

# Algoritma Dan Struktur Data

**M. Haviz Irfani, S.Si** - STMIK MDP PALEMBANG

## 1. Algoritma

Algoritma yang akan dibahas terdiri dari : a. Algoritma Pseudocode dan b. Algoritma Flowchart

### a. Algoritma Pseudocode

Kata Algoritma berasal dari kata Algorithm-algorithm, berasal dari nama Abu Ja'far Muhammad Ibnu Musa Al-Khuwarizmi (Ilmuan Persia tahun 825 M/**Pernah menulis Kitab Al-Jabar w'al-muqobala / ilmu pemugaran dan Pengurangan**). **Algoritma** merupakan kumpulan pernyataan-pernyataan (intruksi) yang tertulis (kalimat, gambar) secara jelas dan masuk akal (logis) dalam pengambilan keputusan untuk menyelesaikan permasalahan. **Pseudocode** merupakan algoritma yang dibuat mendekati suatu bahasa pemrograman. Jadi, **Pseudocode** adalah algoritma yang menggunakan bahasa (langkah-langkah ringkas) menyerupai (Pseudo) sembarang bahasa Pemrograman untuk menyelesaikan permasalahan.

Implementasi dari algoritma dinamakan Program. **Program** merupakan kumpulan instruksi-instruksi yang berkaitan sistematis dan logis dalam komputasi (bahasa pemrograman tertentu) untuk memecahkan permasalahan.

PROGRAM = ALGORITMA + STRUKTUR DATA

### Ciri-ciri Algoritma:

1. Ada INPUT
2. Ada PROSES
3. Ada OUTPUT
4. Memiliki Instruksi yang jelas dan tidak AMBIGU
5. Harus mempunyai titik Stopping Role.

Syarat algoritma dikatakan baik:

Tingkat kepercayaan tinggi (Realibility), Hasil yang diperoleh dari proses harus berakurasi yang tinggi dan benar

Pemrosesan yang efisien (Low Cost), Proses harus diselesaikan secepat mungkin dan jumlah kalkulasi sesingkat mungkin.

Bersifat General, Dapat menyelesaikan bermacam-macam kasus secara general.

Bisa dikembangkan (Expandable), haruslah sesuatu yang bisa dikembangkan lebih lanjut dengan perubahan-perubahan sesuai dengan aturan (permasalahan) yang berkaitan.

Mudah dimengerti, memudahkan dalam pengelolaan untuk para pembaca.

Portabilitas yang tinggi (Portability), dapat diimplementasi diberbagai platform computer.

**Sifat-sifat Algoritma:**

1. Tidak menggunakan simbol atau sintaks dari suatu bahasa Pemrograman
2. Tidak Tergantung pada suatu bahasa Pemrograman
3. Notasi-notasinya dapat digunakan untuk seluruh bahasa manapun

**Contoh Algoritma bukan Peudecode " memasak Mie Instan".**

1. *Mulai.*
2. *Rebus air hingga mendidih.*
3. *Masukkan mie instan kedalam air mendidih tersebut.*
4. *Tunggu beberapa menit sampai mie terlihat matang.*
5. *Jika mie sudah matang, angkat dan tiriskan.*
6. *Campurkan bumbu-bumbu dan aduk sampai rata.*
7. *Makan siap dihidangkan*
8. *Selesai.*

Setiap algoritma harus dapat diimplementasikan ke dalam bahasa pemrograman apapun, karena setiap bahasa pemrograman dapat melakukan operasi-operasi dasar seperti membaca dan menulis data, operasi perbandingan aritmatika/logika dan sebagainya. Perkembangan komputer hanya merubah kecepatan membaca, biaya dan tingkat ketelitian.

