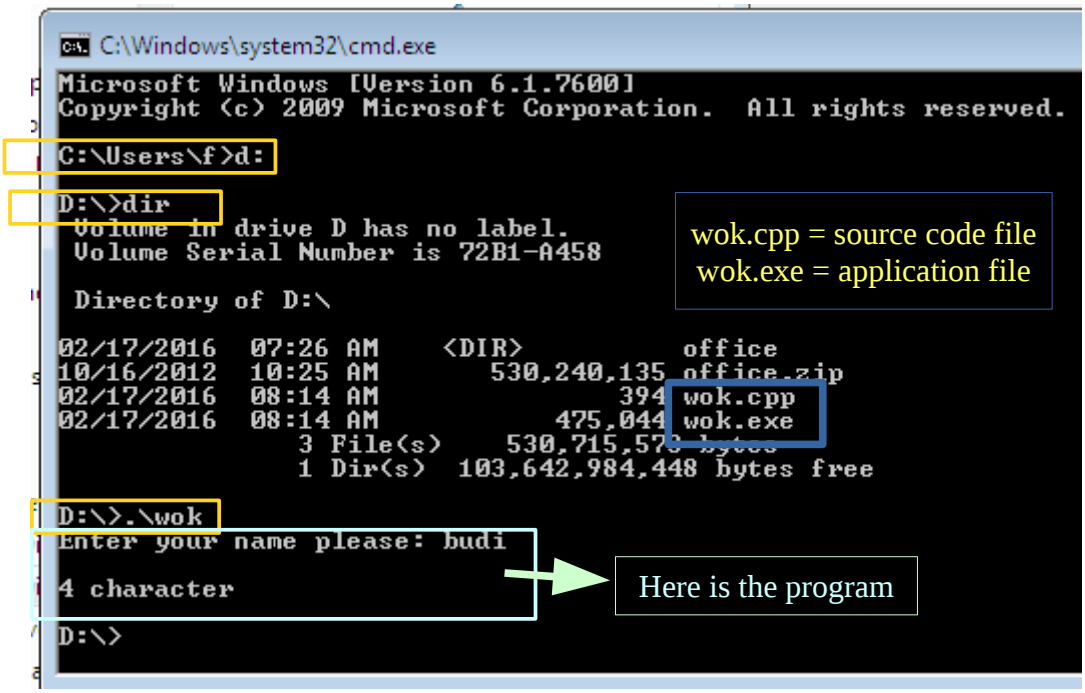
Compiling & Running C++

Windows

Setelah mengetikkan kode pada editor Devcpp, klik tombol compile & run.

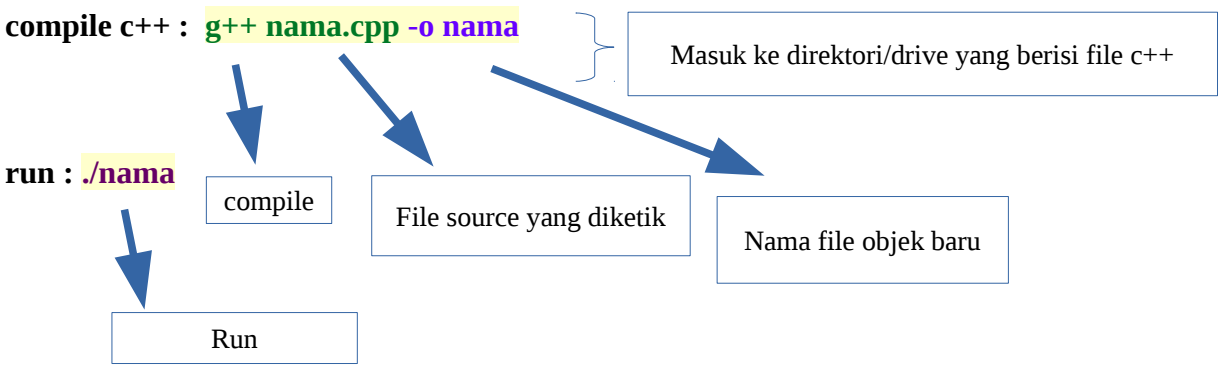
Untuk menjalankan aplikasi lewat command prompt:

1. Masuk ke drive tempat simpan file file c++→ C:\Users\f**d:**
2. Lihat isi drive→ D:\>**dir**
3. Jalankan file exe (harus compile dulu pada editor Devcpp)→ D:\>.**wok**



GNU/Linux Ubuntu

Compile, built and run c++ dari terminal



Operators

Combined assignment operators

operator	example	equivalent to
+=	x += 5;	x = x + 5;
-=	x -= 5;	x = x - 5;
*=	x *= 5;	x = x * 5;
/=	x /= 5;	x = x / 5;
%=	x %= 5;	x = x % 5;

Relational and comparison operators

operator	description
==	Equal to
!=	Not equal to
<	Less than
>	Greater than
<=	Lest han or equal to
>=	Greater than or equal to

Logical Operators : and (&&), or (||), xor (^), not (~)

a	b	a && b
1	1	1
1	0	0
0	1	0
0	0	0

a	b	a || b
1	1	1
1	0	1
0	1	1
0	0	0

a	b	a ^ b
1	1	0
1	0	1
0	1	1
0	0	0

a	~
1	0
0	1

Struktur utama C++

Output / Cout (using namespace std;)

```
#include <iostream> // atau #include "iostream"
using namespace std;
int main(void) // atau  int main()
{
  cout << "Test"  << " " << "Ha.ha.ha" << endl;
  cout << "Ha";
  return 0;
}
```

Fungsi utama m mengembalikan nilai 0
Boleh tidak digunakan

Output / Cout (std::cout)

```
#include <iostream>
int main(void)
{
   std::cout << "Test" << std::endl;
}
```

Output / Cout (using std::cout;)

```
#include <iostream>
using std::cout;
using std::endl;
int main(void)
{
   cout << "Test" << endl;
}
```

Output (c style)

```
#include <stdio.h>
int main()
{
    printf("test");
}
```

#include <iostream> = library/pustaka yang berisi perintah cin, cout, endl

int main(void) = fungsi utama
Void = null / kosong / tidak ada nilai
{ } = mulai, selesai

cout = cetak
endl = baris baru

using namespace std = sebagai pengganti std::cout atau std::endl

<< = kirim dari kanan ke kiri

return 0 = fungsi mengembalikan nilai nol

Variabel dengan tipe data short, long

```
#include "iostream"
signed short a, b, c;

int main(void)
{
   signed short d;

   std::cout << "Test" << std::endl;
    a = 5;
    b = 4;
    c = a + b;
    d = 100;
    std::cout << c << std::endl;
    std::cout << d;
}
```

Variabel global

Variabel lokal

Short 2 bytes
Range :
signed : -32768 to 32767
unsigned : 0 to 65535

Long 4 bytes
Range :
signed -2147483648 to 2147483647
unsigned 0 to 4294967295

int = short / long

Nilai default signed

Variabel dengan tipe data integer

```
#include <iostream>
int main()
{
  int x=5;
  int y(8); //int y=(8);
  int z{9}; //int z={9};
  std::cout << x << "\n" << y << "\n" << z;
}
```

C style

C++ style

C++ standard 2011

\n = baris baru alternatif endl

```
#include <iostream>
int main()
{
  int x, y, z;
  x = 5;
  y = (8);
  z = {9};
  std::cout << x << "\n" << y << "\n" << z;
}
```

Signed Int : -2.147.483.647 to 2.147.483.647 → 4 bytes
Unsigned int : 0 to 4.294.967.295 → 4 bytes

Variabel dengan tipe data string

```
#include <iostream>
int main()
{
 std::string x = "wokki's lab";
 std::string y = ("c++");
 std::string z(3, '!');
 std::string p = x + " " + y + z;

 std::cout << x << "\n" << y;
 std::cout << "\n" << z << "\n" << p;
}
```

RUN

```
wokki's lab
c++
!!!
wokki's lab c++!!!
```

Variabel dengan tipe data float, double, long double

```
#include <iostream>
using std::cout;
int main()
{
    float a, b, c;
    int d, e, f, g;

    a = 7;
    b = 3;
    c = a / b;

    d = 7;
    e = 3;
    f = d / e;
    g = d % e;

   cout << c << " ; " << f << " ; " << g;
}
```

Float : 3.4E +/- 38 (7 significant digits) → 4 bytes
Double : 1.7E +/- 308 (15 significant digits) → 8 bytes
Long double : 1.7E +/- 308 (15 significant digits) → 8 bytes

Hasil c
2.33333

Hasil f	Hasil g
2	1

Variabel dengan tipe data char & boolean

```
#include <iostream>
using namespace std;
int main()
{
    char a = 'y';
    bool b = false;
    bool c = true;
      if (a == 'y')    cout << b;
      else cout << c;
}
```

Char : 1 karakter (huruf atau angka dalam tanda petik)
Bool : true (1) or false (0)

RUN

0

Input / Cin

```
#include <iostream>
int main()
{
   std::string  name;

   std::cout << "Enter your name, please : ";
   std::cin >> name;
   std::cout << "Your name is " << name << " thankyou";
}
```

Menerima input string & integer tanpa spasi
std::cin >> variabel

Menerima input string termasuk spasi
std::getline(std::cin,variabel);

Menerima input array char termasuk spasi
std::cin.getline(variabel,sizeof(variabel));

