

Struktur Data

Materi Sesi 2: Array

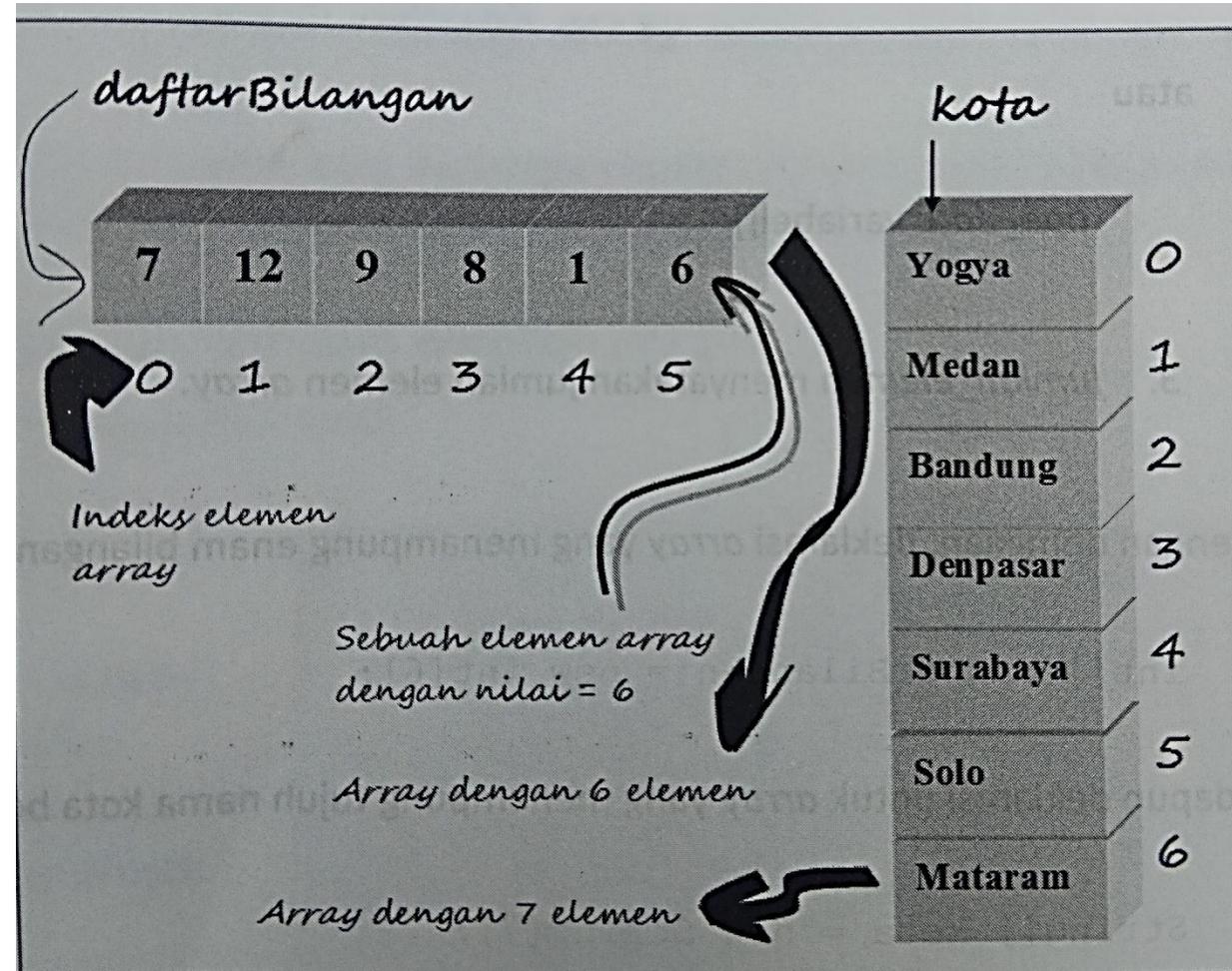
Fakultas Ilmu Komputer
Universitas Esa Unggul

Apakah array itu?

