



POLITEKNIK STATISTIKA STIS

For Better Official Statistics

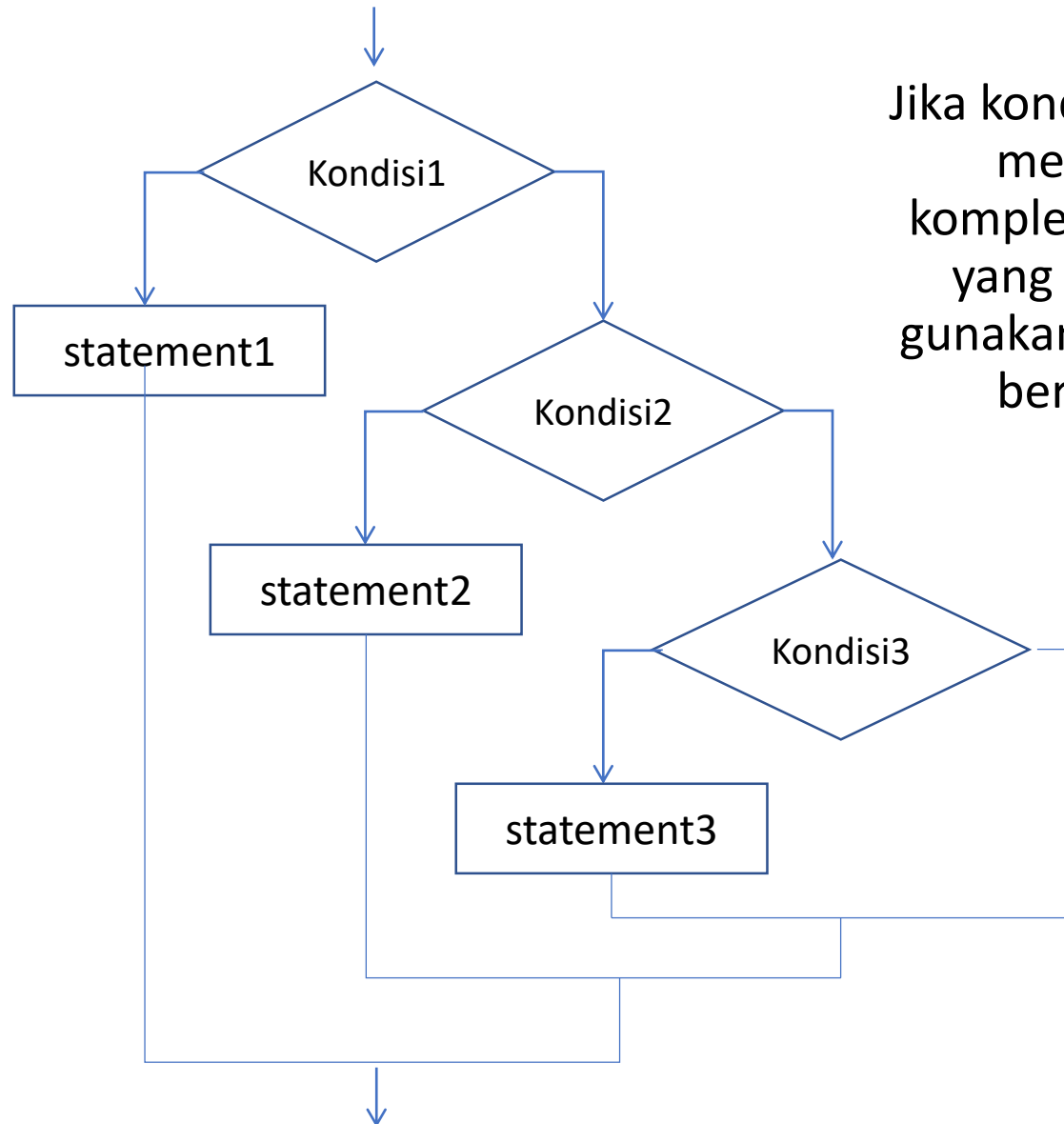
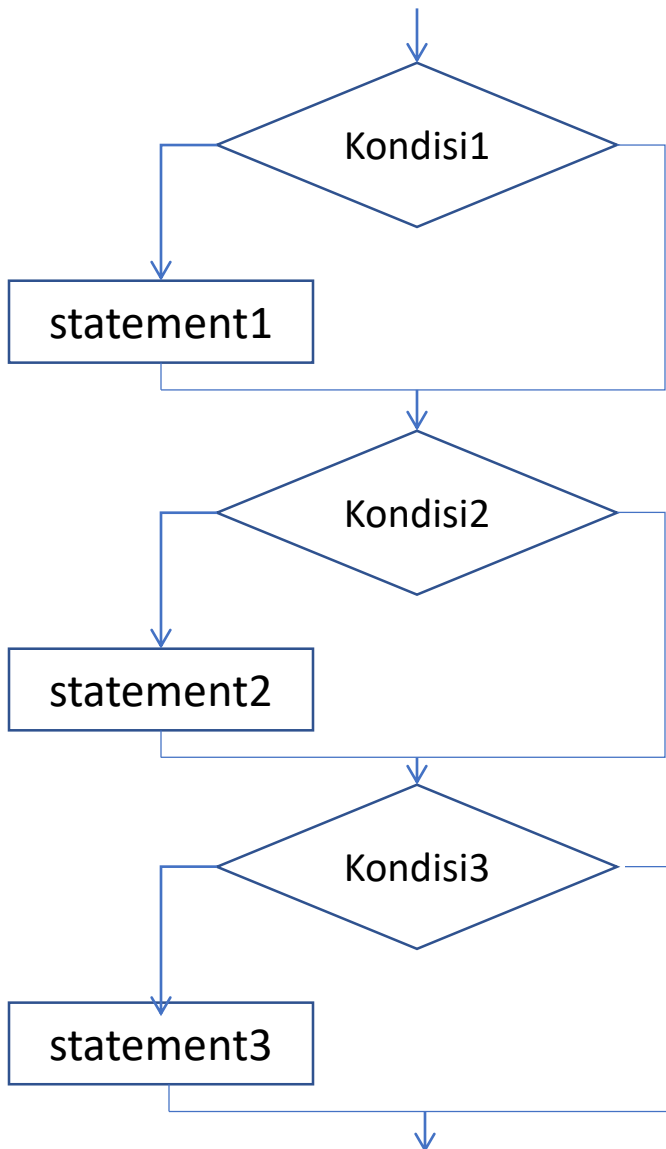
KONSEP PEMILIHAN (BAG. 2)

