



Sequence

(TIB26 – ALGORITMA PEMROGRAMAN)

Pertemuan 5, 6

Sub-CPMK

- Mahasiswa mampu menambahkan input, output dalam suatu urutan algoritma (C3, A3)

Materi

- Sequence
- Input
- Output
- Argumen

Perhatian

- Tidak disarankan copy-paste kode program dari presentasi ini, karena ada beberapa symbol yang dianggap sebagai Unicode oleh editor yang anda gunakan, sehingga akan dianggap sebagai symbol yang salah oleh compiler, sebaiknya diketik ulang saja



1.

Sequence

1.1 Konstruksi Dasar Algoritma

algoritma dibangun dari tiga konstruksi atau struktur dasar, yaitu:

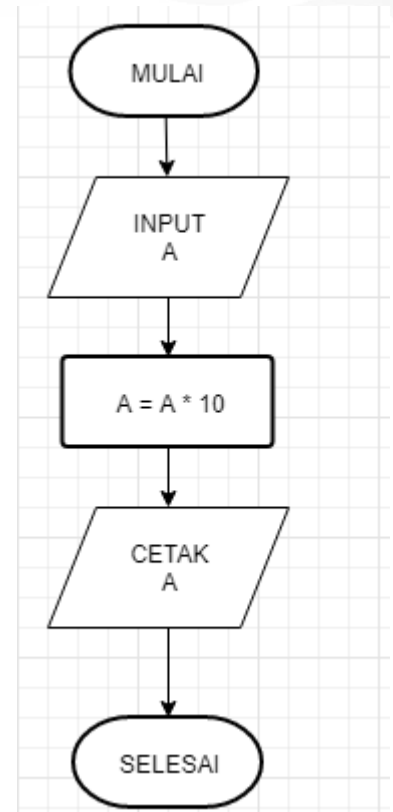
- ***Sequence*** (runtunan/berurutan).
- ***Selection*** (pemilihan/pencabangan).
- ***Repetition*** (pengulangan/*loop*).

1.2 Sequence

- Sequence merupakan sequence / runtutan perintah
- Algoritma merupakan runtutan program yang dilakukan perintah per perintah, dikerjakan satu persatu-satu
- Perintah berikutnya akan menunggu perintah sebelumnya selesai dikerjakan terlebih dahulu
- Secara normal, urutan perintah dilakukan dari atas ke bawah

1.3 Contoh Sequence

- Sequence di samping akan dikerjakan satu-persatu mulai dari Input A, dilanjutkan dengan menghitung $A = A + 10$ dan terakhir mencetak A
- Urutan dilakukan sebagai berikut:
 - Pernyataan Input A
 - Hitung $A = A + 10$
 - Cetak A



1.4 Contoh Sequence pada Pseudo-code

PROGRAM Contoh

DEKLARASI:

A : Integer;

ALGORITMA:

input(A);

A = A +10;

cetak (A);



2.

Input

2.1 Input

- Suatu algoritma akan mendapat input nilai yang akan diproses pada algoritma.
- Nilai input akan disimpan pada variabel
- Input bisa didapatkan dari inisialisasi, statement ataupun perintah input melalui keyboard ataupun perangkat input lainnya, argumen, parameter maupun penugasan

2.2 Perintah Masukan

Beberapa perintah masukan pada C antara lain:

- `cin>>variable;`
- `scanf()`
- `gets()`
- `getch()`
- `getche()`

2.3 cin

- cin merupakan keyword pada c yang dipergunakan untuk input, untuk mengambil input lewat keyboard
- Syntax:
`cin>>variable;`
- perhatikan arah operator >> menuju ke variabelnya
- Pembacaan data oleh cin adalah per nilai datanya
- Tidak dapat membaca spasi (adanya spasi malah dianggap sebagai pembacaan input data kedua)

2.3.1 Contoh cin

```
char nama[10];  
int nilai;
```

```
cout<<"Ketik Nama: ";  
cin>>nama;  
Cout<<"Ketik Nilai: ";  
cin>>nilai;
```

```
Cout<<"Nilai dari "<<nama<<" adalah "<<nilai;
```

2.4 scanf()

- scanf() adalah fungsi yang digunakan untuk input data melalui keyboard.
- fungsi scanf() berada di file header "stdio.h" yang merupakan library milik bahasa C, maka dapat juga digunakan pada C++.
- Syntax: scanf(format, &nama-variabel);
Keterangan : simbol "&" adalah pointer yang dipakai untuk menunjuk kealamat variabel memori yang dituju

2.4.1 Format scanf()

Tipe Data	Penentuan Format untuk printf()
Integer	%d
Floating Point	
- Bentuk desimal	%f
- Bentuk Berpangkat	%e
- Bentuk desimal dan angka	%g
Double Precision	%lf
Unsigned Integer	%u
Long Integer	%ld
Long Unsigned Integer	%lu
Unsigned Hexadecimal Integer	%x
Unsigned Octal Integer	%o
Charackter	%c
String	%s

