

03

Introduction to Algorithm

using
C / C++

teknik dasar
Algoritma

teknik dasar

Algoritma

menggunakan

Control Statement

if

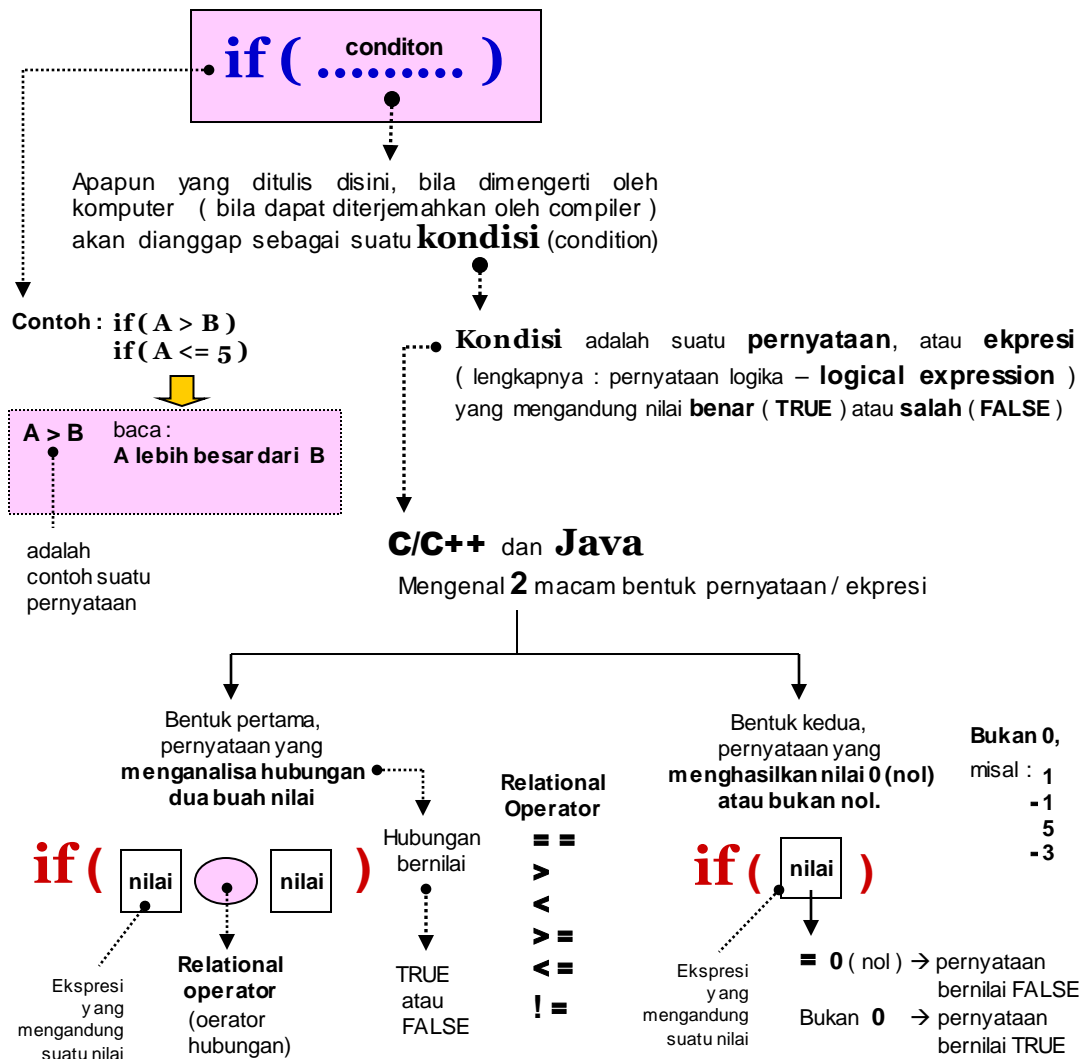
Melakukan Seleksi dengan Control Statement **if** dan **switch**

Book is like a
window

if you open it,
you can see
the sky

3.1 Syntax

Syntax statement **if** :



Relational operator

Relational operator (operator hubungan) yang digunakan dalam bahasa **C / C++** dan **Java** adalah :

Simbol operator	Maksud
<code>==</code>	Sama dengan (Equal to)
<code>></code>	Lebih besar dari (Greater than)
<code><</code>	Lebih kecil dari (Less than)
<code>>=</code>	Lebih besar atau Sama dengan (Greater than or Equal to)
<code><=</code>	Lebih kecil dari atau Sama dengan (Less than or Equal to)
<code>!=</code>	Tidak Sama dengan (Not Equal to)

Penulisan Operator Relatonal dalam instruksi **if()**

<code>if (A<=B)</code>	Benar, tanpa ada satu spasi pun
<code>if (A<=B)</code>	Benar, ada satu atau lebih spasi antara "if" dan "("
<code>if (A<=B)</code>	Benar, ada satu atau lebih spasi antara "(" dan "A" dan antara "B" dan ")"
<code>if (A <= B)</code>	Benar, ada satu atau lebih spasi antara "A" dan "<"
<code>if (A < = B)</code>	Error , ada satu atau lebih spasi antara "<" dan "="

Ada spasi disini ➡ Error

if (nilai relational operator nilai)

Contoh :

```
if( A > 5 )
if( A == B )
if( A+1 < 5 )
if( A+5 > B )
if( A > B+2 )
```

```
if( A-5 > B+3 )
if( A != 5 )
if( X == 'C' )
if( X == ' ' )
```

Memeriksa hubungan dua buah nilai

Memeriksa nilai 0 (nol) atau bukan nol

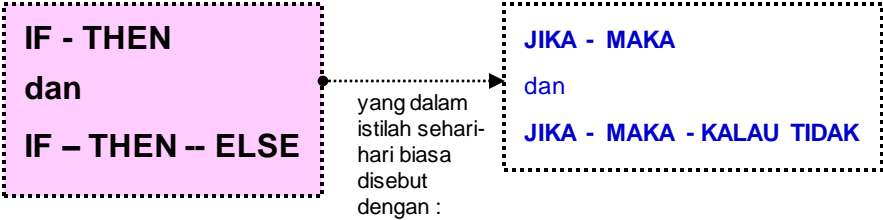
Contoh :

```
if( A )
if( A=B )
if( -1 )
if( A+1 )
if( A=B>C )
```

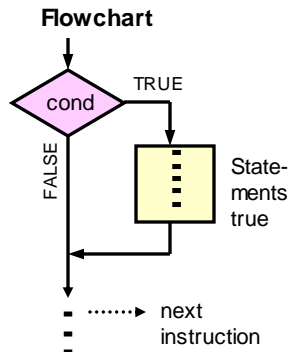
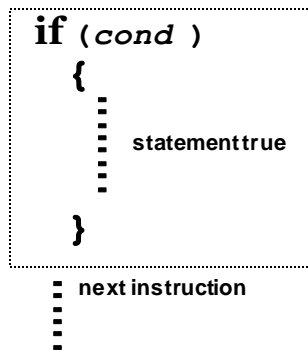
bukan ==

3.2. Bentuk Umum Statement if ()

Bentuk umum statement **if** ada 2 macam , yang dalam pelajaran **algoritma** disebut :



3.2.1. Bentuk Umum instruksi **if ()** bentuk IF-THEN. **if then**



cond = condition,
yang mempunyai nilai
TRUE atau FALSE

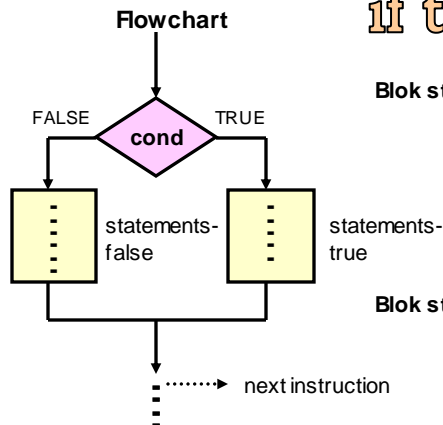
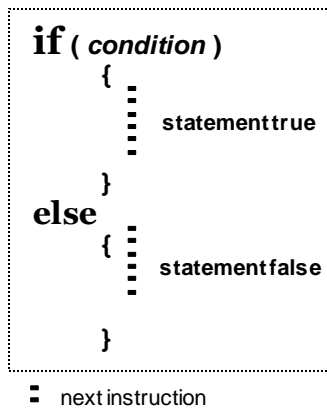
statements-true

terdiri dari satu atau lebih dari satu statement atau instruksi Minimal satu statement. Instruksi-instruksi dibuat menjadi sebuah blok statement yang diapit oleh kurung { dan }

Cara-Kerja :

1. Memeriksa nilai **condition**.
2. ● Bila condition bernilai **TRUE**, maka kerjakan statement-statement yang berada dalam blok **statement-true**. Setelah selesai mengerjakan semua statement yang ada dalam blok **statement-true**, langsung meloncat mengerjakan **next-instruction**
- Bila condition bernilai **FALSE** maka langsung meloncat mengerjakan next- instruction

3.2.2. Bentuk Umum instruksi **if ()** bentuk IF-THEN-ELSE. **if then else**



if then else

Blok statements-true terdiri dari satu atau lebih dari satu statement. Minimal satu statement.

Blok statement-false terdiri dari satu atau lebih dari satu statement. Minimal satu statement

Cara Kerja :

1. Memeriksa nilai **condition**.
2. ● Bila condition bernilai **true**, maka kerjakan statement-statement yang berada dalam blok **statement-true**. Setelah selesai mengerjakan semua statement yang ada dalam blok ini, maka langsung meloncat mengerjakan **next-instruction**
- Bila condition bernilai **false**, maka kerjakan statement-statement yang berada dalam blok **statement-false**. Setelah selesai mengerjakan statement dalam blok ini, maka langsung meloncat mengerjakan next-instruction

3.3. Contoh penulisan / penggunaan if()

3.3.1. Beberapa contoh penulisan / penggunaan
if() bentuk IF-THEN

if then

```
int A= 5, B= 7;
if( A< B )
{
    cout << "Jakarta\n";
}
cout << "Bandung";
```

5<7
true

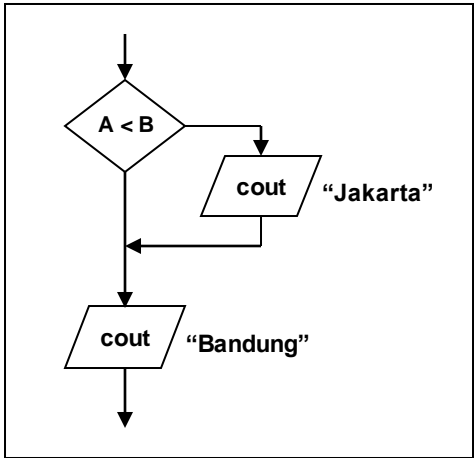
Tercetak : Jakarta
Bandung

```
int A= 9, B= 7;
if( A< B )
{
    cout << "Jakarta\n";
}
cout << "Bandung";
```

9<7
false

Tercetak : Bandung

Statement **if()** untuk semua program pada halaman ini bila dituangkan dalam flowchart, dapat digambar sebagai berikut :



Penulisan instruksi **if()** diatas, dapat dibuat sebagai berikut :

Atau seperti ini :

```
if(A < B)
{ cout << "Jakarta\n";
}
cout << "Bandung";
```

```
if(A<B)
{ cout << "Jakarta\n"; }
cout << "Bandung";
```

```
if(A<B)
    cout << "Jakarta\n";
Cout << "Bandung";
```

Bila dalam satu blok hanya ada satu instruksi maka blok tersebut boleh tidak diapit oleh kurung buka dan kurung tutup. Dengan perkataan lain, bila sebuah blok tidak diapit oleh kurung buka dan kurung tutup, maka milik blok tersebut hanya satu instruksi

```
if(A<B)
    cout << "Jakarta\n";
    cout << "Bandung";
```

```
if(A<B)
    cout <<"Jakarta\n";
    cout <<"Bandung";
```

```
int A=5, B=7;
if(A<B) cout << "Jakarta\n"; cout << "Bandung";
```

```
int A=5, B=7; if(A<B) cout<"Jakarta\n"; cout << "Bandung";
```

3.3.2. Beberapa contoh penulisan / penggunaan if() bentuk IF-THEN-ELSE

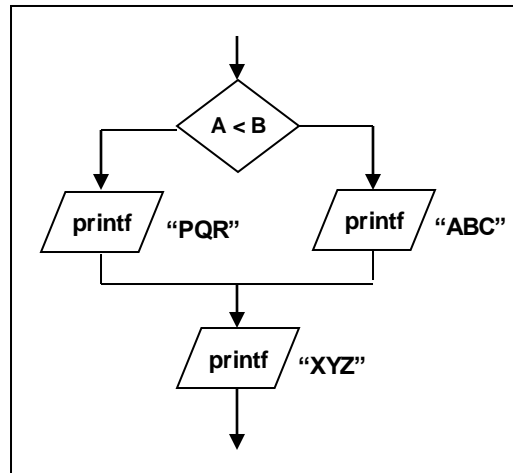
if then else

```
int A=5, B=7;
if(A<B)
{
    cout << "ABC";
}
else
{
    cout << "PQR";
}
cout << "\nXYZ";
```

5 < 7
true

Tercetak : ABC
XYZ

Statement if() untuk semua program pada halaman ini bila dituangkan dalam flowchart, dapat digambar sebagai berikut:



Program diatas dapat ditulis seperti ini

```
if(A<B)
{ cout << "ABC";
}
else
{ cout << "PQR";
}
cout << "\nXYZ";
```

atau ditulis seperti ini

```
if(A<B)
{ cout << "ABC"; }
else
{ cout << "PQR"; }
cout << "\nXYZ";
```

atau ditulis seperti ini

```
if(A<B) { cout << "ABC"; }
else { cout << "PQR"; }
cout << "\nXYZ";
```