**Contoh Algoritma bukan Peudecode menghitung sisi miring :**

1. *Mulai.*
2. *Masukkan nilai a.{sisi datar}*
3. *Masukkan nilai b.{sisi Tegak}*
4. *Hitung  $c^2 = a^2 + b^2$  .*
5. *Cetak/tampilkan nilai c.{sisi miring}*
6. *Selesai.*

Algoritma Biasa	Pseudecode	Bhs C++
Tipe dasar tidak ada	<u>Boolean</u> <u>Integer</u> <u>Real</u> <u>Char</u> <u>String</u> <u>Record</u> <field1 :type, field2 :type, :: fieldn :type>	Secara khusus tidak ada. Unsigned char,char (1 byte),unsigned int,int,short int (2 byte), long int(4 byte). Float, double Char Char[n] Struct {Type field1, type field2, :: Type fieldn;}
Operator aritmatika biasa	+ - * / Div Mod	+ - * / / %
Operator Perbandingan biasa	< > ≤ ≥ = ≠	< > <= >= == !=

Algoritma Biasa	Pseudocode	Bhs C++
Operator Logika biasa	Not and or xor	! &&    Tidak ada
{Komentar}	{komentar}	/*komentar*/
	<u>Const</u> <u>Type</u> <u>True</u> <u>false</u>	#define Typedef Dapat diganti 1 Dapat diganti 0
Masukan bilangan	<u>Read</u> (bil)	Cin>>"masukkan bilangan =">>bil;
Cetak bilangan	<u>Write</u> (bil)	Cout<<"bilangan ="<<bil;
A=B.C	A ← B*C	A = B*C ;
Deklarasi tipe tidak ada	Deklarasi : a,b,c: <u>integer</u> z,x : <u>Real</u>	Int a,b,c; Float z,x;
Mulai,akhir	Tidak baku	{' dan '}

Contoh algoritma Pseudocode:

Algoritma menghitung sisi miring

Deklarasi :

a,b,c : byte

Deskripsi:

read(a)

read(b)

$c \leftarrow \text{sqrt}(a*a + b*b)$

write(c)

Algoritma dalam pembahasan dalam modul ini menggunakan pendekatan dengan bahasa Pascal. Contoh diatas memperlihatkan bentuk algoritma Pseudocode yang menyerupai bahasa Pascal, tetapi tidak sama dengan sintak(tata cara penulisan) bahasa Pascal. Walaupun demikian algoritma tersebut masih dapat diubah ke dalam bahasa pemrograman manapun.

#### b. Algoritma Flowchart

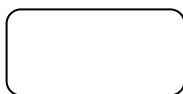
Merupakan algoritma yang disajikan dalam bentuk simbol2x yang mempunyai aliran yang sekuensial (terurut) untuk menyelesaikan permasalahan. Bagan alir program (program flowchart) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Bagan alir program dibuat dari derivikasi bagan alir sistem.

**Kegunaan :**

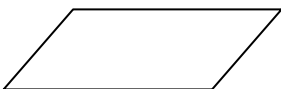
1. **Untuk mendesain program**
2. **Untuk merepresentasikan program**

**Simbol-simbol dalam Flowchart :**

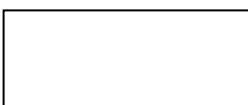
Bagan alir program dibuat dengan menggunakan simbol-simbol sebagai berikut ini:



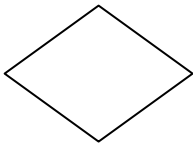
Simbol titik terminal untuk menunjukkan awal dan akhir dari suatu proses



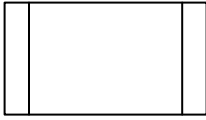
Simbol input / output digunakan untuk data input/output



Simbol proses digunakan untuk mewakili suatu proses



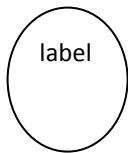
Simbol keputusan (decision symbol) untuk menyeleksi kondisi didalam program.



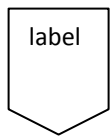
Simbol proses terdefinisi, digunakan untuk menunjukkan suatu proses operasi yang rinciannya ditunjukkan di tempat lain.



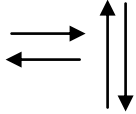
Simbol perulangan, digunakan untuk memberikan batasan nilai iterasi/ perulangan.



Simbol penghubung dalam halaman yang sama



Simbol penghubung dalam halaman yang berbeda



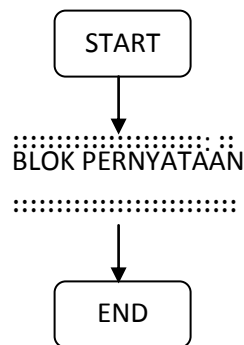
Simbol garis alir untuk menunjukkan arus dari suatu proses