Konstanta = variabel yang nilainya tidak bisa berubah/tetap

```
#include <iostream>

#define ss "102034"
#define a 20
const int b = 50;

int main()
{
   int c;
   c = a + b;
   std::cout << "Your NISN " << ss << std::endl;
   std::cout << a << " + " << b << " = " << c;
}
```

konstanta string

konstanta integer → cara 1

konstanta integer → cara 2

Variabel lokal tipe integer

RUN

Your NISN 102034
20 + 50 = 70

Operator increment

```
#include <iostream>
int main()
{
  int x,y;
  x = 4;
  y = ++x; //prefix

  std::cout << y;
}
```

```
int x,y;
x = 4;
x+=1;
y=x;
```

```
int x,y;
x = 4;
x=x+1;
y=x;
```

RUN
5

```
#include <iostream>
int main()
{
  int x,y;
  x = 4;
  y = x++; //postfix

  std::cout << y;
}
```

RUN
4

Operator decrement

```
#include <iostream>
using namespace std;

int main()
{
    int a = 10;
    int b = 0;

    while (a > 0)
    {
        b = a-- * 2; //postfix
        cout << b << " ";
    }
}
```

RUN (postfix)
20 18 16 14 12 10 8 6 4 2

RUN (prefix)
18 16 14 12 10 8 6 4 2 **0**

```
#include <iostream>
using namespace std;

int main()
{
    int a = 10;
    int b = 0;

    while (a > 0)
    {
        b = --a * 2; //prefix
        cout << b << " ";
    }
}
```

Condition / Kondisi (If else)

```
#include <iostream>
using namespace std;

int main()
{
  int a, b, c;
  cout << "First number : "; cin >> a;
  cout << "Second number : "; cin >> b;
  c = a + b;

  if (c == 10)
    {
      cout << c;
      cout << " good";
    }
  else if (c < 5)
    {
      cout << c;
      cout << " bad";
    }
  else
      cout << c;
}
```

RUN

First number : 5
Second number : 5
10 good

Condition / Kondisi (If else → true/false)

```
#include <iostream>
using namespace std;

int main ()
{

 int year = 2016;

 if (true)
    cout << "Copyright" << endl;
 else
    cout << "-";

 if (year)
    cout << year;
}
```

Selalu cetak pernyataan ini

Cetak variabel tahun

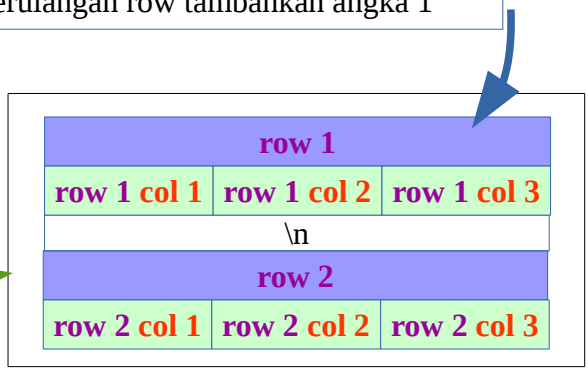
RUN

Copyright
2016

Looping / Perulangan (for .. for)

```
#include <iostream>
using namespace std;

int main()
{
    int row, col;

    for (row = 1; row<=2; row++ )
    {
        for (col = 1; col<=3; col++)
            cout << row << col << " ";

        cout << "\n";
    }
}
```

Mulai dari row=1;
sampai row lebih kecil atau sama dengan 2;
tiap perulangan row tambahkan angka 1



Pindah baris

RUN

11 12 13
21 22 23

Looping (for hanya satu pernyataan / just one statement)

```
#include <iostream>
using namespace std;

int main ()
{
    int a=10;
  for(; a>0;)
  {
      --a;
      cout << a;
  }
}
```

Satu statement/pernyataan

atau bisa
a--

RUN

9876543210

Clear screen

```
#include <iostream>
using namespace std;
int main ()
{
        cout << string(50, '\n');
}
```

Looping / Perulangan (while & do .. while)

```
#include <iostream>
using namespace std;

int main()
{
  int x;
  cout << "Enter number : ";
  cin >> x;

  while (x > 0)
  {
     cout << x << endl;
     --x;
  }
  cout << "Finish";
}
```

atau bisa
x - 1;

atau bisa
x--;

Selama pernyataan benar, kerjakan antara { }

```
#include <iostream>
using namespace std;

int main()
{
  int x;
  cout << "Enter number : ";
  cin >> x;

  do
  {
     cout << x << endl;
     --x;
  }
  while (x > 0);

  cout << "Finish";
}
```

Kerjakan antara { } selama pernyataan benar

RUN

```
Enter number : 3
3
2
1
Finish
```

Operator sizeof = mendapatkan ukuran byte dari berbagai tipe data

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Size of char : " << sizeof(char) << endl;
    cout << "Size of int : " << sizeof(int) << endl;
    cout << "Size of short int : " << sizeof(short int) << endl;
    cout << "Size of long int : " << sizeof(long int) << endl;
    cout << "Size of float : " << sizeof(float) << endl;
    cout << "Size of double : " << sizeof(double) << endl;
    cout << "Size of wchar_t : " << sizeof(wchar_t) << endl;
}
```

RUN

```
Size of char : 1
Size of int : 4
Size of short int : 2
Size of long int : 4
Size of float : 4
Size of double : 8
Size of wchar_t : 4
```

Break

```
#include <iostream>
using namespace std;

int main()
{
  int x, y, z, t, s;
  cout << "Enter high number : ";     cin >> x;
  cout << "Enter low number : ";      cin >> y;
  cout << "Enter stop number : ";     cin >> s;
  if (x < y)
    {
      t = x;
      x = y;
      y = t;
    }
  for (z = x; z >= y; z--)
    {
        if (z == s)
            {
                cout << z << " : founded " << s << endl;
                break;
            }
        else
                cout << z << " : did not found " << s << endl;
    }
  cout << "Finish";
}
```

RUN

```
Enter high number : 5
Enter low number : 1
Enter stop number : 4
5 : did not found 4
4 : founded 4
Finish
```

RUN

```
Enter high number : 2
Enter low number : 8
Enter stop number : 5
8 : did not found 5
7 : did not found 5
6 : did not found 5
5 : founded 5
Finish
```

break = berhenti looping ketika kondisi ditemukan

Continue

```
#include <iostream>
using namespace std;

int main()
{
  int x, y, z, t, s;
  cout << "Enter high number : ";    cin >> x;
  cout << "Enter low number : ";      cin >> y;
  cout << "Enter jumping number : "; cin >> s;
  cout << endl;
  if (x < y)
    {
      t = x;
      x = y;
      y = t;
    }
  for (z = x; z >= y; z--)
    {
        if (z == s)
        {
            cout << "-";
            continue;
        }
            cout << z;
    }

  cout << " Finish";
}
```

RUN

```
Enter high number : 8
Enter low number : 3
Enter jumping number : 6

87-543 Finish
```

Continue
ketika kondisi ditemukan, jangan cetak variabel pada looping ini namun lompat ke akhir looping lalu lanjutkan looping berikutnya

Condition / Kondisi (switch .. case)

```
#include  <iostream>
using namespace std;

int main()
{
    int menu;
    string hh;

    cout << "MENU" << endl;
    cout << "1.Sate" << endl;
    cout << "2.Nasi goreng" << endl;
    cout << "3.Soto ayam" << endl;
    cout << "4.Bakso" << endl;
    cout << endl;

    again:

    cout << "Pilih menu : ";  cin >> menu;

    switch (menu)
    {
        case 1:   cout << "Anda pilih sate";    break;
        case 2:   cout << "Anda pilih nasi goreng";    break;
        case 3:   cout << "Anda pilih soto ayam";     break;
        case 4:   cout << "Anda pilih bakso";       break;
        default:  cout << "Kami tidak tahu apa pilihan Anda";
    }

    cout << endl;
    cout << "Mau pilih lagi [y or n] : "; cin >> hh;
    cout << endl;
    if (hh == "y") goto again;
    cout << "Terima kasih";
}
```

RUN

```
MENU
1.Sate
2.Nasi goreng
3.Soto ayam
4.Bakso

Pilih menu : 2
Anda pilih nasi goreng
Mau pilih lagi [y or n] : y

Pilih menu : 3
Anda pilih soto ayam
Mau pilih lagi [y or n] : n

Terima kasih
```

Goto

```
#include <iostream>
using namespace std;

int main()
{
  int x, y, t;
  cout << "Enter high number : ";    cin >> x;
  cout << "Enter low number : ";   cin >> y;
  cout << endl;

  if (x < y)
    {
     t = x;
     x = y;
     y = t;
    }

   here:

   cout << x;
   x--;

  if (x >= y) goto here;

  cout << " Finish";
}
```

RUN

Enter high number : 1
Enter low number : 9

987654321 Finish

label

Goto = ketika kondisi terpenuhi, program lompat ke posisi label yang telah ditentukan sebelumnya

Rreferences / Referensi

```
#include <iostream>
using namespace std;

int main()
{
    int a;
    int& b = a;

    a = 5;
    cout << b;
}
```

Referensi digunakan untuk merujuk ke suatu variabel yang nilainya sama

& = references

atau bisa
int &b = a;

RUN

5

