

Contoh Algoritma Insertion Sort untuk Latihan

written by [Teddy](#)

Dalam dunia programming, penguasaan algoritma adalah sebuah hal penting. Karena algoritma adalah dasar dari segala jenis pemrograman, baik untuk pemrograman Android, Windows, dan lainnya. [Pengertian algoritma pemrograman](#) adalah urutan langkah logis tertentu untuk memecahkan suatu masalah. Pengertian lainnya dari algoritma pemrograman adalah urutan langkah-langkah logis penyelesaian masalah yang disusun sistematis.

Dalam algoritma, alur pemikiran dalam menyelesaikan suatu persoalan dituangkan secara tertulis. Algoritma seseorang bisa berbeda dari algoritma orang lain. Sedangkan, alur pemikiran tersebut bisa berupa kalimat, gambar, atau tabel tertentu.

Algoritma sendiri terdiri dari beberapa jenis. Salah satu jenisnya adalah algoritma insertion sort. [Fungsi algoritma](#) insertion sort adalah untuk mengurutkan sebuah array data yang tidak terurut agar menjadi sebuah array yang terurut. Algoritma insertion sort adalah algoritma pengurutan yang menggunakan dua buah list untuk proses pengurutannya. dua list tersebut yaitu sorted list dan unsorted list.

Pada kondisi awal, semua bilangan yang hendak diurutkan berada dalam kondisi “unsorted list”. Lalu, index “0” dari unsorted list dipindahkan ke sorted list. Kemudian berlanjut ke index “1” dan seterusnya. Lalu, index “0” dan “1” yang sudah dipindahkan akan dibandingkan berdasarkan nilai terkecil. Index yang memiliki terkecil itu selanjutnya akan menjadi index 0 atau index paling awal. Begitu juga yang terjadi pada index seterusnya, dimana tiap index yang baru saja dimasukan dari unsorted list ke sorted list akan dibandingkan dengan semua index yang sudah duluan masuk ke sorted list. Proses ini akan berulang sampai semua index yang ada di unsorted list berpindah ke sorted list dan dibandingkan. Barulah kita dapat melihat hasil pengurutan tersebut.