Berikut ini gambaran sintak bahasa C dengan flowchart secara terpenggal:

```

void main()
{
.....
BLOK PERNYATAAN
.....
}

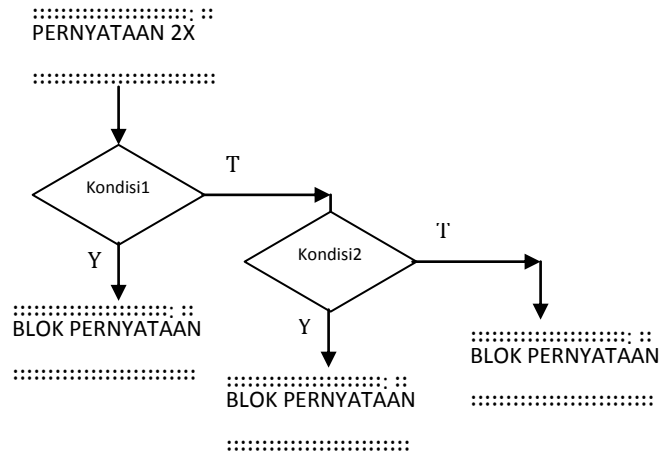
```



```

.....
.....
switch (var)
  case kondisi1 : pernyataan;
  case kondisi2 : pernyataan;
  default : pernyataan;
.....
.....

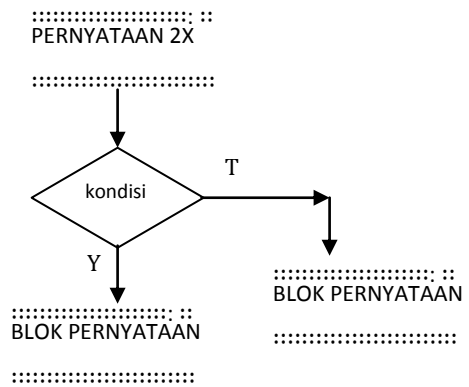
```



```

.....
.....
If (kondisi)
  Pernyataan;
.....
.....

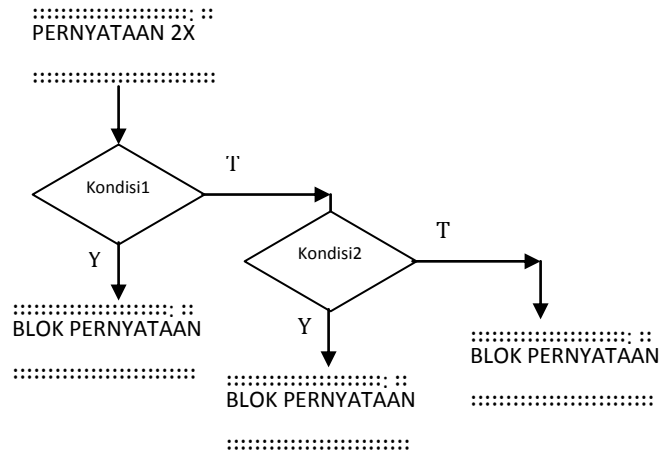
```



```

.....
.....
if (kondisi1) pernyataan;
else if (kondisi2) pernyataan;
else pernyataan;
.....
.....

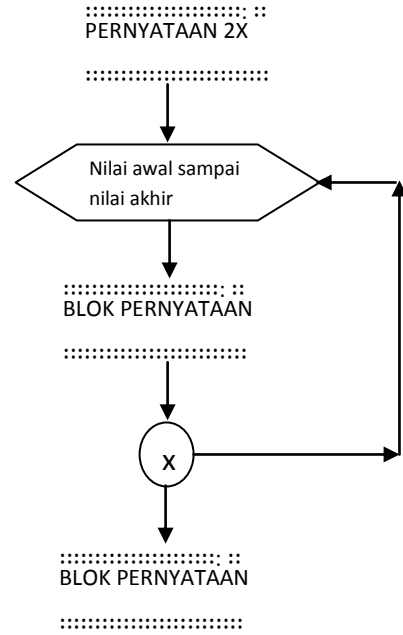
```



```

.....
.....
for (nilai awal;kondisi;kenaikan/penurunan nilai counter)
    pernyataan;
.....
.....

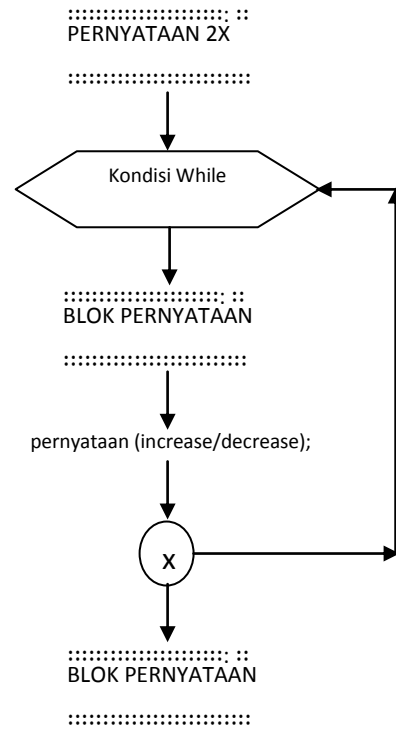
```



```

.....
.....
while (kondisi)
{
.....
BLOK PERNYATAAN
.....
    pernyataan (increase/decrease);
}
.....
.....