```
int main()
{
    int a;
    int b;

    a = 5;
    b = a;
    cout << b;
}
```

Variabel tanpa referensi

Tipe data Enumerations

```
#include <iostream>
using namespace std;

int main()
{
    enum difficulty {easy = 10, normal = 100, hard = 1000};
    int options;

    cout << "\nOPTIONS";
    cout << "\n1.Easy" << " , " << "2.Normal" << " , " << "3.Hard";
    cout << endl;
    cout << "\nChoose difficulty : "; cin >> options;

    difficulty diffEasy = easy;
    difficulty diffNormal = normal;
    difficulty diffHard = hard;
    cout << endl;

    switch (options)
    {
        case 1:
            cout  << "Enemy Power : " << diffEasy; break;
        case 2:
            cout << "Enemy Power : " << diffNormal; break;
        case 3:
            cout << "Enemy Power : " << diffHard; break;
        default:
            cout << "Wrong";
    }
}
```

enumeration adalah kumpulan dari konstanta bertipe unsigned int

RUN

```
OPTIONS
1.Easy , 2.Normal , 3.Hard

Choose difficulty : 2

Enemy Power : 100
```

Tipe data Structures / Tipe data Record

```
#include <iostream>
using namespace std;

int main ()
{
  struct student
  {
      string nisn, name;
      int value;
  };

    student wokki;    student wakka;

    wokki.nisn = "100000";     wokki.name = "wokkis";   wokki.value = 80;
    wakka.nisn = "750802060";   wakka.name = "wakka";    wakka.value = 90;

    cout << "NISN : " << wokki.nisn << "\n";
    cout << "Name : " << wokki.name << "\n";
    cout << "Value : " << wokki.value << "\n" << "\n";

    cout << "NISN : " << wakka.nisn << "\n";
    cout << "Name : " << wakka.name << "\n";
    cout << "Value : " << wakka.value;
}
```

Structures adalah kumpulan variabel dengan tipe data yang berbeda

objek student

RUN

```
NISN : 100000
Name : wokkis
Value : 80

NISN : 750802060
Name : wakka
Value : 90
```

Structures sebagai parameter dari fungsi

```
#include <iostream>
using namespace std;

 struct student
 {
     string nisn, name;
     int value;
 };

void printdata(struct student mstudent)
{
    cout << "NISN : " << mstudent.nisn << "\n";
    cout << "Name : " << mstudent.name << "\n";
    cout << "Value : " << mstudent.value;
}

int main ()
{
    struct student wokki;     struct student wakka;

    wokki.nisn = "100000";    wokki.name = "wokkis";   wokki.value = 80;
    wakka.nisn = "750802060";  wakka.name = "wakka";   wakka.value = 90;

    printdata(wokki);
    cout << "\n" << "\n";
    printdata(wakka);
}
```

Tempatkan structure diluar fungsi utama

Buat fungsi tanpa tipe / void dan tempatkan struct **student** sebagai parameter

objek student

panggil fungsi **printdata**

RUN

```
NISN : 100000
Name : wokkis
Value : 80

NISN : 750802060
Name : wakka
Value : 90
```

Exceptions try..catch..throw → using integer

```
#include <iostream>
using namespace std;

int division(int a, int b)
{
    if(b == 0)
    {    throw "Division by zero";    }
    return a/b;
}

int main()
{
    int a,b,c;
    cout << "Enter first number : "; cin >> a;
    cout << "Enter second number : "; cin >> b;

        try
        {
            c=division(a,b);
            cout << c;
        }
        catch (const char* mess)
        {      cerr << mess;     }
}
```

RUN

```
Enter first number : 6
Enter second number : 0
Division by zero
```

Exceptions try..catch..throw → using string

```
#include <iostream>
using namespace std;

string num(string a)
{
    if(a == " ")
    {    throw "Don't enter space";   }
    return a;
}

int main()
{
    string a, c;

    again:
    cout << "Enter sentences : "; getline(cin,a);

        try
        {
            c=num(a);
            cout << c;
        }
        catch (const char* mess)
        {
            cerr << mess;
            cout << "\n";
            goto again;
        }
}
```

Arrays integer

```
#include <iostream>
using namespace std;

int main()
{
    int number[3];

    number[0] = 2;
    number[1] = 3;
    number[2] = 10;

    cout << number[0] << "\n";
    cout << number[1] << "\n";
    cout << number[2] << "\n";
}
```

arrays = sekumpulan variabel dengan tipe sama

arrays tipe integer dengan 3 variabel

mengakses arrays

cetak arrays

RUN

2
3
10

Arrays integer 2

```
#include <iostream>
using namespace std;

int main()
{
    int number[3] = { 2, 3,10 };

    cout << number[0] << "\n";
    cout << number[1] << "\n";
    cout << number[2] << "\n";
}
```

atau bisa
int number[] = { 2, 3,10 };

Memasukkan arrays ke dalam variabel

```
#include <iostream>
using namespace std;

int main()
{
    int number[] = { 2,3,10 };
    int a, b, c;

    a = number[0];
    b = number[1];
    c = number[2];

    cout << a << "\n" << b << "\n" << c;
}
```

masukkan arrays ke dalam variabel

cetak variabel

Arrays string

```
#include <iostream>
using namespace std;

int main()
{
    string text[3];
    unsigned short a;

    text[0] = "cow";
    text[1] = "horse";
    text[2] = "dog";

    for (a=0; a<=2; a++)
        cout << text[a] << "\n";
}
```

arrays string dengan 3 variabel

atau bisa
string text[3] = {"cow", "horse", "dog"};

memasukkan nilai ke arrays

cetak arrays

RUN

Cow
Horse
dog

Arrays char

```
#include <iostream>
using namespace std;

int main()
{
    char text[3] = {'c', 'o' ,'w'};
    unsigned short a;

    for (a=0; a<=2; a++)
        cout << text[a] << "\n";
```

or
char text[4] = {"cow"};

or
char text[] = {"cow"};

or
char text[] = "cow";

or
char text[4] = "cow";

RUN

c
o
w

Arrays char dengan petik dua

```
#include <iostream>
using namespace std;

int main()
{
    char a[8] = "c++ lab";
    cout << a;
}
```

Char : hanya 7 karakter digunakan

1	2	3	4	5	6	7	8	Char a[8]
c	+	+		l	a	b	\0	hanya 7 digunakan
0	1	2	3	4	5	6	\0	mengakses array

Arrays char, input semua karakter termasuk spasi

```
#include <iostream>
using namespace std;

int main()
{
    char name[20];

    cout << "Your name : "; cin.getline(name,sizeof(name));
    cout << name;
}
```

cin.getline(name,sizeof(name));
Terima lebih dari satu kata

sizeof(name) → 20

RUN

Your name : wokki's lab c++
wokki's lab c++

Arrays sebagai paramater dari fungsi

```
#include <iostream>
using namespace std;

void celick(int c[], int d)
{
    int n;

    for (n=0; n<=d; n++)
        cout << c[n] << " ";
}

int main()
{
    int a[] = { 1, 4, 6 };
    int b;

    b = 2;

    celick(a, b);
}
```

Arrays sebagai parameters c[]

Fungi yang tidak mengembalikan nilai menggunakan tipe data void

arrays

Panggil fungsi celick

RUN

1 4 6

Hitung jumlah variabel dalam array integer dan string

```
#include <iostream>
using namespace std;

int main()
{
    int arr[] = {110,5,10,23,22,100,1,23};
    int amount = sizeof(arr)/sizeof(arr[0]);
    cout << amount;
    return 0;
}
```

string arr[] = {"sapi", "kuda", "ayam"};

Mencari jumlah array

RUN
3

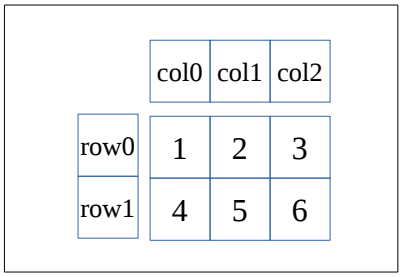
Jumlah string

RUN
8

Jumlah integer

Arrays 2d, rows=baris, columns=kolom

```
#include <iostream>
using namespace std;

int main()
{
    int number[2][3] =
    { // 2 rows(0,1) and 3 columns(0,1,2)
        {1,2,3}, //row 0
        {4,5,6} //row 1
    };

    int row, col;

    for (row = 0; row<=1; row++ )
    {
        for (col = 0; col<=2; col++)
        {
            cout << number[row][col];
        }
        cout << "\n";
    }
}
```



RUN

123
456

Copy Arrays

```
#include <iostream>
using namespace std;

int main()
{
    int a[50]={100,200,300};
    int b[50];
    int x;

    for(x=0; x<3; x++)
    {   b[x]=a[x];   }

    for(x=0; x<3; x++)
    {   cout << b[x] << " "; }
}
```

RUN

100,200,300

Copy Arrays 2d

```
#include <iostream>
using namespace std;

int main()
{
    int a[50][50]= {
                    {100,200,300}, //x0y0,x0y1,x0y2
                    {400,500,600}, //x1y0,x1y1,x1y2
                    {700,800,900} //x2,y0,x2y1,x2y2
                   };
    int b[50][50];
    int x,y;


    for(x=0; x<3; x++)
    {
        for(y=0; y<3; y++)
        {
            b[x][y]=a[x][y];
        }
    }

    fo<3;r(x=0; x x++)
    {
        for(y=0; y<3; y++)
        {
            cout << b[x][y] << " ";
        }
        cout << endl;
    }
}
```

RUN