Dr. Abdul Ghofar, S.Si., M.Tl., Nori Wilantika, S.S.T., M.T.I.



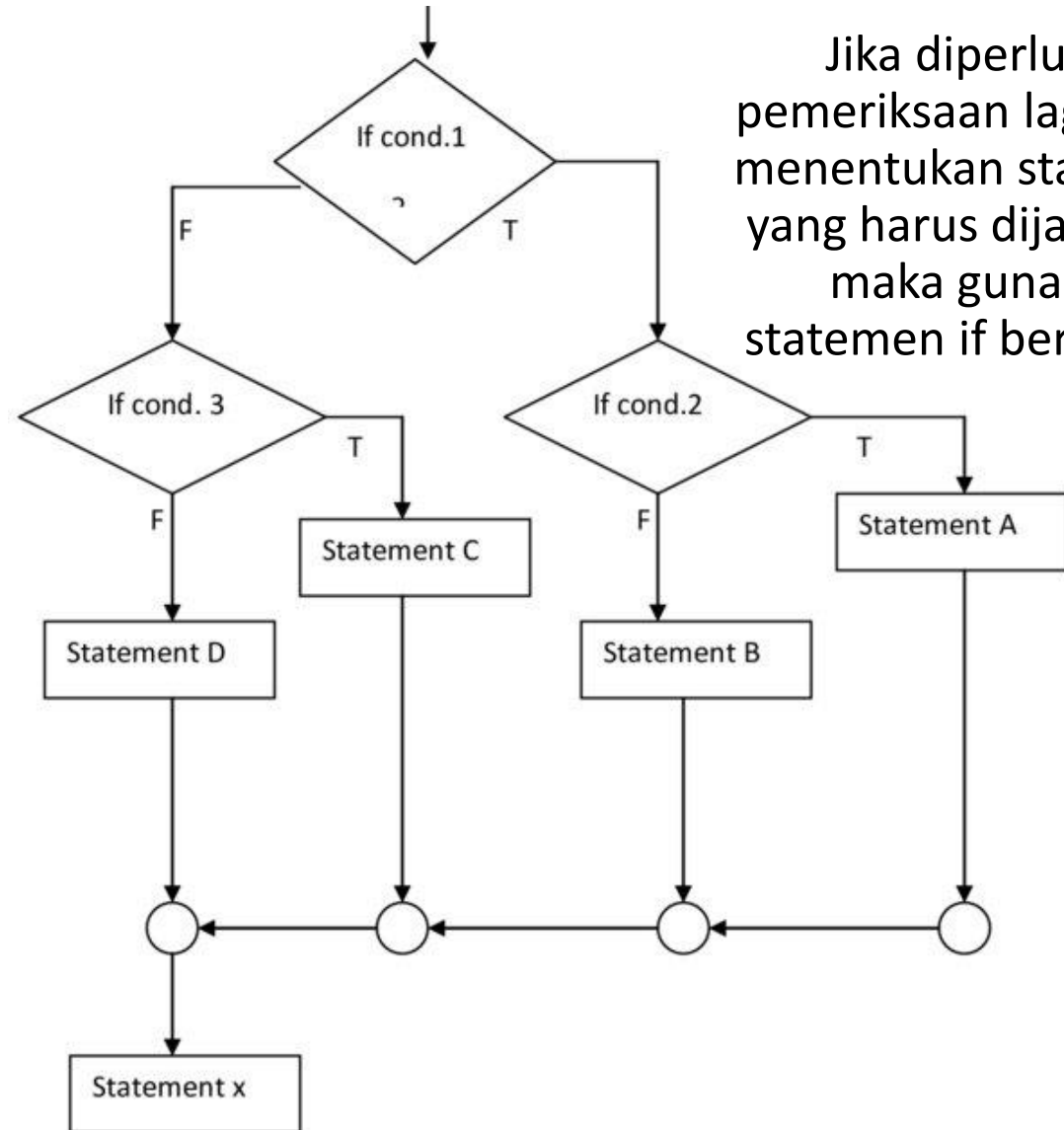