2.4.2 Contoh scanf()

```
#include <stdio.h>
using namespace std;
int main()
{
    char nama[50];
    int nilai;

    printf("Masukkan Nama   : ");
    scanf("%s", &nama);
    printf("Masukkan Nilai  : ");
    scanf("%d", &nilai);
    printf("\nMahasiswa atas nama %s nilainya %d",nama,nilai);
    return 0;
}
```

2.5 gets()

- Function gets() merupakan perintah untuk memasukkan data string ke dalam variable,
- Perintah ini dirancang untuk dapat menerima adanya spasi pada input keyboard dan dijadikan sebagai satu kesatuan input
- fungsi gets() berada di file header "stdio.h" yang merupakan library milik bahasa C, maka dapat juga digunakan pada C++.
- Syntax: gets(nama_variabel_string);

2.5.1 Contoh gets()

```
#include <stdio.h>
using namespace std;

int main(){
    char nama[50];
    int nilai;

    printf("Masukkan Nama   : ");
    gets(nama);
    printf("Masukkan Nilai  : ");
    scanf("%d", &nilai);
    printf("\nMahasiswa bernama %s nilainya %d",nama,nilai);
    return 0;
}
```

2.6 getch()

- getch() merupakan fungsi yang diperuntukan membaca sebuah karakter.
- Terdapat pada file header <conio.h>
- Berbeda dengan cin, scanf() dan gets() yang menunggu penekanan enter, getch() tidak memerlukan penekanan enter, setelah sebuah karakter diinput lewat keyboard, maka karakter tersebut langsung di simpan pada variable
- Karakter yang diinputkan tidak akan ditampilkan pada layar

2.6 getch() (lanj...)

- Nilai balikan (return value) dari function getch() adalah karakter yang diinputkan dari keyboard.
- Syntax: `variable_char = getch();`

2.6.1 Contoh getch()

```
#include <stdio.h>
#include <conio.h>

using namespace std;

int main(){
    char kar;

    printf("Tekan Tombol Keyboard : ");
    kar = getch();
    printf("\n\nAnda Menekan tombol : %c", kar);
    return 0;
}
```

2.7 getch()

- Jika kita menginginkan karakter yang diinput di tampilkan, maka kita dapat mempergunakan function getch()
- Funtion ini terdapat pada file header <conio.h>
- Penggunaan sama seperti getch()
- Nilai balikan (return value) dari function getch() adalah karakter yang diinputkan dari keyboard.
- Syntax: `variable_char = getch();`

2.7.1 Contoh getch()

```
#include <stdio.h>
#include <conio.h>

using namespace std;

int main(){
    char kar;

    printf("Tekan Tombol Keyboard : ");
    kar = getch();
    printf("\n\nAnda Menekan tombol : %c", kar);
    return 0;
}
```




3.

Output

3.1 cout()

- cout merupakan fungsi yang digunakan untuk mencetak output ke peralatan standar (screen) / layar console program.
- Terdapat pada file header <iostream.h>
- Syntax: cout<<data_atau_variable;
- Perhatikan arah operator <<
- Dalam satu perintah cout dapat mencetak lebih dari satu data hanya dengan menambahkan operator <<
- Syntax: cout<<data_atau_variable1<< data_atau_variable2;

3.1.1 Perpindahan Baris

- Meskipun ditulis pada kode terpisah, teks yang dicetak dengan `cout` akan ditulis pada baris yang sama dengan keluaran dari `cout` di atasnya, sehingga memerlukan perintah untuk pindah baris pada penggunaan `cout`
- Contoh:

```
cout<<"baris pertama";  
cout<<"baris kedua";
```
- Hasil eksekusi akan menampilkan

```
baris pertamabaris kedua
```

3.1.2 Perpindahan Baris dengan endl

- Output di layar dapat dipisahkan pada baris yang berbeda dengan menambahkan perintah endl

- Contoh:

```
cout<<"baris pertama";
```

```
cout<<endl;
```

```
cout<<"baris kedua";
```

- Hasil keluaran:

```
baris pertama
```

```
baris kedua
```

3.1.2 Perpindahan Baris dengan endl (lanj...)

- Perintah endl dapat digabungkan pada perintah dalam satu baris

- Contoh:

```
cout<<"baris pertama"<<endl<<"baris kedua";
```

- Hasil keluaran:

```
baris pertama
```

```
baris kedua
```

3.1.3 Perpindahan Baris dengan \n

- Output di layar dapat dipisahkan pada baris yang berbeda dengan menambahkan tesk “\n”

- Contoh:

```
cout<<"baris pertama";
```

```
cout<<"\n";
```

```
cout<<"baris kedua";
```

- Hasil keluaran:

```
baris pertama
```

```
baris kedua
```

3.2 printf()

- Function printf merupakan fungsi yang digunakan untuk mencetak output
- Terdapat pada file header <stdio.h>
- **Syntax:** `printf("pernyataan %penentuan_format pernyataan", argumen);`
- **Contoh:**

```
nilai = 98;  
printf("Nilai UAS : %d point", nilai);
```
- Akan menghasilkan output
Nilai UAS : 98 point