Bila sebuah blok instruksi baik blok statement-true, maupun blok statement-false hanya terdiri dari satu instruksi, maka blok tersebut boleh tidak diapit oleh kurung buka "{" dan kurung tutup "}". Dengan perkataan lain, bila sebuah blok tidak diapit oleh tanda kurung, maka blok tersebut hanya terdiri dari satu instruksi.

```
if(A<B)
    cout << "ABC";
else
    cout << "PQR";
cout << "\nXYZ";
```

```
int A=5, B=7;
if(A<B) cout << "ABC";
else cout << "PQR";
cout << "\nXYZ";
```

```
int A=5, B=7;
if(A<B) cout << "ABC"; else cout << "PQR";
cout << "\nXYZ"
```

```
int A=5, B=7;
if(A<B) cout << "ABC"; else cout << "PQR"; cout << "\nXYZ";
```

```
int A=5, B=7; if(A<B) cout << "ABC"; else cout << "PQR"; cout << "\nXYZ";
```

```
int A=5, B=7;
if(A<B)
    cout << "ABC";
else
    cout << PQR";
    cout << nXYZ";
```

tercetak: ABC
 XYZ

Walaupun ditulis satu garis vertikal dengan printf("PQR");, tapi bukan merupakan milik statements-else karena milik statements-else hanya satu instruksi yaitu printf("PQR");

Satu blok yang terdiri lebih dari satu instruksi, bila tidak diapit oleh tanda kurung maka akan menimbulkan **error**, seperti dicontohkan berikut ini.

```
int A=5, B=7;
if(A<B)
    cout << ABC";
    cout << EFGH";
else
    cout << PQR";
    cout << nXYZ";
```

Error,
karena cout ABC"; dan cout << "EFGH";
Tidak digabung menjadi satu block
yaitu tidak diapit oleh tanda kurung "{ dan "}",
maka compiler menganggap statements-true hanya terdiri dari satu instruksi, yaitu cout << ABC;
sehingga instruksi cout << EFGH"; dianggap sebagai next instruction
Dengan demikian instruksi else tak punya hubungan dengan instruksi if diatasnya.
else yang berdiri sendiri, tanpa **if** akan menyebabkan **error**

3.3.3. Beberapa contoh aspek penulisan condition

1.

```
int A = 5;
if(A > 2)
    cout << ABC";
else
    cout <<"XYZ";
    cout << "\n%i", A;
```

Tercetak: ABC
 5

kondisi (5 > 2)
bernilai TRUE dan Nilai A tidak berubah

2.

```
int A = 5;
if( A )
    cout << ABC";
else
    cout << XYZ";
    cout << n%i", A;
```

Tercetak: ABC
 5

kondisi (5) adalah TRUE karena 5 bukan nol dan Nilai A tidak berubah

Relational operator

nilai nilai

if ()

==
>
<
>=
<=
!=

if (nilai)

- Bila **nilai = 0 (nol)** maka **kondisi** bernilai FALSE
- Bila **nilai bukan 0 Positip** atau **negatip** maka **kondisi** bernilai TRUE

Untuk program contoh 1 dan 2, Flow chartnya dapat digambarkan sebagai berikut :

```
graph TD
    START([START]) --> A5[A = 5]
    A5 --> Kondisi{ kondisi }
    Kondisi --> XYZ[/cout "XYZ"/]
    Kondisi --> ABC[/cout "ABC"/]
    XYZ --> CoutA[/cout A/]
    ABC --> CoutA
    CoutA --> END([END])
```


3.

```
int A = 5;
if( A-5 )
    cout << "ABC \n";
else
    cout << "XYZ \n";
cout << A;
```

kondisi
(0)
adalah FALSE
tapi
(A-5)
Tidak menyebabkan
Nilai A berubah

Tercetak : XYZ
5

4.

```
int A = 5;
if( A-7 )
    cout << "ABC \n";
else
    cout << "XYZ \n";
cout << A;
```

kondisi
(-2)
adalah TRUE
dan
(A-7)
Tidak menyebabkan
Nilai A berubah

Tercetak : ABC
5

5.

```
int A = 5;
if(A = 3)
    cout << "ABC \n";
else
    cout << "XYZ \n";
cout << A ;
```

A, yang
sebelumnya
nilainya = 5,
sekarang
bernilai 3

3
3 bukan nol
sehingga kondisi
bernilai TRUE

Tercetak : ABC
3

6.

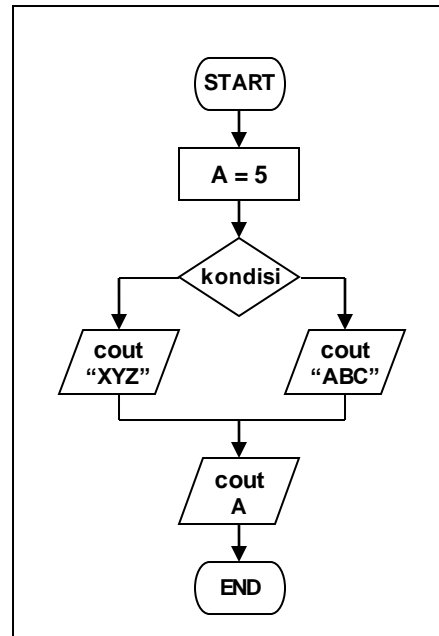
```
int A = 5
if(A = A-5 )
    cout << "ABC \n";
else
    cout << "XYZ \n";
cout << A;
```

$A = A - 5$
 $5 - 5 = 0$
 $A = 0$

Kondisi
bernilai
FALSE

Tercetak : XYZ
0

Untuk program no. 3, 4, 5, 6, pada halaman ini, flowchartnya sama sebagai berikut. Tiap-tiap program mempunya kondisi **if** tersendiri



Perhatikan :

<p>1. <pre>cout << (9>5);</pre></p> <p>Tercetak : 1</p> <p>(9>5) bernilai TRUE Nilai TRUE kalau dinyatakan dengan bilangan integer adalah = 1</p>	<p>2. <pre>int A; A = (9>5); cout << A ;</pre></p> <p>Tercetak : 1</p> <p>1 A</p>	<p>3. <pre>int A; A = (9<5); cout << A ;</pre></p> <p>Tercetak : 0</p> <p>0 A</p>
---	--	--

7.

```
int A=5, B=7;
if(A+4 > B+1)
    cout << ABC";
else
    cout << XYZ";
cout << "\n" << A ;
cout << "\n" << B;
```

•

(9 > 8)

bernilai TRUE

tercetak: ABC

5

7

•

Nilai A & B tidak berubah

8.

```
int A=5, B=7;
if(A+4 > B+1)
    cout << ABC";
else
    cout << XYZ";
cout << "\n" << A ;
cout << "\n" << B;
```

•

(9 > 8)

bernilai TRUE

tercetak: ABC

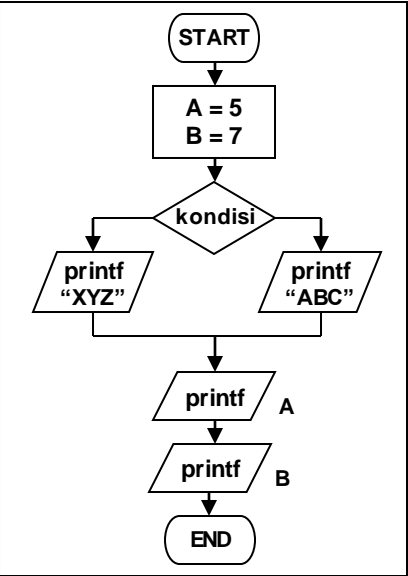
5

7

•

Nilai A & B tidak berubah

Semua program pada halaman ini, flowchartnya sama sebagai berikut. Tiap-tiap program mempunyai kondisi **if** tersendiri



9.

```
int A=5, B=7;
if(A=A+4 > B+1)
    cout << ABC";
else
    cout << XYZ";
cout << "\n" << A ;
cout << "\n" << B;
```

•

Bagaimana bila diganti dengan:

if (A=A+4 > B+3)

Dalam program tertulis : **if (A=A+4 > B+1)**
Oleh Compiler dibaca : **if (A = A+4 > B+1)**

A+4=9

B+1=8

Sehingga menjadi : **if (A = 9>8)**

9 > 8 bernilai True

nilai True adalah 1

Sehingga menjadi : **if (A = 1)**

1

A

A diisi dengan 1, kemudian dites sebagai suatu condition

Sehingga menjadi : **if (1)**

Menurut bahasa C, 0 (nol) berarti FALSE selain 0 (nol) berarti TRUE

Disini terlihat operator ">" dilaksanakan lebih dulu dari operator "=" . Kalau tidak mahir dengan urutan operator, lebih baik menghindari penulisan seperti ini

tercetak: ABC

1

7

3.4. Contoh program sederhana menggunakan instruksi if ()

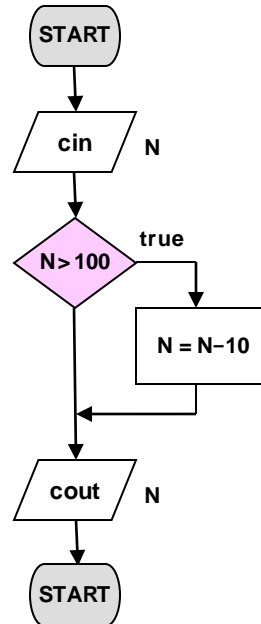
Soal-11.

Diberikn program sebagai berikut :

```
#include <iostream.h>
void main()
{ int N;
  cin >> N;;
  if (N > 100)
    N = N - 10;
  cout << N ;
}
```

Apa yang tercetak , bila diinput nilai untuk N :

- 100 → Jawab : 100
- 105 → Jawab : 95
- 98 → Jawab : 90



Soal yang cocok untuk program seperti ini, misalnya :



Sebuah toko memberikan potongan harga yang harus dibayar sebesar 10 rupiah, bila nilai belanjaan lebih besar dari 100 rupiah.



Susun program untuk menginput nilai belanjaan, dan mencetak harga yang dibayar.

Soal-12.

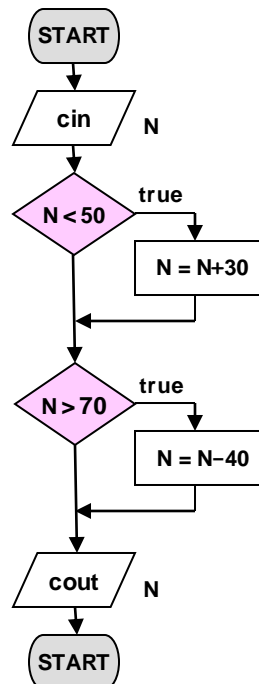
Diberikan program sebagai berikut :

```
#include <iostream.h>
void main()
{ int N;
  cin >> N ;
  if (N < 50)
    N = N + 30;
  if (N > 70)
    N = N - 40;
  cout << N ;
}
```

Catatan :
Nilai N yang diinput berkisar dari 1 - 100

Apa yang tercetak , bila diinput nilai untuk N :

- 35 → Jaw ab : 65
- 45 → Jaw ab : 35
- 55 → Jaw ab : 55
- 75 → jaw ab : 35



Program ini menyebabkan nilai **N** yang diinput (yang kemungkinan bernilai 1 -100), dibuat menjadi nilai-nilai yang mempunyai rentang **30 - 70**

1	}	Berubah Menjadi 31 - 79
2		
-		
-		
48	}	Tetap, tidak berubah
49		
50		
51		
52	}	Berubah menjadi 31 - 60
-		
-		
69		
70	}	
71		
72		
-		
-	}	
-		
100	}	
-		

Soal-13.

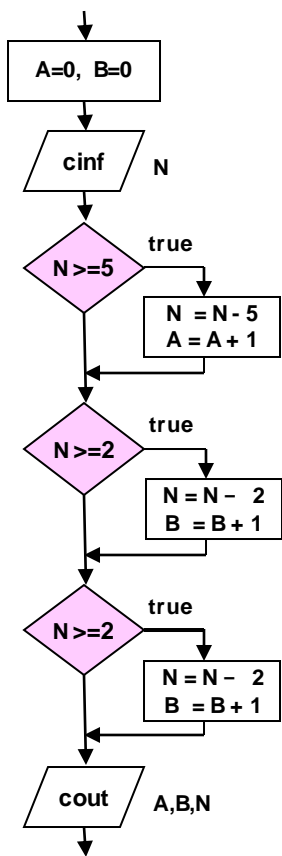
Diberikan program sebagai berikut :

Catatan :
Nilai N yang Diinput berkisar antara 1 - 9

```
#include <iostream.h>
void main()
{ int N, A=0, B=0 ;
  cin >> N ;
  if ( N >= 5 )
  { N = N - 5;
    A = A + 1;
  }
  if ( N >= 2 )
  { N = N - 2;
    B = B + 1;
  }
  if ( N >= 2 )
  { N = N - 2;
    B = B + 1;
  }
  cout << A << B << N;
}
```

Apa yang tercetak , bila diinput nilai untuk N :

5 → Jaw ab : 1 0 0
4 → Jaw ab : 0 2 0
2 → Jaw ab : 0 1 0



Pada sebuah toko tersedia uang koin dengan pecahan nilai :

5 sen
2 sen , dan
1 sen

Yang digunakan untuk kembalian uang yang bernilai 1- 9 sen

Program ini, menginput nilai uang kembalian, yang diterima oleh variabel N, dan mencetak jumlah keping koin untuk kembalian uang 1- 9 sen

Bila diinput Untuk N	Maka tercetak
9	1 2 0
8	1 1 1
7	1 1 0
6	1 0 1
5	1 0 0
4	0 2 0
3	0 1 1
2	0 1 0
1	0 0 1

Soal-14.

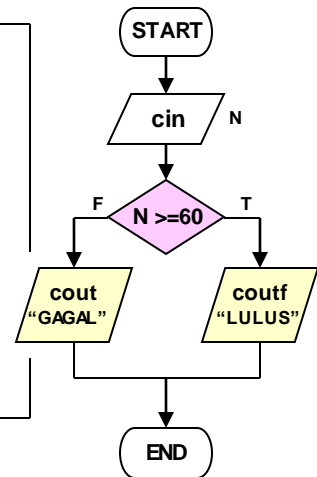
Susun program untuk menginput sebuah nilai integer (nilai ujian mahasiswa) kemudian cetak perkataan **"LULUS"**, bila nilai tersebut ≥ 60 atau cetak perkataan **"GAGAL"** bila nilai tersebut < 60 .