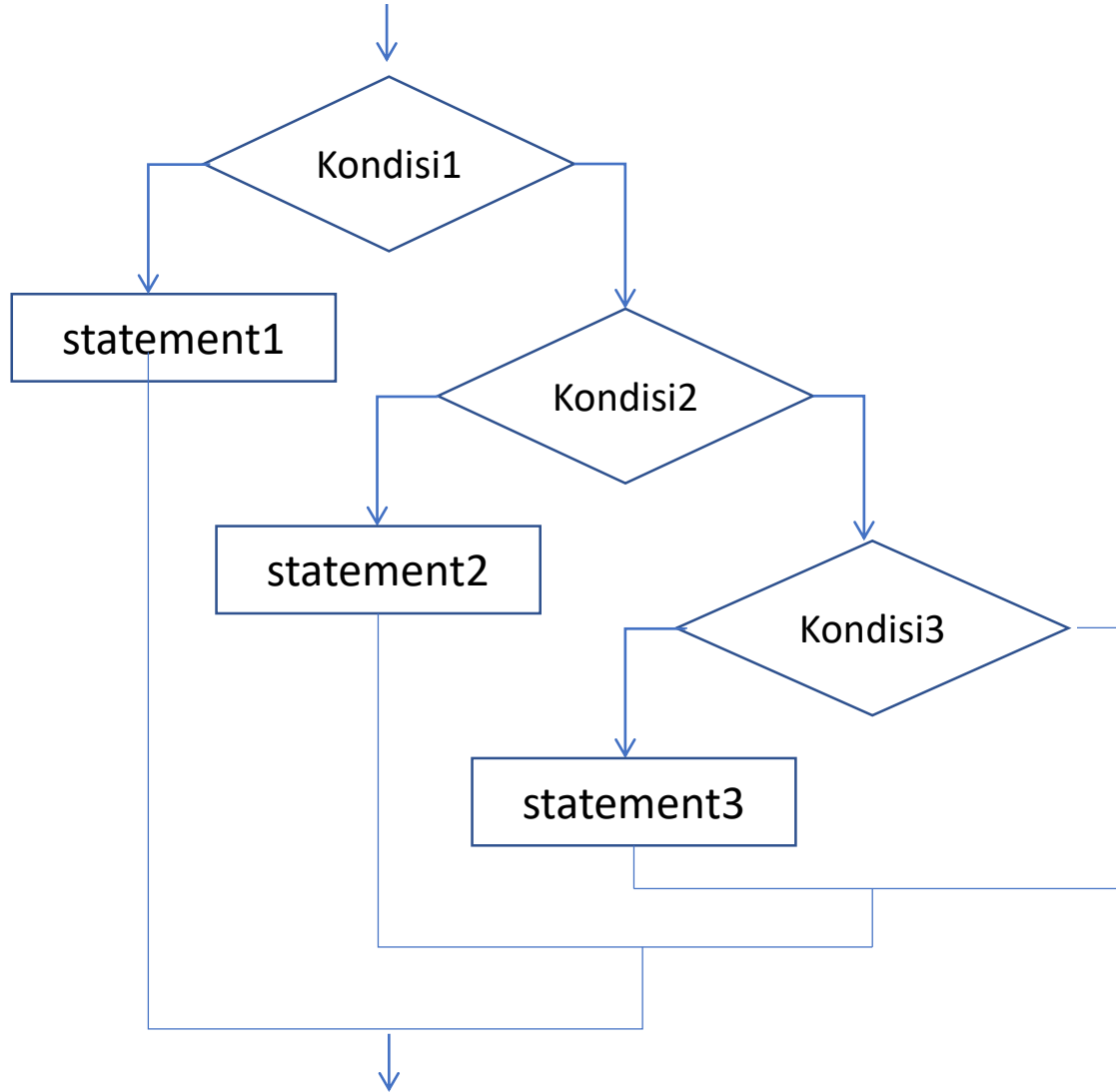
- Review IF
- CASE
- Pemilihan Tanpa Syarat: EXIT, GOTO

