



IPB University
— Bogor Indonesia —

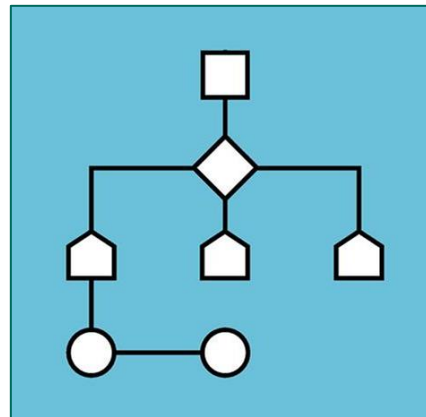
KOM120B #4

Algoritme Perulangan

Tim Pengajar

KOM120B – Algoritme dan Dasar Pemrograman

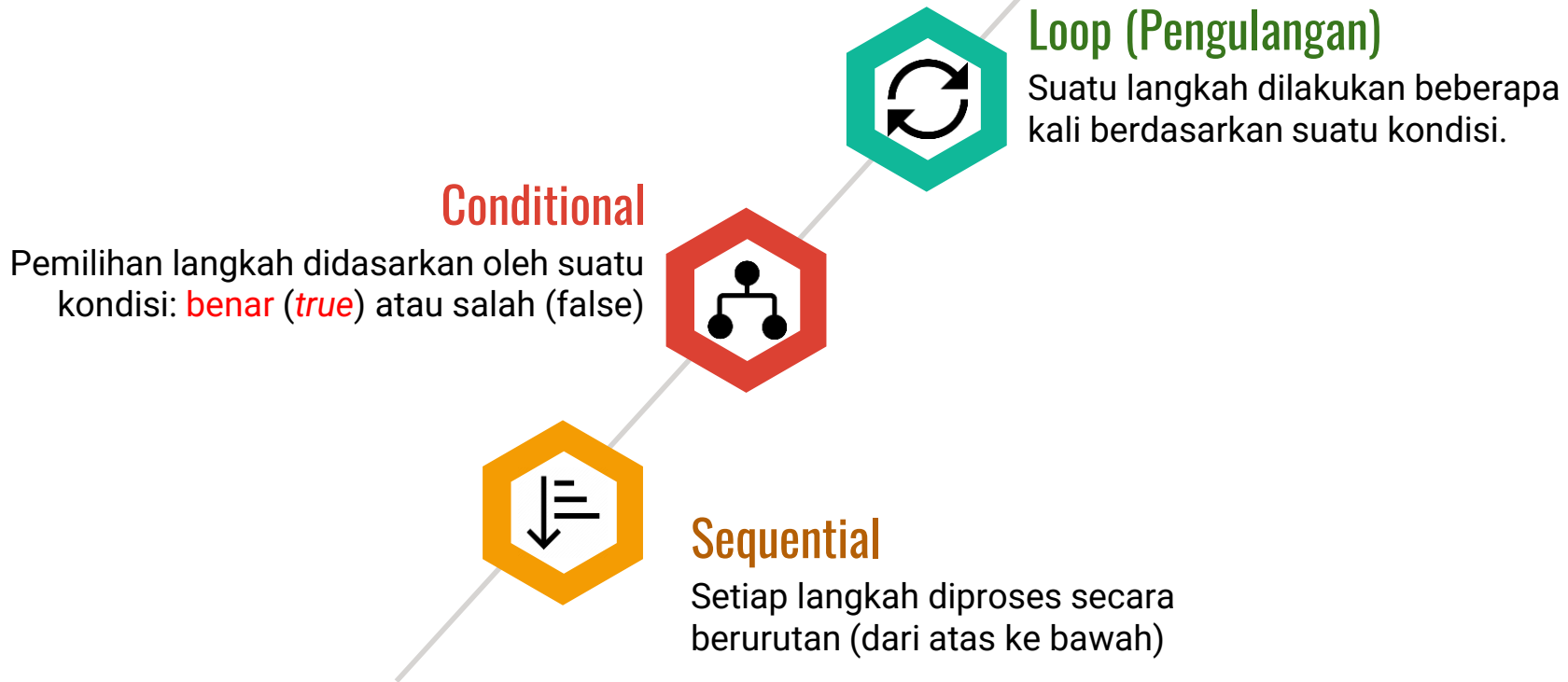
Departemen Ilmu Komputer - FMIPA



```
7 string sInput;  
8 int iLength, iN;  
9 double dblTemp;  
10 bool again = true;  
21  
22 while (again) {  
23     iN = -1;  
24     again = false;  
25     getline(cin, sInput);  
26     system("cls");  
27     stringstream(sInput) >> dblTemp;  
28     iLength = sInput.length();  
29     if (iLength < 4) {  
30         again = true;  
31         continue;  
32     } else if (sInput[iLength - 3] != '.') {  
33         again = true;  
34         continue;  
35     } while (++iN < iLength) {  
36         if (isdigit(sInput[iN])) {  
37             continue;  
38         } else if (iN == (iLength - 3)) {  
39             continue;  
40         }  
41     }  
42 }
```