Perhatikan blok instruksi **if** pada program ini. Empat baris instruksi ini, dapat ditulis menjadi satu baris:

Jawab Soal-4. Cara-1:

```
#include <iostream.h>
void main()
{ int N;
  cout << "Input sebuah nilai: ";
  cin >> N;

  if (N >= 60)
    cout << "LULUS" ;
  else
    cout << "GAGAL" ;
}
```



```
if (N >= 60) cout << "LULUS"; else cout << "GAGAL";
```

Bahkan bisa ditulis menjadi :

```
(N>=60)? cout << "LULUS": cout << "GAGAL";
```

atau

```
N>=60 ? cout << "LULUS": cout << "GAGAL";
```

atau menjadi :

```
cout << ( (N >= 60)? "LULUS" : "GAGAL" );
```

atau

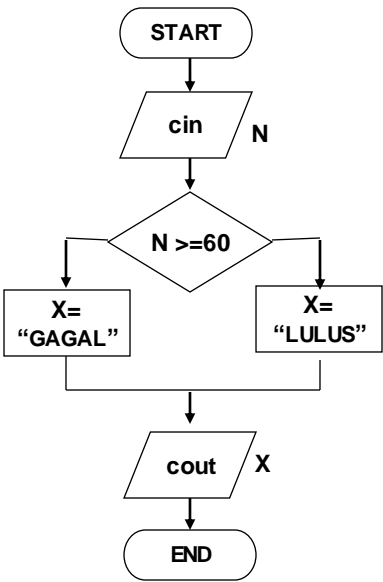
```
cout << ( (N >= 60)? ("LULUS<< "): ("GAGAL<< " ) );
```

atau

```
cout << ( N >= 60? "LULUS" : "GAGAL" );
```

Jawab-14. Cara-2:

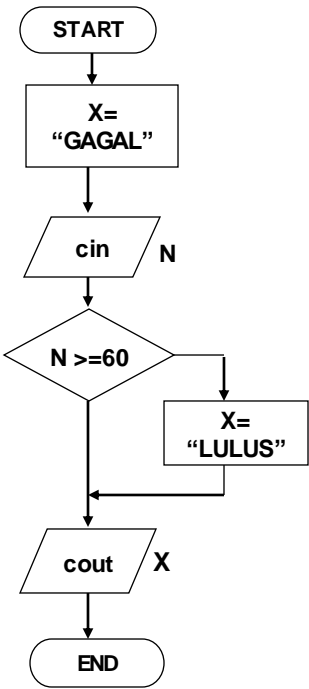
```
#include <iostream.h>
#include<string.h>
void main()
{
    int N;
    char X[10];
    cout << "\n Input  Nilai :
";
    cin >> N ;
    if (N >=60)
        strcpy (X, "LULUS");
    else
        strcpy(X, "GAGAL");
    cout << "\n " << X;
}
```



Disini untuk mengisi string X, digunakan instuksi **strcpy()** , suatu fungsi pustaka untuk **string copy**
Instruksi ini memerlukan **#include<string.h>**
Instruksi ini berlaku juga pada **C++**.

Jawab-14. Cara-3:

```
#include <iostream.h>
#include <string.h>
main()
{int N;
    char X[10];
    strcpy(X, "GAGAL");
    cout << "\n Input nilai : ";
    cin >> N ;
    if (N >= 60)
        strcpy(X, "LULUS");
    cout << "\n " << X;
}
```



Soal-15.

Susun program untuk menginput sebuah nilai integer, bilangan bulat positif lebih besar dari nol, kemudian cetak perkataan "EVEN", bila bilangan tersebut merupakan bilangan **genap**, sebaliknya cetak perkataan "ODD" bila bilangan tersebut merupakan bilangan **ganjil**.

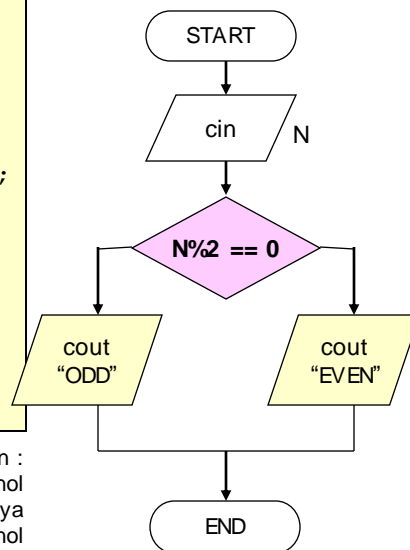
Jawab Soal-15. Cara-1:

```
#include <iostream.h>
void main()
{ int N;
  cout << "Input nilai : ";
  cin >> N;

  if (N % 2 == 0)
    cout << "EVEN";
  else
    cout << "ODD";
}
```

Catatan :
Bilangan bulat positif lebih besar dari nol
genap : 2, 4, 6, 8, dan seterusnya
Bilangan bulat positif lebih besar dari nol
ganjil : 1, 3, 5, 7, dan seterusnya

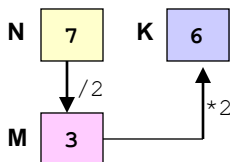
Flowchart



Keterangan :

Bilangan genap bila dibagi dengan 2,
maka sisa pembagiannya = 0 (nol)
(Secara umum dikatakan $N \text{ Mod } 2 = 0$)

Bila tidak menggunakan operator "%" atau
operator Modulus, dapat juga
menggunakan pembagian biasa dengan
cara sebagai berikut:



$M = N / 2$;
 $K = M * 2$;
Bila $N == K$, berarti N
bilangan bulat.
(disini perlu variabel
M dan K)

```
int N, M, K;
Cin >> N;
M = N / 2;
K = M * 2;
if (N == K)
  cout << "EVEN";
else
  cout << "ODD";
```

atau:

```
int N;
cin >> N;
if (N == (N/2) * 2 )
  cout << "EVEN";
else
  cout << "ODD";
```

Cara-2 dan
seterusnya dapat
d disesuaikan
dengan
contoh yang
sudah ada

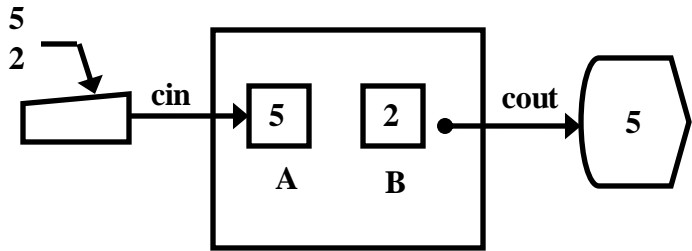
Instruksi: `if (N == (N/2) * 2)`

Dapat diganti menjadi : `if (N == N/2 * 2)`

Soal-16

Susun program dalam bahasa C++
untuk menginput 2 (dua) buah bilangan bulat,
kemudian mencetak salah satu bilangan yang nilainya terbesar.

MISAL NILAI YANG DIINPUT ADALAH 5 DAN 2



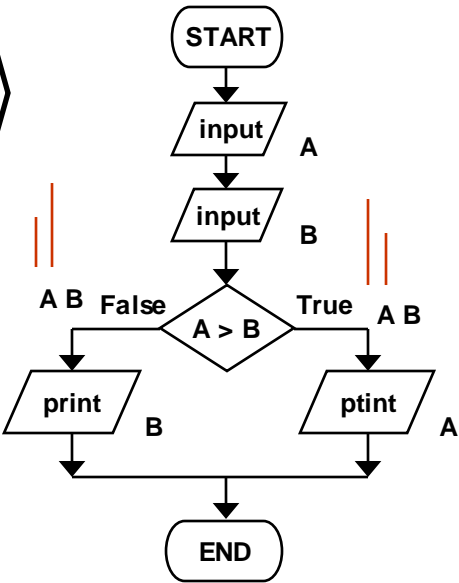
Cara-1

```
#include <iostream.h>
void main()
{ int A, B;
  cin >> A;
  cin >> B;
  if (A > B)
  { cout << A;
  }
  else
  { cout << B;
  }
}
```

```
if (A > B)
{ cout << A; }
else
{ cout << B; }
```

```
if (A > B)
  cout << A;
else
  cout << B;
```

```
if (A > B) cout << A;
else cout << B;
```



```
if (A > B)
  cout << A ;
else
  cout << B ;
```

```
if (A < B)
  cout << B ;
else
  cout << A ;
```

```
if (B > A)
  cout << B ;
else
  cout << A ;
```

```
if (B < A)
  cout << A ;
else
  cout << B ;
```


Cara-2

```
#include <iostream.h>
void main()
{
    int A,B,C, MAX;

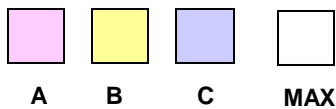
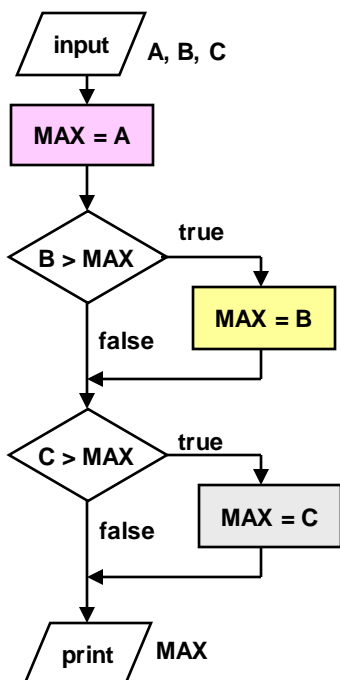
    cin >> A >> B >> C;

    MAX = A;

    if (B > MAX)
        MAX = B;

    if (C > MAX)
        MAX = C;

    cout << MAX;
}
```

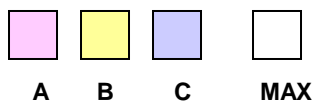
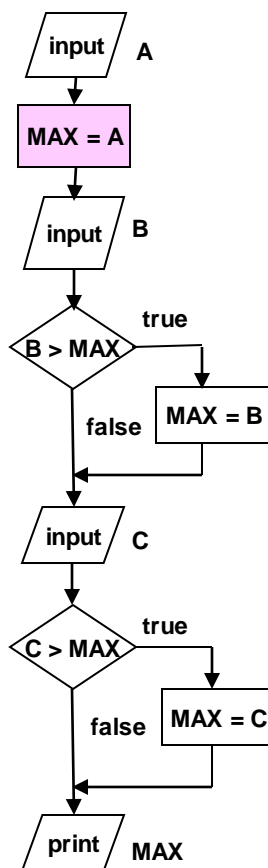


Menggunakan 4 variabel

Cara-3

```
#include <iostream.h>
void main()
{
    int A,B,C, MAX;

    cin >> A;
    MAX = A;
    cin >> B;
    if (B > MAX)
        MAX = B;
    cin >> C;
    if (C > MAX)
        MAX = C;
    cout << MAX;
}
```

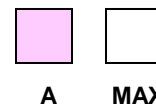
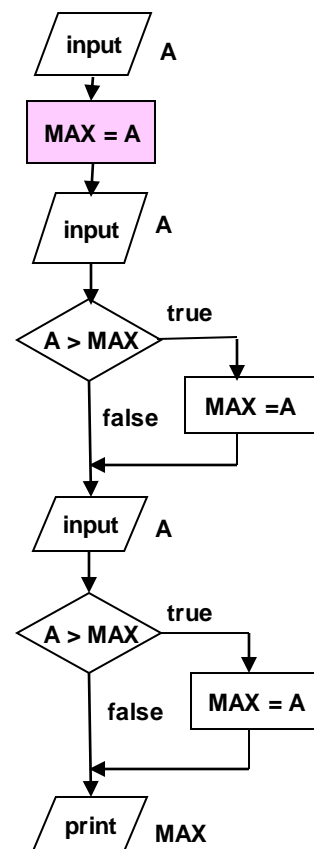


Menggunakan 4 variabel

Cara-4

```
#include <iostream.h>
void main()
{
    int A, MAX;

    cin >> A;
    MAX = A;
    cin >> A;
    if (A > MAX)
        MAX = A;
    cin >> AC;
    if (A > MAX)
        MAX = A;
    cout << MAX;
}
```



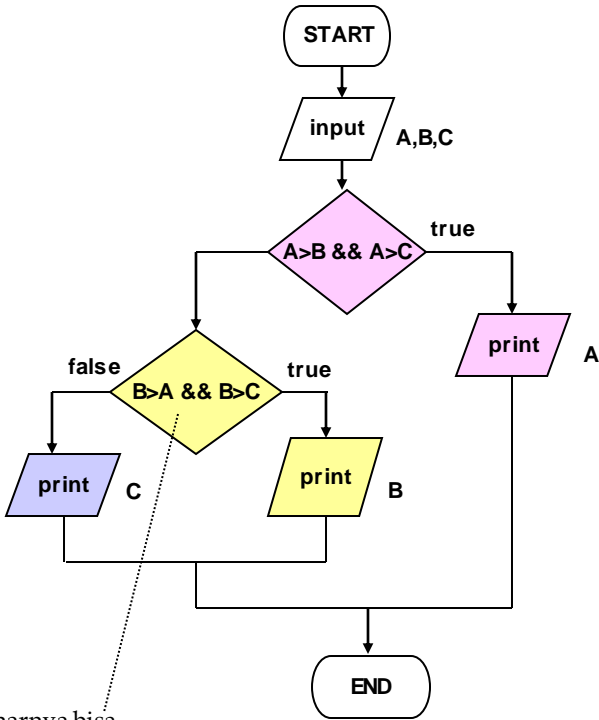
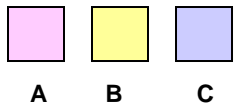
Hanya menggunakan 2 variabel

Cara-5

Dengan menggunakan
Logical Operator AND (&&)

```
#include <iostream.h>
void main()
{
    int A,B,C;
    cin >> A >> B >> C;

    if (A>B && A>C)
        cout << A;
    else
        { if (B>A && B>C)
            cout << B;
          else
            cout << C;
        }
}
```



Ini sebenarnya bisa diganti cukup dengan :

