BAB IV
INPUT DAN OUTPUT

Input Data

Umumnya suatu program mempunyai proses pemasukan data. Dalam program berbahasa C, pemasukan data dapat dilakukan dengan perintah scanf. Fungsi **scanf** merupakan fungsi yang dapat digunakan untuk memasukan berbagai jenis data, tergantung dengan format penentunya. Format-format penentu tipe data yang umum dipakai adalah :

Format	Kegunaan
%c	Digunakan untuk pemasukan data bertipe char
%i atau %d	Digunakan untuk pemasukan data bertipe int, char.
%f	Digunakan untuk pemasukan data berupa bi;angan pecahan
%u	Digunakan untuk pemasukan data berupa unsigned int atau unsigned char
%s	Digunakan untuk pemasukan data berupa string

Bentuk umum penggunaan fungsi scanf adalah

```
scanf ("format",&namavariabel);
```

Contoh :

```
int i,jam,menit,detik;
unsigned int j;
float f;
char nama[60];
scanf("%i",&i);
scanf("%u",&j);
scanf("%f",&f);
scanf("%i %i %i",&jam,&menit,&detik);
scanf("%s",nama);
```

Fungsi **scanf()** kurang begitu bagus jika dipakai untuk pembacaan string. Karena data yang tersimpan adalah hanya sampai menemukan spasi, maksudnya jika kita mengisikan 2 buah kata dengan pemisah spasi, maka data yang masuk ke variabel tersebut hanyalah kata yang pertama. Oleh karena itu, pembacaan data bertipe string biasanya menggunakan
perintah **gets()** yang bentuk umumnya adalah :

```
gets(namavariabel);
```

Contoh :

```
gets(nama);
gets(alamat);
```

Untuk pembacaan data bertipe char, selain dengan menggunakan scanf() dengan format %c, bisa juga dengan menggunakan fungsi getch() atau getche(). Perbedaan dari getch() dan **getche()** adalah **getch()** digunakan untuk membaca data bertipe char tanpa menampilkannya di layar, dan **getche()** digunakan untuk membaca data bertipe char dengan menampilkan data karakternya di layar.

Contoh :

```
#include <stdio.h>
#include <conio.h>
main()
   {
     char a,b;
     printf("Masukan Huruf pertama : ");
     a=getch();
     printf("Masukan Huruf kedua : ");
     b=getche();
     printf("Data yang dimasukan adalah %c dan %d\n",a,b);
     getch();
     return 0;
   }
```

Output Data

Untuk output data, perintah yang bisa dipakai adalah **printf()**. Untuk menampilkan data dengan fungsi **printf()**, kita harus mengatur format tampilannya, dengan format-format penentu.

Untuk menentukan panjang medan yang disediakan bagi tampilan data, maka sesudah tanda % dalam penentu format dapat disisipi dengan bilangan bulat yang menyatakan panjang medan.

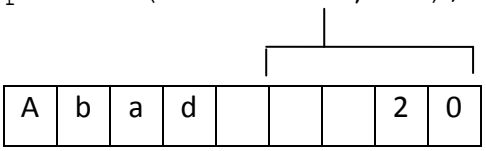
Untuk data yang berupa bilangan bulat, misalnya pada :

```
printf ("Abad %4d", 20);
```

Ket :

%4d menyatakan medan untuk menampilkan bilangan 20 adalah sepanjang 4 karakter.

```
printf ("Abad %4d", 20);
```



Untuk data yang berupa bilangan real, spesifikasi medannya berupa

```
m.n
```

Ket :

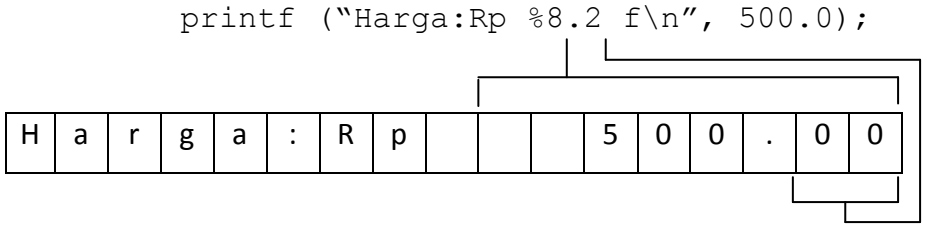
m = panjang medan

n = jumlah digit pecahan

Contoh :

```
printf ("Harga:Rp %8.2 f\n", 500.0);
```

%8.2f menyatakan panjang medan dari bilangan real yang akan ditampilkan adalah 8 karakter dengan jumlah digit pecahan 2 buah.



