POLITEKNIK STATISTIKA STIS

For Better Official Statistics

STRUKTUR PENGULANGAN

BAGIAN 1

Ibnu Santoso, S.S.T., M.T., Nori Wilantika, S.S.T., M.T.I.

Materi

- Struktur **WHILE – DO**
- Struktur **REPEAT-UNTIL**
- **WHILE-DO** atau **REPEAT-UNTIL**?

Pengantar

- Komputer dapat melaksanakan instruksi berulang tanpa rasa bosan dengan kinerja yang sama.
- Manusia gampang bosan dan cenderung rentan melakukan kesalahan
- Contoh:
  - Menulis “Saya berjanji akan belajar Alpro dengan rajin” sebanyak 100 lembar.

Struktur Pengulangan

- Terdapat 2 bagian struktur pengulangan:
  - Kondisi pengulangan
  - Badan (body) pengulangan
- Struktur pengulangan biasanya disertai bagian:
  - Inisialisasi
  - Terminasi
- Di dalam algoritma, pengulangan (*repetition* atau *loop*) dapat dilakukan sejumlah *n* kali, atau sampai kondisi pengulangan berhenti tercapai.
- Perulangan harus berhenti.

Notasi Struktur Pengulangan

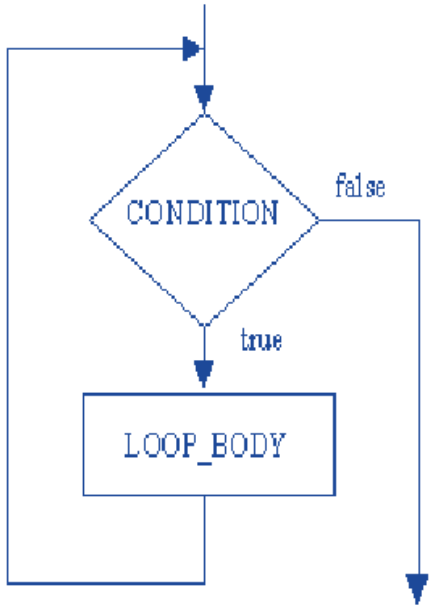
- Struktur **WHILE-DO**
- Struktur **REPEAT-UNTIL**
- Struktur **FOR**

WHILE - DO

Struktur WHILE-DO

Bentuk Umum algoritma:

```
while <kondisi> do
    aksi
endwhile
```



Translasi dalam Bahasa Pascal:

```
while kondisi do
    aksi;
```

- Aksi akan dilaksanakan berulang-ulang sepanjang `<kondisi> boolean` bernilai `true`
- Jika `<kondisi> boolean` bernilai `false`, badan pengulangan tidak dilakukan lagi (selesai).

Algoritma Cetak Halo

```
algoritma cetak_halo;
{mencetak 'HALO' sebanyak 10 kali}
deklarasi
  n:integer {pencacah pengulangan}
deskripsi
  n ← 1
  while n<=10 do
      write('HALO')
      n ← n+1;
  endwhile
  {kondisi berhenti: n>10}
```

Cetak Halo dalam Pascal

```
program cetak_halo;
var
  n:integer;
begin
  n:=1;
  while n<=10 do
      begin
          writeln('HALO');
          n:=n+1;
      end;
  readln;
end.
```

- Output yang dihasilkan:

```
HALO
HALO
HALO
HALO        n=1
HALO
HALO        N<=10?      OUTPUT      n
HALO          T           HALO      2
HALO          T           HALO      3
HALO
HALO          ...
HALO          T           HALO     11
              F
```

Penjelasan Cetak Halo

- Pada mulanya `n` diisi nilai `1`
- Sebelum memasuki badan pengulangan, kondisi `n<=10` diperiksa apakah bernilai `true`
- Karena `1<=10` bernilai true maka pernyataan `writeln('HALO')` dilaksanakan sehingga output yang tercetak:

```
HALO
```

- Selanjutnya pernyataan `n:=n+1` menyebabkan nilai `n` bertambah `1` menjadi `2`, lalu siklus pengulangan dimasuki lagi
- Sebelum melaksanakan `writeln('HALO')`, kondisi pengulangan `n<=10` diperiksa terlebih dahulu
- Karena `2<=10` maka badan pengulangan dimasuki, `writeln('HALO')` dilaksanakan sehingga output yang tercetak:

```
HALO
HALO
```

- Demikian seterusnya sebanyak `10` kali. Setiap kali badan pengulangan dimasuki,nilai `n` bertambah satu sampai `n=11`.
- Karena `11>10` maka kondisi pengulangan `n<=10` bernilai false
- Pengulangan dihentikan.

Apa yang akan terjadi?

- Jika programmer lupa melakukan inisiasi nilai `n`?

```
program cetak_halo;
var
  n:integer;
begin
  n:=1;
  while n<=10 do
      begin
          writeln('HALO');
          n:=n+1;
      end;
  readln;
end.
```

Apa yang akan terjadi?