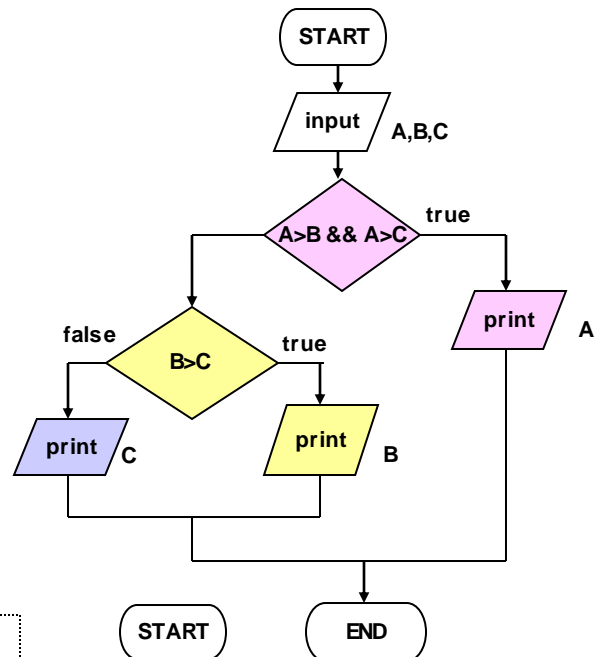


Sebenarnya disini cukup membanding-kan B dengan C karena bila pemeriksaan pertama if (A>B && A>C) hasilnya FALSE maka A sudah bukan yang terbesar, tinggal B atau C yang terbesar

Hanya untuk keseragaman saja dibuat lagi **if** (B>A && B>C), dengan cara ini berpikirnya lebih mudah

```
#include <iostream.h>
void main()
{  int A,B,C;
   cin >> A >> B >> C;

   if (A>B && A>C)
       cout << A;
   else
       { if (B>C)
           cout << B;
         else
           cout << C;
       }
}
```



Cara-6

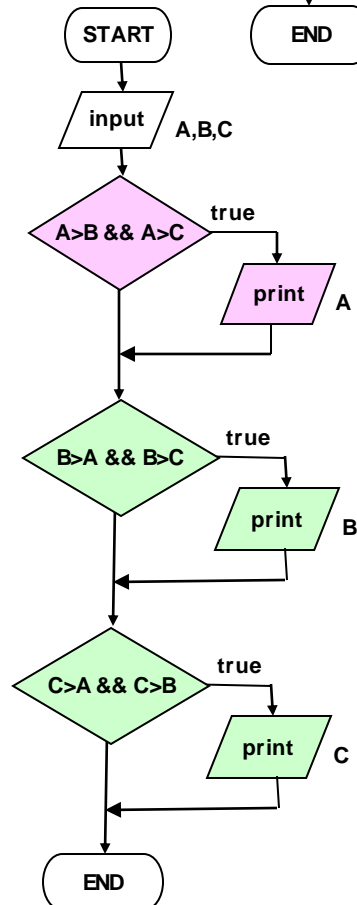
```
#include <iostream.h>
void main()
{  int A,B,C;
   cin >> A >> B >> C;
   if (A>B && A>C)
       cout << A;
   if (B>A && B>C)
       cout << B;
   if (C>A && C>B)
       cout << C;
}
```

Ketiga if () diatas dapat ditulis sebagai berikut :

```
if (A>B && A>C)  cout << A;
if (B>A && B>C)  cout << B;
if (C>A && C>B)  cout << C;
```

Berpikir lebih mudah, walaupun computer time lebih besar karena selalu melalui atau melaksanakan 3 if

Dalam pemrograman selalu ada kompromi antara kemudahan membuat program dengan efisiensi.



Soal-17

Susun program untuk menginput empat (4) buah bilangan bulat kemudian mencetak salah satu bilangan yang nilainya terbesar:

Jawab Cara-1

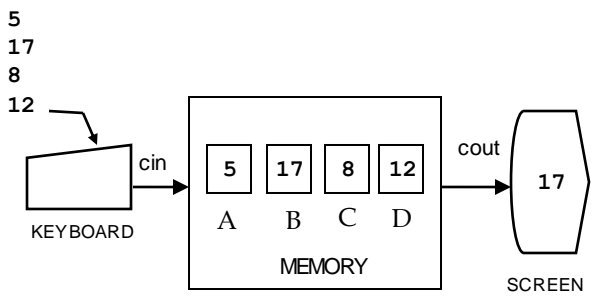
Menggunakan 5 variabel, 4 variabel untuk input.
1 variabel untuk MAX

```
cin >> A >> B >> C >> D;

if(A > B)
{
  if(A > C)
  {
    if(A > D)
    {
      cout << A;
    }
    else
    {
      cout << D;
    }
  }
  else
  {
    if(C > D)
    {
      cout << C;
    }
    else
    {
      cout << D;
    }
  }
}
else
{
  if(B > C)
  {
    if(B > D)
    {
      cout << B;
    }
    else
    {
      cout << D;
    }
  }
  else
  {
    if(C > D)
    {
      cout << C;
    }
    else
    {
      cout << D;
    }
  }
}
}
```

Terlihat disini, makin banyak data yang diinput, dan dipilih salah satu yang terbesar cara-1 semakin rumit dan semakin panjang programnya.

Oleh karena itu dipilih teknik atau cara lain yang penulisannya bisa disingkat dengan membuat LOOP menggunakan control statement FOR atau WHILE

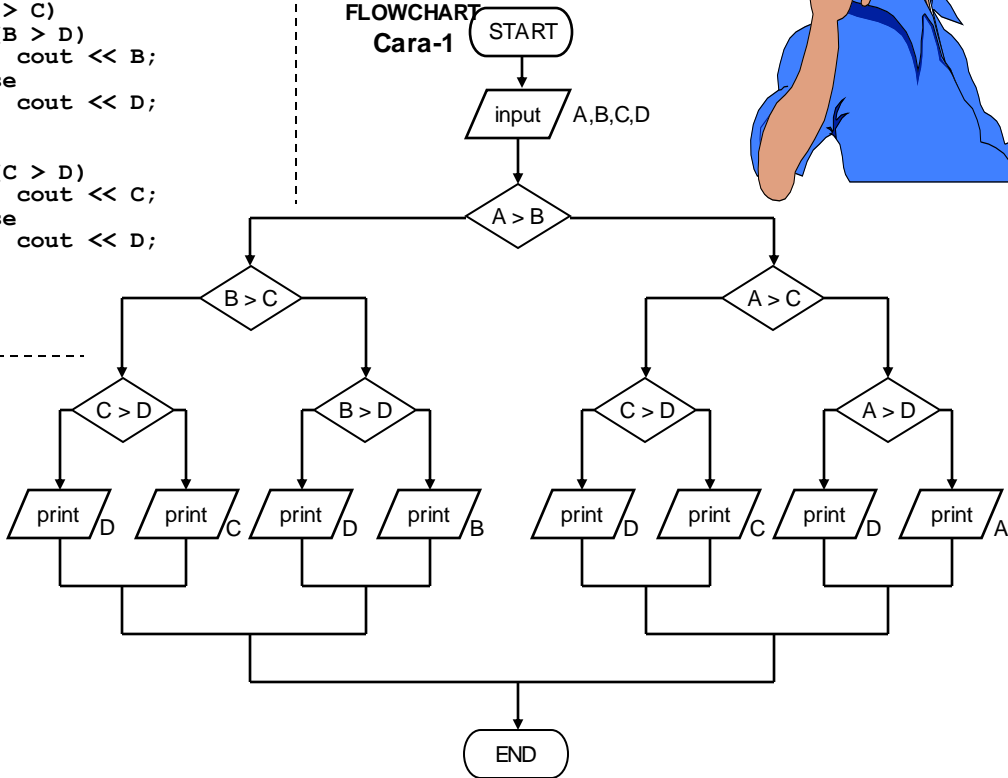


Bila bilangan yang diinput adalah 5, 17, 8 dan 12, maka tentunya akan tercetak 17

Ini untuk 4 bilangan bagaimana kalau 100 bilangan

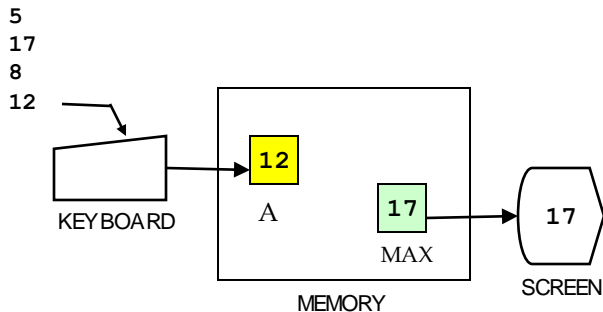


FLOWCHART
Cara-1



Jawab Cara-4 (Cara-2 dan Cara-3 seperti pada Soal-16, tidak diperlihatkan)

Dalam menjawab Soal-1 sebelumnya, sudah diberikan 6 contoh cara menyelesaikan soal. Karena Soal-2 tidak berbeda logikanya dengan Soal-1, maka untuk menjawab Soal-2, hanya diambil cara-1 yaitu cara klasik, dan Cara-4, yaitu cara yang diambil sebagai dasar penyelesaian masalah. Cara lain dapat digunakan untuk persoalan yang khusus.



Cara ini hanya menggunakan 2 variabel.
Variabel A khusus menerima input
Variabel MAX khusus menampung nilai terbesar

```
#include <iostream.h>
void main()
{ int A, MAX;
  cin >> A;
  MAX = A;

  cin >> A;
  if (A > MAX)
    MAX = A;

  cin >> A;
  if (A > MAX)
    MAX = A;

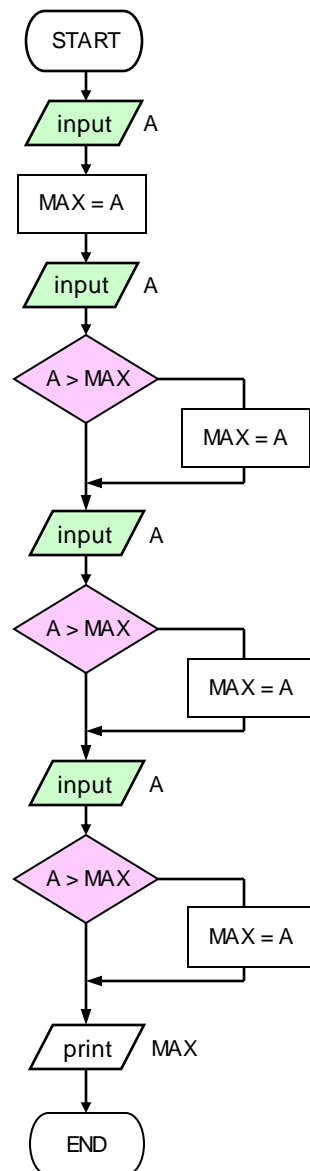
  cin >> A;
  if (A > MAX)
    MAX = A;

  cout<< MAX;
}
```

Terlihat ada 3 block instruksi yang sama

Apabila block instruksi ini ditulis dalam suatu LOOP, maka penulisannya cukup satu kali, tapi pelaksanaannya 3 kali.

Loop dapat dibuat dengan control statement FOR, atau WHILE, atau DO WHILE



Jawab Cara-7

Seperti terlihat pada Cara-4, ada k 3 kelompok instruksi yang sama yaitu :

```
cin >> A;
if( A> MAX)
    MAX = A;
```

Dalam bahasa pemrograman, ada fasilitas untuk melaksanakan suatu kelompok instruksi secara berulang ulang sebanyak yang kita ingini. Pekerjaan yang dilaksanakan secara berulang-ulang disebut loop. Dua diantara instruksi yang dapat melakukan loop adalah control statement **for()** dan **while()**. Walaupun teori for dan while belum dipelajari, tapi disini langsung digunakan sebagai contoh.

Perhatikan program berikut ini.

Cara-7a.

Membuat LOOP dengan control statement **FOR**.....

```
#include <iostream.h>
void main()
{
    int A, i, MAX;
    cin >> A;
    MAX = A;

    for (i=1; i <= 3; i=i+1)
    {
        cin >> A;
        if (A > MAX)
            MAX = A;
    }

    cout << MAX;
}
```

Cara-7b.

Membuat LOOP dengan control statement **WHILE**.....

```
#include <iostream.h>
void main()
{
    int A, i, MAX;
    cin >> A;
    MAX = A;

    i = 1;
    while (i <= 3)
    {
        cin >> A;
        if (A > MAX)
            MAX = A;
        i = i + 1;
    }

    cout << MAX;
}
```

Dari contoh diatas, bila yang diinput 100 buah bilangan, maka cukup mengganti angka 3 pada instruksi for() atau while() dengan 99.

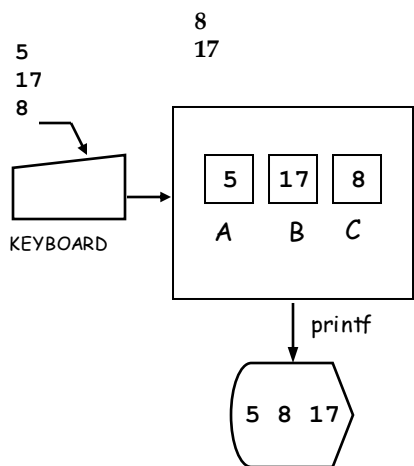
Sehingga menjadi :

```
for(i=1; i<=99; i=i+1)    dan    while(i<=99)
```

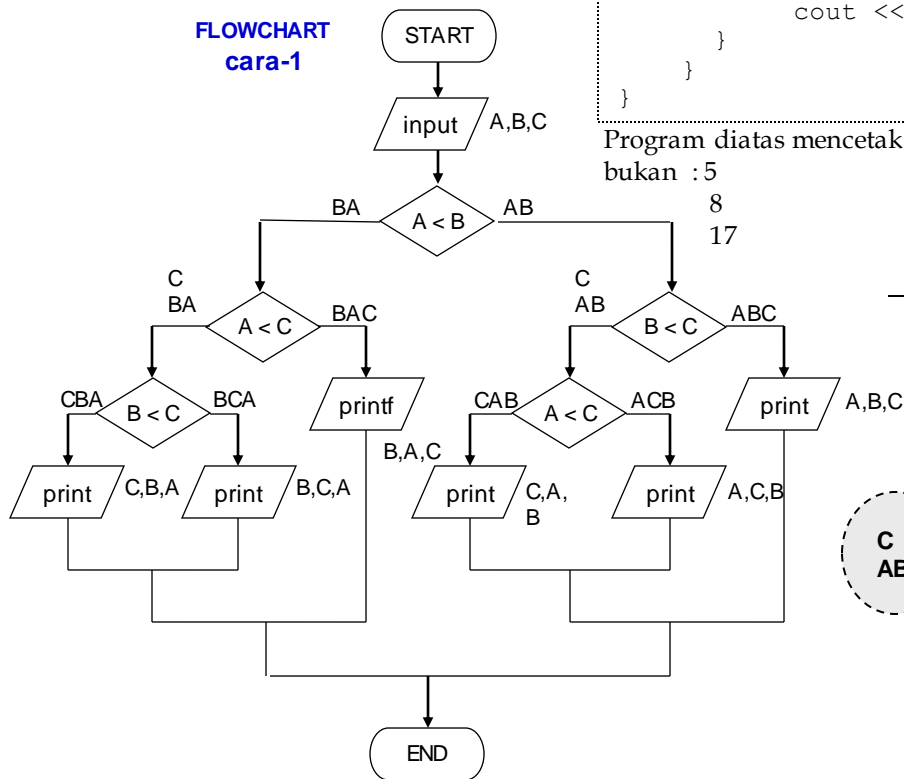
Soal-18

Susun program untuk menginput tiga (3) buah bilangan bulat (misal A, B dan C dimana $A < B < C < A$), kemudian mencetak ketiga nilai tersebut urut dari kecil ke besar.