3.2.1 Format Penugasan printf()

Tipe Data	Penentuan Format untuk printf()
Integer	%d
Floating Point	
- Bentuk desimal	%f
- Bentuk Berpangkat	%e
- Bentuk desimal dan angka	%g
Double Precision	%lf
Unsigned Integer	%u
Long Integer	%ld
Long Unsigned Integer	%lu
Unsigned Hexadecimal	%x
Integer	
Unsigned Octal Integer	%o
Charackter	%c
String	%s

3.2.2 Mencetak lebih dari satu data dengan printf()

- Function printf dapat mencetak lebih dari satu data dalam satu function selama disediakan penentu format sesuai jumlah argumen
- **Syntax:** `printf("pernyataan %penentuan_format1 pernyataan %penentuan_format2 pernyataan %penentuan_format3", argumen1, argumen2, argumen3);`
- **Contoh:**

```
nilai = 89;  
nama = "Nohara Shinosuke";  
Printf("Nilai UAS mhs %s adalah %d point", nama, nilai);
```
- **Keluaran dari perintah tsb adalah:**
Nilai UAS mhs **Nohara Shinosuke** adalah **89** point



4.

Argumen

4.1 Argumen

- Argumen dapat ditambahkan pada fungsi main(), sehingga pemanggilan program untuk dieksekusi dapat disertai argument sebagai input
- Contoh dari command line kita memberi perintah sbb
`D:\>contoh 19 kiri 55`
- Maka program contoh tersebut memiliki 4 argument yang terdiri dari “contoh”, dan parameter “19”, “kiri” dan “55”
- Nama program dianggap argumen ke 0

4.2 Argumen (lanj...)

- Argumen dapat diwujudkan dengan parameter pada function main
- Umumnya argument dibaca sebagai string karena itu pada function main mempergunakan parameter bertipe pointer dari array char.
- Parameter pertama merupakan informasi jumlah argumen

4.3 Argumen (lanj...)

- Misalkan Contoh program “contoh.cpp”:

```
int main(int argc, char *argv[])  
{  
    printf("Jumlah Argumen %d", argc);  
    printf("Argumen ke 1 %s", argv[1]);  
    printf("Argumen ke 2 %s", argv[2]);  
    printf("Argumen ke 3 %s", argv[3]);  
}
```

- Maka pemanggilan
D:\>contoh 19 kiri 55
- Akan menghasilkan keluaran
Jumlah Argumen 4
Argumen ke 1 19
Argumen ke 2 kiri
Argumen ke 3 55

4.2 Contoh Argumen

```
int main(int argc, char *argv[])
{
    int i;
    for(i=1;i<argc;i++)
    {
        printf("Argumen ke &d adalah %s", i, argv[i]);
    }
    return 0;
}
```



5.

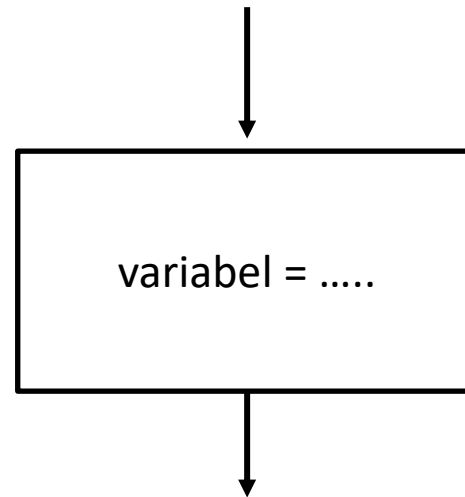
Penugasan

5.1 Penugasan

- Data dimasukkan ke dalam variable dapat melalui perintah input, argument, parameter, maupun penugasan.
- Penugasan memanipulasi data dan menyimpannya ke dalam variabel
- Penugasan data dilakukan mempergunakan operator aritmatika ataupun operator logika.

5.2 Notasi Flowchart Penugasan

- Notasi flowchart mempergunakan notasi untuk proses berupa bangun segi empat
- Di dalamnya diberi rumus yang merupakan penugasan



5.3 Contoh penugasan

- `A = 10;` //mengisi variabel A dengan nilai 10
- `B = A + 5;` //mengisi variabel B dengan penjumlahan data pada variabel A + 5

Ringkasan

- Algoritma merupakan runtutan program yang dilakukan perintah per perintah, dikerjakan satu persatu-satu
- Suatu algoritma akan mendapat input nilai yang akan diproses pada algoritma.
- Nilai input akan disimpan pada variabel
- Input bisa didapatkan dari inisialisasi, statement ataupun perintah input melalui keyboard ataupun perangkat input lainnya, argumen, parameter maupun penugasan
- Argumen dapat ditambahkan pada fungsi main(), sehingga pemanggilan program untuk dieksekusi dapat disertai argument sebagai input
- Penugasan memanipulasi data dan menyimpannya ke dalam variabel



Terimakasih

TUHAN Memberkati Anda

Teady Matius Surya Mulyana (tmulyana@bundamulia.ac.id)