```

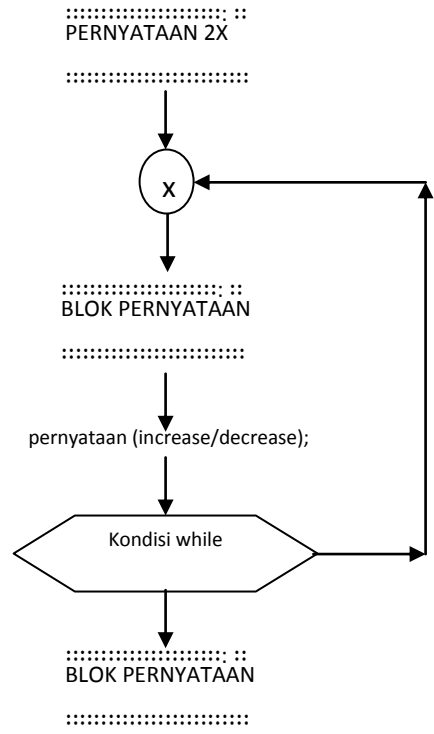




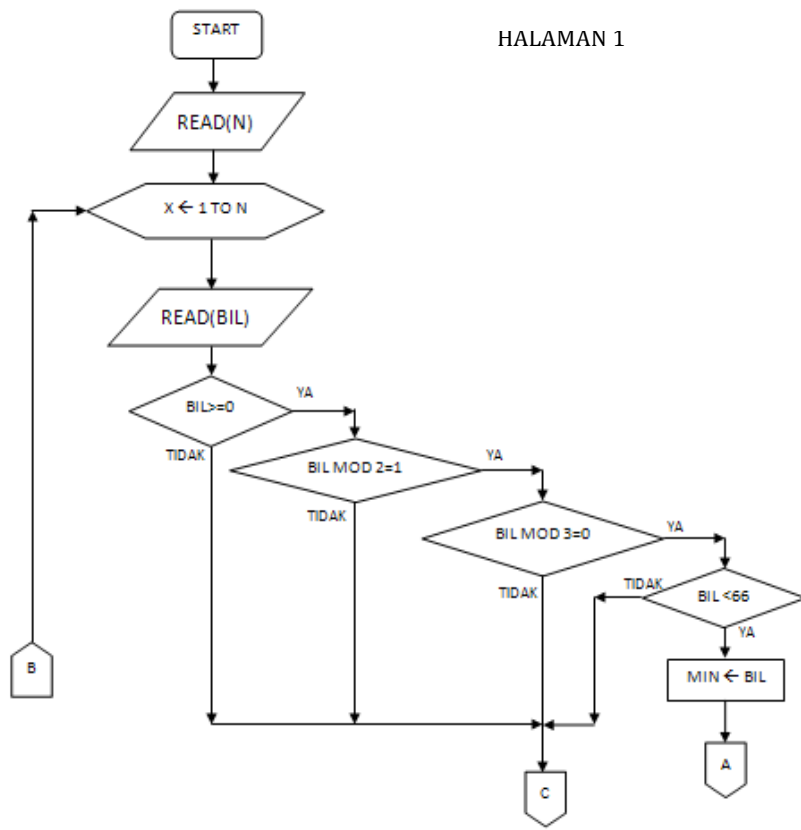
```

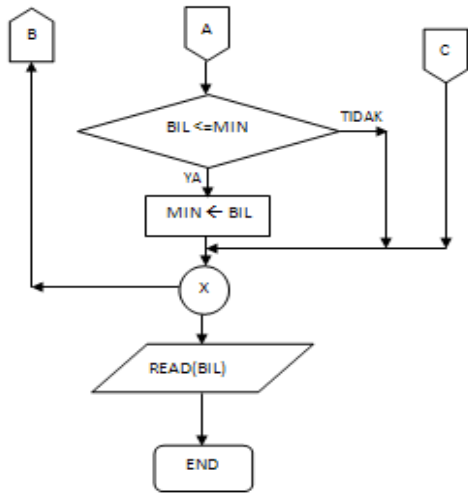
....
do
{
    .....
    BLOK PERNYATAAN
    .....
    pernyataan (increase/decrease);
}
while (kondisi);
....
....