```
100 200 300
400 500 600
700 800 900
```

Fungsi tidak mengembalikan nilai &tanpa parameter

```
#include <iostream>
using namespace std;

void say()
{
    cout << "Function without parameters";
}

int main()
{
    say();
}
```

Function without parameters using **void** as type data

No return value

Call function here

Fungsi yang mengembalikan nilai dengan parameter nilai / call by value

```
#include <iostream>
using namespace std;

int addition (int x, int y)
{
    int z;
    z = x + y;
    return z;
}

int main()
{
    int zzz, a, b;
    cout << "Enter first number : "; cin >> a;
    cout << "Enter second number : "; cin >> b;

    zzz = addition(a,b);
    cout << "Result : " << a << " + " << b << " = " << zzz;
}
```

Function addition bertipe integer dengan dua parameter x,y bertipe integer

Return **z** adalah nilai yang akan dicetak

Panggil fungsi addition dengan parameter baru a,b

```
RUN

Enter first number : 3
Enter second number : 4
Result : 3 + 4 = 7
```

Fungsi tidak mengembalikan nilai dengan parameter nilai / call by value)

```
#include <iostream>
using namespace std;

void swap(int x ,int y) //fungsi swap
{
    int temporary;
    temporary = x;
    x = y;
    y = temporary;
    cout << "\n";
    cout << "After swap a : " << x << endl;
    cout << "After swap b : " << y << endl;
}

int main()
{
    int a = 100;
    int b = 200;

    cout << "Before swap a : " << a << endl;
    cout << "Before swap b : " << b << endl;

    swap(a,b);
}
```

RUN

```
Before swap a : 100
Before swap b : 200

After swap a : 200
After swap b : 100
```

Parameter pointer / call by pointer)

```
#include <iostream>
using namespace std;

void swap(int *x ,int *y) //fungsi swap
{
    int temporary;
    temporary = *x;
    *x = *y;
    *y = temporary;
}

int main()
{
    int a = 100;
    int b = 200;

    cout << "Before swap a : " << a << endl;
    cout << "Before swap b : " << b << endl;

    swap(&a,&b);
    cout << "\n";
    cout << "After swap a : " << a << endl;
    cout << "After swap b : " << b << endl;
}
```

Parameter reference / call by reference

```
#include <iostream>
using namespace std;

void swap(int &x ,int &y)  //fungsi swap
{
    int temporary;
    temporary = x;
    x = y;
    y = temporary;
}

int main()
{
    int a = 100;
    int b = 200;

    cout << "Before swap a : " << a << endl;
    cout << "Before swap b : " << b << endl;

    swap(a,b);
    cout << "\n";
    cout << "After swap a : " << a << endl;
    cout << "After swap b : " << b << endl;
}
```

Fungsi tidak mengembalikan nilai dengan parameter pointer (cetak array)

```
#include <iostream>
using namespace std;

void print(int *a, int hh)
{
    int ss=0;
    while(ss<hh)
    {
        cout << a[ss] << " ";
        ss++;
    }
}

int main()
{
    int aa[100];
    int x,y;

    cout << "Enter amount of number : "; cin >> x;

    for(y=0; y<x; y++)
    { cin >> aa[y];     }

    print(aa,x); //output
}
```

Tanpa pointer
void print(int a[], int hh)

or

```
for(int ss=0; ss<hh; ss++)
{
    cout << a[ss] << " ";
}
```

RUN

```
Enter amount of number : 5
1 2 3 4 5
1 2 3 4 5
```

Recursivity

```
#include <iostream>
using namespace std;

double factorial(int fac)
{
    if (fac > 1)
        return(fac * factorial(fac - 1));
    else
        return(1);
}

int main()
{
    int fac;

    cout << "Enter number : "; cin >> fac;
    cout << "Factorial : " << factorial(fac);
}
```

Recursivityadalah fungsi yang memanggil dirinya secara terus menerus hingga kondisi tidak terpenuhi

RUN

Enter number : 5
Factorial : 120

return (5 * factorial(5-1))
return (20 * factorial(4-1))
return (60 * factorial(3-1))
return (120 * factorial(2-1))
return (**120**)

Menempatkan fungsi dibawah fungsi utama / prototyping function

```
#include <iostream>
using namespace std;

double factorial(int fac);

int main()
{
    int fac;

    cout << "Enter number : "; cin >> fac;
    cout << "Factorial : " << factorial(fac);
}

double factorial(int fac)
{
    if (fac > 1)
        return(fac * factorial(fac - 1));
    else
        return(1);
}
```

Deklarasikan fungsi disini tanpa peryataan apapun

Fungsi utama (Main)

Fungsi factorial berada dibawah fungsi utama (main)

Fungsi yang saling memanggil satu sama lain

```
#include <iostream>
using namespace std;

void odd(int a);
void even(int a);

int main()
{
    int a;
    string hh;

    again:

    cout << "\nEntering number : "; cin >> a;
    odd(a);
    cout << "\n";

    cout << "\nChoose again [y/n] : "; cin >> hh;

    if (hh=="y") goto again;

    cout << "Ok";
}

void odd(int a)
{
    if (a%2 > 0)
    cout << "Odd";
    else even(a);
}

void even(int a)
{
    if (a%2 == 0)
    cout << "Even";
    else odd(a);
}
```

Deklarasikan 2 nama fungsi disini tanpa pernyataan apapun

RUN

```
Entering number : 5
Odd

Choose again [y/n] : y

Entering number : 6
Even

Choose again [y/n] : n
Ok
```

Panggil satu fungsi (odd)

Kedua fungsi saling memanggil satu sama lain

Fungsi strlen = menghitung panjang karakter

```
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
    char a[100];
    int b;

    cout << "Enter name : "; cin.getline(a,sizeof(a));
    b = strlen(a);
    cout << "Length : " << b;
}
```

Fungsi **strlen** butuh pustaka / library **<cstring>**

RUN

Enter name : sepeda tua
Length : 10

Fungsi length = menghitung panjang string

```
#include <iostream>
using namespace std;

int main()
{
    string name;
    int a;

    cout << "Enter sentences : "; getline(cin,name);
    a = name.length();
    cout << a;
}
```

RUN

Enter sentences : sepeda tua
10

Fungsi strcpy = kopi karakter

```
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
    char a[101]; char b[101];
    string c;

   cout << "Enter name : ";    cin.getline(a,sizeof(a));
   cout << "\n";
   c = strcpy(b,a);
   cout << "copy : " << c;
}
```

Fungsi **strcpy** butuh pustaka **<cstring>**

strcpy = kopi karakter a ke b dan masukkan dalam variabel tipe sring

RUN

Enter name : wokki

copy : wokki

Fungsi atoi / atol & atof = konversi char ke int / char ke long & char ke float

```
#include <iostream>
#include <cstdlib>
using namespace std;

int main()
{
   char a[11];
   int b;

   cout << "Enter number : "; cin >> a;
   b = atoi(a) * 2;
   cout << b;
}
```

Fungsi **atoi / atol** , **atof** butuh pustaka **<cstdlib>**

```
b = atol(a) * 2;
```

```
b = atof(a) * 2;
```

RUN atoi
Enter number : 34
68

RUN atof
Enter number : 12.5
25

Fungsi strcmp = bandingkan 2 variabel tipe char apakah sama atau tidak

```
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
   char a[50]; //0 sampai 49
   char b[50];
   int c;

   cout << "Enter character : "; cin.getline(a,sizeof(a));
   cout << "Enter character : "; cin.getline(b,sizeof(b));
   c = strcmp(a, b);

   if (c == 0)
     cout << "\nCharacter is same";
   else
     cout << "\nCharacter is different";
}
```

strcmp butuh pustaka **<cstring>**

Jika tidak diketemukan perbedaan maka

RUN

Enter character : wokki's lab
Enter character : wokki's lab

Character same

Fungsi strcat = menggabung dua array tipe character

```
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
    char aa[9] = "sepeda";
    char bb[4] = "tua";
    cout << strcat(aa, bb);
}
```

Function **strcat** butuh pustaka **<cstring>**

or
char **aa**[7] = "sepeda";
char **bb**[9] = "tua";

RUN

sepedatua

array of char 1

array of char 2

Ukuran max array 9 karakter

Fungsi strstr = mencari array karakter dalam array karakter

```
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
    char aa[11] = "sepeda tua";
    char bb[4] = "tua";

    if (strstr(aa,bb))
        cout << "Character " << bb << " founded";
    else
        cout << "Character " << bb << " not founded";
}
```

Fungsi **strstr** butuh pustaka**<cstring>**

RUN

Character tua founded

Cari array bb pada array aa

If ditemukan

Fungsi toupper, tolower = mengubah karakter ke huruf besar atau kecil

```
#include <iostream>
using namespace std;
int main()
{
    char nama[11]= "sepeda tua";

    nama[0] =toupper(nama[0]);
    cout << nama;
}
```

or

string nama= "sepeda tua";

RUN

Sepeda tua

tolower

Fungsi isalpha = cek jika variabel angka atau karakter

```
#include <iostream>
using namespace std;

int main()
{
    char b = 'd' ;

    if (isalpha(b))
        cout << "it's not number";
    else
        cout << "it's number";
}
```

isalnum

RUN
it's not number