- Suatu struktur data yang dapat memuat **sejumlah data** dengan **tipe yang sama**. Dengan perkataan lain, sejumlah data dapat diwakili dengan sejumlah nama.
- Kumpulan elemen yang bertipe sama dalam urutan tertentu yang menggunakan nama yang sama. Letak atau posisi dari elemen array ditunjukkan oleh index atau posisi.

Contoh berikut memperlihatkan dua Array dengan tipe yang berbeda.

Variabel **daftarBilangan** merujuk ke array yang mempunyai enam elemen yang bertipe **int**, sedangkan variabel **kota** merujuk ke array yang memiliki tujuh elemen bertipe **String**.



Contoh diatas memperlihatkan dua Array dengan tipe yang berbeda. Variabel **daftarBilangan** merujuk ke array yang mempunyai enam elemen yang bertipe **int**, sedangkan variabel **kota** merujuk ke array yang memiliki tujuh elemen bertipe **String**.

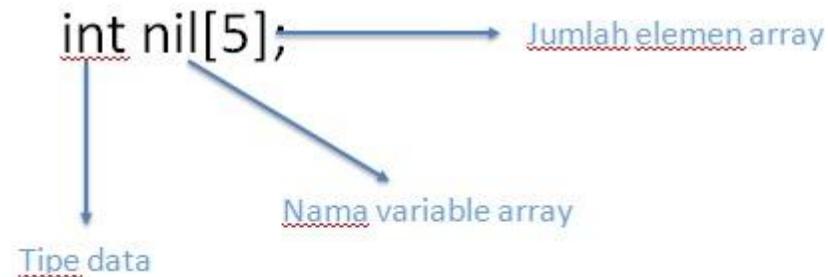
Bagaimana Deklarasi Array ?

- Variabel array dideklarasikan dengan mencantumkan tipe data dan nama variabel yang diikuti dengan banyaknya lokasi memori yang ingin dibuat.
- Bentuk umum pendeklarasian array :

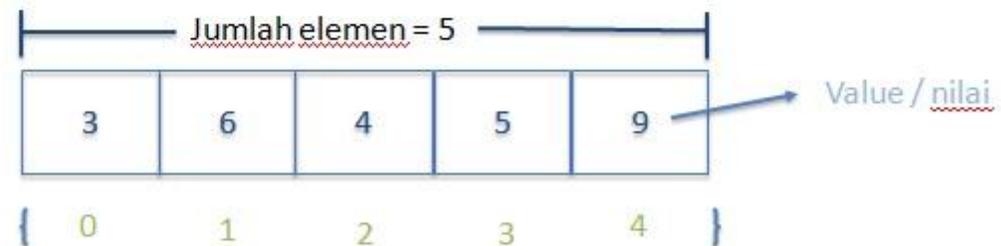
Tipe_Data Nama_Variabel [Ukuran/jumlah _elemen];

- Contoh :

```
int nil[5];
```



Ilustrasi / gambar array



Inisialisasi Array

- Nilai suatu variabel array dapat juga diinisialisasikan secara langsung pada saat deklarasi

Misalnya:

```
int nil [5] = {1, 3, 6, 12, 24};
```

Maka di penyimpanan ke dalam array dapat digambarkan sebagai berikut :

	0	1	2	3	4
nil	1	3	6	12	24

□ Pemberian nilai pada Variabel Array pada C++

Variabel pada array diberi nilai ada berbagai cara:

- **Cara Pertama:**

```
int a[5] = {4,5,8,9,12};    // Contoh dengan tipe Integer
```

- **Cara Kedua:**

```
int a[0] = 4;
```

```
int a[1] = 5;
```

```
int a[2] = 8;
```

```
int a[3] = 9;
```

```
int a[4] = 12
```

- **Cara Ketiga: (memanfaatkan struktur pemrograman REPEATITION, boleh memakai FOR, While, Do...)**

```
for (i=0;i<10;i++)
```

```
{
```

```
    cin>>a[i];
```

```
}
```