Bila bilangan yang diinput adalah 5, 17 dan 8, maka tentunya akan tercetak 5 8 17 atau tercetak :



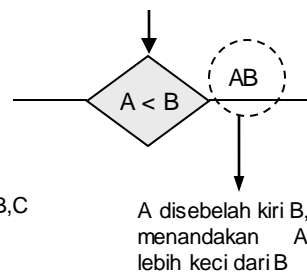
FLOWCHART
cara-1



Jawab Cara-1

```
#include <iostream.h>
void main()
{int A,B,C;
cin >> A >> B >> C;
if(A<B)
{if(B<C)
cout << A << B << C;
else
{if(A<C)
cout << A << C << B;
else
cout << C << A << B;
}
}
else
{if(A<C)
cout << B << A << C;
else
{if(B<C)
cout << B << C << A;
else
cout << C << B << A;
}
}
}
```

Program diatas mencetak 5 8 17
bukan : 5



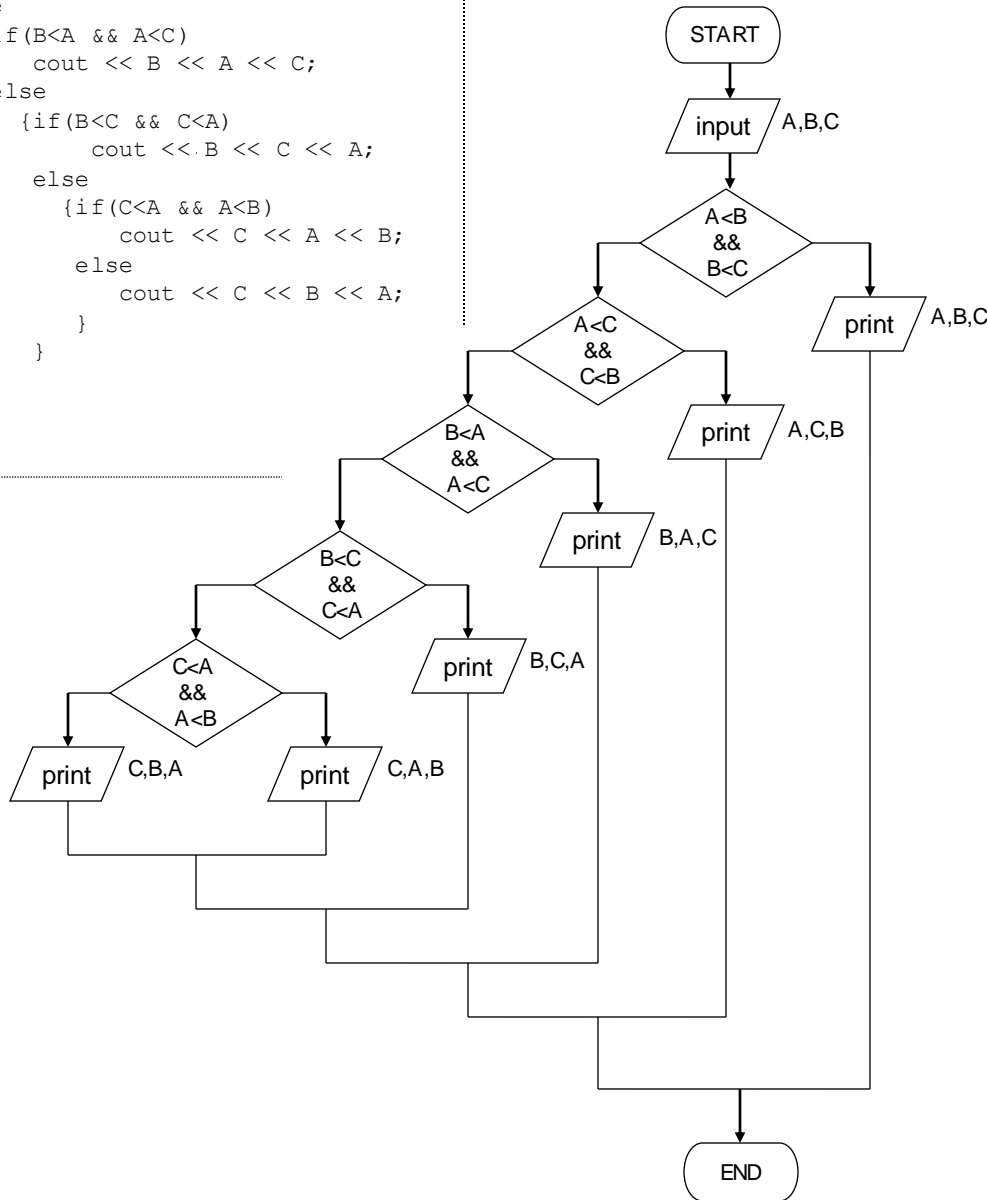
Ini menandakan bahwa A dan C kedua-duanya lebih kecil dari B, sedangkan A dan C sendiri belum diperiksa mana yang lebih besar atau mana yang lebih kecil

Jawab Cara-2

```
#include <iostream.h>
void main()
{int A,B,C;
cin >> A >> B >> C;
if(A<B && B<C)
    cout << A << B << C;
else
    {if(A<C && C<B)
        cout << A << C << B;
    else
        {if(B<A && A<C)
            cout << B << A << C;
        else
            {if(B<C && C<A)
                cout << B << C << A;
            else
                {if(C<A && A<B)
                    cout << C << A << B;
                else
                    cout << C << B << A;
                }
            }
        }
    }
}
```

Kemungkinan nilai yang di-input

A	B	C	Tercetak :
5	7	9	5 7 9
5	9	7	5 7 9
7	5	9	5 7 9
7	9	5	5 7 9
9	5	9	5 7 9
9	7	5	5 7 9



Soal-19

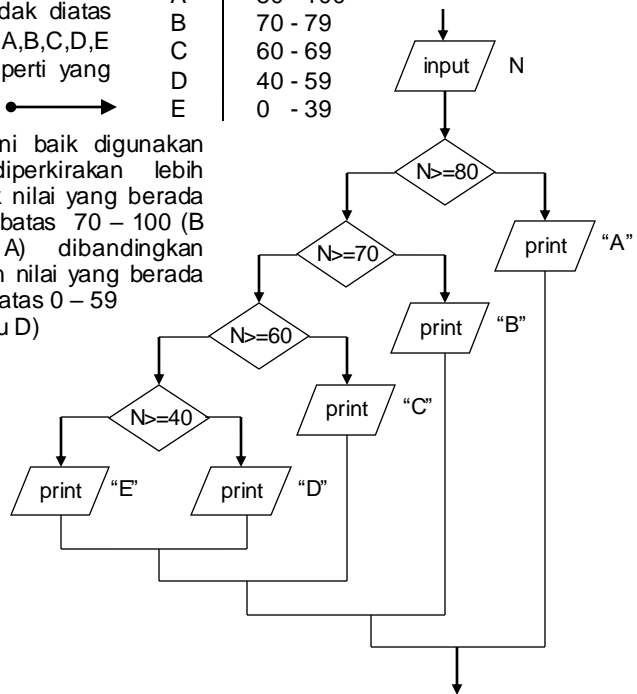
Susun program untuk menginput sebuah nilai integer (nilai ujian mahasiswa) Anggap nilai yang diinput berada dalam batas (0-100), artinya tidak dibawah 0 dan tidak diatas 100. Kemudian cetak nilai dalam bentuk : A,B,C,D,E sesuai dengan nilai numerik yang diinput, seperti yang dijelaskan dalam tabel yang diberikan.

Cetak huruf	Bila Nilai berada dalam batas :
A	80 - 100
B	70 - 79
C	60 - 69
D	40 - 59
E	0 - 39

Jawab. Cara-1

```
Cin >> N;
if(N >= 80)
    cout << "A" ;
else
    {if(N >= 70)
        cout << "B";
    else
        {if(N >= 60)
            cout << "C";
        else
            {if(N >= 40)
                cout << "D";
            else
                cout << "E";
            }
        }
    }
}
```

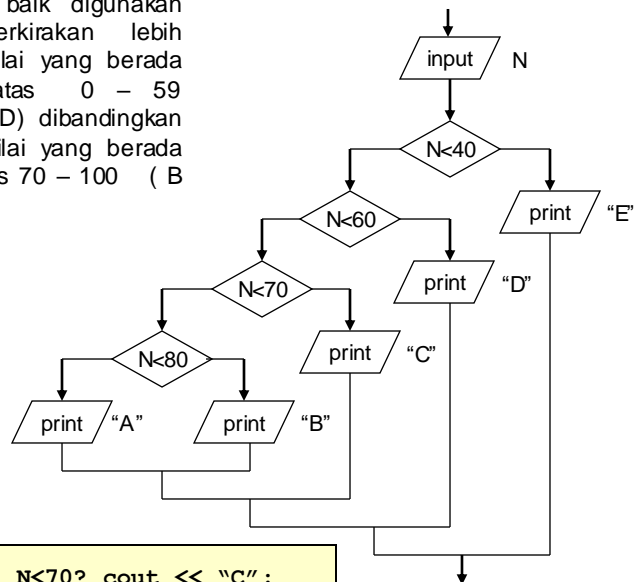
Cara ini baik digunakan bila diperkirakan lebih banyak nilai yang berada dalam batas 70 – 100 (B atau A) dibandingkan dengan nilai yang berada pada batas 0 – 59 (E atau D)



Jawab. Cara-2

```
Cin >> N;
if(N < 40)
    cout << "E";
else
    {if(N < 60)
        cout << "D";
    else
        {if(N < 70)
            cout << "C";
        else
            {if(N < 80)
                cout << "B";
            else
                cout << "A";
            }
        }
    }
}
```

Cara ini baik digunakan bila diperkirakan lebih banyak nilai yang berada dalam batas 0 – 59 (E atau D) dibandingkan dengan nilai yang berada pada batas 70 – 100 (B atau A)



Bagian **if** dan **printf** pada program diatas dapat ditulis dengan cara lain sebagai berikut:

```
N<40? cout << "E" : N<60? cout << "D": N<70? cout << "C":
N<80? cout << "B" : coutt << "A";
```

atau

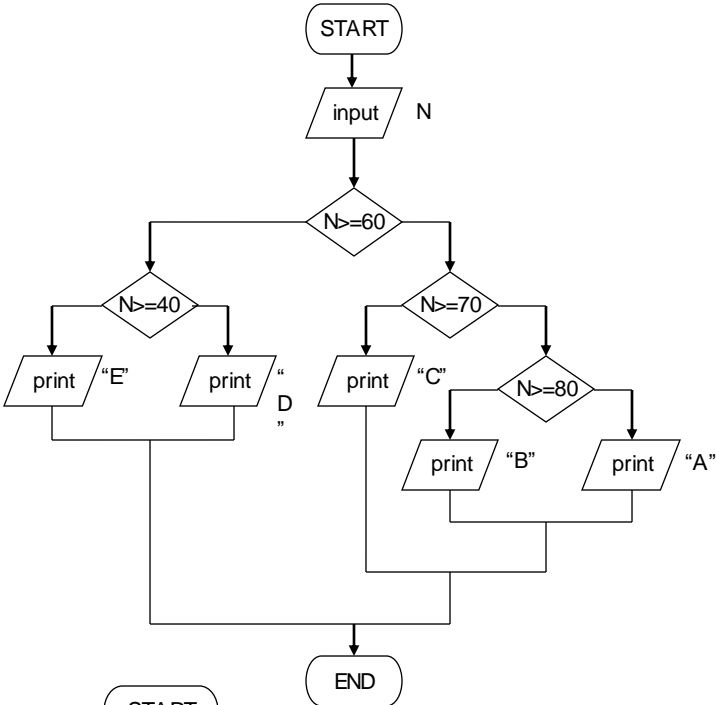
```
cout << (N<40? "E": N<60? "D": N<70? "C": N<80? "B": "A") ;
```

Mirip menggunakan IF pada Excel

Jawab. Cara-3

```
#include <iostream.h>
void main()
{int N;
 cin >> N;
 if(N >= 60)
   if(N>=70)
     if (N>=80)
       cout << "A";
     else
       cout << "B";
   else
     cout << "C";
 else
   if(N >= 40)
     cout << "D";
   else
     cout << "E";
}
```