Jika hanya jumlah digit pecahan yang perlu ditentukan, panjang medan tak perlu disertakan, misal :

```
printf ("%.2f\n", 600.0);
printf ("%.2f\n", 7500.25);
```

hasilnya :

```
600.00
7500.25
```

Untuk data yang berupa string, contoh :

```
printf ("%12s", "Bahasa C");
```

maka akan ditampilkan sebagai berikut

				B	a	h	a	s	a		C

Tampak dalam berbagai jenis data diatas, penentu format yang mengandung panjang medan, secara default akan menampilkan data dalam bentuk rata kanan terhadap panjang medan yang diberikan. Untuk data string yang biasanya dikehendaki untuk ditampilkan dalam bentuk rata kiri, maka sesudah tanda % pada penentu format %s perlu disisipkan tanda – (minus), contoh :

```
printf ("%-12s", "Bahasa C");
```

menyatakan bahwa string akan ditampilkan dalam medan dengan panjang 12 karakter dan diatur rata kiri. Sehingga tampilan diatas berubah menjadi :

B	a	h	a	s	a		C				

Untuk lebih jelas perhatikan program di bawah ini.

```
#include <stdio.h>
#include <conio.h>
main()
{
     int a=25000;
     unsigned int b=50000;
     float c=12345.678;
     char nama[50]="Universitas Komputer Indonesia";
     char alamat[10]="Bandung";
     clrscr();
     printf("Penampilan data tanpa di format\n");
     printf("Nilai a : %d\n",a);
     printf("Nilai b : %u\n",b);
     printf("Nilai c : %f\n",c);
     printf("String : %s %s\n",nama,alamat);//rata kanan
     printf("\nPenampilan data setelah di format\n");
     printf("\n");
     printf("Nilai a : %8d\n",a);
     printf("Nilai b : %8u\n",b);
     printf("Nilai c : %11.2f\n",c);
     printf("String 1: %40s %10s\n",nama,alamat);//rata kanan
     printf("String 2: %-40s %-10s\n",nama,alamat);//rata kiri
     getch();
     return 0;
}
```



Contoh Kasus :

Di suatu perusahaan, data penggajian dihitung dengan ketentuan sebagai berikut :

Gaji Pokok : Rp. 5000000

Gaji Lembur/jam : Rp. 5000

Total Gaji Lembur : Lama Lembur * Gaji Lembur/jam

Gaji Kotor : Gaji Pokok + Total Gaji Lembur

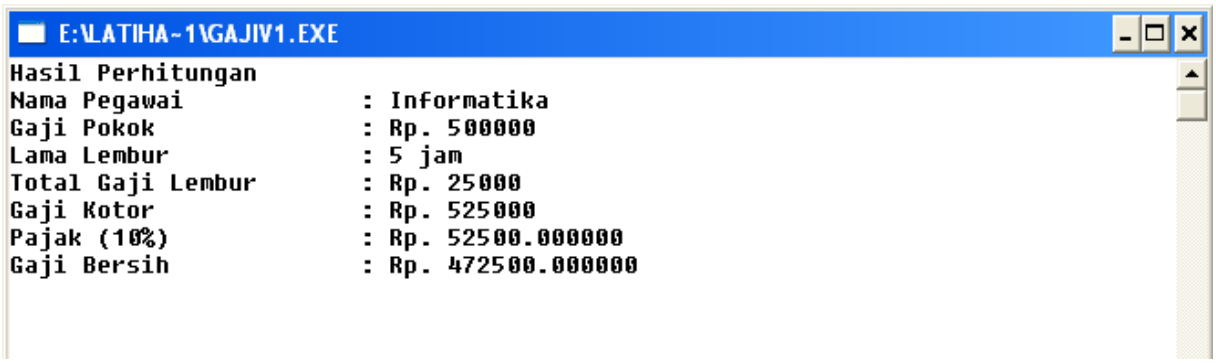
Pajak : 10%

Gaji Bersih : Gaji Kotor - Pajak

Data yang diinputkan adalah : Nama Pegawai, Lama Lembur.