- Jika programmer lupa menambahkan nilai `n` dengan `1`?

```
program cetak_halo;
var
  n:integer;
begin
  n:=1;
  while n<=10 do
      begin
          writeln('HALO');
          n:=n+1;
      end;
  readln;
end.
```

Algoritma Cetak Angka sebanyak N

```
algoritma cetak_n_angka;
{mencetak 1,2,3,...,n}
deklarasi
  n:integer
  angka:integer
Deskripsi
  read(n)
  angka ← 1
  while angka <= n do
      write(angka)
      angka ← angka+1;
  endwhile
  {kondisi berhenti: angka>N}
```

Program Cetak Angka Sebanyak N

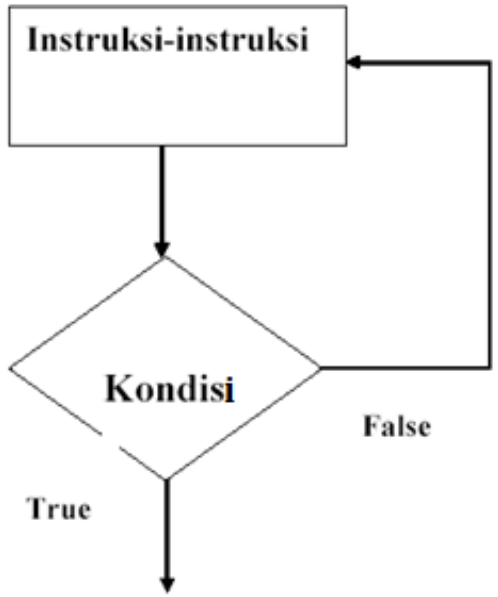
```
program cetak_angka;
var
     n,angka:integer;
Begin
     write('masukkan jumlah angka: ');
     readln(n);
     angka:=1;
     while angka<=n do
          begin
               writeln(angka);
               angka:=angka+1;
          end;
     readln;
end.
```

REPEAT - UNTIL

Struktur REPEAT-UNTIL

Bentuk umum algoritma:

```
repeat
    aksi
until <kondisi>
```



Translasi dalam Bahasa Pascal:

```
repeat
    aksi;
until kondisi;
```

- Aksi di dalam badan pengulangan terus diulang sampai kondisi boolean bernilai `true`
- Dengan kata lain, jika kondisi berhenti masih `false`, pengulangan masih terus dilakukan.
- Karena pengulangan harus berhenti, di dalam badan pengulangan harus ada aksi yang mengubah nilai kondisi

Algoritma Cetak Halo

```
algoritma cetak_halo;
{mencetak 'HALO' sebanyak 10 kali}
deklarasi
  n:integer {pencacah pengulangan}
deskripsi
  n ← 1
  repeat
      write('HALO')
      n ← n+1;
  until n > 10
  {kondisi berhenti: n>10}
```

Cetak Halo dalam Pascal

- Mencetak HALO sebanyak sepuluh kali menggunakan REPEAT-UNTIL

```
program cetak_halo;
var
  n:integer;
begin
  n:=1;
  repeat
      writeln('HALO');
      n:=n+1;
  until n>10;
  readln;
end.
```

```
n=1

OUTPUT        n      n>10?
 HALO         2        F
 HALO         3        F
 ...
 HALO         11       T
```

- Struktur REPEAT-UNTIL memiliki makna yang mirip dengan WHILE-DO
  - Namun pada struktur REPEAT-UNTIL, aksi (atau sekumpulan aksi) dilaksanakan minimal satu kali, karena kondisi pengulangan diperiksa pada akhir struktur.
  - Pada struktur WHILE-DO kondisi pengulangan diperiksa pada awal struktur sehingga memungkinkan badan pengulangan sama sekali tidak dilaksanakan.

Algoritma Cetak Angka sebanyak N

```
algoritma cetak_n_angka;
{mencetak 1,2,3,...,n}
deklarasi
  n:integer
  angka:integer
Deskripsi
  read(n)
  angka ← 1
  repeat
      write(angka)
      angka ← angka+1;
  until angka > n
  {kondisi berhenti: angka>N}
```

Program Cetak Angka Sebanyak N

```
program cetak_angka;
var
    n,angka:integer;
Begin
    write('masukkan jumlah angka: ');
    readln(n);
    angka:=1;
    repeat
        writeln(angka);
        angka:=angka+1;
    until angka>n;
    readln;
end.
```

- Perhatikan bahwa contoh cetak angka tersebut hanya benar jika `n` positif.
- Jika pengguna mencoba memasukkan nilai `n` negatif atau `0`,maka struktur REPEAT-UNTIL tetap dimasuki.
- Pemilihan struktur pengulangan yang tepat mempengaruhi kebenaran logika program.
- Pemilihan struktur pengulangan bergantung pada pemeriksaan kondisi pengulangan, apakah di awal atau di akhir.