```

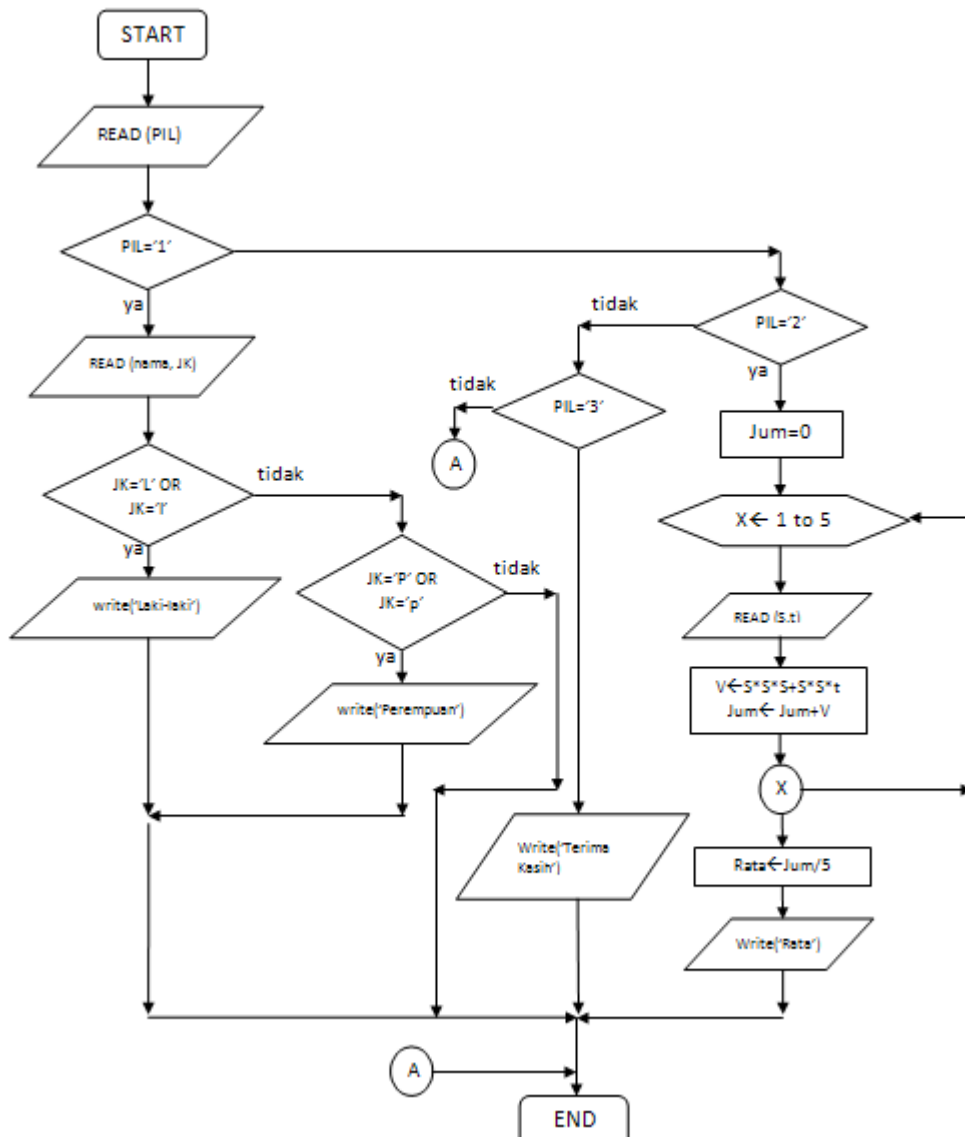


HALAMAN 1





FLOWCHART :