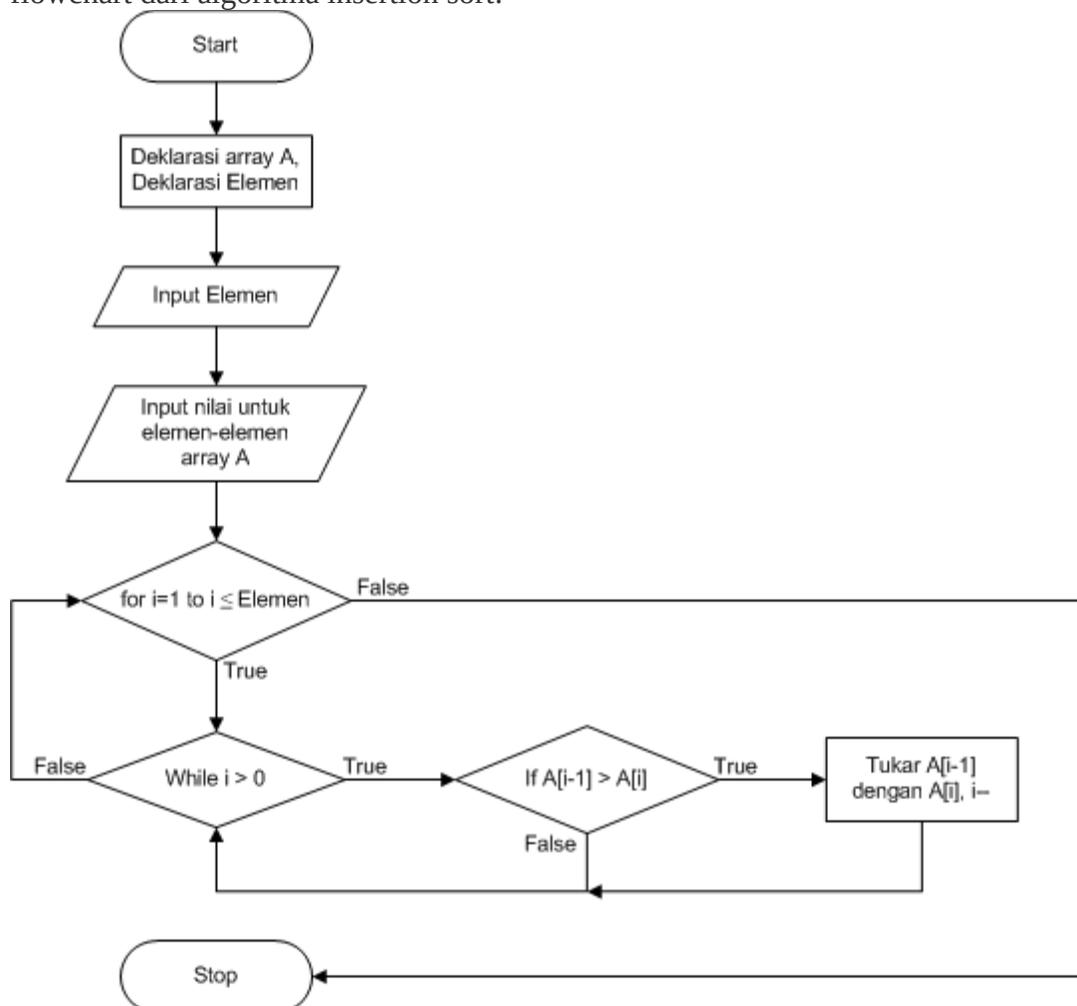
Untuk mempermudah pemahaman Anda, algoritma insertion sort kita analogikan dengan pengurutan kartu. Anggaplah Anda hendak mengurutkan satu set kartu dari kartu yang bernilai paling kecil hingga yang bernilai paling besar. Berikut penganalogiannya:

1. Semuanya dimulai dari posisi tangan kosong dan semua kartu masih tersimpan di atas meja. Anggap saja Anda hendak menyusun kartu ke tangan kiri Anda.
2. Anda lalu mengambil kartu pertama dari meja dan menyimpannya di tangan kiri.
3. Kemudian, Anda mengambil kartu kedua dari meja. Lalu, Anda membandingkan kartu tersebut dengan kartu yang ada di tangan kiri.
4. Jika kartu yang baru saja diambil dari meja memiliki nilai yang lebih kecil daripada kartu di tangan kiri, maka kartu tersebut akan diletakan di depan kartu pembanding
5. Kartu yang telah dibandingkan akan bergeser mundur.
6. Proses ini akan terus berlangsung sampai semua kartu terurut dengan benar sesuai kriteria pengurutannya.
7. Setelah semua kartu terurut, satu set kartu yang sudah terurut akan disimpan kembali ke meja. Berikut adalah contoh algoritma insertion sort jika dituliskan:

Insertion_Sort(A)

- Deklarasi Array A
- Deklarasi Elemen
- Input elemen array A
- Input nilai elemen-elemen array A
- For $i=1$ sampai $i <$ elemen
- While $i > 0$
- If $A[i-1] > A[i]$
- Tukar $A[i-1]$ dengan $A[i]$
- $i-1$
- end if
- end while
- end for

Algoritma di atas bisa digambarkan dalam sebuah flowchart. [Fungsi flowchart pada pemrograman](#) adalah untuk memudahkan perancangan sebuah program. Dan berikut ini adalah flowchart dari algoritma insertion sort:



Flowchart lainnya untuk bahan latihan bisa Anda lihat di artikel kami seputar [contoh flowchart program](#).