- ✓ Pemilihan (seleksi) digunakan pada masalah yang tidak dapat dipecahkan dengan perhitungan yang pasti.
- ✓ Inti dari pemecahan masalah jenis ini adalah menganalisis semua kondisi yang mungkin terjadi dan menentukan statemen yang dilakukan untuk setiap kondisi tersebut.
- ✓ Terdapat dua skema pemilihan, yaitu menggunakan statemen if dan menggunakan statemen case.
- ✓ Statemen if dapat digunakan dengan bagian else atau tanpa else, tergantung masalah yang dipecahkan dan algoritma yang dipilih.
- ✓ Jika terdapat dua kondisi yang komplementer, maka statemen if diikuti bagian else.
- ✓ Kondisi dalam statemen if dituliskan dalam ekspresi boolean. Statemen if juga dapat digunakan secara berulang.



Jika kondisi yang satu merupakan komplemen kondisi yang lain, maka gunakan statemen if berjenjang.

Jika diperlukan pemeriksaan lagi untuk menentukan statement yang harus dijalankan, maka gunakan statemen if bersarang.





IF-ELSE PUZZLES



```
PROGRAM else_bermasalah;  
VAR  
    angka : INTEGER;  
BEGIN  
    Write('Ketik sembarang integer:');  
    Readln (angka);  
    If Angka > 10 THEN  
        If Angka > 20 then  
            Writeln ('A')  
        ELSE  
            Writeln ('B');  
            Writeln ('C');  
    END.  
END.
```

- Jika kita masukkan sembarang integer: 30, 15 dan 5 apa hasilnya?
- Else pada program ini menjadi “milik” statemen if yang mana?

Jika bagian else menjadi “milik” statemen if yang pertama:

```
If Angka > 10 then
    If Angka > 20 then
        Writeln ('A')
Else
    Writeln ('B')
Writeln ('C')
```

Jika bagian else adalah “milik” statemen if yang kedua:

```
If Angka > 10 then
    If Angka > 20 then
        Writeln ('A')
    Else
        Writeln ('B')
Writeln ('C')
```

Apakah indentasi cukup untuk menggambarkan logika pemecahan masalah?

Dalam Pascal, bagian else dianggap sebagai “milik” statemen if yang terdekat.

Indentasi aja tidak cukup, karena tap tidak di eksekusi dalam program

Jika bagian else menjadi “milik” statemen if yang pertama:

```
If Angka > 10 then
  Begin
    If Angka > 20 then
      Writeln ('A')
    End
  Else
    Writeln ('B');
  Writeln ('C');
```

Jika bagian else adalah “milik” statemen if yang kedua:

```
If Angka > 10 then
  If Angka > 20 then
    Writeln ('A')
  Else
    Writeln ('B');
  Writeln ('C')
```

Penentuan ekspresi boolean pada statemen if sangat penting, terutama untuk meningkatkan efisiensi pemeriksaan dan menghindari adanya statemen yang tidak pernah dijalankan apapun kondisi yang terjadi.