```
#include <iostream>
using namespace std;

int main()
{
    int a;
    a = 6;

    if (isalpha(a))
        cout << "it'is not number";
    else
        cout << "it's number";
}
```

RUN
it's number

Fungsi isspace = cek jika variabel berupa spasi

```
#include <iostream>
using namespace std;

int main()
{
    int a;
    a = ' ';
    if (isspace(a)) cout << "it's space";  else    cout << a+1;
}
```

RUN

it's space

Fungsi islower, isupper = cek jika karakter berupa huruf kecil atau huruf besar

```
#include <iostream>
using namespace std;

int main()
{
    char a = 'd' ;

    if (islower(a))
    {
        a = toupper(a);
        cout <<  a;
    }
    else
        cout << a;
}
```

isupper

RUN

D

Fungsi setw = menentukan lebar data yang dicetak

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
   string  name;

   cout << "Enter sentences : ";
   getline(cin,name);
   cout << "Your text is " << setw(20) << name;
}
```

Fungsi **setw** butuh pustaka **<iomanip>**

std::setw

RUN

```
Enter sentences : sepeda tua
Your text is                sepeda tua
```

10 10

Fungsi setfill = tambahkan karakter ke area kosong akibat fungsi setw

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
   string  name;

   cout << "Enter sentences : ";
   getline(cin,name);
   cout << "Your text is " << setw(20) << setfill('.') << name;
}
```

Fungsi **setfill** butuh pustaka **<iomanip>**

RUN

```
Enter sentences : sepeda tua
Your text is .....sepeda tua
```

Fungsi clear = membersihkan string

```
#include <iostream>
using namespace std;

int main()
{
    string name, clearname;
    int a;

    cout << "Enter sentences : "; getline(cin,name);
    a = name.length();
    cout << a;
    cout << endl;
    cout << "Clear string [y/n] : "; getline(cin,clearname);

    if (clearname == "y" or clearname == "Y")
        name.clear();
    else
        cout << name;
}
```

```
clearname[0] = toupper(clearname[0]);
if (clearname == "Y")
    name.clear();
```

RUN

```
Enter sentences : sepeda
6
Clear string [y/n] : n
sepeda
```

Fungsi find = mencari posisi karakter pada sebuah variabel tipe string

```
#include <iostream>
using namespace std;

int main()
{
    string a;
    char b;

    cout << "Enter sentences : "; getline(cin,a);
    cout << "Enter character to find : "; cin >> b;
    string::size_type position = a.find(b);

    if(position!=string::npos)
    {
        cout << position;
    }
    else
        cout << "Character not found";
}
```

RUN

Enter sentences : sepeda tua
Enter character to find : t
7

s=0
e=1
p=2
e=3
d=4
a=5
spasi = 6
t=7
u=8
a=9

Cari karakter b pada kalimat a
lalu simpan pada variabel position
variabel posisi bertipe string::size_type

Jika variabel posisi tidak sama dengan string::npos
Cetak position
Jika variabel posisi sama dengan string::npos
Cetak character not found

Fungsi time_t = date time

```
#include <iostream>
//#include <ctime>
using namespace std;

int main()
{
    time_t mytime = time(0);
    char* mytimeis = ctime(&mytime);

    cout << mytimeis;
}
```

RUN

Mon Feb 22 13:30:09 2016

fungsi time_t

Konversi **mytime** ke char

Fungsi floor, ceil, pow, sqrt, abs = fungsi matematika/math

```
#include <iostream>
#include <cmath>
using namespace std;

int main()
{
    float a = 5.8;
    float b = 5.8;
    int c = 2;
    int d = 4;
    int e = -3;

    cout << "Sebelum kena fungsi floor " << a << "\n";
    cout << "Sebelum kena fungsi ceil " << b << "\n";
    cout << "Sebelum kena fungsi pow " << c << "\n";
    cout << "Sebelum kena fungsi sqrt " << d << "\n";
    cout << "Sebelum kena fungsi abs " << e << "\n";

    a = floor(a);
    b = ceil(b);
    c = pow(c,4); // c*c*c*c
    d = sqrt(d); //square root
    e = abs(e);

    cout << "\n";
    cout << "Setelah kena fungsi floor " << a << "\n";
    cout << "Setelah kena fungsi ceil " << b << "\n";
    cout << "Setelah kena fungsi pow " << c << "\n";
    cout << "Setelah kena fungsi sqrt " << d << "\n";
    cout << "Setelah kena fungsi abs " << e << "\n";
}
```

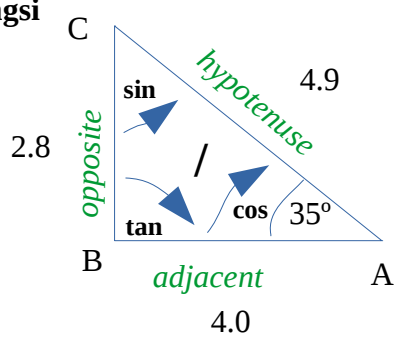
Butuh pustaka <**cmath**>

Cat : di windows
Jk tidak bisa tipe int maka
gunakan float, double, long double

RUN

```
Sebelum kena fungsi floor 5.8
Sebelum kena fungsi ceil 5.8
Sebelum kena fungsi pow 2
Sebelum kena fungsi sqrt 4
Sebelum kena fungsi abs -3

Setelah kena fungsi floor 5
Setelah kena fungsi ceil 6
Setelah kena fungsi pow 16
Setelah kena fungsi sqrt 2
Setelah kena fungsi abs 3
```

Cari sin, cos, tan tanpa fungsi




```
#include <iostream>
using namespace std;

int main()
{
    float hyp, adj, opp; //hypotenuse,  adjacent,  opposite
    float sin,cos,tan;

    hyp = 4.9;   adj = 4.0;    opp = 2.8;

    sin = opp/hyp;
    cos = adj/hyp;
    tan = opp/adj;

    cout << "sin : " << sin << "\n";
    cout << "cos : " << cos << "\n";
    cout << "tan : " << tan << "\n";
}
```

RUN

```
sin : 0.571429
cos : 0.816326
tan : 0.7
```

Fungsi sin, cos, tan

```
#include <iostream>
#include <cmath>
using namespace std;

#define PI 3.14159265
int main()
{
    float aa,bb,cc;
    int num;

    num = 35;

    aa = sin(num*PI/180);
    bb = cos(num*PI/180);
    cc = tan(num*PI/180);

    cout << "sin " << num << " = " << aa << "\n";
    cout << "cos " << num << " = " << bb << "\n";
    cout << "tan " << num << " = " << cc << "\n";
}
```

Butuh pustaka <**cmath**>

RUN

```
sin 35 = 0.573576
cos 35 = 0.819152
tan 35 = 0.700208
```

Konversi string ke integer (stringstream)

```
#include <iostream>
#include <sstream>
int main()
{
    std::string a = "1234";
    int b = 1;
    int c, d;
    std::stringstream(a) >> d;

    c = d+b;
    std::cout << c;
}
```

stringstream butuh pustaka <**sstream**>

kirim string a ke int d

RUN
1235

konversi string/teks ke integer/angka
(istringstream)

```
#include <iostream>
#include <sstream> //istringstream
using namespace std;

int convertstrtoint(string aa)
{
    int bb;

    istringstream wek(aa); //stream wek digunakan untuk konversi string aa
    if(!(wek >> bb)) //kirim wek ke int bb
        bb = 0; //jika gagal hasil = 0 (contoh string ='12345' bukan 'sepeda')
    return(bb); //kembalikan hasil ke main function
}

int main()
{
    string aa;
    int result;

    cout << "Enter string 0-9 : "; getline(cin,aa); //input string number 0-9

    result = convertstrtoint(aa); //konversi string ke integer
    cout << result;
}
```

RUN

Enter string 0-9 : 1234567890
1234567890

konversi integer/angka ke string/teks, length of integer
(ostringstream)

```
#include <iostream>
#include <sstream> //ostringstream
using namespace std;

string convertinttostr(int aa)
{
    string bb;

    ostringstream wek; //stream wek digunakan untuk konversi
    wek << aa; //masukkan int aa ke stream wek
    bb = wek.str(); //string bb sebagai penampung stream wek
    return(bb); //kirim string bb ke main function
}

int main()
{
    int aa;
    string result;
    int cc;

    cout << "Enter number 0-9 : "; cin >> aa; //input integer

    result = convertinttostr(aa); //konversi  integer ke string
    cc=result.length(); //panjang string
    cout << "Amount : " << cc << " number";
}
```

RUN

```
Enter number 0-9 : 1234567890
Amount : 10 number
```

Operator bitwise xor(^)