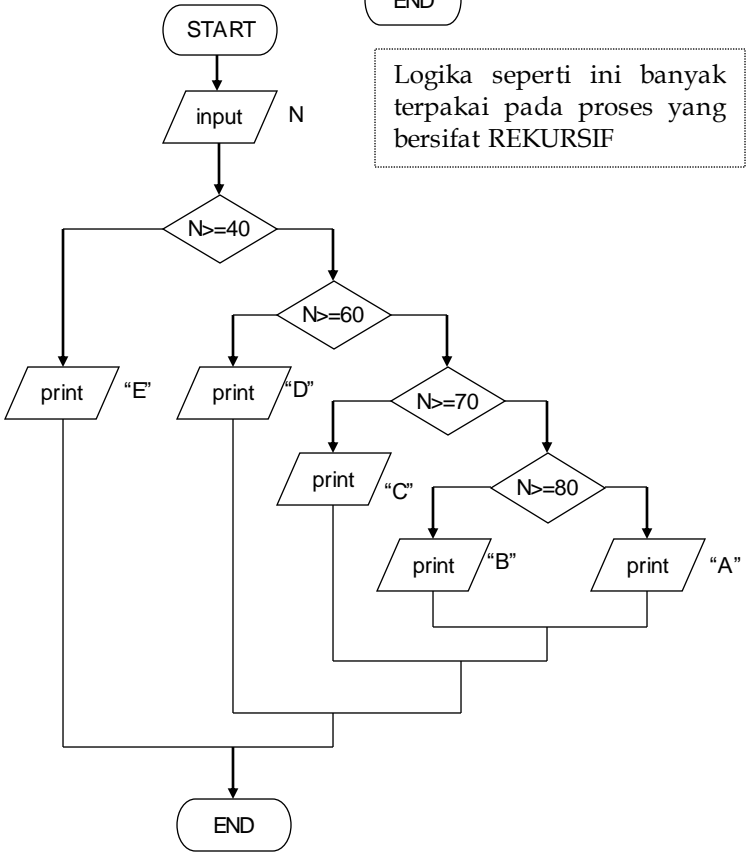
Karena semua instruksi hanya satu baris, maka kurung buka dan kurung tutup dapat diabaikan



Logika seperti ini banyak dipakai pada proses yang bersifat REKURSIF

Jawab. Cara-4

```
#include <iostream.h>
void main()
{int N;
 cin >> N;
 if(N >= 40)
   if(N>=60)
     if(N>=70)
       if(N >= 80)
         cout << "A";
       else
         cout << "B";
     else
       cout << "C";
   else
     cout << "D";
 else
   cout << "E";
}
```



Soal-20

Susun program untuk menginput 3 buah nilai integer (nilai ujian). Cetak perkataan "DITERIMA", bila ketiga nilai tersebut berkategori "lulus" (≥ 60). Selainnya bila ada nilai yang "tidak Lulus", maka cetak perkataan "TIDAK DITERIMA".

```
{int A,B,C;
cin >> A >> B >> C;
if(A >= 60)
    if(B >= 60)
        if(C >= 60)
            cout << "DITERIMA";
        else
            cout << "TIDAK DITERIMA";
    else
        cout << "TIDAK DITERIMA";
else
    cout << "TIDAK DITERIMA";
}
```

atau

```
{ int A,B,C;
int Jum = 0;
cin >> A >> B >> C;
if(A >= 60) Jum = Jum + 1;
if(B >= 60) Jum = Jum + 1;
if(C >= 60) Jum = Jum + 1;
if(Jum == 3 )
    cout << "DITERIMA";
else
    cout << "TIDAK DITERIMA";
}
```

Soal-21

Susun program untuk menginput 3 buah nilai integer (nilai ujian). Cetak perkataan "DITERIMA", bila cukup salah satu nilai berkategori "lulus" (≥ 60). Bila ketiga buah nilai tersebut berkategori "tidak lulus" maka cetak perkataan "TIDAK DITERIMA".

```
{int A,B,C;
cin >> a >> B >> C;
if(A >= 60)
    cout << "DITERIMA";
else
    if(B >= 60)
        cout << "DITERIMA";
    else
        if(C >= 60)
            cout << "DITERIMA";
        else
            cout << "TIDAK DITERIMA";
}
```

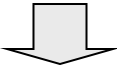
Kalau nilai A lulus (≥ 60) maka langsung cetak "DITERIMA", kemudian selesai tanpa memeriksa nilai yang lain.

atau

```
{ int A,B,C;
int Jum = 0;
cin >> A >> B >> C;
if(A >= 60) Jum = Jum + 1;
if(B >= 60) Jum = Jum + 1;
if(C >= 60) Jum = Jum + 1;
if(Jum > 0 )
    cout << "DITERIMA";
else
    cout << "TIDAK DITERIMA";
}
```

Soal-22

Susun program untuk menginput 4 buah nilai integer yang menyatakan nilai ujian mahasiswa. Kemudian cetak perkataan tertentu bila nilai ujian yang lulus seperti yang dinyatakan atau diperlihatkan dalam tabel berikut :



Cetak perkataan	Bila nilai yang dinyatakan lulus
"BAIK SEKALI"	Semua nilai (4 nilai)
"BAIK"	Hanya 3 diantara 4 nilai
"CUKUP"	Hanya 2 diantara 4 nilai
"KURANG"	Hanya 1 diantara 4 nilai
"BURUK"	Tidak ada satupun

Jawab Soal-22

```
int A,B,C,D;
int Jum = 0;
cin >> A >> B >> C;
if(A >= 60) Jum = Jum + 1;
if(B >= 60) Jum = Jum + 1;
if(C >= 60) Jum = Jum + 1;
if(C >= 60) Jum = Jum + 1;
```

Bersambung ke :

```
if(Jum == 4) cout << "BAIK SEKALI";
else if(Jum == 3) cout << "BAIK";
else if(Jum == 2) cout << "CUKUP";
else if(Jum == 1) cout << "KURANG";
else cout << "BURUK";
```

3.5. Seleksi dengan menggunakan Switch

Soal-23

Susun program untuk menginput sebuah nilai integer (misal N). Kemudian cetak peringkat nilai sesuai dengan nilai N sebagai berikut : →

N	Nama Peringkat yang dicetak
1	Kurang Sekali
2	Kurang
3	Cukup
4	Baik
5	Baik Sekali

Bila yang diinput bukan seperti nilai-nilai N diatas, maka cetak perkataan "Data Salah "

Jawab Soal-23.

Cara-1

Menggunakan Nested IF

```
#include <stdio.h>
void main()
{int N;
cin >> N;
if(N == 1)
    cout << "Kurang Sekali";
else
{if(N == 2)
    cout << "Kurang";
else
{if(N == 3)
    cout << "Cukup";
else
{if(N == 4)
    cout << "Baik";
else
{ if(N == 5)
    cout << "Baik Sekali";
else
    cout << "Data Salah";
}
}
}
}
}
```

Atau dapat juga ditulis sebagai berikut:

```
#include <stdio.h>
void main()
{int N;
cin >> N;
if(N == 1)
    cout << "Kurang Sekali";
else if(N == 2)
    cout << "Kurang";
else if(N == 3)
    cout << "Cukup";
else if(N == 4)
    cout << "Baik";
else if(N == 5)
    cout << "Baik Sekali";
else
    cout << "Data Salah";
}
```

Bila setiap kondisi "true", hanya terdiri dari satu instruksi maka kurung buka dan kurung tutup boleh tidak dipakai

Persoalan diatas dapat juga diselesaikan dengan menggunakan **switch() & case** sebagai berikut :

Jawab Soal-23. Cara-2 Menggunakan switch & case

```
int N;
cin >> N;
switch(N)
{ case 1:
    cout << "Kurang Sekali";
    break;
  case 2:
    cout << "Kurang";
    break;
  case 3:
    cout <<("Cukup";
    break;
  case 4:
    cout <<("Baik";
    break;
  case 5:
    cout << "Baik Sekali"
    break;
  default:
    cout <<"Data Salah";
}
```

Bagaimana bila semua case tidak menggunakan instruksi : break

```
int N;
cin >> N;
switch(N)
{case 1:
    cout << "\nKurang Sekali");
  case 2:
    cout << "\nKurang";
  case 3:
    cout << "\nCukup";
  case 4:
    cout << "\nBaik";
  case 5:
    cout << "\nBaik Sekali";
  default:
    cout << "\nData Salah";
}
```

Bila N = ..., bukan 1, 2, 3, 4 atau 5, maka akan tercetak :
Data Salah

Bila **N = 1**, maka akan tercetak :
Kurang Sekali
Kurang
Cukup
Baik
Baik Sekali
Data Salah

Bila **N = 4**, maka akan tercetak :
Baik
Baik Sekali
Data Salah

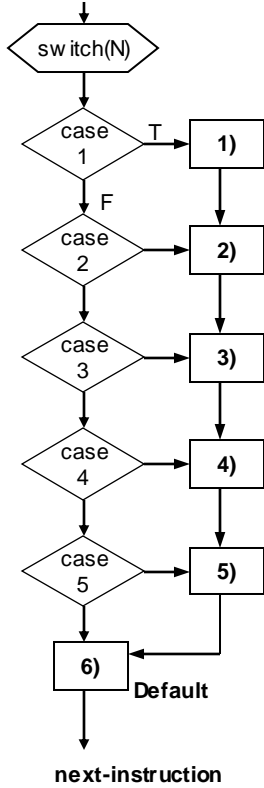
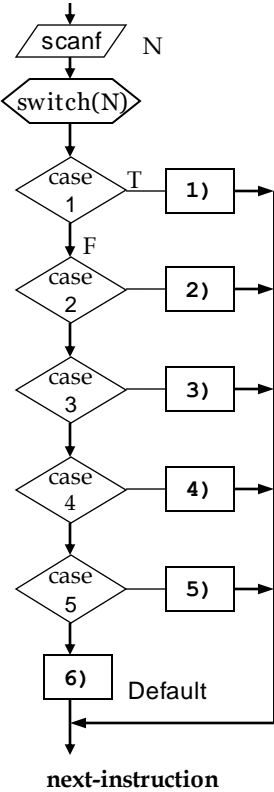
- 1) mencetak "Kurang Sekali"
- 2) mencetak "Kurang"
- 3) mencetak "Cukup"
- 4) mencetak "Baik"
- 5) mencetak "Baik Sekali"
- 6) mencetak "Data Salah"

Perhatikan instruksi :

a. **default.**
Kalau semua kondisi di case tidak ada yang TRUE, maka kerjakan blok instruksi yang ada di default

b. **break.**
Setelah mengerjakan blok instruksi yang ditunjuk oleh case yang bernilai TRUE, dengan instruksi break, maka akan langsung meloncat ke next-instruction

1) mencetak "Kurang Sekali"
2) mencetak "Kurang"
3) mencetak "Cukup"
4) mencetak "Baik"
5) mencetak "Baik Sekali"
6) mencetak "Data Salah"



Bagaimana bila **sebagian case**
tidak menggunakan instruksi : **break**

```
#include <stdio.h>
void main()
{int N;
cin >> N;
switch(N)
{
case 1:
cout << "\nKurang Sekali";
case 2:
cout << "\nKurang";
break;
case 3:
cout << "\nCukup";
break;
case 4:
cout << "\nBaik";
case 5:
cout << "\nBaik Sekali";
default:
cout << "\nData Salah";
}
}
```

Bila N = 1,
maka akan
tercetak :

Kurang Sekali
Kurang

Bila N = 2,
maka akan
tercetak :

Kurang

Bila N = 3,
maka akan
tercetak :

Cukup

Bila N = 4,
maka akan
tercetak :

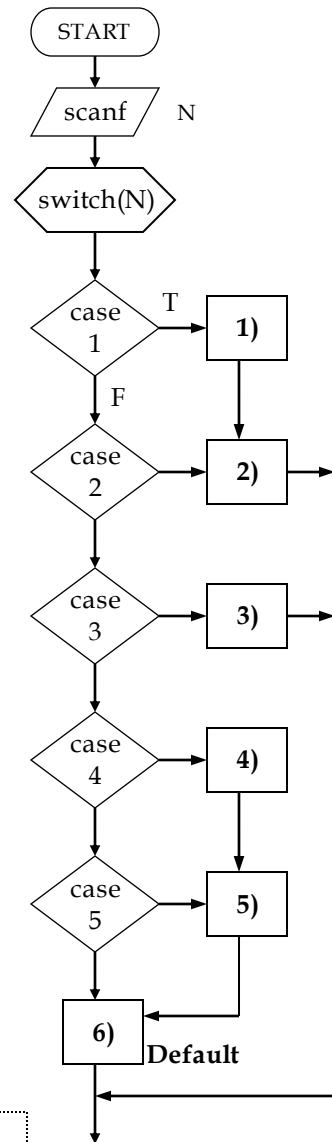
Baik
Baik Sekali
Data Salah

Bila N = 5,
maka akan
tercetak :

Baik Sekali
Data Salah

Bila N = ...,
bukan 1, 2, 3,
4 atau 5,
aka akan
tercetak :

Data Salah



next-instruction

- 1) mencetak "Kurang Sekali"
- 2) mencetak "Kurang"
- 3) mencetak "Cukup"
- 4) mencetak "Baik"
- 5) mencetak "Baik Sekali"
- 6) mencetak "Data Salah"

Teori :
Perhatikan contoh berikut :

Contoh-1 :

```
#include <stdio.h>
void main()
{int N;
 cin >> N;
 switch(N)
 {
  case 1:
  case 2:
    cout << "\nTidak Lulus");
    break;
  case 3:
  case 4:
  case 5:
    cout << "\nL u l u s");
    break;
  default:
    cout << "\nData Salah");
  }
}
```

Bila **N = 1**, atau = **2**
maka akan
tercetak :

Tidak Lulus

Bila **N = 3**, atau = **4**,
atau = **5**
maka akan
tercetak :

L u l u s

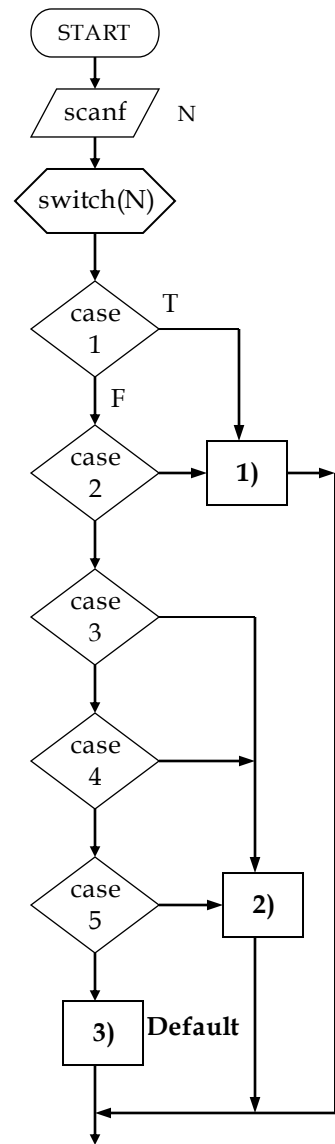
Bila **N = ...** ,
bukan 1, 2, 3,
4 atau 5,
aka akan
tercetak :

