

Algoritme dan Pemrograman

Kuliah #5

- Compound statement
- Kontrol Program : WHILE, DO..WHILE, FOR

Loop control structures

- Program akan **mengulang** satu atau lebih statement untuk diproses atau tidak diproses berdasarkan kondisi tertentu.
- Kondisi dinyatakan sebagai ekspresi.
- Pernyataan yang digunakan disebut loop statement.
- Ada tiga jenis:
 - Pernyataan WHILE
 - Pernyataan DO ... WHILE
 - Pernyataan FOR

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Loop control structures: WHILE

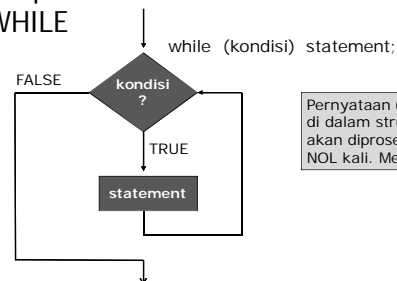
- Implementasi dari pernyataan: lakukan sesuatu berulang-ulang selama kondisi bernilai true (berarti juga hentikan melakukan itu jika kondisi bernilai false).

- Format:

```
while (kondisi)
    simple or compound statement;
```

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Loop control structures: WHILE



Pernyataan (*statements*)
di dalam struktur WHILE
akan diproses minimum
NOL kali. Mengapa?

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Loop control structures: WHILE

- Perhatikan potongan program berikut:

```
int i=0;
while (i<5) {
    printf("%d", i);
    i++;
}
```

- Output program tersebut adalah:
01234

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Loop control structures: WHILE

- Apa output potongan program berikut?

```
int i=0;
while (i<5)
    printf("%d", i++);    01234
```

- Bagaimana dengan potongan program berikut?

```
int i=0;
while (i++<5)
    printf("%d", i);    12345
```

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Latihan

- Dengan menggunakan pernyataan looping WHILE dan jumlah instruksi sesedikit mungkin, tuliskan cuplikan program (bukan program lengkap, hanya instruksi utama) untuk mencetak deret nilai 20, 14, 8, 2, -4, dan -10

Loop control structures:

DO ... WHILE

- Implementasi dari pernyataan: Lakukan sesuatu. Kemudian ulangi sesuatu itu selama kondisi bernilai true
- Berarti memproses pernyataan di dalamnya terlebih dahulu, baru kemudian memeriksa kondisi yang ada.
- Pernyataan akan diproses minimal SATU kali.
- Format:

```
do {
    pernyataan;
} while (kondisi);
```

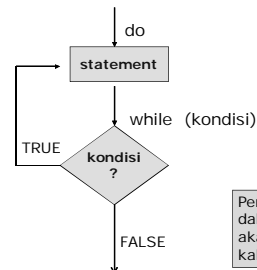
Loop control structures: DO ... WHILE

```
#include <stdio.h>
int main() {
    int i, n=5;
    i=0;
    do {
        printf("%d ", i);
        i++;
    } while (i<n);
    return 0;
}
```

Output program tersebut adalah:

0 1 2 3 4

Loop control structures: DO ... WHILE



Pernyataan (*statements*) di dalam struktur DO ... WHILE akan diproses minimum SATU kali. Mengapa?

Contoh: Program menulis bilangan

```
#include <stdio.h>
int main()
{
    int n;
    scanf("%d", &n);
    do {
        printf("%d", n%10);
        n /= 10;
    } while (n);
    return 0;
}

#include <stdio.h>
int main()
{
    int n;
    scanf("%d", &n);
    while (n){
        printf("%d", n%10);
        n /= 10;
    }
    return 0;
}
```

Jika diberikan input 120, apa output program tsb? 021
Ubah program menggunakan **while**

Loop control structures: FOR

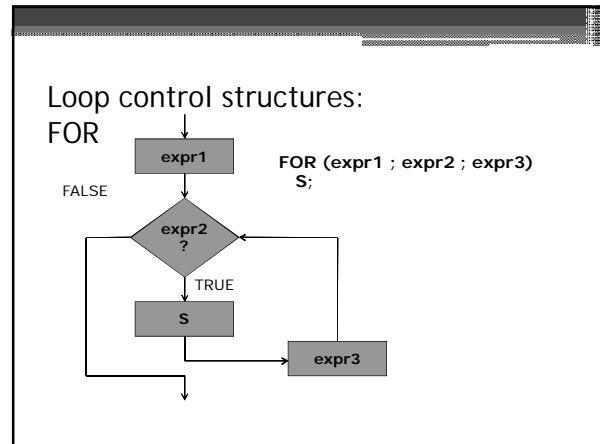
- Pernyataan for umumnya digunakan untuk memproses pernyataan secara berulang-ulang dan kita sebelumnya sudah mengetahui berapa kali pernyataan tersebut akan diulang.
- Contoh:
 - Menuliskan bilangan bulat 1 sampai dengan 5
 - Membaca n bilangan bulat.