Perhatikan contoh berikut:

```
If Jumlah < 200 THEN
    statemen1
Else {jumlah >= 200}
    If Jumlah >= 100 then
        statemen2
    Else {jumlah < 100}
        statemen3;
```

- Statemen3 tidak pernah dijalankan berapapun harga dari jumlah.
- Jika jumlah < 200, maka statemen1 dijalankan
- Jika jumlah >= 200, maka bagian **else** akan dijalankan.
- Bagian **else** pada statemen if tersarang, akan dijalankan jika jumlah < 100. Karena sebuah harga tidak mungkin >=200 dan sekaligus < 100, maka dapat disimpulkan bahwa statemen3 tidak mungkin pernah dijalankan.
- Statemen seperti ini disebut dengan **kode mati (dead code)**.

```

If jumlah > 200 then
    statemen1
Else {jumlah <= 200}
    If jumlah > 100 then
        statemen2
    Else {jumlah <= 100}
        If harga <= 100 then
            statemen3;
  
```

- Ekspresi boolean pada statemen if yang terakhir, harga ≤ 100 , tidak diperlukan.
- Jika ekspresi boolean pada statemen if sebelumnya, jumlah > 100 , bernilai **false**, maka ekspresi boolean harga ≤ 100 pasti bernilai **true**, dan karenanya cukup dengan else saja tanpa pemeriksaan lanjutan.
- Bentuk ini akan menghasilkan efek yang sama dengan bentuk sebelumnya tetapi tanpa melakukan pemeriksaan yang tidak diperlukan.

```
If jumlah > 200 then  
    statemen1  
Else {jumlah <= 200}  
    If jumlah > 100 then  
        statemen2  
    Else {jumlah <= 100}  
        statemen3;
```



CASE



- Untuk masalah dengan dua kasus atau lebih
- Struktur CASE dapat menyederhanakan penulisan **ELSE-IF berjenjang** sebagaimana pada contoh-contoh sebelumnya

label dan ungkapan dapat berupa integer, char, boolean, atau enumerated

Real tidak bisa

CASE ungkapan OF

```
label-1 : statemen-1 ;  
label-2 : statemen-2 ;  
label-3 : statemen-3 ;  
  
...  
  
...  
label- $n$  : statemen- $n$  ;  
[ELSE statemen  $n+1$ ]  
  
END ;
```

Statemen **ELSE** bersifat opsional dan baru dijalankan jika nilai ungkapannya diluar label-1 s.d label- n

Struktur CASE di atas menggantikan analisis kasus yang ekuivalen dengan struktur IF-THEN-ELSE berikut ini:

```
If kondisi-1 then  
    statement-1  
else if kondisi-2 then  
    statement-2  
else if kondisi-3 then  
    statement-3  
...  
...  
else if kondisi-n then  
    statement-n  
else  
    statement-x;
```

Buatlah algoritma yang:

- membaca sebuah angka bulat yang nilainya terletak antara 1 sampai 4, lalu menuliskan ke layar angka tersebut dalam kata-kata.
- Misalkan bila dibaca angka 1, maka tercetak tulisan 'satu', bila dibaca 2, maka tercetak di layar tulisan 'dua', dst.
- Jika angka yang dimasukkan selain 1 sampai 4, tuliskan pesan bahwa "angka yang dimasukkan salah"!

Algoritma KATA_UNTUK_ANGKA

Deklarasi

Angka : integer {angka yang dibaca}

Deskripsi:

```
read(angka)
If angka=1 THEN
    write('satu')
ELSE IF angka = 2 THEN
    write('dua')
ELSE IF angka = 3 THEN
    write('tiga')
ELSE IF angka = 4 THEN
    write('4')
ELSE
    write('Angka yang dimasukkan salah')
endif
endif
endif
endif
```

Dengan struktur CASE, algoritma untuk masalah tersebut dapat dibuat menjadi lebih singkat sbb:

Algoritma KATA_UNTUK_ANGKA

Deklarasi

A: integer {angka yang dibaca}

Deskripsi:

Read(A)

CASE (A)

A=1 : write('satu')

A=2 : write('dua')