Data Salah

Catatan :

Bila **N** bertipe char, misal **char N**;
maka digunakan :

case '1' :



- next-instruction**
- 1) mencetak "Tidak Lulus"
 - 2) mencetak "L u l u s"
 - 3) mencetak "Data Salah"

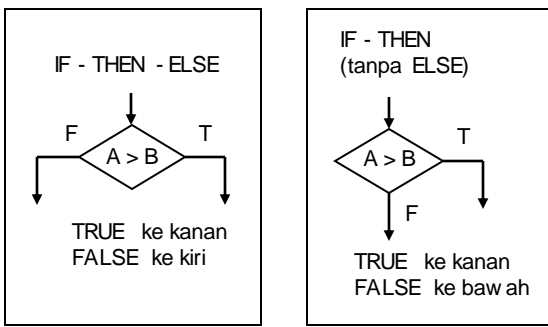
Contoh-2

Switch & Case berjenjang

```
#include <stdio.h>
void main()
{char KodePulau;    int KodeKota;
  cin >> KodePulau;
  switch(KodePulau)
  {
    case 'J':
      cout << "Pulau Jawa");
      cin >> KodeKota );
      switch(KodeKota)
      { case 1: cout << "\nJakarta";
        break;
        case 2: cout << "\nSurabaya";
        break;
        case 3: cout << "\nBandung";
        break;
        case 4: cout << "\nSemarang";
        break;
        case 5: cout << "\nYogyakarta";
        break;
        default: cout << "Kode Kota di Pulau Jawa Salah";
      }
    case 'S':
      cout << "Pulau Sumatera";
      cin >> KodeKota ;
      switch(KodeKota)
      { case 1: cout << "\nMedan";
        break;
        case 2: cout << "\nPalembang";
        break;
        case 3: cout << "\nPadang";
        break;
        default: cout << "Kode Kota di Sumatera Salah";
      }
    case 'K':
      cout << "Pulau Kalimantan";
      cin >> KodeKota );
      switch(KodeKota)
      { case 1: cout << "\nBanjarmasin";
        break;
        case 2: cout << "\nPontianak";
        break;
        default: cout << "Kode Kota di Kalaimantan Salah";
      }
    default: cout << "Kode Pulau Salah";
```

Soal-Soal Latihan Mandiri

Catatan : Untuk semua flowchart dalam soal-soal berikut ini, berlaku pengertian TRUE (T) dan FALSE (F) sebagai berikut :



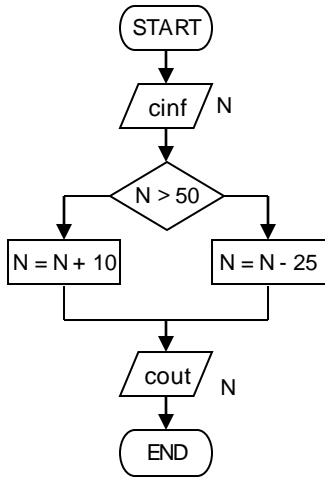
Soal-01.
Tulis program (penggalan program) dalam bahasa C untuk menyatakan algoritma yang digambarkan oleh flowchart **Gambar-1** dan **Gambar-2** :

Soal-02.
Untuk program yang algoritmanya digambarkan dengan Flowchart **Gambar-1**, apa yang tercetak bila untuk nilai N diinputkan nilai :

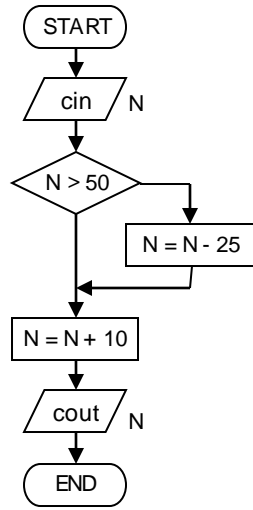
- a. 30
- b. 50
- c. 65

Soal-03.
Untuk program yang algoritmanya digambarkan dengan Flowchart **Gambar-2**, apa yang tercetak bila untuk nilai N diinputkan nilai :

- a. 30
- b. 50
- c. 65



Gambar-1



Gambar-2

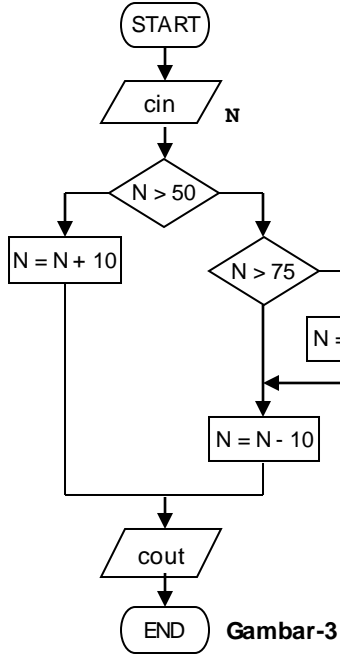
Soal-04.
Tulis program (penggalan perogram) dalam bahasa C untuk menyatakan algoritma yang digambarkan oleh flowchart **Gambar-3** dan **Gambar-4**.

Soal-05.
Untuk program yang algoritmanya digambarkan dengan Flowchart **Gambar-3** diatas, apa yang tercetak bila untuk nilai N diinputkan nilai :

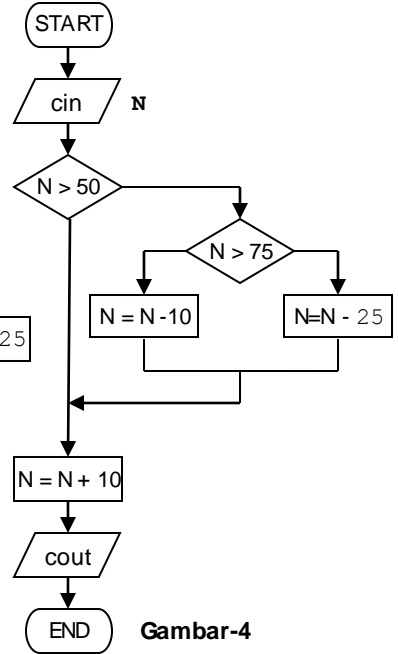
- a. 30
- b. 50
- c. 65
- d. 75
- e. 90

Soal-06.
Untuk program yang algoritmanya digambarkan dengan Flowchart **Gambar-4** diatas, apa yang tercetak bila untuk nilai N diinputkan nilai :

- a. 30
- b. 50
- c. 65
- d. 75
- e. 90



Gambar-3



Gambar-4

Soal-07.

Tulis program (penggalan program) dalam bahasa C untuk menyatakan algoritma yang digambarkan oleh flowchart **Gambar-5** dan **Gambar-6**.

Soal-08.

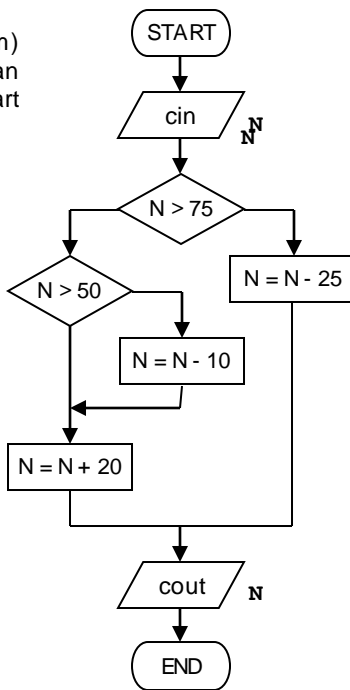
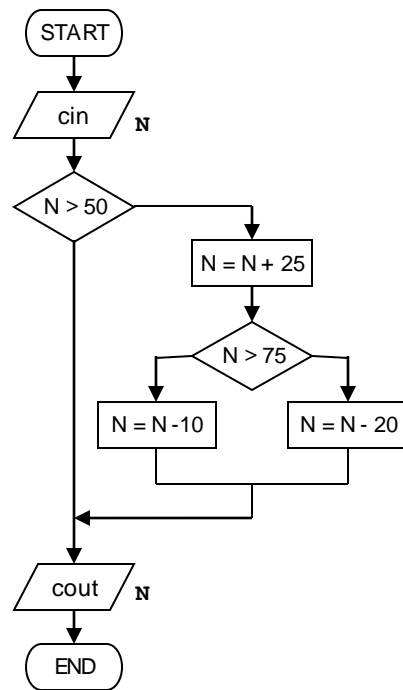
Untuk program yang algoritmanya digambarkan dengan Flowchart **Gambar-5** diatas, apa yang tercetak bila untuk nilai N diinputkan nilai :

- a. 30
- b. 50
- c. 65
- d. 75
- e. 90

Soal-09.

Untuk program yang algoritmanya digambarkan dengan Flowchart **Gambar-6** diatas, apa yang tercetak bila untuk nilai N diinputkan nilai :

- a. 30
- b. 50
- c. 65
- d. 75
- e. 90

**Gambar-5****Gambar-6****Soal-10.**

Apa yang tercetak oleh masing-masing penggalan program berikut ini bila untuk nilai N diinputkan nilai :

- a. 30
- b. 50
- c. 75

(Setiap program dites 3 kali dengan nilai N yang berbeda).

10.1 Program-1

```

cin >> N;
if(N>50)
    {N = N + 10;}
else
    {N = N - 25;}
cout << N;
  
```

10.2 Program-2

```

cin >> N;
if(N>50)
    N = N + 10;
else
    N = N - 25;
cout << N;
  
```

10.3 Program-3

```

cin >> N;
if(N>50)  N = N + 10;
else  N = N - 25;
cout << N;
  
```

10.4 Program-4

```

cin >> N;

if(N < 50)
    {N = N + 30;}
if(N > 50)
    {N = N - 25;}
N = N + 10;
cout << N;
  
```

Soal-11.

Gambarkan flowchart untuk penggalan program pada Soal 10.1 (Program-1) dan Soal 10.4 (Program-4) diatas.

10.5 Program-5

```

cin >> N;
if(N < 50)
    N = N + 30;
if(N > 50)
    N = N - 25;
N = N + 10;
cout << N;
  
```

10.6 Program-6

```

cin >> N;
if(N < 50) N = N + 30;
if(N > 50) N = N - 25;
N = N + 10;
cout << N;
  
```

Soal-12.

Gambarkan flowchart untuk penggalan program berikut ini :

12.1 Program-7

```
cin >> N;
if(N >= 80)
    cout << "A";
else if(N>= 70)
    cout << "B";
else if(N >= 60)
    cout << "C";
else if(N >= 40)
    cout << "D";
else cout << "E";
```

Apa yang tercetak bila untuk Program-7 diinput nilai N =

- a. 85
- b. 64

12.2 Program-8

```
cin >> N;
if(N >= 40)
    if(N >= 60)
        if(N >= 70)
            if(N >= 80)
                cout << "A";
            else cout << "B";
        else cout << "C";
    else cout << "D";
else cout << "E";
```

Apa yang tercetak bila untuk program-8 diinput nilai N =

- a. 85
- b. 64

Soal-13.

Gambarkan flowchart untuk penggalan program berikut ini :

13.1 Program-9

```
cin >> N;
if(N > 50)
    {if(N>75)
        N = N -25;
        N = N -10;
    }
else
    N = N + 10;
cout << N;
```

13.2 Program-10

```
cin >> N;
if(N > 50)
    {if( N>75 )
        N = N -25;
        else
            N = N -10;
    }
N = N + 10;
cout << N;
```

13.3 Program-11

```
cin >> N;
if(N > 50)
    {if( N>75 )
        N = N -25;
        else
            N = N -10;
    }
else
    N = N + 10;
cout << N;
```

13.4 Program-12

```
cin >> N;
if(N > 50)
    { if( N>75 )
        N = N -25;
    }
else
    N = N + 10;
cout << N;
```

13.5 Program-13

```
cin >> N;
if(N > 75)
    N = N -25;
else
    {if ( N>50 )
        N = N -10;
        N = N + 20; }
cout << N;
```

13.6 Program-14

```
cin >> N;
if(N > 50)
    {N = N - 25;
    if( N>75 )
        N = N -20;
    else
        N = N -10;
    }
cout << N;
```

Apa yang tercetak bila untuk program-program diatas diinput nilai N =

- a. 90
- b. 75
- c. 50
- d. 35

Soal-14. Susun program (penggalan program) dalam Bahasa C untuk flowchart **Gambar-7**.

Soal-15. Periksa Flowchart **Gambar-7**. Apa yang tercetak bila untuk nilai A, B, dan C berturut-turut diinputkan nilai sebagai berikut:

- a. 5, 7, 12 Tercetak :
- b. 5, 10, 7 Tercetak :
- c. 7, 5, 12 Tercetak :
- d. 7, 10, 5 Tercetak :
- e. 10, 5, 7 Tercetak :
- f. 10, 7, 5 Tercetak :

Soal-16. Periksa apakah flowchart **Gambar-7**, maksudnya adalah mencetak salah satu bilangan yang nilainya terbesar dari tiga buah bilangan yang diinput.