Loop control structures:
FOR

- Bentuk umum

```
for (expr1; expr2; expr3) {
    s1;
    s2;
    ...
}
```

- Proses yang dilakukan mengikuti alur berikut:
 - expr1 dieksekusi hanya satu kali sebelum mulai looping.
 - expr2 merupakan ekspresi Boolean. Jika tidak dituliskan, diasumsikan bernilai true.
 - Jika expr2 bernilai false, maka proses looping dihentikan.
 - Jika expr2 bernilai true, maka s1, s2, ... dieksekusi.
 - Setelah eksekusi s1, s2, ..., maka expr3 dieksekusi.
- expr1, expr2, dan expr3 bersifat optional (dapat dituliskan, juga dapat tidak dituliskan).



Pernyataan FOR:
Contoh #1

```
#include <stdio.h>
int main(){
    int i,n = 5;

    for (i=0; i<n; i++) {
        printf("%d", i);
    }
    return 0;
}
```

Apa output program tersebut ? (ikuti diagram FOR pada slide sebelumnya) 01234

Pernyataan FOR - loop backward:
Contoh #2

```
#include <stdio.h>
int main(){
    int i=5;

    for ( ; i; i--) {
        printf("%d", i);
    }
    return 0;
}
```

Apa output program tersebut ? 54321
Bagaimana jika baris for menjadi "for (; i--;)"? 43210

Pernyataan FOR - loop backward:
Contoh #3

```
#include <stdio.h>
int main(){
    int i, n=5, sum=0;

    for (i=1 ; i<=n; sum+=i++);

    printf("%d\n", sum);
    return 0;
}
```

Apa output program tersebut ? 15
Apa yang dilakukannya?
Bagaimana jika menggunakan "for (; i++<n; sum+=i);"?

Pernyataan FOR: Contoh #4

```
#include <stdio.h>
int main()
{
    int i, n, bil, sum=0;
    scanf("%d", &n);
    for (i=1; i<=n; i++) {
        scanf("%d", &bil);
        sum+=bil;
    }
    printf("%d\n", sum);
    return 0;
}
```

Apa output program jika diberi input 3 13 43 32?
Apa yang dilakukan program tersebut? 88

Output?

```
3 2
2 1
1 0
```

...

```
int a, b=3;
for ( ; a=b, b-- ; )
    printf("%d %d\n",a,b);
```

...

Loop control structures: Nested FOR

- Pernyataan for yang bersarang, artinya terdapat pernyataan for lainnya di dalam pernyataan for.
- Contoh, apa output potongan program berikut?

```
int i, j;
for (i=1; i<=2; i++) {
    for (j=1; j<=3; j++) {
        printf("*");
    }
    printf("\n");
}
```

```
***
***
```

Nested FOR: Menulis bilangan bentuk matrik

- Misalkan ingin menuliskan bilangan 1, 2, ... dalam bentuk matriks ukuran 3x4. Setiap baris hanya diakhiri oleh newline.
- Digunakan nested for:

```
#include <stdio.h>

int main() {
    int i, j, c=1;
    for (i=1; i<=3; i++)
        for (j=1; j<=4; j++) {
            printf("%d", c++);
            if (j!=4) printf(" ");
            else printf("\n");
        }
    return 0;
}
```

Nested FOR: Menggunakan variabel sebelumnya

- Perhatikan contoh berikut. Struktur for untuk variabel *j* menggunakan variabel *i*.
- Apa outputnya? Apa yang dilakukan?

```
int i, j, sum;
for (i=1; i<=3; i++) {
    sum=0;
    for (j=1; j<=i; j++)
        sum+=j;
    printf("%d %d\n", i, sum);
}
```

```
1 1
2 3
3 6
```

Nested FOR: Menggunakan variabel sebelumnya

- Apa output potongan program berikut?

```
int i, j, n=4;
for (i=1; i<=n; i++) {
    for (j=i; j<=n; j++)
        printf("*");
    printf("\n");
}
```

```
****
***
**
*
```

Men-trace algoritme

- Men-trace algoritme adalah suatu proses untuk menelusuri langkah demi langkah dari suatu algoritme dan menentukan output akhirnya.
- Dalam pemrograman, proses ini disebut juga sebagai proses debug.
- Tips:
 - Ikuti langkah demi langkah
 - Catat setiap perubahan nilai dari suatu variabel
 - Jika menemukan suatu pola tertentu, maka output akhir dapat langsung ditentukan tanpa mengikuti algoritme sampai selesai (hanya dilakukan oleh orang yang sudah sangat memahami algoritme)

Contoh men-trace algoritme

// Misal input n=4

```
procedure printN {
  read(n);
  c = 0;
  while (c<n) {
    c=c+1;
    print(c);
  }
}
```

n

c

output

1

2

3

4

STOP

Latihan

Latihan #1

- Berapa kali teks IPB dicetak oleh program berikut?

```
#include <stdio.h>

int main() {
  int i;
  for (i=1; i<=20; i++) {
    if (i=5)
      printf ("IPB\n");
  }
  return 0;
}
```

- Apa yang dapat disimpulkan dari kasus ini ?