```
#include <iostream>
using namespace std;

int main()
{
    cout << (12 ^ 23);
}
```

RUN

27

1 = benar	0 = salah		
1	1	=	0
1	0	=	1
0	1	=	1
0	0	=	0

Cara :

Desimal 12

bagi = /, sisa = %

12 bagi 2 = **6** sisa **0**
6 bagi 2 = **3** sisa **0**
3 bagi 2 = **1** sisa **1**
1 bagi 2 = 0 sisa **1**
jadi biner ambil angka dari bawah ke atas = **1100**

Desimal 23

bagi = /, sisa = %

23 bagi 2 = **11** sisa **1**
11 bagi 2 = **5** sisa **1**
5 bagi 2 = **2** sisa **1**
2 bagi 2 = **1** sisa **0**
1 bagi 2 = 0 sisa **1**

jadi biner ambil angka dari bawah ke atas = **10111**

1 = benar | 0 = salah

tambahkan agar biner 12 sama digitnya dengan biner 23

0	1	1	0	0
1	0	1	1	1

12 biner

23 biner

1	1	0	1	1
2^4	2^3	2^2	2^1	2^0
16	8		2	1

jadi 16 + 8 + 2 + 1 = **27**

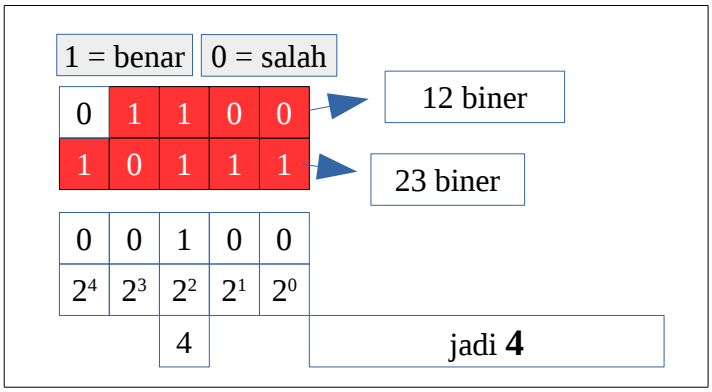
Operator bitwise and (&)

```
#include <iostream>
using namespace std;

int main()
{
    cout << (12 & 23);
}
```

RUN

4

1 = benar	0 = salah		
1	1	=	1
1	0	=	0
0	1	=	0
0	0	=	0



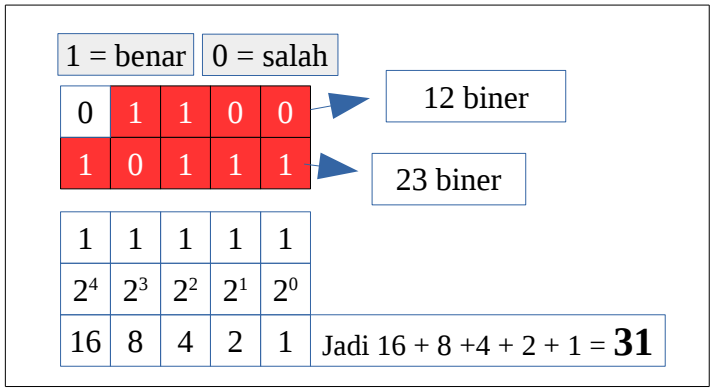
Operator bitwise or (|)

```
#include <iostream>
using namespace std;

int main()
{
    cout << (12 | 23);
}
```

RUN

31

1 = benar	0 = salah		
1	1	=	1
1	0	=	1
0	1	=	1
0	0	=	0



Menggeser bit ke kiri (<<)

```
#include <iostream>
using namespace std;

int main()
{
    cout << (12 << 2);
}
```

```
RUN

48
```

Tinggal tambahkan 0 dibelakang bilangan biner yang ada

Desimal 1 =	1	biner	<< 7	=	1	0	0	0	0	0	0	0
Cara cepat 1x2x2x2x2x2x2x2 = 128				=	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
				=	128							
Desimal 12 =	1100	biner	<< 2	=	1	1	0	0	0	0		
Cara cepat 12x2x2 = 48				=	2^5	2^4	23	22	2^1	2^0		
				=	32	16						
				=	32 + 16 =					48		

Menggeser bit ke kanan (>>)

```
#include <iostream>
using namespace std;

int main()
{
    cout << (12 >> 2);
}
```

```
RUN

3
```

Tambahkan 0 didepan bilangan biner

Desimal 1 =	1	biner	>> 7	=	0				
Cara cepat 1 / 2 = 0				=	2^0				
Desimal 12 =	1100	biner	>> 2	=	1	1	0	0	awal
Cara cepat 12 / 2 = 6		bagi 2 ke-1		=	0	1	1	0	geser 1x
6 / 2 = 3		bagi 2 ke-2		=	0	0	1	1	geser 2x
Hasil = 3				=	2^3	2^2	2^1	2^0	
				=			2	1	
				=	2 + 1 =			3	

Pointer

Pointer adalah variabel yang berisi alamat variabel lain pada memori komputer. Variabel pointer akan menunjukkan alamat memori dari suatu variabel. Fungsi pointer ialah untuk menyimpan alamat memori dari sebuah variabel atau alamat memori sebuah fungsi.

Pointer → variabel tipe integer

```
#include <iostream>
using namespace std;

int main()
{
    int a;
    int *b; //int*b  or int* b
    b = &a;

    a=1;
    cout << a << "\n" << b;
}
```

```
RUN
1
0xbfbbaa58
```

Isi variabel a
Alamat memori dari variabel a
(alamat memori kemungkinan berubah-ubah tiap kali run, tergantung sistem)

Angka **1** tersimpan sementara pada alamat **0xbfbbaa58** di memori komputer

a= variabel biasa/umum bertipe integer
b= variabel bertipe pointer (isinya menunjuk pada alamat di memori komputer dari variabel a)

variabel tipe float

```
#include <iostream>
using namespace std;

int main()
{
    float a;
    float *b;
    b = &a;

    a=1.8;

    cout << a << "\n" << b;
}
```

Pointer → variabel tipe string

```
#include <iostream>
using namespace std;

int main()
{
    string a;
    string *b;
    b = &a;

    a="sepeda tua";
    cout << a << "\n" << b;
}
```

```
RUN

sepeda tua
0xbfebfdd8
```

Pointer → variabel tipe char

```
#include <iostream>
using namespace std;

int main()
{
    char a;
    char *b;
    b = &a;

    a='a';

    cout << a << "\n" << b;
}
```

Array pointer → variabel tipe integer

```
#include <iostream>
using namespace std;

int main()
{
    int x;

    int a[20];
    int *b[20];
    for(x=0;x<3; x++)
        b[x] = &a[x];

    a[0]=100;
    a[1]=230;
    a[2]=400;

    for(x=0;x<3; x++)
    cout << a[x] << ", " << b[x] << "\n";
}
```

RUN

```
100, 0xbfbad0c0
230, 0xbfbad0c4
400, 0xbfbad0c8
```

```
#include <iostream>
using namespace std;

int main()
{
    int x;

    int a[20];
    int *b;
    b = a;

    a[0]=100;
    a[1]=230;
    a[2]=400;

    for(x=0;x<3; x++)
    cout << a[x] << ", " << &b[x] << "\n";
}
```

atau langsung
b

Array pointer → variabel tipe string

```
#include <iostream>
using namespace std;

int main()
{
    int x;

    string a[20];
    string *b[20];
    for(x=0;x<3; x++)
        b[x] = &a[x];

    a[0]="sepeda tua";
    a[1]="mobil baru";
    a[2]="motor rusak";

    for(x=0;x<3; x++)
    cout << a[x] << ", " << b[x] << "\n";
}
```

RUN

```
sepeda tua, 0xbf9d8250
mobil baru, 0xbf9d8254
motor rusak, 0xbf9d8258
```

```
#include <iostream>
using namespace std;

int main()
{
    int x;

    string a[20];
    string *b;
    b = a;

    a[0]="sepeda tua";
    a[1]="mobil baru";
    a[2]="motor rusak";

    for(x=0;x<3; x++)
    cout << a[x] << ", " << &b[x] << "\n";
}
```

atau langsung b

Array pointer → variabel tipe char

```
#include <iostream>
using namespace std;

int main()
{
    int x;

    char a[20]; // or char a[20] = "tua";
    char *b[20];
    for(x=0;x<3; x++)
        b[x] = &a[x];

    a[0]='t';
    a[1]='u';
    a[2]='a';

    for(x=0;x<3; x++)
    cout << a[x] << ", " << b[x] << "\n";
}
```

```
t, tua◆-◆F◆◆◆◆
u, ua◆-◆F◆◆◆◆
a, a◆-◆F◆◆◆◆
```

Hapus ini jika array langsung diisi diatas

```
#include <iostream>
using namespace std;

int main()
{
    int x;

    char a[20]; // or char a[20] = "tua";
    char *b;
    b = a;

    a[0]='t';
    a[1]='u';
    a[2]='a';

    for(x=0;x<3; x++)
    cout << a[x] << ", " << &b[x] << "\n";
}
```

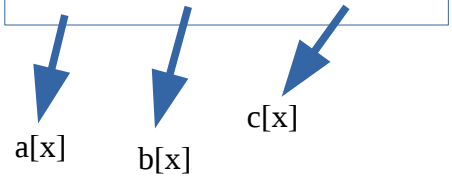
Hapus ini jika array langsung diisi diatas

Pointer pada pointer

```
#include <iostream>
using namespace std;

int main()
{
    int x;

    int a[20];
    int *b[20];
    int **c[20];
    for(x=0;x<3; x++)
        b[x] = &a[x];
    for(x=0;x<3; x++)
        c[x] = &b[x];

    a[0]=100;
    a[1]=230;
    a[2]=400;

    for(x=0;x<3; x++)
    cout << a[x] << ", " << b[x] << ", " << c[x] << "\n";
}
```

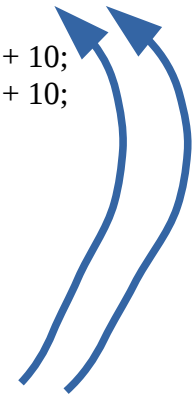
RUN

100, 0xbfe90b30, 0xbfe90b80
230, 0xbfe90b34, 0xbfe90b84
400, 0xbfe90b38, 0xbfe90b88



Pointer pada fungsi

