

METODE INPUT DATA

C-Project #1

Kemampuan Prasyarat:

Untuk mempelajari metode input data ini, Anda harus sudah paham dengan pemrograman dasar dalam bahasa C.

Struktur program
Input Output
Perintah if
Perintah for, while,
dan do ... while
Operator
Tipe data

PENDAHULUAN

Ada banyak cara untuk input data di dalam bahasa pemrograman C, mulai dari tipe data dasar hingga tipe data terstruktur.

Tipe data dasar, adalah tipe data bawaan yang langsung tersedia di C untuk merepresentasikan nilai sederhana. Contohnya: **int** (integer), **float**, **double**, dan **char**.

Tipe data terstruktur adalah tipe data yang memungkinkan pengelompokan data menjadi bentuk yang lebih kompleks. Contoh **array** untuk pengelompokan data dengan tipe data yang sama dan **structure** jika data yang dikelompokkan memiliki tipe data yang berbeda.

BAGAIMANA CARA INPUT DATA DALAM C ?

Disamping adalah perintah yang dapat kita gunakan untuk memasukkan data di dalam program C.

- **scanf**
- **fgets**
- **fscanf**
- **sscanf**
- **strtok**
- **format specifier:**
 - **%d** : integer
 - **%ld** : long integer
 - **%f** : float
 - **%lf** : double
 - **%s** : string
 - **%c** : character

SCANF

Ini adalah perintah dasar dalam pemrograman C dan dapat digunakan untuk semua tipe data dasar.

Format perintah scanf : **scanf("format_specifier",&nama_variabel);**

Contoh:

```
scanf("%d", &angkaInteger);  
scanf("%ld", &angkaLongInteger);  
scanf("%u", &angkaUnsigned);  
scanf("%f", &angkaFloat);  
scanf("%lf", &angkaDouble);  
scanf("%c",&nilaiHuruf);  
char nama[30]; scanf("%s",nama);
```

INPUT BEBERAPA DATA SEKALIGUS

Di dalam program C,
beberapa input data dapat
langsung dibaca sekaligus:

```
scanf("%d %d %f",  
&angka1, &angka2, &nilai);
```

FORMAT INPUT:

10 20 57.5

*gunakan pemisah tanda “ “ spasi
antar angka input*

INPUT DATA TERSTRUKTUR

Input data array 1 dimensi. Contoh array 1 dimensi:

```
int arrNilai[5]={10,20,30,40,50};
```

Pada contoh diatas, angka 5 di dalam kurung kotak adalah banyak data dalam variabel array arrNilai. Posisi data di dalam variabel array arrNilai ini disebut **index** dan dimulai dari index 0.

Data di dalam arrNilai dimasukkan menggunakan perulangan. Contoh disamping menggunakan perulangan for.

```
int arrNilai[5];  
for (int i=0; i<5; i++){  
    scanf("%d",&arrNilai[i]);  
}
```

Penggalan program diatas juga dapat diganti dengan perulangan while.

INPUT DATA TERSTRUKTUR

Input data array 2 dimensi, yang memiliki index baris dan index kolom. Contoh: `int arrMatriks[2][2]={1,2,3,4};`

Pada contoh diatas, angka 2 di dalam kurung kotak menunjukkan arrMatriks memiliki 2 baris dan 2 kolom. Pada baris pertama (index 0) terdapat 2 data (kolom), yaitu 1 dan 2. Sedang baris kedua (index 1) terdapat 2 data (kolom) juga, yaitu 3 dan 4.

```
int arrMatriks[2][2];
for (int baris=0; baris<2; baris++){
    for (int kolom=0; kolom<2; kolom++){
        scanf("%d",&arrMatriks[baris][kolom]);
    }
}
```

Pada penggalan program diatas terdapat 2 kali perulangan, yaitu perulangan index baris dan perulangan index kolom.

INPUT DATA STRING SCANF

Ada beberapa metode yang dapat kita gunakan untuk memasukkan data string. Berikut pembahasan berdasarkan format input.

Program memasukkan nama siswa dan umur.

FORMAT INPUT:

Euginia

18

Lalita Euginia

19

Dalam bahasa C:

```
char nama[20];
```

```
int umur;
```

```
scanf("%s",nama);
```

```
scanf("%d",&umur);
```

Penggalan program diatas jika dijalankan dan diberi input nama "Lalita Euginia" yang terdiri dari 2 kata dengan jeda spasi, maka akan terjadi kesalahan

INPUT DATA STRING SCANF

Ada beberapa metode yang dapat kita gunakan untuk memasukkan data string. Berikut pembahasan berdasarkan format input.

Program memasukkan nama siswa dan umur.

FORMAT INPUT:

Euginia

18

Lalita Euginia

19

Program sebelumnya dapat diperbaiki dengan mengganti format specifier

```
char nama[20];
```

```
int umur;
```

```
scanf("%[^\\n]",nama);
```

```
scanf("%d",&umur);
```

Dengan format specifier di atas maka program akan membaca hingga menemukan karakter new line atau \\n

INPUT DATA STRING SCANF

Perintah scanf untuk memasukkan data string yang mengandung spasi dapat menggunakan format specifier **%[^\n]**.