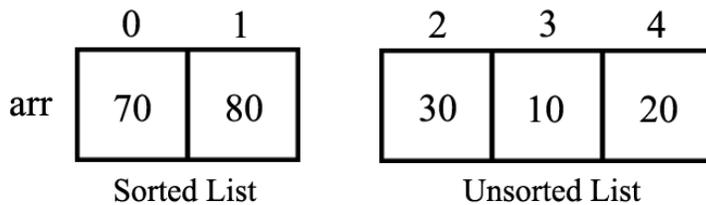
Pada artikel kali ini, kami akan memberikan beberapa contoh algoritma insertion sort. Mari kita simak contoh-contoh berikut ini:

- **Contoh 1**

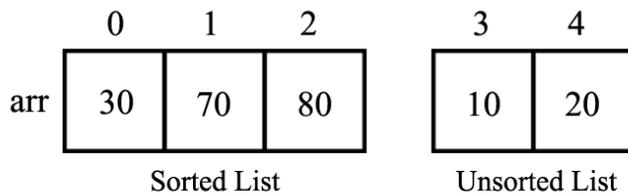
Langkah 1. Terdapat array dengan 5 elemen seperti di bawah ini:

	0	1	2	3	4
arr	70	80	30	10	20

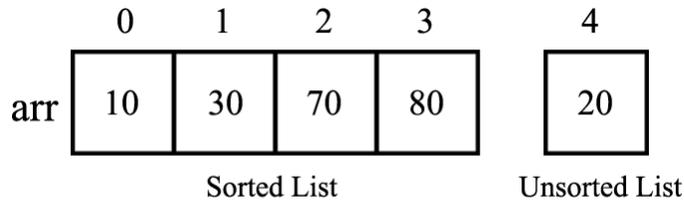
Selanjutnya data pada index 0 dan 1 akan dipindahkan ke sorted list. Setelahnya, data pada index 0 dan 1 di sorted list akan dibandingkan untuk mencari index yang memiliki nilai terkecil. Dari data di atas, terlihat bahwa nilai pada index 0 lebih kecil dari nilai index 1. Maka pada sorted list, tidak terjadi perubahan posisi. Hasilnya dapat dilihat pada gambar di bawah ini :



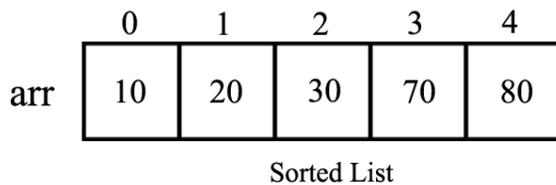
Langkah 2. Data pada index 2 dipindahkan ke sorted list. Lalu, data tersebut dibandingkan dengan data-data pada index 0 dan 1. Pada contoh kali ini, hasilnya data pada index 2 memiliki nilai lebih kecil ketimbang data di index 0 dan 1. Hasilnya, data yang baru masuk sorted list tersebut diposisikan ke index 0 dan data-data sebelumnya bergerak mundur. Hasil komparasinya akan seperti gambar berikut:



Langkah 3. Data pada index 3 masuk ke sorted list dan dibandingkan dengan data-data yang sudah ada di sorted list. Karena nilai data pada index 3 lebih kecil daripada nilai data di index 0 sampai 2, maka data pada index 3 diposisikan ke index 0 dan data-data lainnya bergerak mundur. Jadinya akan seperti berikut:



Langkah 4. Langkah terakhir adalah memasukkan data pada index 4 di unsorted list ke dalam sorted list. Setelah dibandingkan, nilai pada data index 4 lebih kecil ketimbang nilai data pada index 1 sampai 3. Hasilnya, data index 4 akan diposisikan ke index 2 dan data setelahnya akan bergerak mundur. Maka, hasilnya akan seperti gambar di bawah ini:



Dengan hasil tersebut, maka proses pengurutan dengan metode insertion sort sudah selesai.