A=3 : write('tiga')

A=4 : write('empat')

else write('angka yang dimasukkan salah')

endcase

```
Var
    Nilai : char ;
Begin
    Write ('Nilai huruf yang didapat ?'); readln (Nilai);
    Case Nilai of
        'A' : writeln ('Sangat baik') ;
        'B' : writeln ('baik') ;
        'C' : writeln ('Cukup') ;
        'D' : writeln ('Kurang') ;
        'E' : writeln ('Gagal') ;
    End ;
End.
```

Algoritma:

Algoritma NAMA_NAMA_BULAN

Deklarasi

AngkaBulan: integer {1..12}

Deskripsi:

Read(AngkaBulan)

case (AngkaBulan)

AngkaBulan=1 : write('Januari')

AngkaBulan=2 : write('Februari')

AngkaBulan=3 : write('Maret')

AngkaBulan=4 : write('April')

AngkaBulan=5 : write('Mei')

AngkaBulan=6 : write('Juni')

AngkaBulan=7 : write('Juli')

AngkaBulan=8 : write('Agustus')

AngkaBulan=9 : write('September')

AngkaBulan=10 : write('Oktober')

AngkaBulan=11 : write('November')

AngkaBulan=12 : write('Desember')

otherwise write('Bukan bulan yang benar')

endcase

Pascal:

```
Program NAMA_NAMA_BULAN;  
var  
    AngkaBulan: integer;  
Begin  
    write('Masukkan angka bulan (1-12): ');  
    Readln(AngkaBulan);  
    case (AngkaBulan) of  
        1 : writeln('Januari');  
        2 : writeln('Februari');  
        3 : writeln('Maret') ;  
        4 : writeln('April') ;  
        5 : writeln('Mei') ;  
        6 : writeln('Juni') ;  
        7 : writeln('Juli') ;  
        8 : writeln('Agustus');  
        9 : writeln('September') ;  
       10 : writeln('Oktober') ;  
       11 : writeln('November');  
       12 : writeln('Desember') ;  
    else writeln('Bukan bulan yang benar');  
    end;  
End.
```

Algoritma:

Algoritma EMPAT_PERSEGI_PANJANG

Deklarasi

NomorMenu : integer
panjang, lebar : real
luas, keliling, diagonal : real

Deskripsi:

```
{cetak menu}  
write(' Menu Empat Persegi Panjang ')  
write('    1. Hitung Luas           ')  
write('    2. Hitung Keliling        ')  
write('    3. Hitung panjang diagonal')  
write('    4. Keluar Program          ')  
write(' Masukkan pilihan anda (1/2/3/4)? ')
```

Read(NomorMenu)

case (NomorMenu)

NomorMenu =1 : read(panjang, lebar)
luas \leftarrow panjang*lebar
write(luas)

NomorMenu = 2 : read(panjang, lebar)
keliling \leftarrow 2*panjang + 2*lebar
write(keliling)

NomorMenu =3 : read(panjang, lebar)
diagonal \leftarrow sqrt(panjang*panjang +
lebar*lebar)
write(diagonal)

NomorMenu =4 : write('Keluar program.....sampai jumpa')
endcase

Pascal:

Program EMPAT_PERSEGI_PANJANG

```
var
    NomorMenu : integer ;
    panjang, lebar : real;
    luas, keliling, diagonal : real;

begin
    {cetak menu}
    writeln('Menu Empat Persegi Panjang');
    writeln('1. Hitung Luas');
    writeln('2. Hitung Keliling');
    writeln('3. Hitung panjang diagonal');
    writeln('4. Keluar Program');
    writeln('Masukkan pilihan anda (1/2/3/4)?');
    readln(nomormenu);
```

```
case NomorMenu of
```

```
    1: begin
```

```
        write('Panjang? '); readln(panjang);
        write('Lebar? ');   readln(lebar);
        luas := panjang*lebar;
        writeln('Luas = ',luas);
```

```
    end;
```

```
    2: begin
```

```
        write('Panjang? '); readln(panjang);
        write('Lebar? ');   readln(lebar);
        keliling := 2*panjang + 2*lebar;
        writeln('Keliling = ', keliling);
```

```
    end;
```

```
    3: begin
```

```
        write('Panjang? '); readln(panjang);
        write('Lebar? ');   readln(lebar);
        diagonal := sqrt(panjang*panjang + lebar*lebar);
        writeln('Diagonal = ',diagonal);
```

```
    end;
```

```
    4 : writeln('Keluar program....sampai jumpa');
```

```
end;
```

```
End.
```