WHILE-DO atau REPEAT-UNTIL?

- Meskipun WHILE-DO dan REPEAT-UNTIL memiliki makna yang sama, namun pemilihan struktur yang tepat **bergantung pada masalah yang akan diprogram.**
- Ingat bahwa pemeriksaan kondisi pada WHILE-DO dilakukan di **awal** pengulangan,
- Sedangkan pada REPEAT-UNTIL pemeriksaan kondisi dilakukan pada **akhir** pengulangan
- Sebagai konsekuensi dari waktu pemeriksaan kondisi, aksi di dalam badan WHILE-DO paling sedikit dikerjakan **0** kali
- Aksi di dalam badan REPEAT-UNTIL paling sedikit dikerjakan **1** kali

Program Memilih Menu

```
program memilih_menu;
var
     nomor_menu:integer;
begin
     writeln('    MENU          ');
     writeln('1. SATE KAMBING   ');
     writeln('2. SATE AYAM      ');
     writeln('3. BAKSO SAPI     ');
     writeln('4. NASI GORENG    ');
     writeln('5. KELUAR PROGRAM');
     write('PILIH MENU: ');
     readln(nomor_menu);
     case (nomor_menu) of
          1:writeln('ANDA MEMILIH MENU SATE KAMBING');
          2:writeln('ANDA MEMILIH MENU SATE AYAM');
          3:writeln('ANDA MEMILIH MENU BAKSO SAPI');
          4:writeln('ANDA MEMILIH MENU NASI GORENG');
          5:writeln('KELUAR PROGRAM');
     end;
     readln;
end.
```

Permasalahan

- Program memilih menu tadi tidak dapat memilih menu secara berulang-ulang.
- Kita menginginkan dapat memilih menu manapun berkali-kali
- Misal pemilihan menu hanya diakhiri jika kita memilih nomor 5.
- Bagaimana hal ini dapat dilakukan?

Menggunakan REPEAT-UNTIL

```
program memilih_menu;
var
      nomor_menu:integer;
begin
      repeat
            begin
                  writeln('    MENU          ');
                  writeln('1. SATE KAMBING   ');
                  writeln('2. SATE AYAM      ');
                  writeln('3. BAKSO SAPI     ');
                  writeln('4. NASI GORENG    ');
                  writeln('5. KELUAR PROGRAM');
                  write('PILIH MENU: ');
                  readln(nomor_menu);
                  case (nomor_menu) of
                        1:writeln('ANDA MEMILIH MENU SATE KAMBING');
                        2:writeln('ANDA MEMILIH MENU SATE AYAM');
                        3:writeln('ANDA MEMILIH MENU BAKSO SAPI');
                        4:writeln('ANDA MEMILIH MENU NASI GORENG');
                        5:writeln('KELUAR PROGRAM');
                  end;
            end;
      until nomor_menu=5;
      readln;
end.
```

Menggunakan WHILE-DO

```
program memilih_menu;
var
      nomor_menu:integer;
begin
      nomor_menu:=0;
      while (nomor_menu <> 5) do
            begin
                  writeln('    MENU          ');
                  writeln('1. SATE KAMBING   ');
                  writeln('2. SATE AYAM      ');
                  writeln('3. BAKSO SAPI     ');
                  writeln('4. NASI GORENG    ');
                  writeln('5. KELUAR PROGRAM');
                  write('PILIH MENU: ');
                  readln(nomor_menu);
                  case (nomor_menu) of
                        1:writeln('ANDA MEMILIH MENU SATE KAMBING');
                        2:writeln('ANDA MEMILIH MENU SATE AYAM');
                        3:writeln('ANDA MEMILIH MENU BAKSO SAPI');
                        4:writeln('ANDA MEMILIH MENU NASI GORENG');
                        5:writeln('KELUAR PROGRAM');
                  end;
            end;
      readln;
end.
```

Keterangan Memilih Menu

- Permasalahan memilih menu di atas sama-sama bisa dilakukan baik menggunakan struktur REPEAT-UNTIL maupun WHILE-DO
- Namun struktur REPEAT-UNTIL lebih tepat,sebab menu ditampilkan lebih dahulu, baru kemudian nomor pilihan menu dibaca. Pemeriksaan kondisi pengulangan dilakukan di akhir struktur sampai `nomor_menu` yang dibaca adalah `5`.
- Penggunaan WHILE-DO pada masalah ini mengharuskan variabel `nomor_menu` harus diinisialisasi dengan sembarang nilai asal bukan `5`, agar kondisi pengulangan bernilai `true`.
- Jadi meskipun program di atas tetap benar, namun penggunaan struktur WHILE-DO untuk masalah ini kurang tepat.

POLITEKNIK STATISTIKA STIS

For Better Official Statistics

TERIMA KASIH