▪ **Contoh 2**

Data awal: [5, 2, 4, 6, 1, 3]. Jumlah index adalah 6, dimulai dari 0 sampai 5. Anggaplah index adalah “I”, dimana untuk setiap proses pengurutan, perbandingan data akan dimulai dari index kedua (dalam hal ini i=1)

Proses I:

i=1, x=1; j=0

$x < j \wedge 2 < 5$? — true = 2, 5, 4, 6, 1, 3

Proses II

i=2, j=1, x=2

$x < j \wedge 4 < 5$ — true = 2, 4, 5, 6, 1, 3; j=j-1. Artinya jika proses benar, maka “x=x-1”

$x < j \wedge 4 < 2$ — false = 2, 4, 5, 6, 1, 3

Proses III

I=3, j=2, x=3

$x < j \wedge 6 < 5$ — false = 2, 4, 5, 6, 1, 3; j=j-1. Artinya jika sebuah proses bernilai false, maka proses tersebut tidak akan dilanjutkan, karena semua data yang ada di sebelah kiri sudah terurut dengan benar secara otomatis .

Proses IV

i=4, j=3, x=4

$x < j \wedge 1 < 6$ — true = 2, 4, 5, 1, 6, 3; j=j-1, jika benar maka $x=x-1$

$x < j \wedge 1 < 5$ — true = 2, 4, 1, 5, 6, 3; j=j-1, jika benar maka $x=x-1$

$x < j \wedge 1 < 4$ — true = 2, 1, 4, 5, 6, 3; j=j-1, jika benar maka $x=x-1$

$x < j \wedge 1 < 2$ — true = 1, 2, 4, 5, 6, 3

Proses V

i=5, j=4, x=5

$x < j \wedge 3 < 6$ — true = 1, 2, 4, 5, 3, 6 j=j-1, jika benar maka $x=x-1$

$x < j \wedge 3 < 5$ — true = 1, 2, 4, 3, 5, 6 j=j-1, jika benar maka $x=x-1$

$x < j \wedge 3 < 4$ — true = 1, 2, 3, 4, 5, 6 j=j-1, jika benar maka $x=x-1$

$x < j \wedge 3 < 2$ — false = 1, 2, 3, 4, 5, 6 j=j-1

Berikut adalah penerapannya di dalam program menggunakan bahasa pemrograman C++:

```
#include<iostream>
#include <conio.h>
int main()
{
int data[]={5, 2, 4, 6, 1, 3};
cout<<"sebelum disorting: ";
for(int i=0; i<6; i++)
cout<<data[i] <<" ";
cout<<endl <<endl;
for(int i=1; i<6; i++)
{
int j=i;
while(data[j]<data[j-1])
{
int tmp=data[j];
data[j]=data[j-1];
data[j-1]=tmp;
j--;
}
}
cout<<"Setelah disorting: ";
for(int i=0; i<6; i++)
cout<<data[i] <<" ";
getch();
}
```

Setelah program di atas dieksekusi, maka hasilnya akan seperti gambar di bawah:

- **Contoh 3**

Contoh algoritma insertion sort dalam sebuah program dengan bahasa pemrograman C#:

Sintaks:

```
using System;
using System.Text;
namespace tester
{
class Insertion
{
public void InsertionSort()
{
Console.Clear();
Console.WriteLine("Masukkan Banyak Elemen : ");
/* Deklarasi variabel untuk input jumlah elemen array yg akan diurutkan */
string Input=Console.ReadLine();
int Elements;
if(int.TryParse(Input, out Elements))
{
Elements = Convert.ToInt32(Input);
}
else
{
Console.WriteLine("Maaf input yang Anda masukkan salah. Silahkan tekan Enter untuk
mengulang");
Console.ReadLine();
InsertionSort();
}
/* Deklarasi array untuk menampung angka-angka yang akan diurutkan dan elemen array
berdasarkan input user (variabel Elements) */
int[] Angka = new int[Elements];
Console.WriteLine("-----");
/* Metode untuk input angka yang akan disimpan pada masing-masing element di Array */
for (int i = 0; i < Elements; i++)
{
Console.WriteLine("Masukkan angka untuk mengisi elemen " + i + ":");
string Input_Temp = Console.ReadLine();
int angka;
if(int.TryParse(Input_Temp, out angka))
{
Angka[i] = Convert.ToInt32(angka);
}
}
else
```

```

{
Console.WriteLine("Maaf input yang Anda masukkan salah. Silahkan tekan Enter untuk
mengulang");
Console.ReadLine();
InsertionSort();
}
}
/* Metode Insertion Sort */
for(int i=1; i < Elements; i++) { while(i>0)
{
if(Angka[i-1] > Angka[i])
{
int Temp = Angka[i-1];
Angka[i-1] = Angka[i];
Angka[i] = Temp;
i--;
}
else
{
break;
}
}
}
/* Menampilkan hasil pengurutan */
Console.WriteLine("");
Console.WriteLine("Hasil pengurutan nilai : ");
for (int i = 0; i < Elements; i++)
{
Console.WriteLine("Elemen" + i + " :");
Console.WriteLine(Angka[i]);
}
}
static void Main(string[] args)
{
Insertion I = new Insertion();
I.InsertionSort();
}
}
}

```

Hasil eksekusi program:

```
C:\Windows\system32\cmd.exe
Masukkan Banyak Elemen yang anda inginkan :
5
-----
Silahkan masukkan angka untuk mengisi elemen 0:
70
Silahkan masukkan angka untuk mengisi elemen 1:
80
Silahkan masukkan angka untuk mengisi elemen 2:
30
Silahkan masukkan angka untuk mengisi elemen 3:
10
Silahkan masukkan angka untuk mengisi elemen 4:
20

Hasil Sorting nilai :
Elemen0:
10
Elemen1:
20
Elemen2:
30
Elemen3:
70
Elemen4:
80
```

Contoh Pemrograman Terstruktur untuk Bahan Belajar Programming

written by [Teddy](#)

Pemrograman terstruktur adalah teknik pemrograman dalam merangkai instruksi-instruksi dalam bahasa komputer agar bisa tersusun secara logis dan sistematis agar mudah dimengerti, diuji, dan dimodifikasi. Pemrograman terstruktur membuat sebuah program sebagai kumpulan prosedur. Dimana prosedur-prosedur tersebut dapat saling memanggil dan dipanggil dari manapun di dalam program serta dapat menggunakan parameter yang berbeda-beda untuk setiap pemanggilan. Bisa dibilang, pemrograman terstruktur ini merupakan implementasi dari [fungsi flowchart dalam pemrograman](#).

Manfaat pemrograman terstruktur dalam pembuatan program komputer adalah sebagai berikut:

- Dapat membuat program yang kompleks
- Menghindari konflik di internal tim pembuat program
- Membagi kerja tim pembuat program berdasarkan modul-modul program yang sudah dirancang
- Kemajuan pengerjaan sistem dapat diawasi dan juga dikaji

Sebuah program yang dibuat dengan pemrograman terstruktur dapat dilihat dari ciri-ciri program yang dibuat tersebut, diantaranya:

- Program tersebut handal dan berjalan dengan lancar (Run correctly)
- Program tersebut sederhana dan tidak rumit (Run efficiently)
- Program tersebut mudah dibaca dan ditelusuri kodingnya (Be easy to read and understand)
- Program tersebut mudah ditelusuri kesalahannya (Be easy to debug)
- Program tersebut mudah dimodifikasi oleh orang lain untuk dikembangkan (Be easy to modify)

Pada kesempatan kali ini, kami akan memberikan beberapa contoh pemrograman terstruktur yang bisa Anda praktekan sendiri di rumah. Bahasa pemrograman yang digunakan adalah C++ dan Java. Berikut ini contoh-contohnya:

Contoh 1

```
#include <iostream.h>
void garis() {
for (i=0; i<49; i++) cout << '-';
cout << endl;
}
void judul(){
int i;
cout << "\t\tDaftar Buku"<< endl; // \t = tabulasi
garis();
cout <<"Judul Buku"<<"\t\tPengarang"<< endl;
garis();
}
void main() {
judul();
}
```

Contoh 2: Program untuk menghitung Luas Segitiga

```

// modul penghitung luas segitiga
float Luas_sgt(float alas, float tinggi)
{
return (alas * tinggi / 2);
}
//main program
void main() {
float A, T;
cout <<"Data Alas      : "; cin >> A;
cout <<"Data Tinggi    : "; cin >> T;
cout <<"Luas segitiga : " << Luas_sgt(A,T)<<endl;
}

```

Contoh 3: Program Java untuk Penghitungan Geometri

```

import java.io.*;
public class geoterstruktur
{
public static void main (String[] args) throws IOException
{
int panjang, lebar, tinggi, volume, luas, keliling;
BufferedReader input=new BufferedReader (new InputStreamReader(System.in));
System.out.println("                G E O M E T R I                ");
System.out.println("=====");
System.out.println("");
System.out.println("");
System.out.print("Masukkan Panjang = ");
panjang=Integer.parseInt(input.readLine());
System.out.print("Masukkan Lebar   = ");
lebar=Integer.parseInt(input.readLine());
System.out.print("Masukkan Tinggi  = ");
tinggi=Integer.parseInt(input.readLine());
luas= panjang*lebar;
keliling=2*(panjang+lebar);
volume=panjang*lebar*tinggi;
System.out.println("                ");
System.out.println("PERSEGI PANJANG");
System.out.println("-----");
System.out.println(" Panjang   : "+panjang);
System.out.println(" Lebar     : "+lebar);
System.out.println(" Tinggi    : "+tinggi);
System.out.println(" Luas      : "+luas);
System.out.println(" Keliling  : "+keliling);
System.out.println("                ");
System.out.println("BALOK ");
System.out.println("-----");
System.out.println("Panjang    : "+panjang);
System.out.println("Lebar      : "+lebar);
System.out.println("Tinggi     : "+tinggi);
System.out.println("Volume     : "+volume);
System.out.println("                ");
System.out.println("PROGRAM SELESAI ");
}
}

```

Contoh 4: Program C++ Hitungan Geometri dengan “If Else”

```

#include <iostream.h>
#include <conio.h>
#include <stdlib.h>
void main()
{
unsigned int pilihan;
cout << "MENU PILIHAN" << endl;
cout << "-----" << endl;
cout << "1. hitung luas lingkaran" << endl;
cout << "2. hitung luas persegi" << endl;
cout << "3. hitung luas segitiga" << endl;
cout << "4. exit" << endl;
cout << "-----" << endl;
while (pilihan>=0){
cout << "pilihan anda (1/2/3/4) : ";
cin >> pilihan;
if (pilihan== 1) {
cout << "-----" << endl;
cout << "Menghitung Luas Lingkaran" << endl;
float r;
cout << "Masukan jari-jari : ";
cin >> r;
float Ll = 3.14*r*r;
cout << "Luas Lingkaran : " << Ll << endl;
cout << "-----" << endl;
} else if (pilihan== 2){
cout << "-----" << endl;
cout << "Menghitung Luas Persegi" << endl;
float p;
cout << "Masukan Panjang : ";
cin >> p;
float l;
cout << "Masukan Lebar : ";
cin >> l;
float Lp = p*l;
cout << "Luas Persegi : " << Lp << endl;
cout << "-----" << endl;
} else if (pilihan== 3){
cout << "-----" << endl;
cout << "Menghitung Luas Segitiga" << endl;
float a;
cout << "Masukan Alas : ";
cin >> a;
float t;
cout << "Masukan Tinggi : ";
cin >> t;
float Ls = 0.5*a*t;
cout << "Luas Segitiga : " << Ls << endl;
cout << "-----" << endl;
} else if (pilihan==4) {
exit(4);
} else {
cout << "Kode Yang Anda Masukan Salah...!!!" << endl;
}
}
}