```
#include <iostream>
using namespace std;

void add(int *x, int *y)
{
    *x = *x + 10;
    *y = *y + 10;
}

int main()
{
    int a,b;
    a=10;
    b=10;

    add(&a, &b);

    cout << "a : " << a << "\n";
    cout << "b : " << b;
}
```



Fungsi lain pointer :
Fungsi void biasanya tidak mengembalikan nilai sehingga ketika nama fungsi dipanggil dari fungsi utama variabel akan di cetak (cout) langsung pada fungsi void.

Dengan menggunakan pointer, cetak variabel bisa langsung dari fungsi utama.

Pointer pada kelas

```
#include <iostream>
using namespace std;

class box
{
    public:
        double length;
        double breadth;
        double height;

        double getVolume() //fungsi anggota
        {
            return length * breadth * height;
        }

        box()
        {   cout << "Thanks \n";}

        ~box()
        {   cout << "Goodbye \n";  }
};


int main()
{
    box box1;
    double vol;

    box *ptrbox; //pointer
    ptrbox = &box1; //isi pointer dengan referensi (&) ke box1

    cout << "Box 1" << endl;
    cout << "Enter length : ";  cin >> box1.length;
    cout << "Enter breadth : ";  cin >> box1.breadth;
    cout << "Enter height : ";  cin >> box1.height;


    cout << "Volume" << endl;
    vol = ptrbox->getVolume();
    cout << "Volume box 1 : " << vol << endl;
}
```

kelas box dengan objek box1

panggil fungsi dengan variabel pointer

RUN

```
Thanks
Box 1
Enter length : 5
Enter breadth : 4
Enter height : 3
Volume
Volume box 1 : 60
Goodbye
```

Memori dinamis (Dynamic Memory)

Sebelum sebuah program di run, semua kebutuhan memori ditentukan lewat definisi berbagai variabel pada aplikasi. Namun adakalanya kebutuhan memori ditentukan ketika program telah di run. Contoh ketika kebutuhan memori tergantung pada inputan user. Pada kondisi ini program butuh alokasi memori secara dinamis. C++ menggunakan operator new dan delete untuk membuat dan menghapus alokasi memori dinamis ini.

Memori pada program C++ yang kita buat dibagi ke dalam dua bagian :

- **Stack** : semua variabel yang dideklarasikan didalam fungsi akan mengambil memori dari stack.
- **Heap** : ini memori yang tidak digunakan oleh program dan dapat digunakan untuk alokasi memori secara dinamis ketika program jalan.

Dua cara memori dialokasikan untuk menyimpan data :

1. Alokasi memori statis (ketika compile time)
   - memori untuk nama variabel dialokasikan oleh compiler.
   - Ukuran yang tepat dan tipe penyimpanan harus diketahui pada waktu compile.
   - Untuk deklarasi array standar, ukuran harus tetap.
2. Alokasi memori dinamis (ketika program telah run)
   - memori dialokasikan pada saat program telah run.
   - Area memori yang dialokasikan diambil pada segment yang disebut heap/free store.
   - Ukuran yang tepat tidak harus diketahui oleh kompiler pada mulanya.
   - Penggunaan Pointer adalah penting.

Kita bisa mengalokasikan penyimpanan secara dinamis ketika program telah run, namu kita tidak dapat membuat variabel setelah program run. Oleh karena itu alokasi dinamis memerlukan dua langkah.

1. Buat area dinamis
2. simpan alamatnya pada pointer (sehingga area bisa diakses).

Untuk dapat mengalokasikan memori kita menggunakan operator new.
Setelah memori dialokasikan/disediakan maka sehabis digunakan (pada variabel) memori perlu dibebaskan/dealokasi agar beban komputer berkurang. Pada alokasi memori statis semua variabel dibebaskan otomatis sehabis digunakan oleh kompiler. Namun pada memori dinamis adalah tugas programmer untuk membebaskan memori akibat penggunaan operator new. Untuk membebaskan memori gunakan operator delete.

Dynamic memory allocation → for integer

```
#include <iostream>
using namespace std;
int main()
{
    int* a; //pointer initialized
    a = new int; //request memory for the variable

    *a = 10000; //store value at allocated address
    cout << "Value of a :" << *a;
    delete a; //free up the memory
}
```

New operator untuk mengalokasikan memori secara dinamis

RUN

Value of a :10000

Delete operator untuk membebaskan memori

Dynamic memory allocation → for string

```
#include <iostream>
using namespace std;
int main()
{
    string* a; //pointer initialized
    a = new string; //request memory for the variable

    *a = "sepeda tua"; //store value at allocated address
    cout << "Value of a :" << *a;
    delete a; //free up the memory
}
```

New operator untuk mengalokasikan memori secara dinamis

RUN

Value of a :sepeda tua

Delete operator untuk membebaskan memori

Dynamic memory allocation → for char

```
#include <iostream>
using namespace std;
int main()
{
    char* a; //pointer initialized
    a = new char; //request memory for the variable

    *a = 'w'; //store value at allocated address
    cout << "Value of a : " << *a;
    delete []a; //free up the memory  / bebaskan memori
}
```

RUN

Value of a : w

Dynamic memory allocation → for array integer

```
#include <iostream>
using namespace std;
int main()
{
    int* a; //pointer initialized / deklarasi variabel pointer
    a = new int[20]; //request memory for the variable array integer  / minta memori

    int mm;
    for(mm=0; mm<=3; mm++)
    {
        cout << "Enter Number " << mm+1 << " : ";
        cin >> a[mm];
    }

    for(mm=0; mm<=3; mm++)
    {
        cout << "Value of " << mm+1 << " : " << a[mm];
        cout << endl;
    }
    delete []a; //free up the memory  / bebaskan memori
}
```

atau
delete a;

RUN

```
Enter Number 1 : 45
Enter Number 2 : 34
Enter Number 3 : 67
Enter Number 4 : 34
Value of 1 : 45
Value of 2 : 34
Value of 3 : 67
Value of 4 : 34
```

Dynamic memory allocation → for array string

```
#include <iostream>
using namespace std;
int main()
{
    string* a; //pointer initialized / deklarasi variabel pointer
    a = new string[20]; //request memory for the variable array integer  / minta memori

    int mm;
    for(mm=0; mm<=3; mm++)
    {
        cout << "Enter Sentences " << mm+1 << " : ";
        getline(cin, a[mm]);
    }

    for(mm=0; mm<=3; mm++)
    {
        cout << "Value of " << mm+1 << " : " << a[mm];
        cout << endl;
    }
    delete []a; //free up the memory  / bebaskan memori
}
```

atau
delete a;

RUN

```
Enter Sentences 1 : sepeda tua
Enter Sentences 2 : mobil tua nenek
Enter Sentences 3 : motor tua kakek
Enter Sentences 4 : sepatu roda adik budi
Value of 1 : sepeda tua
Value of 2 : mobil tua nenek
Value of 3 : motor tua kakek
Value of 4 : sepatu roda adik budi
```

Dynamic memory allocation→ for array character

```
#include <iostream>
using namespace std;
int main()
{
    char* a; //pointer initialized / deklarasi variabel pointer
    a = new char[20]; //request memory for the variable array integer / minta memori

    int mm;
    for(mm=0; mm<=5; mm++)
    {
        cout << "Enter Character " << mm+1 << " : ";
        cin >> a[mm];
    }

    cout << "Value of 1 - 6" << " : ";
    for(mm=0; mm<=5; mm++)
    {
        cout << a[mm];
    }
    delete []a; //free up the memory / bebaskan memori
}
```

Hanya bisa satu kata

RUN

```
Enter Character 1 : s
Enter Character 2 : e
Enter Character 3 : p
Enter Character 4 : e
Enter Character 5 : d
Enter Character 6 : a
Value of 1 - 6 : sepeda
```

Dynamic memory allocation→ for array character (langsung tanpa for)

```
#include <iostream>
using namespace std;
int main()
{
    char* a;  //pointer initialized / deklarasi variabel pointer
    a = new char[20]; //request memory for the variable array integer / minta memori


    cout << "Enter Character : ";
    cin >> a;

    cout << "Value : " << a;

    delete []a; //free up the memory  / bebaskan memori
}
```

Hanya bisa satu kata

Menentukan banyak variabel integer array ketika program telah run

```
#include <iostream>
using namespace std;
int main()
{

    int size;
    cout << "How many variabel integer : "; cin >> size;
    int *list = new int[size]; //request memory for the variable array integer

    int a;

    cout << endl;

    for(a=0; a<size; a++) //input data array ke array list
    {   cout << "Number " << a+1 << " : "; cin >> list[a]; }

    cout << endl;

    for(a=0; a<size; a++) //output data array ke array temp
    {   cout << list[a] << " ";   }

    delete []list; //free up the memory
}
```

Ketika program run diminta masukkan dulu banyak variabel array

Banyak variabel array integer list tidak diketahui ketika proses coding dan compile banyak variabel = size

RUN

```
How many variabel integer : 3

Number 1 : 1000
Number 2 : 50000
Number 3 : 250000

1000 50000 250000
```

Kelas

Perbedaan utama antara bahasa prosedural (c) dan bahasa berorientasi objek (c++) terletak pada adanya fitur pembuatan kelas.