Struktur Algoritme



Struktur Perulangan

- Dalam struktur perulangan (loop), sebuah pernyataan mengizinkan program untuk mengeksekusi pernyataan berulang-ulang selama memenuhi kondisi BENAR.
- Kebanyakan program melibatkan algoritme perulangan
- Terdapat 2 tipe algoritme perulangan, yaitu:
 - *counter-controlled repetition* → mengetahui secara tepat berapa kali perulangan dilakukan, menggunakan variabel counter
 - *sentinel-controlled repetition* → tidak tahu berapa kali perulangan akan dilakukan, menggunakan variabel kontrol.

Contoh Kasus

- Buat algoritme untuk menghitung jumlah dari N bilangan bulat yang dimasukkan.

Contoh input:

5

10 50 30 20 40

Contoh output:

150

counter-controlled repetition

- Buat algoritme untuk menentukan rata-rata tinggi (bilangan bulat, cm) dari beberapa tanaman jagung yang diinputkan. Input data berhenti dengan tanda nilai -1.

Contoh input:

10 50 30 20 40 -1

10 15 -1

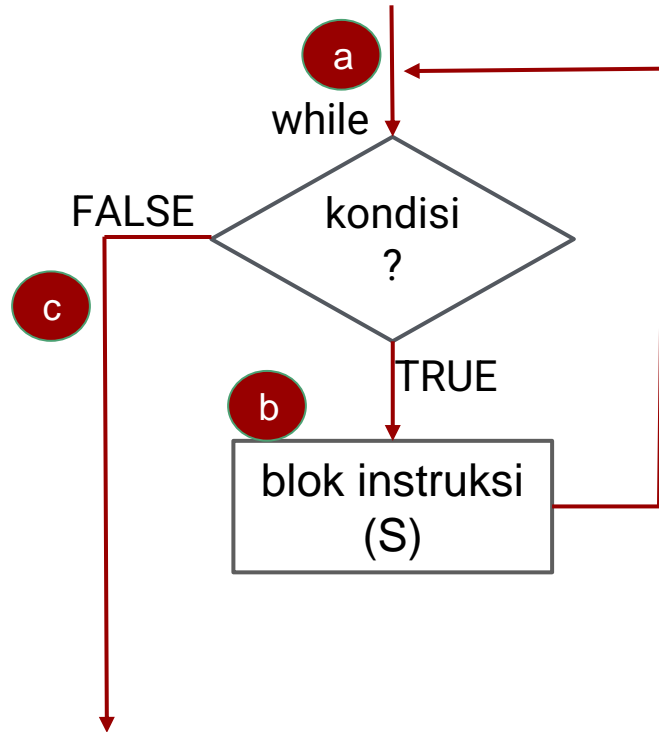
Contoh output:

5 30.00

2 12.50

sentinel-controlled repetition

Perulangan WHILE



- Sintaks C:

```
while (kondisi)
    S
```

- Tahapan proses:

- a. Masuk while, cek kondisi
- b. Jika kondisi TRUE, lakukan S. Kembali ke (a)
- c. Jika kondisi FALSE, keluar dari perulangan (lanjut ke instruksi berikutnya)

Contoh #1

Buat program untuk menuliskan teks "Dies IPB" sebanyak N baris ($1 \leq N \leq 2$ Milyar).

Contoh Input:

5

Contoh Output:

Dies IPB
Dies IPB
Dies IPB
Dies IPB
Dies IPB

```
read(n) ;  
counter=0;  
while (counter<n)  
{  
    print("Dies IPB");  
    counter++;  
}
```

Contoh #1 :: Program C

```
read(n);  
counter=0;  
while (counter<n)  
{  
    print("Dies IPB");  
    counter++;  
}
```

```
#include <stdio.h>  
int main()  
{  
    int n;  
    scanf("%d", &n);  
    counter=0;  
    while (counter<n)  
    {  
        printf("Dies IPB\n");  
        counter++;    // counter=counter+1  
    }  
    return 0;  
}
```

Contoh #2

Buat algoritme untuk menghitung jumlah dari N bilangan bulat yang dimasukkan, ($1 \leq N \leq 2$ Milyar).