```

```
getch();
}
```

Contoh 5: Progra C++ Hitungan Geometri dengan “Switch Case”

```
#include <iostream.h>
#include <conio.h>
#include <stdlib.h>
void main()
{
    unsigned int pilihan;
    float r, Ll, p, l, Lp, a, t, Ls;
    cout << "MENU UTAMA" << endl;
    cout << "-----" << endl;
    cout << "1. Menghitung luas lingkaran" << endl;
    cout << "2. Menghitung luas persegi" << endl;
    cout << "3. Menghitung luas segitiga" << endl;
    cout << "4. Keluar" << endl;
    cout << "-----" << endl;
    while (pilihan>=0){
        cout << "pilihan anda (1/2/3/4) : ";
        cin >> pilihan;
        switch(pilihan) {
            case 1:
                cout << "-----" << endl;
                cout << "Menghitung Luas Lingkaran" << endl;
                cout << "Masukan jari-jari : ";
                cin >> r; Ll = 3.14*r*r;
                cout << "Luas Lingkaran : " << Ll << endl;
                cout << "-----" << endl;
                break;
            case 2:
                cout << "-----" << endl;
                cout << "Menghitung Luas Persegi" << endl;
                cout << "Masukan Panjang : ";
                cin >> p;
                cout << "Masukan Lebar : ";
                cin >> l; Lp = p*l;
                cout << "Luas Persegi : " << Lp << endl;
                cout << "-----" << endl;
                break;
            case 3:
                cout << "-----" << endl;
                cout << "Menghitung Luas Segitiga" << endl;
                cout << "Masukan Alas : ";
                cin >> a;
                cout << "Masukan Tinggi : ";
                cin >> t; Ls = 0.5*a*t;
                cout << "Luas Segitiga : " << Ls << endl;
                cout << "-----" << endl;
                break;
            case 4:
                exit(4);
                break;
            default:
                cout << "Kode Yang Anda Masukan Salah...!!!" << endl;
        }
    }
}
```

```
}  
getch();  
}
```

Contoh 6

```
#include <stdio.h>  
main()  
{  
int A=1;  
clrscr();  
switch(A)  
{  
case 1: printf("Satu \n"); break;  
case 2: printf("Dua \n"); break;  
default: printf("Bukan Satu atau Dua...!\n"); break;  
}  
}
```

Contoh 7

```
#include <stdio.h>  
main()  
{  
int i,  
clrscr();  
for(i=0;i<5;i++)  
/*pengulangan diproses sebanyak 0 sampai 4, kenapa bisa begitu...? */  
{  
printf("%d\n",i);  
}  
}
```