Algoritma:

Algoritma JUMLAH_HARI {menentukan jumlah hari dalam satu bulan }

Deklarasi

AngkaBulan : integer {1..12}

Tahun : integer {>0}

JumlahHari : integer

Deskripsi:

read(AngkaBulan, Tahun)

case(AngkaBulan)

AngkaBulan= [1, 3, 5, 7, 8, 10, 12] : JumlahHari \leftarrow 31

AngkaBulan= [4, 6, 9, 11] : JumlahHari \leftarrow 30

AngkaBulan= 2 : case Tahun

Tahun mod 4 = 0 : JumlahHari \leftarrow 29

Tahun mod 4 \neq 0 : JumlahHari \leftarrow 28

endcase

Endcase

Write(JumlahHari)

Pascal:

```

Program JUMLAH_HARI; {menentukan jumlah hari dalam satu bulan }

(*Deklarasi*)
var
    AngkaBulan : integer; {1..12}
    Tahun       : integer; {>0}
    JumlahHari  : integer;

(*Deskripsi:*)
begin
    write('Bulan (1-12)?'); readln(AngkaBulan);
    case AngkaBulan of
        1, 3, 5, 7, 8, 10, 12 : JumlahHari := 31;
        4, 6, 9, 11          : JumlahHari := 30;
        2                     :
                                BEGIN
                                write('Tahun?'); readln(Tahun);
                                if Tahun mod 4 = 0 then
                                    JumlahHari :=29
                                else
                                    JumlahHari :=28;
                                END;
    end;{case}
    Writeln('Jumlah Hari dalam Bulan ', AngkaBulan,'adalah ',JumlahHari);
End.

```

Algoritma:

Algoritma IBUKOTA

{diberikan nama ibukota provinsi di pulau Jawa. Harus ditentukan nama provinsinya }

Deklarasi

ibk : string {nama ibukota provinsi di Jawa}

Deskripsi:

read(ibk)

case(ibk)

ibk = 'Jakarta' : write('DKI Jakarta')

ibk = 'Bandung' : write('Jawa Barat')

ibk = 'Semarang' : write('Jawa Tengah')

ibk = 'Yogyakarta' : write('DI Yogyakarta')

ibk = 'Surabaya' : write('Jawa Timur')

ibk = 'Serang' : write('Banten')

endcase

Free Pascal memungkinkan penggunaan string sebagai CASE label

```
program IBUKOTA;  
Var  
    ibk:string;  
Begin  
    Read(ibk);  
    Case lowercase(ibk) of  
        'jakarta': WriteLn ('DKI Jakarta');  
        'bandung': WriteLn ('Jawa Barat');  
        'semarang': WriteLn ('Jawa Tengah');  
        'yogyakarta': WriteLn ('Yogyakarta');  
        'surabaya': WriteLn ('Jawa Timur');  
        'serang': WriteLn ('Banten');  
    end;  
End.
```



PEMILIHAN TANPA SYARAT



```

Program JUMLAH_HARI; {menentukan jumlah hari dalam satu bulan }

(*Deklarasi*)
var
  AngkaBulan : integer; {1..12}
  Tahun       : integer; {>0}
  JumlahHari  : integer;

(*Deskripsi:*)
begin
  write('Bulan (1-12)?'); readln(AngkaBulan);
  case AngkaBulan of
    1, 3, 5, 7, 8, 10, 12 : JumlahHari := 31;
    4, 6, 9, 11          : JumlahHari := 30;
    2                    :
      BEGIN
        write('Tahun?'); readln(Tahun);
        if Tahun mod 4 = 0 then
          JumlahHari :=29
        else
          JumlahHari :=28;
        END;
      end;{case}
  Writeln('Jumlah Hari dalam Bulan ', AngkaBulan,'adalah ',JumlahHari);
End.

```

Apa yang terjadi jika pengguna memasukkan angka selain 1-12?

Pernyataan EXIT diklasifikasikan sebagai pernyataan pemilihan tanpa syarat.