*Program memasukkan
nama siswa dan hobi.*

FORMAT INPUT:

Eugenia

Bulutangkis

Lalita Eugenia

Sepak bola

Dalam bahasa C:

```
char nama[20];
```

```
char hobi[20];
```

```
scanf("%[^\n]",nama);
```

```
scanf("%[^\n]",hobi);
```

Pada penggalan program diatas, nama dan hobi dapat dibaca meskipun mengandung karakter spasi. Namun pada input nama, akan meninggalkan new line pada stdin, sehingga data hobi tidak dapat dimasukkan.

INPUT DATA STRING SCANF

Perintah `scanf("%[^\\n]",nama);` meninggalkan karakter new line pada buffer atau `stdin`.

Program memasukkan nama siswa dan hobi.

FORMAT INPUT:

Euginia

Bulutangkis

Lalita Euginia

Sepak bola

Gunakan perintah `getchar()` untuk menghapus karakter new line

```
char nama[20];
```

```
char hobi[20];
```

```
scanf("%[^\\n]",nama);
```

```
getchar();
```

```
scanf("%[^\\n]",hobi);
```

INPUT DATA STRING FGETS

Ada beberapa metode yang dapat kita gunakan untuk memasukkan data string. Berikut pembahasan berdasarkan format input.

*Program memasukkan
nama siswa dan hobi.*

FORMAT INPUT:

Euginia

Bulutangkis

Lalita Euginia

Sepak bola

Dalam bahasa C:

```
char nama[20];
```

```
char hobi[20];
```

```
fgets(nama, sizeof(nama), stdin);
```

```
fgets(hobi, sizeof(hobi), stdin);
```

```
printf("%s punya hobi: %s",nama,hobi);
```

INPUT DATA STRING FGETS

Perintah `fgets()` dapat membaca karakter spasi, namun memiliki kelemahan yaitu menyimpan karakter new line pada buffer.

Program memasukkan nama siswa dan hobi.

FORMAT INPUT:

Lalita Euginia
Sepak bola

Karakter new line pada nama dapat dihilangkan dengan perintah disamping.

Dalam bahasa C, hilangkan new line dalam buffer dengan perintah:

```
char nama[20];  
char hobi[20];  
fgets(nama, sizeof(nama), stdin);  
nama[strlen(nama)-1]=' ';  
fgets(hobi, sizeof(hobi), stdin);  
printf("%s punya hobi: %s",nama,hobi);
```

INPUT DATA DARI STRING (TEKS)

Perintah `sscanf()` digunakan untuk membaca beberapa data dari sebuah teks atau string.

- Baca data string/ teks.
- Baca setiap data pada string/ teks yang dipisahkan dengan karakter tertentu.

FORMAT INPUT:

Lalita Euginia,95,Lulus

```
char data[20];  
scanf("%[^\\n]",data);  
char nama[20];  
int nilai;  
char keterangan[20];  
sscanf(data,"%[^,],%d,%s",  
        nama,&nilai,keterangan);
```

INPUT DATA DARI STRING (TEKS)

Metode lain untuk membaca beberapa data dari teks adalah menggunakan token

- Buat variabel pointer token,
- Gunakan strtok untuk mengisi dengan bagian dari baris input hingga delimiter,
- Jika token != NULL, isi variabel data dengan token.

CONTOH PROGRAM C:

```
fgets(input, sizeof(input), stdin);  
char *tokenData = strtok(input, "$");  
if (tokenData != NULL) strcpy(nama, tokenData);  
tokenData = strtok(NULL, "$");  
if (tokenData != NULL) nilai = atoi(tokenData);
```


INPUT DATA DARI FILE TEXT

Data yang dimasukkan ke dalam program, dapat dimasukkan secara manual, tetapi juga dapat dimasukkan melalui file teks yang terdiri dari kumpulan data.

Tahapan input data dari file teks:

1. Baca file. Deklarasi structure dalam array.
2. Gunakan perintah fgets untuk melewati header jika ada.
3. Lakukan perulangan selama masih bisa membaca baris data:
 - a. Hilangkan new line dari buffer
 - b. Gunakan tokenisasi untuk memindah data, ulangi sebanyak kolom data yang ada.
 - c. Tambah index structure array.

CONTOH PROGRAM:

```
FILE *fileData=fopen("data.txt","r");
if (fileData==NULL){
    perror("File tidak dapat dibuka.\n");
    return 1;
}
apartment kosan[5000];
char inputBaris[256];
int index=0;
fgets(inputBaris, sizeof(inputBaris), fileData); //melewati header
while (fgets(inputBaris, sizeof(inputBaris), fileData)){
    inputBaris[strcspn(inputBaris, "\n")] = 0; //menghilangkan new line
    char *tokenData = strtok(inputBaris, ",");
```

CONTOH PROGRAM:

```
if (tokenData != NULL) strcpy(kosan[index].loc1, tokenData);  
tokenData = strtok(NULL, ",");  
if (tokenData != NULL) strcpy(kosan[index].loc2, tokenData);  
index++;  
}
```

**SEMANGAT
NGODING**