SOAL1 :  
**BUATLAH PSEUDOCODE, FLOWCHART DAN PROGRAM DALAM BAHASA C++ UNTUK MENAMPILKAN BILANGAN 1-50 YANG TIDAK HABIS DIBAGI 3 DAN 5. DAN Mencari KELILING BALOK. GUNAKAN PERNYATAAN IF, FOR DAN SWITCH CASE.!**

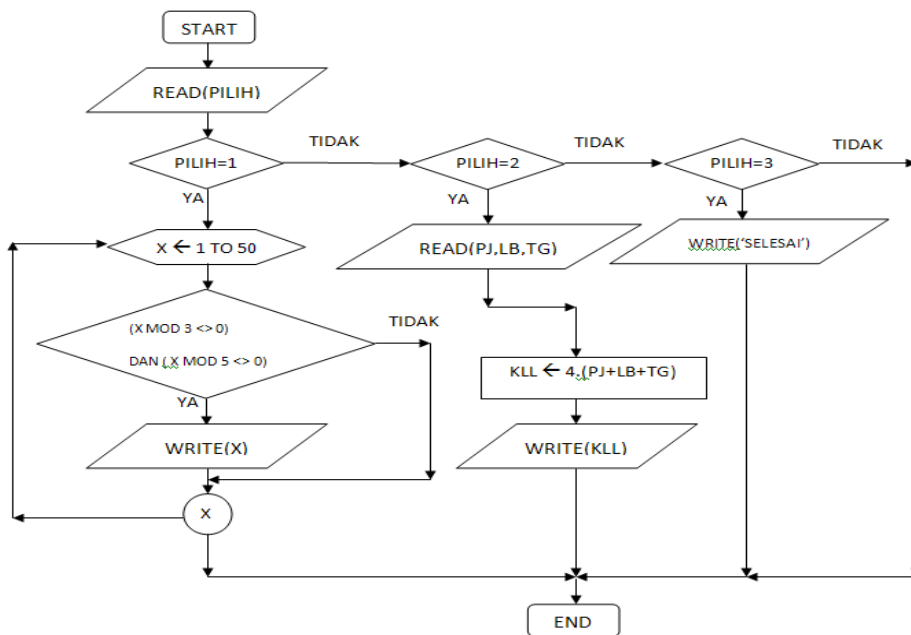
**M E N U**

- ```
=====
1. MENAMPILKAN BILANGAN 1-50
   YANG TIDAK HABIS DIBAGI 3
   DAN 5
2. Mencari KELILING BALOK
3. EXIT
=====
```

MASUKKAN PILIHAN ANDA :

ALGORITMA SOAL1\_

```
DEKLARASI :
    PILIH,X,PJ,LB,TG,KLL:INTEGER
DESKRIPSI :
    READ(PILIH)
    CASE PILIH
    1:    FOR X<-- 1 TO 100
          IF (X MOD 3<>0) AND (X MOD 5<>0)
            WRITE(X)
          ENDIF
        ENDFOR
    ENDCASE1
    2:    READ(PJ,LB,TG)
          KLL<-- 4*(PJ+LB+TG)
          WRITE(KLL)
    ENDCASE2
    3:    WRITE('SELESAI')
    ENDCASE
```



```

//PROGRAM SOAL1
#include<iostream>
#include<conio.h>
using namespace std;
void main()
{
    //DEKLARASI
    PILIH,X,PJ,LB,TG,KLL:INTEGER
    //DESKRIPSI
    cin>>PILIH
    switch (PILIH)
    {
        case 1: for (X=0;x<=50;x++)
                IF (X % 3!= 0 AND X % 5!= 0)
                    cout<<X;
                break;
        case 2: cin>>PJ>>LB>>TG;
                KLL= 4*(PJ+LB+TG); cout<<KLL;
                break;
        case 3: cout<<"SELESAI";
                break;
    }
}

