

Algoritma dan Pemrograman

Bab IV – Struktur Pemilihan



```
> internet
```

Adam Mukharil Bachtiar, S.Kom.
Universitas Komputer Indonesia

Pengertian Pemilihan

Struktur algoritma yang memungkinkan melakukan **statement (pernyataan)** jika suatu **kondisi** terpenuhi.



Pemilihan Satu Kasus

Format algoritma:

if kondisi then

statement

endif

Pemilihan Satu Kasus

Format Pascal:

```
if kondisi then  
    statement;
```

Atau

```
if kondisi then  
begin  
    statement 1;  
    statement 2;  
end;
```

Contoh Pemilihan Satu Kasus

```
1  Algoritma Bilangan_Ganjil
2  {I.S: Diinputkan satu bilangan oleh user}
3  {F.S: Menampilkan statement apabila bilangannya ganjil}
4
5  Kamus:
6      bil:integer
7
8  Algoritma:
9      input(bil)
10     if bil mod 2 = 1 then
11         output('Bilangan ',bil,' adalah bilangan ganjil')
12     endif
```


Contoh Pemilihan Satu Kasus

```
1  program Bilangan_Ganjil;
2  uses crt;
3
4  var
5      bil:integer;
6
7  begin
8      write('Masukan sebuah bilangan bulat: ');
9      readln(bil);
10     if bil mod 2 = 1 then
11         writeln('Bilangan ',bil,' adalah bilangan ganjil');
12     writeln();
13     writeln('Ketik sembarang tombol untuk menutup...');
14     readkey();
15 end.
```

Pemilihan Dua Kasus

Format Algoritma:

if kondisi then

statement 1

else

statement 2

endif

Pemilihan Dua Kasus

Format Pascal:

```
if kondisi then  
    statement 1  
else  
    statement 2;
```

```
if kondisi then  
begin  
    statement 1;  
    statement 2;  
end  
else  
begin  
    statement 3;  
    statement 4;  
end;
```


Contoh Pemilihan Dua Kasus

```
1  Algoritma Bilangan_Genap_Ganjil
2  {I.S: Diinputkan satu bilangan oleh user}
3  {F.S: Menampilkan statement bilangan ganjil atau genap}
4
5  Kamus:
6      bil:integer
7
8  Algoritma:
9      input(bil)
10     if bil mod 2 = 1 then
11         output('Bilangan ',bil,' adalah bilangan ganjil')
12     else
13         output('Bilangan ',bil,' adalah bilangan genap')
14     endif
```

Contoh Pemilihan Dua Kasus

```
1  program Bilangan_Genap_ganjil;
2  uses crt;
3
4  var
5      bil:integer;
6
7  begin
8      write('Masukkan sebuah bilangan bulat: ');
9      readln(bil);
10     if bil mod 2 = 1 then
11         writeln('Bilangan ',bil,' adalah bilangan ganjil')
12     else
13         writeln('Bilangan ',bil,' adalah bilangan genap');
14     writeln();
15     writeln('Tekan sembarang tombol untuk menutup...');
16     readkey();
17 end.
```

Pemilihan Tiga Kasus atau Lebih

Format Algoritma:

```
if kondisi 1 then  
    statement 1  
else  
    if kondisi 2 then  
        statement 2  
    else  
        if kondisi 3 then  
            statement 3  
        else  
            statement 4  
        endif  
    endif  
endif
```

Pemilihan Tiga Kasus atau Lebih

Format Pascal:

```
if kondisi 1 then
    statement 1
else
    if kondisi 2 then
        statement 2
    else
        if kondisi 3 then
            statement 3
        else
            statement 4;
```

```
if kondisi 1 then
begin
    statement 1;
end
else
    if kondisi 2 then
begin
    statement 2;
end
    else
        if kondisi 3 then
begin
    statement 3;
end
        else
begin
    statement 4;
end;
```

Contoh Pemilihan Tiga Kasus atau Lebih

```
1  Algoritma Lampu_Lalu_Lintas
2  {I.S: Diinputkan satu warna lampu oleh user}
3  {F.S: Menampilkan statement sesuai warna lampu}
4
5  Kamus:
6      warna:string
7
8  Algoritma:
9      input(warna)
10     if warna = 'MERAH' then
11         output('Berhenti!')
12     else
13         if warna = 'KUNING' then
14             output('Hati-Hati!')
15         else
16             if warna = 'HIJAU' then
17                 output('Jalan!')
18             else
19                 output('Warna salah!')
20             endif
21         endif
22     endif
```