```
#include <iostream>
using namespace std;

class box
{
    public:
        double length;
        double breadth;
        double height;
};

int main()
{
    box box1;
    box box2;
    double vol;

    cout << "Box 1" << endl;
    cout << "Enter length : ";  cin >> box1.length;
    cout << "Enter breadth : ";  cin >> box1.breadth;
    cout << "Enter height : ";  cin >> box1.height;
    cout << "Box 2" << endl;
    cout << "Enter length : ";  cin >> box2.length;
    cout << "Enter breadth : ";  cin >> box2.breadth;
    cout << "Enter height : ";  cin >> box2.height;

    cout << "Volume" << endl;
    vol = box1.length * box1.breadth * box1.height;
    cout << "Volume box 1 : " << vol << endl;

    vol = box2.length * box2.breadth * box2.height;
    cout << "Volume box 2 : " << vol;
}
```

buat kelas / class **Box** dengan 3 variabel publik

access	public	protected	private
Same class	yes	yes	yes
Derived classes	yes	yes	no
Outside classes	yes	no	no

RUN

```
Box 1
Enter length : 3
Enter breadth : 2
Enter height : 4
Box 2
Enter length : 2
Enter breadth : 1
Enter height : 3
Volume
Volume box 1 : 24
Volume box 2 : 6
```

Fungsi dalam kelas / class member function

```
#include <iostream>
using namespace std;

class box
{
    public:
        double length;
        double breadth;
        double height;

        double getVolume(void) //member function
        {
            return length * breadth * height;
        }
};

int main()
{
    box box1;
    box box2;
    double vol;

    cout << "Box 1" << endl;
    cout << "Enter length : ";  cin >> box1.length;
    cout << "Enter breadth : ";  cin >> box1.breadth;
    cout << "Enter height : ";  cin >> box1.height;
    cout << "Box 2" << endl;
    cout << "Enter length : ";  cin >> box2.length;
    cout << "Enter breadth : ";  cin >> box2.breadth;
    cout << "Enter height : ";  cin >> box2.height;

    cout << "Volume" << endl;
    vol = box1.getVolume();
    cout << "Volume box 1 : " << vol << endl;

    vol = box2.getVolume();
    cout << "Volume box 2 : " << vol;
}
```

atau
double getVolume()

Class member function

RUN

```
Box 1
Enter length : 3
Enter breadth : 2
Enter height : 4
Box 2
Enter length : 2
Enter breadth : 3
Enter height : 4
Volume
Volume box 1 : 24
Volume box 2 : 24
```

Panggil fungsi

:: → scope resolution operator

atau

```
class box
{
    public:
        double length;
        double breadth;
        double height;

        double getVolume(); //member function declaration
};

double box::getVolume() //member function definition
        {
            return length * breadth * height;
        }
```

konstruktor dan destruktor

Konstruktor kelas adalah fungsi dalam kelas yang spesial dimana dieksekusi kapanpun kita menciptakan objek baru dari kelas tersebut.
Konstruktor mempunyai nama sama seperti nama kelas dan tidak mengembalikan nilai sama sekali bahkan nilai kosong (void)
Destruktor (~) sangat berguna untuk melepaskan sumber daya sebelum keluar aplikasi
Ketika menutup program, agar memori yang digunakan program bisa dibebaskan.

```
#include <iostream
using namespace std;

class box
{
    public:
        double length;
        double breadth;
        double height;

        double getVolume(void) //member function
        {
            return length * breadth * height;
        }

        box()
        {   cout << "Thanks \n"; }

        ~box()
        {   cout << "Goodbye \n";  }
};

int main()
{
    box box1;
    double vol;

    cout << "Box 1" << endl;
    cout << "Enter length : ";  cin >> box1.length;
    cout << "Enter breadth : ";  cin >> box1.breadth;
    cout << "Enter height : ";  cin >> box1.height;

    cout << "Volume" << endl;
    vol = box1.getVolume();
    cout << "Volume box 1 : " << vol;
}
```

atau
double getVolume()

Class member function

konstruktor & destruktor

RUN

```
Thanks
Box 1
Enter length : 3
Enter breadth : 2
Enter height : 4
Volume
Volume box 1 : 24
```

atau

```
class box
{
    public:
        double length;
        double breadth;
        double height;

        double getVolume(); //deklarasi fungsi anggota
        { return length * breadth * height; }
        box();
        ~box();
};

box::box() //konstruktor dan destruktor
    {  cout << "Thanks \n"; }
box::~box()
    {  cout << "Goodbye \n";  }
```

Kelas dasar dan kelas turunan atau disebut pewarisan (inheritance)

```
#include <iostream>
using namespace std;

//base class / kelas dasar
class Shape
{
    public:
        void setWidth(int w)
        {   width = w; }
        void setHeight(int h)
        {   height = h; }
    protected:
        int width;
        int height;
};

//derived class / kelas turunan
class Rectangle: public Shape
{
    public:
        int getArea()
        {   return (width * height);    }
};

int main()
{
    int ww, hh;

    Rectangle Rect;
    cout << "Enter width : "; cin >> ww;
    cout << "Enter hight : "; cin >> hh;

    Rect.setWidth(ww);
    Rect.setHeight(hh);

    cout << "Total area: " << Rect.getArea();
}
```

Base class = Shape

Derived class = Rectangle

Class **Rectangle** dengan objek baru **Rect**

RUN

```
Enter width : 5
Enter hight : 4
Total area: 20
```

Enkapsulasi

Enkapsulasi adalah konsep mengikat data dan fungsi yang memanipulasi data dan menjaga keduanya aman dari campur tangan user akhir. C++ menggunakan kelas untuk enkapsulasi yang berisi anggota private, protected dan public. Secara default, semua item pada kelas ditentukan sebagai private.

```
#include <iostream>
using namespace std;

class Adder //base class / kelas dasar
{
    public:
        Adder(int aa = 0) //konstruktor
        {   total = aa;  }

        void addNum(int number) //yang bisa diakses dari user akhir
        {   total += number;     } //total = total + number

        int getTotal() //yang bisa diakses dari user akhir
        {   return total;     }

    private:
        int total; //data tersembunyi dari user akhir
};


int main()
{
    Adder a; //objek a
    int number, bb, cc[100];

    cout << "Enter number to add : "; cin >> number;

    for(bb=1; bb<=number; bb++)
    {
        cout << "Number " << bb << " : ";  cin >> cc[bb];
        a.addNum(cc[bb]);
    }

    cout << "Total : " << a.getTotal();
}
```

RUN

```
Enter number to add : 4
Number 1 : 3
Number 2 : 2
Number 3 : 3
Number 4 : 5
Total : 13
```

```
for(bb=0; bb<number; bb++)
{
    ..... << bb+1 << " : "; ....
    ........
}
```

Template

Template adalah kerangka atau rancangan. Fungsinya mempercepat dalam proses pembuatan isi. Ibarat seperti framework.

Template Function

```
#include <iostream>
using namespace std;

template<class T>
T jumlah(T a, T b)
{
    return (a+b);
}

int main()
{
    int hi = jumlah<int>(1,2);
    long hl = jumlah<long>(1,2);
    double hd = jumlah<double>(1.3,2.6);

    cout << hi << endl;
    cout << hl << endl;
    cout << hd;
}
```

Template function

One template can handle 3 different variable type

RUN

```
3
3
3.9
```

Fungsi tanpa template

```
#include <iostream>
using namespace std;

int addinteger(int a, int b)
{   return (a+b);   }

long addlong(long a, long b)
{   return (a+b);   }

double adddouble(double a, double b)
{   return (a+b);   }

int main()
{
    cout << addinteger(1,2) << endl;
    cout << addlong(1,2) << endl;
    cout << adddouble(1.3,2.6);
}
```

3 function

RUN

```
3
3
3.9
```

Class Template

```
#include <iostream>
using namespace std;

template <class T>
class box
{
    public:
    T whereisbig(T a, T b)
    {
        return (a>b) ? a : b;
    }
};

int main()
{
    box <int>box1;
    box <long>box2;
    box <double>box3;
    cout << box1.whereisbig(1,2) << endl;
    cout << box2.whereisbig(11,2) << endl;
    cout << box3.whereisbig(3,2);
}
```

or

```
if(a>b)
    return a;
else
    return b;
```

RUN

```
2
11
3
```

Kelas tanpa template

```
#include <iostream>
using namespace std;

class box
{
    public:
    int whereisbigint(int a,int b)
    {   return (a>b) ? a:b;   }
    long whereisbiglong(int a, int b)
    {   return (a>b) ? a:b;   }
    double whereisbigdouble(int a, int b)
    {   return (a>b) ? a:b;   }
};

int main()
{
    box box1;
    box box2;

    cout << box1.whereisbigint(1,2) << endl;
    cout << box1.whereisbiglong(11,2) << endl;
    cout << box1.whereisbigdouble(3,2) << endl;

    cout << box2.whereisbigint(5,2) << endl;
    cout << box2.whereisbiglong(10,2) << endl;
    cout << box2.whereisbigdouble(1,2);
}
```

3 function inside class box

RUN

```
2
11
3
5
10
2
```