Latihan #2

- Buat program untuk membaca sebuah bilangan bulat n , dan menampilkan output seperti dalam contoh. Setiap baris output hanya diakhiri dengan *newline*.

- Contoh input:

5

- Contoh output:

```
*
**
***
****
*****
```

```
int main() {
  int i, j, n=5;
  for (i=1; i<=n; i++) {
    for (j=1; j<=i; j++)
      printf("*");
    printf("\n");
  }
  return 0;
}
```

Latihan #4

Apa output potongan program berikut jika diberikan input data 8 20? Apa yang dilakukan?

```
int a, b, t;
scanf("%d %d", &a, &b);
while (b) {
  t = a*b;
  a = b;
  b = t;
}
printf("%d\n", a);
```

4

Latihan #5

Apa output potongan program berikut jika diberikan input data 8 20 3 -99? Apa yang dilakukan?

```
int bil, c=0;
scanf("%d", &bil);
while (bil!=-99) {
  c++;
  scanf("%d", &bil);
}
printf("%d\n", c);
```

3

Latihan #6

Apa output potongan program berikut jika diberikan input data 9872 ? Apa yang dilakukan?

```
int bil, t=0;
scanf("%d", &bil);
while (bil) {
    t = (t*10) + (bil%10);
    bil/=10;
}
printf("%d\n", t);
```

2789

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

Latihan #7

Suatu alat sensor dipasang pada tanaman apel untuk mengatur suplai air. Alat secara periodik menerima data berupa kandungan air tanah di sekitar tanaman. Jika alat tersebut menerima data kandungan air tanah kurang dari 4.5 satuan, maka alat akan menyuplai air dengan volume 0.12 liter. Demikian seterusnya sampai diperoleh data kandungan air tanah lebih besar atau samadengan 4.5 satuan dan alat akan berhenti sementara. Buat program C untuk menghitung total volume air yang disuplai sampai alat tersebut berhenti sementara. Output ditulis dalam dua desimal di belakang titik.

Contoh input : 1.2 4.2 4.3 4.4 5.2
Contoh output : 0.48

Contoh input : 4.9
Contoh output : 0.00

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

```
float bil=0, jum=0;
while (bil<4.5) {
    jum=jum+0.12;
    scanf("%d", &bil);
}
printf("%d\n", jum);
```

Latihan #8

Seorang anak suka bermain dengan bilangan bulat. Jika diberikan bilangan bulat lebih besar atau samadengan 100, dia hanya akan mengambil digit paling kiri dan paling kanan untuk membentuk suatu bilangan baru dengan dua digit. Misalkan diberikan nilai 7895, dia membentuk bilangan baru yaitu 75. Selanjutnya, dia menuliskan sisa pembagian bilangan baru tersebut dengan 6 di kertas. Jika bilangan bulat yang diberikan kurang dari 100, maka dia tidak melakukan apa-apa. Buat program C untuk mendapatkan nilai seperti yang dituliskan anak tadi.

Contoh input : 7895
Contoh output : 3

Contoh input : 37
Contoh output :

DEPARTEMEN ILMU KOMPUTER
INSTITUT PERTANIAN BOGOR

```
int kiri,kanan,bil,jum, i, sisa;
scanf("%d", &bil);

kanan = bil % 10;

while (bil>10) {
    bil = bil/10;
}
kiri = bil * 10;
bil = kiri + kanan;
bil=bil%6;
printf("%d\n", bil);
```

Latihan #9

```
• int a=3, b=0;
  while (b++<a) printf("%d", b);    1 2 3
int a=3, b=0;
  while (++b<a) printf("%d", b);    1 2
int a=5, b=0;
  while (b++<a) printf("%d", b++);  1 3 5
int a=3, b=0;
  while (b=a--) printf("%d %d\n", a,b);    2 3
  1 2
int a=3, b=0;
  while (b-->a) printf("%d %d\n", a,b);    0 1
int a=5, b, c;
  while (b-->a, b=a--) printf("%d %d %d\n", a,b,c);    2 2
  1 1
  3 4 4
  1 2 2
```