Contoh input:

5

10 50 30 20 40

Contoh output:

150

```
read(n);  
counter=0;  
jumlah=0;  
while (counter<n)  
{  
    read(dt);  
    jumlah=jumlah+dt;  
    counter++;  
}  
print(jumlah);
```


Contoh #2 :: Program C

```
read(n);
counter=0;
jumlah=0;
while (counter<n)
{
    read(dt);
    jumlah=jumlah+dt;
    counter++;
}
print(jumlah);
```

```
#include <stdio.h>
int main()
{
    int n, counter, jumlah, dt;
    scanf("%d", &n);
    counter=0; jumlah=0;
    while (counter<n)
    {
        scanf("%d", &dt);
        jumlah=jumlah+dt;
        counter++;
    }
    print(jumlah);
    return 0;
}
```

Contoh #3

Buat algoritme untuk menentukan rata-rata tinggi (bilangan bulat, cm) dari beberapa tanaman jagung yang diinputkan. Input data berhenti dengan tanda nilai -1.

Contoh input:

10 50 30 20 40 -1

Contoh output:

5 30.00

```
n=sum=0;
read(nilai);
while (nilai!=-1)
{
    n++;
    sum=sum+nilai;
    read(nilai);
}
rataan=sum/n;
print(n, rataan);
```

Contoh #3

```
n=sum=0;
read(nilai);
while (nilai!=-1)
{
    n++;
    sum=sum+nilai;
    read(nilai);
}
rataan=sum/n;
print(n, rataan);
```

```
#include <stdio.h>
int main()
{
    int n,sum,nilai;
    float rataan;
    n=sum=0;
    scanf("%d", &nilai);
    while (nilai!=-1)
    {
        n++;
        sum=sum+nilai;
        scanf("%d", &nilai);
    }
    rataan=1.0*sum/n;
    printf("%d %.2f\n", n, rataan);
    return 0;
}
```

INCREMENT dan DECREMENT

- Ada dua tipe:
 - Pre-increment/decrement, misalnya ++c, --d
Nilai ekspresi adalah nilai setelah dilakukan operasi increment/decrement
 - Post-increment/decrement, misalnya c++, d--
Nilai ekspresi adalah nilai sebelum dilakukan operasi increment/decrement
- Contoh (misal nilai awal a=5, b=8):
 - Nilai ekspresi a++ adalah 5, tetapi nilai a setelahnya menjadi 6.
 - Nilai ekspresi ++b adalah 9, dan nilai b setelahnya juga 9

ASSIGNMENT OPERATOR

- Assignment operator (operator penugasan) dilambangkan dengan tanda "=" untuk memberikan **nilai** sebelah kanan ke **variabel** sebelah kiri.

- Contoh:

```
int a=5,b,c;  
b=a++;  
c=a<5;  
printf("%d %d %d",a,b,c);
```

COMPOUND OPERATOR

- Menggabungkan operator penugasan dengan operator aritmatika (+, -, *, /, %)
- Contoh :
 - `a+=5;` `// a=a+5`
 - `b%=10;` `// b=b%10;`

Latihan #1

Tuliskan output program berikut:

```
#include <stdio.h>
int main()
{
    int a = 10;
    printf("\n%d", a += 3);
    printf("\n%d", a -= 3);
    printf("\n%d", a *= 3);
    printf("\n%d", a /= 3);
    printf("\n%d", a %= 3);
    printf("\n%d", a); printf("\n%d", ++a);
    printf("\n%d", a); printf("\n%d", a++);
    printf("\n%d", a); printf("\n%d", --a);
    printf("\n%d", a); printf("\n%d", a--);
    printf("\n%d\n", a);
    return 0;
}
```

Latihan #2

Tuliskan output program berikut jika diberikan input data: 25000000 171

```
#include <stdio.h>
int main()
{
    int a,b,res;
    scanf("%d %d",&a,&b);
    while (b!=0) {
        res=a%b;
        a=b; b=res;
    }
    printf("%d\n", a);
    return 0;
}
```


Latihan #3

Buat program untuk menentukan nilai terkecil dari N bilangan bulat,
($1 \leq N \leq 2$ Milyar)

Contoh input:

5

100 50 30 20 40

Contoh output:

20