Jika pengguna memasukkan nomor selain nomor 1 sampai 12, bagian ELSE dan pernyataan EXIT akan mengakhiri program.

```

Program JUMLAH_HARI; {menentukan jumlah hari dalam satu bulan }

(*Deklarasi*)
var
  AngkaBulan : integer; {1..12}
  Tahun       : integer; {>0}
  JumlahHari  : integer;

(*Deskripsi:*)
begin
  write('Bulan (1-12)?'); readln(AngkaBulan);
  case AngkaBulan of
    1, 3, 5, 7, 8, 10, 12 : JumlahHari := 31;
    4, 6, 9, 11          : JumlahHari := 30;
    2                    :
                                BEGIN
                                write('Tahun?'); readln(Tahun);
                                if Tahun mod 4 = 0 then
                                  JumlahHari :=29
                                else
                                  JumlahHari :=28;
                                END;
  else
    EXIT
  end;{case}
  Writeln('Jumlah Hari dalam Bulan ', AngkaBulan,' adalah ',JumlahHari);
End.
  
```

- Pada umumnya, kita hanya membicarakan keadaan dimana komputer melaksanakan baris demi baris secara beruntun atau berurutan, dan tiap baris dilaksanakan hanya satu kali
- Namun sebenarnya kita dapat memerintahkan komputer untuk melaksanakan baris secara melompat/meloncat, serta melaksanakan lebih dari satu kali. Untuk itu kita dapat memanfaatkan statemen alih kontrol GOTO.
- Pernyataan GOTO merupakan instruksi untuk mengarahkan eksekusi program menuju pernyataan yang diawali dengan suatu **Label**.

```
GOTO label;
```

kalau pakai goto harus ada label. dan label harus dideklarasikan dulu di awal. Harus paling atas, di atas var dan const

memerintahkan program untuk lompat ke baris mana gitu.

- Label merupakan suatu pengenalan (identifier) yang diikuti dengan tanda titik dua (:).
- Label adalah bilangan bulat positif hingga empat digit sebelum statemen yang diperlukan (dalam Turbo Pascal label dapat berupa identifier yang valid dan dapat dimulai dengan digit/angka).
- Label harus dideklarasikan di bagian deklarasi program.
- Deklarasi label dimulai dengan kata kunci LABEL dan dideklarasikan di bagian pertama pada bagian deklarasi


```
PROGRAM GoToDemo (INPUT, OUTPUT);  
LABEL  
    1000;  
VAR  
    InputChar: CHAR;  
BEGIN  
    write('Masukkan huruf (atau 0 untuk berhenti):');  
    Readln (InputChar);  
    If InputChar = '0' THEN  
        GOTO 1000;  
        {other statement may go here  
        ...}  
  
    1000:  
END.
```

- Pada lingkungan DOS, posisi tulisan di layar dicetak berdasarkan posisi cursor yang sedang aktif.
- GOTO dapat digunakan untuk memindahkan posisi cursor.
- Misalnya sekarang posisi cursor berada di kolom 10, baris 5. Anda ingin mencetak sebuah tulisan pada kolom 20, baris 13, maka anda harus memindahkan cursor ke posisi kolom 20, baris 13, diikuti dengan perintah pencetakan tulisan tersebut.

- Contoh :

```
GotoXY(20,13);
```

```
WriteLn('Tulisan ini dicetak mulai kolom 20, di baris 13');
```

- Contoh lainnya:

```
GotoXY(5,5);
```

```
Write('Turbo');
```

```
GotoXY(WhereX+10,3);
```

```
Write('Pascal');
```

1. Berbeda dengan statemen if yang digunakan untuk memilih dari dua kondisi yang berbeda berdasarkan eskpresi boolean, statemen case digunakan untuk memilih jika terdapat lebih dari dua kondisi yang masing-masing memerlukan penanganan yang berbeda.
2. Variabel yang terlibat dalam statemen case harus bersifat ordinal. Hal ini berbeda untuk statemen if. Yang menjadi pokok perhatian dalam pemilihan adalah menentukan skema yang cocok untuk sebuah masalah yang ditemui berdasarkan analisis kondisi yang telah dilakukan.
3. Statemen GOTO sangat mudah untuk digunakan. Namun Anda jarang melihatnya dalam program Pascal karena menghancurkan struktur dalam penulisan program. Dalam beberapa kasus, GOTO dapat berguna untuk keluar dari IF bersarang (Nested IF) dalam satu lompatan.



POLITEKNIK STATISTIKA STIS

For Better Official Statistics

TERIMA KASIH