Contoh Pemilihan Tiga Kasus atau Lebih

```
1  program Lampu_Lalu_Lintas;
2  uses crt;
3
4  var
5      warna:string;
6
7  begin
8      write('Masukkan sembarang warna: ');
9      readln(warna);
10     warna:=upcase(warna); {membuat uppercase}
11
12     if warna='MERAH' then
13         writeln('Berhenti!')
14     else
15         if warna='KUNING' then
16             writeln('Hati-Hati!')
17         else
18             if warna='HIJAU' then
19                 writeln('Jalan!')
20             else
21                 writeln('Warna salah!');
22             writeln();
23             writeln('Tekan sembarang tombol untuk menutup...');
24             readkey();
25     end.
```

Penggunaan Lebih Dari Satu Kondisi

1. Ada beberapa kasus yang menuntut pemeriksaan kondisi **lebih dari satu kondisi**.
2. Pemecahan:
 - a. Kata kunci **and** apabila semua kondisi harus terpenuhi.
 - b. Kata kunci **or** apabila cukup satu kondisi saja yang terpenuhi.

Contoh Penggunaan Lebih Dari Satu Kondisi

```
1  Algoritma Huruf_Konsonan
2  {I.S: Diinputkan satu huruf oleh user}
3  {F.S: Menampilkan pesan huruf konsonan jika konsonan}
4
5  Kamus:
6      k: char
7
8  Algoritma:
9      input(k)
10     if (k≠' a' ) and (k≠' i' ) and (k≠' u' ) and (k≠' e' ) and (k≠' o' ) then
11         output('Huruf ',k,' adalah huruf konsonan')
12     else
13         output('Huruf ',k,' adalah huruf vokal')
14     endif
```

Contoh Penggunaan Lebih Dari Satu Kondisi

```
1  program Huruf_Konsonan;
2  uses crt;
3
4  var
5      k:char;
6
7  begin
8      write('Masukkan satu huruf: ');
9      readln(k);
10     k:=lowercase(k);
11     if (k<>'a') and (k<>'i') and (k<>'u') and (k<>'e') and (k<>'o')
12     then
13         writeln('Huruf ',k,' adalah huruf konsonan')
14     else
15         writeln('Huruf ',k,' adalah huruf vokal');
16     writeln();
17     writeln('Tekan sembarang tombol untuk menutup...');
18     readkey();
19 end.
```

Struktur Case

Format Algoritma:

case ekspresi

nilai 1 : statement 1

nilai 2 : statement 2

nilai 3 : statement 3

.

.

.

nilai n : statement n

otherwise : statement x

endcase

Struktur Case

Format Pascal:

```
case ekspresi of
    nilai 1 : statement 1;
    nilai 2 : statement 2;
    nilai 3 : statement 3;
    .
    .
    .
    nilai n : statement n;
else statement x;
end;
```

Struktur Case

1. Ekspresi bisa berupa aritmatika atau boolean.
2. Ekspresi menghasilkan suatu konstanta.
3. Nilai harus bertipe ordinal (char, boolean, dan integer).
4. Otherwise akan dijalankan apabila nilai yang lain tidak ada yang dipenuhi.



Contoh Struktur Case

```
1  Algoritma Ukuran_Baju
2  {I.S: Diinputkan satu huruf untuk ukuran baju oleh user}
3  {F.S: Menampilkan arti ukuran baju}
4
5  Kamus:
6      size: char
7
8  Algoritma:
9      input(size)
10     case size
11         'S' : output('Kecil');
12         'M' : output('Sedang');
13         'L' : output('Besar');
14         otherwise : output('Ukuran salah!')
15     endcase
```

Contoh Struktur Case

```
1  program Ukuran_Baju;
2  uses crt;
3
4  var
5      size:char;
6
7  begin
8      write('Masukkan ukuran baju [S/M/L]: ');
9      readln(size);
10     size:=upcase(size);
11     case size of
12         'S':writeln('Kecil');
13         'M':writeln('Sedang');
14         'L':writeln('Besar');
15         else writeln('Ukuran salah!');
16     end;
17     writeln();
18     writeln('Tekan sembarang tombol untuk menutup...');
19     readkey();
20 end.
```

Kegunaan Struktur Pemilihan

1. Membuat struktur menu.
2. Memvalidasi inputan.
3. Mencegah error.