Contoh 8: Program Kalkulator dengan C++

```
#include <stdio.h>  
#include <math.h>  
int nilai_A, nilai_B;  
long nilai_C;  
void InputData()  
{  
clrscr();  
printf("Input Data \n");  
printf("Masukan Nilai A = ");scanf("\n%d",&nilai_A);  
printf("Masukan Nilai B = ");scanf("\n%d",&nilai_B);  
}  
int Tambah(int a,int b)  
{  
int c;  
c=a+b;  
printf("Melaksanakan Program Penambahan \n");  
return c;  
}  
int Kurang(int a,int b)  
{  
printf("Melaksanakan Program Pengurangan \n");  
return (a-b);  
}  
long Kali(int a,int b)  
{  
printf("Melaksanakan Program Perkalian \n");
```

```

return (a*b);
}
float Bagi(int a,intb)
{
printf("Melaksanakan Program Pembagian \n");
return a/b;
}
long Pangkat(int a,int b) /* untuk pemangkatan, Anda dapat menggunakan fungsi
pow() dari lib
<math.h> */
{
long temp;
int i;
printf("Melaksanakan Program Pangkat \n");
temp=1;
for(i=0;i<b;i++)
{
temp=temp*a;
}
return temp;
}
main()
{
InputData();
nilai_C=Tambah(nilai_A,nilai_B);
printf("\nHasil A + B = %d",nilai_C);
printf("\nHasil A - B = %d",Kurang(nilai_A,nilai_B);
printf("\nHasil A x B = %d",Kali(nilai_A,nilai_B);
printf("\nHasil A / B = %d",Bagi(nilai_A,nilai_B);
printf("\nHasil A ^ B = %d",Pangkat(nilai_A,nilaiB);
}

```

Contoh 9

```

#include <stdio.h>
main()
{
int angka[10];
int i;
clrscr();
for (i=0;i<5;i++)
{
printf("Masukan angka sembarang : ");
scanf("\n%d",angka[i]);
}
printf("\n");
for (i=0;i<5;i++)
{
printf("Angka ke-%d = %d",i,angka[i]);
}
}

```

Contoh 10: Program konversi bilangan desimal ke bilangan biner

```

#include <iostream.h>
#include <conio.h>
#include <math.h>
void main()
{

```

```

int p,n,i=0;
cout<<"Masukkan bilangan desimal: ";
cin>>p;
double A[100];
do
{
A[++i]=p%2;
p=p/2;
floor(p);
}
while (p>1);
cout<<"Nilai binernya adalah: ";
cout<<p;
for(n=i;n>=1;n--)
{
cout<<A[n];
}
getch();
}

```

Contoh 11: Program Perulangan (Loop)

```

#include <iostream.h>
void main()
{
int n;
for(;;)
{
cout<<"Masukkan bilangan integer: "; cin>>n;
if(n % 2 == 0) continue;
else if(n % 5 == 0) break;
cout<<"\tLanjutkan loop selanjutnya.\n";
}
cout<<"Akhir Loop.\n";
}

```

Dari semua contoh program yang telah kami berikan di atas, sebenarnya jika Anda sebelumnya telah mencoba [contoh bahasa pemrograman C++](#) pada situs kami sebelumnya, Anda juga telah mempraktekkan pemrograman terstruktur. Jadi, untuk mempelajari [macam-macam bahasa pemrograman](#), Anda harus memiliki pola pikir tersusun layaknya pemrograman terstruktur. Karena, pemrograman terstruktur bisa diaplikasikan di semua bahasa pemrograman. Namun, pemrograman terstruktur sendiri direkomendasikan untuk diterapkan pada beberapa bahasa pemrograman, diantaranya:

1. Cobol Turbo Prolog
2. C atau C++
3. Pascal
4. Delphi
5. Borland Delphi

Untuk program yang sederhana, biasanya digunakan pemrograman terstruktur karena masih mudah untuk dilakukan dan tidak banyak perubahan yang berdampak besar. Sedangkan untuk program yang jumlah barisnya lebih dari 100 atau rumit, sebaiknya menggunakan pemrograman berorientasi objek atau OOP (Object Oriented Programming) OOP sendiri [tingkatan bahasa pemrogramannya](#) lebih tinggi ketimbang pemrograman terstruktur. Untuk penjelasan mengenai

OOP, Anda bisa membaca beberapa bahasan kami seputar bahasa pemrograman yang berbasis OOP, salah satunya adalah ulasan kami seputar [kelebihan dan kekurangan Visual Basic](#). Sekian contoh-contoh pemrograman terstruktur yang bisa kami berikan kepada Anda pada artikel kali ini. Selamat mencoba dan jangan menyerah dalam mempelajari programming.