Jawab : Benar / Salah

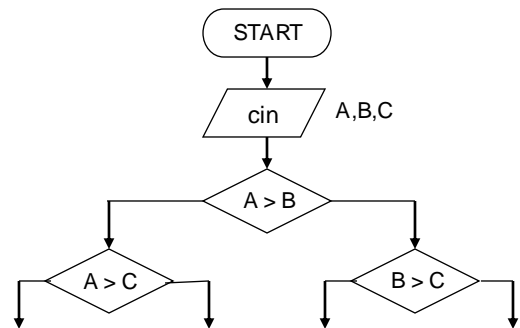
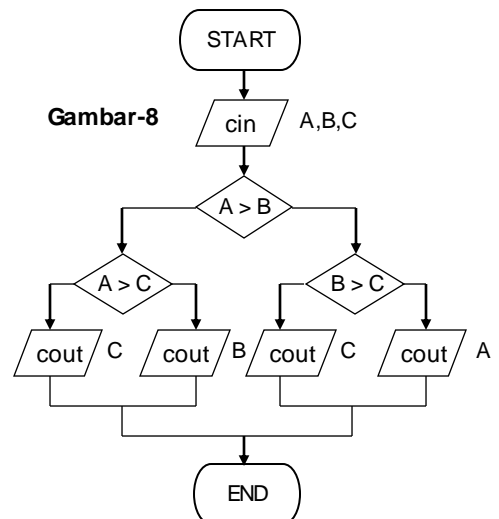
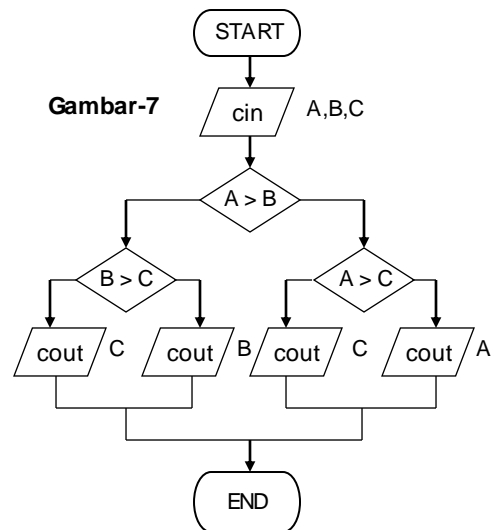
Soal-17. Periksa Flowchart **Gambar-8**. Apa yang tercetak bila untuk nilai A, B, dan C berturut-turut diinputkan nilai sebagai berikut:

- a. 5, 7, 12 Tercetak :
- b. 5, 10, 7 Tercetak :
- c. 7, 5, 12 Tercetak :
- d. 7, 10, 5 Tercetak :
- e. 10, 5, 7 Tercetak :
- f. 10, 7, 5 Tercetak :

Soal-18. Dari hasil (jawaban) yang diperoleh dari Soal-17, dapat diambil kesimpulan bahwa Flowchart **Gambar-8** menggambarkan algoritma mencetak salah satu bilangan yang nilainya terbesar dari 3 buah bilangan yang diinput.

Jawab : Benar / Salah.

Soal-19. Lanjutkan Flowchart **Gambar-9**, sehingga maksudnya menggambarkan algoritma mencetak salah satu bilangan yang nilainya terbesar dari 3 buah bilangan yang diinput.

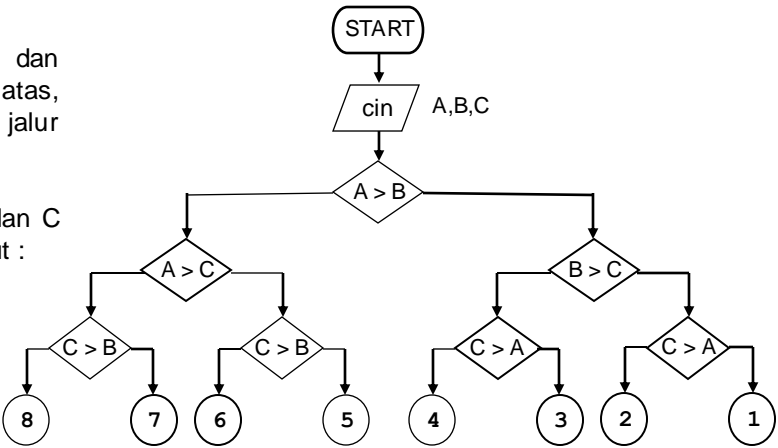


Soal-20.

Untuk semua kemungkinan nilai A, B, dan C pada Flowchart **Gambar-10** diatas, dari 8 jalur yang ada, tunjukkan jalur mana saja yang tidak pernah dilalui.

Kemungkinan kombinasi nilai A, B dan C sebagai contoh adalah sebagai berikut :

- a. 5, 7, 10
- b. 5, 10, 7
- c. 7, 5, 10
- d. 7, 10, 5
- e. 10, 5, 7
- f. 10, 7, 5



Gambar-10

Periksa untuk semua kemungkinan nilai A,B, dan C seperti yang dicontohkan, lalu temukan jalur mana saja yang tidak pernah dilalui.

Soal-21. Susun algoritma untuk menginput 3 buah bilangan (anggap ketiga buah bilangan tersebut nilainya tidak sama satu dengan yang lainnya). Kemudian cetak salah satu bilangan yang mempunyai nilai yang terkecil.

- a. Menggunakan logical operator AND atau OR (&& atau ||).
- b. Tidak boleh menggunakan logical operator AND dan OR (&& dan ||).

Soal-22. Susun algoritma untuk menginput 3 buah bilangan (anggap ketiga buah bilangan tersebut nilainya tidak sama satu dengan yang lainnya). Kemudian cetak ketiga buah bilangan tersebut urut mulai bilangan yang nilainya terbesar sampai dengan nilai yang terkecil.

- a. Menggunakan logical operator AND atau OR (&& atau ||).
- b. Tidak boleh menggunakan logical operator AND dan OR (&& dan ||).

Soal-23 Susun algoritma untuk menginput tiga buah bilangan. Cetak perkataan “BENAR”, bila salah satu bilangan merupakan jumlah dari dua buah bilangan yang lain., selainnya (bila tidak) cetak perkataan “SALAH”.

Soal-24. Susun algoritma untuk menginput 3 buah bilangan yang masing-masing menyatakan panjang sebuah garis. Kemudian periksa ketiga buah garis tersebut apakah dapat membentuk sebuah segitiga atau tidak. Bila ketiga buah garis tersebut dapat membentuk sebuah segitiga, maka cetak perkataan “SEGI TIGA”. Sebaliknya bila ketiga buah garis tersebut tidak mungkin membentuk sebuah segitiga, maka cetak perkataan : “BUKAN SEGITIGA”.
Catatan : Pada sebuah segitiga. berlaku ketentuan yang pasti, yaitu panjang sebuah sisi selalu lebih kecil dibandingkan jumlah panjang kedua buah sisi yang lain. Atau, jumlah panjang dua buah sisi selalu lebih besar dari panjang sisi yang lain.

- a. Menggunakan logical operator AND atau OR (&& atau ||).
- b. Tidak boleh menggunakan logical operator AND dan OR (&& dan ||).

- Soal-25** Susun algoritma untuk menginput 3 buah bilangan yang masing-masing menyatakan panjang sisi sebuah segitiga. Kemudian periksa ketiga buah garis tersebut. Bila ketiga buah garis (sisi) tersebut panjangnya sama maka cetak perkataan "SAMA SISI". Bila hanya dua sisi yang sama maka cetak perkataan "SAMA KAKI ". Tapi bila ketigaitiganya tidak sama maka cetak perkataan "SEMBARANG".
- Menggunakan logical operator AND atau OR (&& atau ||).
 - Tidak boleh menggunakan logical operator AND dan OR (&& dan ||).
- Soal-26.** Susun algoritma untuk menginput tiga buah bilangan yang masing-masing menyatakan besar sebuah sudut (dalam derajat). Periksa ketiga buah bilangan tadi. Bila total ketiganya = 180, maka cetak perkataan "SEGI TIGA", tapi bila totalnya bukan 180, maka cetak perkataan "BUKAN SEGI TIGA".
- Soal-27.** Susun algoritma untuk menginput tiga buah bilangan yang masing-masing menyatakan besar sudut dalam sebuah segitiga.. Periksa ketiga buah bilangan tadi. Cetak perkataan "SEGI TIGA SIKU SIKU", bila ada salah satu bilangan yang nilainya sama dengan 90., bila tidak ada yang bernilai 90, maka cetak perkataan "SEGI TIGA BUKAN SIKU-SIKU"
- Soal-28.** Susun algoritma untuk menginput tiga buah bilangan yang masing-masing menyatakan besar sebuah sudut (dalam derajat). Periksa ketiga buah bilangan tadi.
- Bila total ketiga buah bilangan tersebut = 180, maka periksa lagi apakah salah satu bilangan ada yang nilainya = 90.
 - Bila ada cetak perkataan "SEGI TIGA SIKU-SIKU",
 - Bila tidak ada sudut yang nilainya = 90 maka cetak perkataan " SEGI TIGA BUKANSIKU-SIKU".
 - Bila total ketiganya bukan 180, maka cetak perkataan "BUKAN SEGITIGA"
- Soal-29.** Susun program untuk menginput tiga buah bilangan yang menyatakan susut-sudut sebuah segitiga. Bila salah satu bilangan nilainya sama dengan jumlah nilai dua bilangan yang lain, maka cetak perkataan "SIKU SIKU", bila tidak maka cetak perkataan "BUKAN SIKU-SIKU".
- Soal-30.** Susun program untuk menginput tiga buah bilangan yang menyatakan nilai ujian tiga buah mata kuliah. Cetak perkataan "LULUS" bila nilai rata-rata ketiga buah bilangan tersebut lebih besar atau sama dengan 60. Bila nilai rata-rata lebih kecil dari 60, maka cetak perkataan "TIDAK LULUS".
- Soal-31.** Susun program untuk menginput tiga buah bilangan yang menyatakan nilai ujian tiga buah mata kuliah. Cetak perkataan "LULUS" bila nilai rata-rata ketiga buah bilangan tersebut lebih besar atau sama dengan 60 dan ketiga buah bilangan tersebut tidak ada yang mendapat nilai mati (nilai dibawah 40). Bila nilai rata-rata dibawah 60, atau ada nilai yang mendapat nilai mati, maka cetak perkataan "TIDAK LULUS".

- Soal-32.** Susun program untuk menginput tiga buah bilangan yang menyatakan nilai ujian tiga buah mata kuliah. Cetak perkataan "LULUS" bila nilai rata-rata ketiga buah bilangan tersebut lebih besar atau sama dengan 60 atau, walaupun nilai rata-rata lebih kecil dari 60, tapi bila ada salah satu mata kuliah yang nilainya lebih besar atau sama dengan 80 maka kategorinya juga dinyatakan lulus dan cetak perkataan "LULUS". Bila syarat diatas tidak terpenuhi maka cetak perkataan "TIDAK LULUS".
- Soal-33.** Susun program untuk menginput tiga buah bilangan yang menyatakan nilai ujian tiga buah mata kuliah.
Cetak perkataan "TIGA" bila ketiga mata kuliah tersebut mendapat nilai lulus.
Cetak perkataan "DUA", bila hanya dua dari dari ketiga mata kuliah tersebut yang mendapat nilai lulus.
Cetak perkataan "SATU" bila hanya satu mata kuliah yang mendapat nilai lulus.
Cetak perkataan "NOL" bila ketiga mata kuliah tersebut dinyatakan tidak lulus
Sebuah mata kuliah dinyatakan mendapat nilai lulus bila nilainya lebih besar atau sama dengan 60.
- Soal-34.** Susun program untuk menginput jam keberangkatan dan jam kedatangan, kemudian menghitung serta mencetak jumlah waktu (lama) dalam perjalanan. Semua waktu dinyatakan dengan notasi JAM, MENIT dan DETIK.
Catatan, perjalanan dilakukan pada hari yang sama, dan notasi waktu dalam satu hari adalah mulai jam 00:00:00 sampai dengan jam 24:00:00
Contoh :
- | | | | | | | |
|------------------------|---|----|---|----|---|----|
| Bila jam keberangkatan | : | 8 | : | 45 | : | 30 |
| dan jam kedatangan | : | 10 | : | 20 | : | 17 |
| maka lama perjalanan | : | 1 | : | 34 | : | 47 |
- Artinya, bila berangkat jam 8 lewat 45 menit 30 detik, dan jam kedatangan adalah jam 10 lewat 20 menit 17 detik, maka lama dalam perjalanan adalah 1 jam 34 menit dan 47 detik.
- Soal-35.** Susun algoritma untuk menginput dua buah bilangan bulat yang masing-masing merupakan **Jumlah** dan **Harga Satuan** suatu barang yang dibeli. Kemudian cetak **Harga Yang Harus Dibayar**, apabila berlaku ketentuan sebagai berikut :
1. $\text{Harga Barang} = \text{Jumlah barang} * \text{Harga Satuan}.$
 2. $\text{Harga Yang Harus Dibayar} = \text{Harga Barang} - \text{Potongan}.$
 3. Potongan dihitung sebagai berikut :
 - a. Bila Harga Barang lebih dari satu juta rupiah, maka mendapat Potongan sebesar 10 % dari Harga Barang.
 - b. Bila Harga Barang 501.000 sampai 1.000.000 rupiah maka mendapat Potongan 5 % dari Harga Barang
 - c. Bila harga Barang kurang dari 501.000 rupiah, maka tidak mendapat Potongan atau Potongan = nol rupiah

Soal-36. Susun algoritma untuk menginput sebuah bilangan bulat yang menyatakan **Jumlah m kubik** air PAM yang digunakan. Kemudian cetak **Harga Yang Harus Dibayar**, apabila berlaku ketentuan sebagai berikut :

50 m kubik pertama, Harga per m kubik = Rp. 1000,-
 50 m kubik kedua, Harga per m kubik = Rp. 1500,-
 50 m kubik ketiga, Harga per m kubik = Rp. 2000,-
 dan untuk m kubik selanjutnya berlaku harga Rp. 3000 per m kubik.

Secara tabel dapat digambarkan sebagai berikut :

0	50	51	100	101	150	151
	Rp. 1000 per m kubik		Rp. 1500 per m kubik		Rp. 2000 per m kubik	
						Rp. 3000 per m kubik

Contoh : Bila menggunakan 40 m kubik,
 maka akan membayar = $40 * 1000$

Bila menggunakan 75 m kubik,
 maka akan membayar = $50 * 1000 + 25 * 1500$

Bila menggunakan 175 m kubik,
 maka akan membayar = $50 * 1000 + 50 * 1500 + 50 * 2000 + 25 * 3000$

selesai
Sessi-3