[i].tgaji = >[i].gaji1 + >[i].upah;
34         }
35         else if(>[i].golongar== 2 && >[i].jam >= 1){
36             >[i].gaji1 = 30000;
37             >[i].upah = (3000*(>[i].jam-1));
38             >[i].tgaji = >[i].gaji1 + >[i].upah;
39         }
40         else if(>[i].golongar== 3 && >[i].jam >= 1){
41             >[i].gaji1 = 25000;
42             >[i].upah = (2000*(>[i].jam-1));
43             >[i].tgaji = >[i].gaji1 + >[i].upah;
44         }
45         else if(>[i].golongar== 4 && >[i].jam >= 1){
46             >[i].gaji1 = 10000;
47             >[i].upah = (1500*(>[i].jam-1));
48             >[i].tgaji = >[i].gaji1 + >[i].upah;
49         }
50         else if(>[i].golongar== 5 && >[i].jam >= 1){
51             >[i].gaji1 = 5000;
52             >[i].upah = (1000*(>[i].jam-1));
53             >[i].tgaji = >[i].gaji1 + >[i].upah;
54         }
55         else{
56             cout<<"\nperusahaan : 'anda bukan gologan kami'\n";
57             break;
58         }
59     }
60 }
61
62 void gajibersih(){
63     for(int i=0; i<r; i++){
64         float t;
65         t=((15*>[i].tgaji)/100.0);

```

```

61
62 void gajiBERSIH(){
63     for(int i=0; i<r; i++){
64         float t,f;
65         t=((1!*>[i].gaji)/100.0);
66         f=((10.0*(t+>[i].gaji))/100.0);
67         >[i].gaji=((t+>[i].gaji)-f);
68     }
69 }
70
71 void output(){
72     garis();
73     garis();
74     for(int i=0; i<r; i++){
75         cout<<"\npegawai ke-"<<(i+1);
76         cout<<"\nNo.Induk Pegawai : " << >[i].nip;
77         cout<<"\nNama Pegawai      : " << >[i].nama;
78         cout<<"\nGaji Pokok       : " << >[i].gaji;
79         cout<<"\nGaji Bersih      : " << >[i].gajiB;
80         cout<<"\nGaji Tetap       : " << >[i].gajiT;
81         cout<<"\nUpah Lembur     : " << >[i].upah;
82         cout<<"\nTotal Gaji       : " << >[i].Tgaji<<endl;
83     }
84 }
85
86 int main(){
87     cout<<"PROGRAM UNTUK MENGHITUNG GAJI BERSIH PEGAWAI";
88     garis();
89     cout<<"\nSyarat : \n1. minimal kerja 5 jam\n2. maksimal golongan yaitu 5, tiap golongan berbeda pendapatan\n\n";
90     cout<<"Masukkan Jumlah Pegawai : ";
91     cin>>r;
92     inputar();
93     gajiBERSIH();
94     totalgaji();
95     output();
96     return 0;
97 }
98

```



## 2.3 Pengujian Program

```
PROGRAM UNTUK MENGHITUNG GAJI BERSIH PEGAWAI
-----

Syarat :
1. minimal kerja 5 jam
2. maksimal golongan yaitu 5, tiap golongan berbeda pendapatan

Masukkan Jumlah Pegawai : 2

Pegawai ke-1
Nama Pegawai      : Maulana
No.Induk Pegawai  : 13020190244
Gaji Pokok        : 500.000
Golongan          : 1
Berapa Jam kerja  : 2

Pegawai ke-2
Nama Pegawai      : Wandu
No.Induk Pegawai  : 13020190123
Gaji Pokok        : 300.000
Golongan          : 2
Berapa Jam kerja  : 1

perusahaan : 'anda bukan golongan kami'

-----
-----
```

```
pegawai ke-1
No.Induk Pegawai : 13020190244
Nama Pegawai     : Maulana
Gaji Pokok       : 500
Gaji Bersih      : 517.5
Gaji Tetap       : 0
Upah Lembur      : 0
Total Gaji       : 0

pegawai ke-2
No.Induk Pegawai : 13020190123
Nama Pegawai     : Wandu
Gaji Pokok       : 300
Gaji Bersih      : 310.5
Gaji Tetap       : 0
Upah Lembur      : 0
Total Gaji       : 0

-----
Process exited after 128.6 seconds with return value 0
Press any key to continue . . .
```

## **BAB III**

### **PENUTUP**

#### **3.1 Kesimpulan**

Untuk menciptakan suatu program yang baik yang memiliki portabilitas yang tinggi, sehingga memudahkan dalam merancang dan merawat program serta meningkatkan efektifitas penggunaan peralatan komputer dibutuhkan suatu standar program. Standar-standar tersebut sering dilihat oleh pemrogram sebagai batasan kreatifitas dan kemampuan untuk menuangkan berbagai ide ke dalam bentuk program. Namun dengan adanya standar akan membuat program menjadi konsisten dan mudah untuk dikembangkan..

#### **3.2 Saran**

Diharapkan makalah ini berguna dan dapat menjadi sarana pembelajaran oleh pembaca dan menjadi lebih paham mengenai program dan aplikasi lain pendukungnya

## **DAFTAR PUSTAKA**

Novianto Andi. 2014. *Pemrograman Dasar*. Jakarta Timur: Erlangga.

Limanto Susana, Muljono Anton. 2006. *Algoritma Dan Pemrograman*. Ciputat: Dinastindo.