```

Contoh :

algoritma Faktorial\_permutasi

Deklarasi:

```

x,n,r ,fak1,fak:integer
permutasi: real

```

Deskripsi:

```

read(n)
read(r)
if (n>=r)
    fak<--1
    for x<-- 1 to n
        fak<--fak*x
    endfor
    write(fak)
    { apakah n=r ??? }
    if (n=r)
        fak1<--1
    endif
    else if (n<>r)
        fak1<--1
        for x<-- 1 to(n-r)
            fak1<--fak1*x
        write(n-r,fak1)
        endif

    permutasi<--fak/fak1
    write(permutasi)
endif

```

PROGRAM Faktorial\_permutasi:

```
#include "iostream"
#include "conio.h"
using namespace std;
void main()

{ //RUMUS PERMUTASI= N!/(N-R)!

    short int x,n,r ,fak1,fak;    float permutasi;

    cout<<"mencari nilai faktorial"<<endl;
    cout<<"n=";cin>>n;    cout<<"r=";cin>>r;
    if (n>=r) // syarat nilai faktorial tidak pernah negatif
    {
        // CARI NILAI N!
        fak=1;
        for (x=1;x<=n;x++)
            fak=fak*x;
        cout<<endl<<n<<"!="<<fak;
        // CARI NILAI (N-R)!

        // apakah n=r ???
        if (n==r)
            fak1=1;//TETAPKAN LANGSUNG FAKT1=1 ATAU 0!=1
        else if (n!=r) // JIKA TIDAK SAMA???
        {
            fak1=1;
            for(x=1;x<=(n-r);x++)
                fak1=fak1*x;
            cout<<endl<<n-r<<"!="<<fak1;
        }
        permutasi=float(fak/fak1);// HITUNG LALU KOVERSI KE REAL(BIL BERKOMA)
        cout<<endl<<n<<"P"<<r<<"="<<permutasi;
    } getch();
} //akhir program
```

## Bandungkanlah :

algoritma Perulangan\_for\_bersarang

```
DEKLARASI :
    X,Y,Z :BYTE
DESKRIPSI :
    for x<-- 1 to 3
        for y<-- 5 downto 3
            for z <-- 41 to 44
                write(x,y,z)
```

**algoritma Perulangan\_while\_bersarang**

DEKLARASI :

X,Y,Z :BYTE

DESKRIPSI :

K←-1

WHILE (K<= 3)

write(K,L,M)

K←-K+1

L←-5

WHILE(L<=3)

write(K,L,M)

L←-L+1

M←-41

WHILE (M <=44)

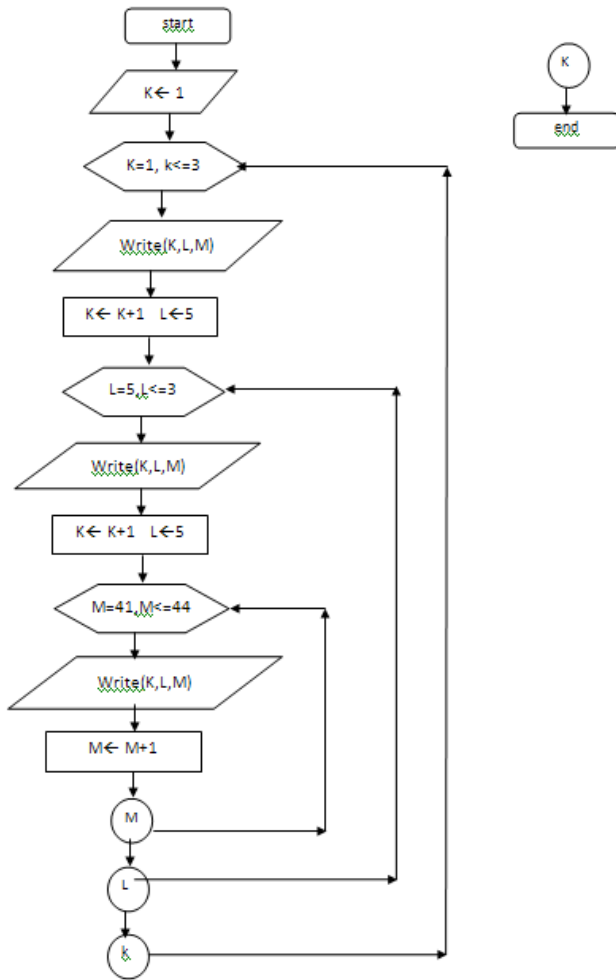
write(x,y,z)

M←-M+1

ENDWHILE

ENDWHILE

ENDWHILE



Selesai.....