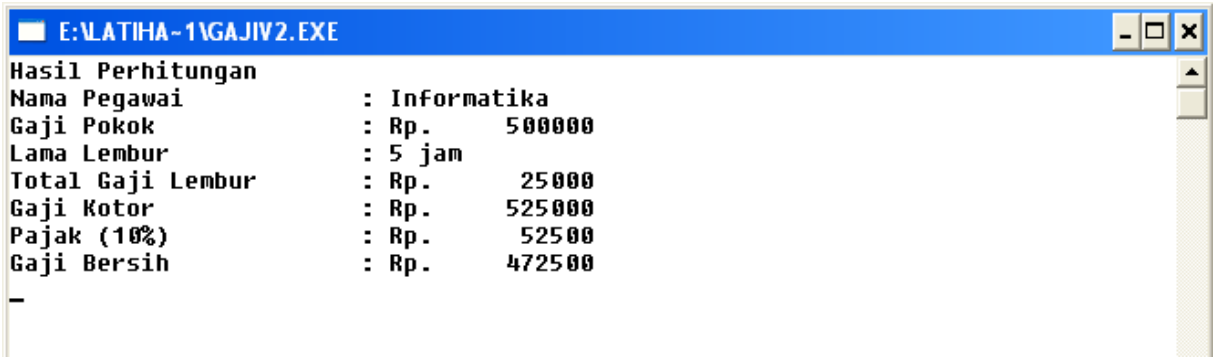
Program ke-1 (tanpa memformat tampilan data)

```
#include <stdio.h>
#include <conio.h>
main()
{
int jamlembur;
long int gajipokok=500000,gajikotor,totalgajilembur;
float pajak,gajibersih;
char nama[50];
printf("Nama Pegawai : ");gets(nama);
printf("Lama Lembur : ");scanf("%i",&jamlembur);
totalgajilembur=(long int)5000*jamlembur;
gajikotor=gajipokok+totalgajilembur;
pajak=0.1*gajikotor;
gajibersih=gajikotor-pajak;
clrscr();
printf("Hasil Perhitungan\n");
printf("Nama Pegawai : %s\n",nama);
printf("Gaji Pokok : Rp. %li\n",gajipokok);
printf("Lama Lembur : %i jam\n",jamlembur);
printf("Total Gaji Lembur : Rp. %li\n",totalgajilembur);
printf("Gaji Kotor : Rp. %li\n",gajikotor);
printf("Pajak (10%%) : Rp. %f\n",pajak);
printf("Gaji Bersih : Rp. %f\n",gajibersih);
getch();
return 0;
}
```



Program ke-2 (dengan memformat tampilan data)

```
#include <stdio.h>
#include <conio.h>
main()
{
     int jamlembur;
     long int gajipokok=500000,gajikotor,totalgajilembur;
     float pajak,gajibersih;
     char nama[50];
     clrscr();
     printf("Nama Pegawai : ");gets(nama);
     printf("Lama Lembur : ");scanf("%i",&jamlembur);
     totalgajilembur=(long int)5000*jamlembur;
     gajikotor=gajipokok+totalgajilembur;
     pajak=0.1*gajikotor;
     gajibersih=gajikotor-pajak;
     clrscr();
     printf("Hasil Perhitungan\n");
     printf("Nama Pegawai          : %s\n",nama);
     printf("Gaji Pokok            : Rp. %10li\n",gajipokok);
     printf("Lama Lembur           : %i jam\n",jamlembur);
     printf("Total Gaji Lembur     : Rp. %10li\n",totalgajilembur);
     printf("Gaji Kotor            : Rp. %10li\n",gajikotor);
     printf("Pajak (10%%)           : Rp. %10.0f\n",pajak);
     printf("Gaji Bersih           : Rp. %10.0f\n",gajibersih);
     getch();
     return 0;
}
```


E:\LATIHA~1\GAJIV2.EXE
Hasil Perhitungan
Nama Pegawai : Informatika
Gaji Pokok : Rp. 500000
Lama Lembur : 5 jam
Total Gaji Lembur : Rp. 25000
Gaji Kotor : Rp. 525000
Pajak (10%) : Rp. 52500
Gaji Bersih : Rp. 472500

TUGAS !

1. Contoh Kasus :

   Di suatu perusahaan, data penggajian dihitung dengan ketentuan sebagai berikut :

   Gaji Pokok : Rp. 5.000.000

   Gaji Lembur/jam : Rp. 5.000

   Total Gaji Lembur : Lama Lembur * Gaji Lembur/jam

   Gaji Kotor : Gaji Pokok + Total Gaji Lembur

   Pajak : 10%

   Gaji Bersih : Gaji Kotor - Pajak

   Data yang diinputkan adalah : Nama Pegawai, Lama Lembur.

   Buat program untuk menghitung besarnya gaji bersih dengan ketentuan diatas,,

2. Buat program untuk menghitung Luas dan Keliling Bujur Sangkar dengan panjang sisi diinputkan dari keyboard.

3. Buatlah program untuk mengkonversi suhu dari Celcius ke Fahrenheit dengan rumus

   F = C * 1.8+ 32

   Input : Suhu dalam celcius
   Output : Suhu dalam Fahrenheit