Deklarasi Array dalam Java

- **Bentuk deklarasinya :**

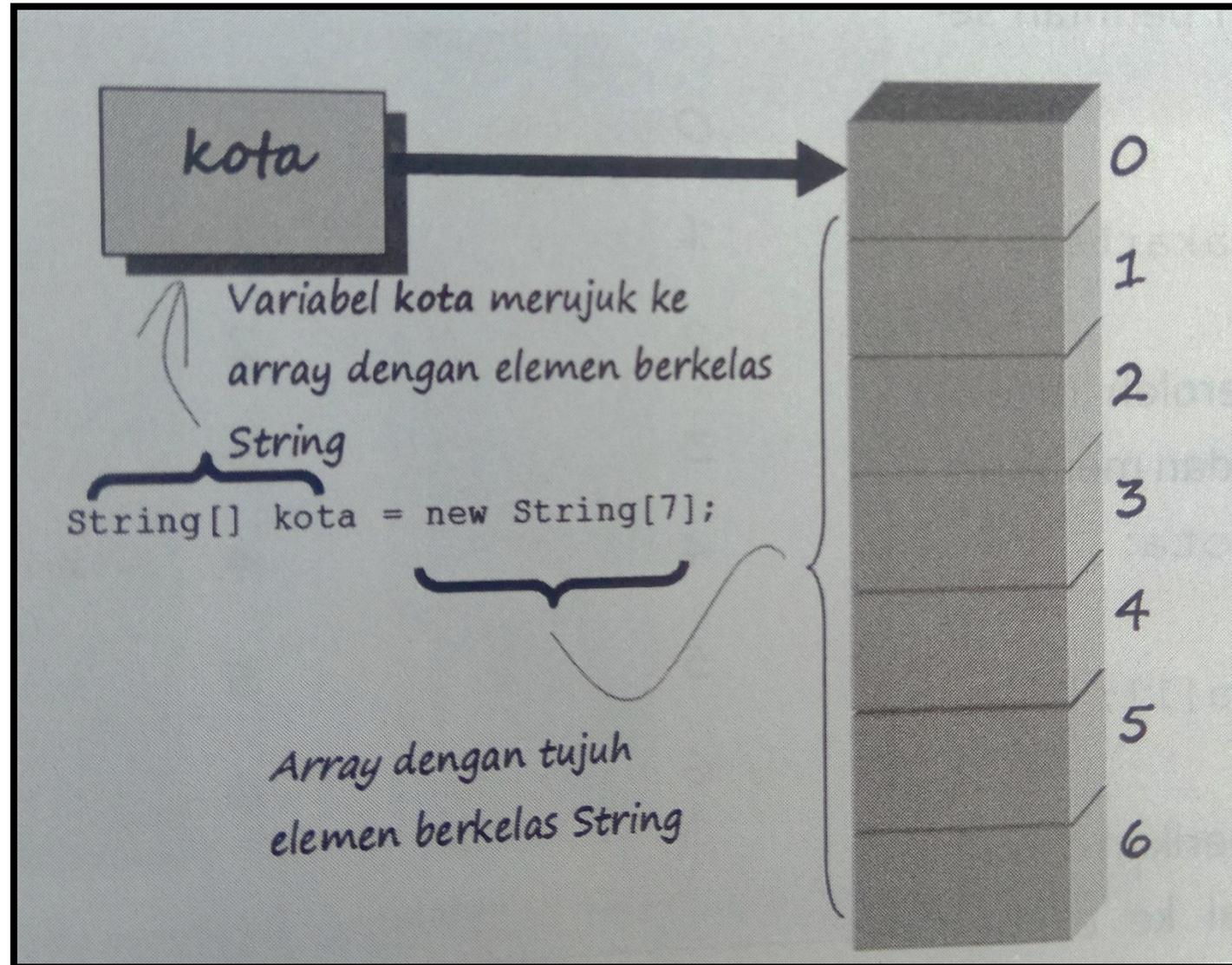
- 1) Variabel = new tipe_data [jumlah_element]; atau
- 2) Tipe_data [] Variabel = new tipe_data[];

Contoh :

- Deklarasi untuk array yang menampung enam bilangan :

Int[] daftarBilangan = new int[6];

Deklarasi untuk array yang menampung nama 7 kota:
String[] kota = new String[7];



Contoh Program mengisi nilai ke elemen array :

```
package array1;
public class Array1 {
    public static void main(String[] args) {
        String[] kota = new String[7];
        //Mengisikan nama kota
        kota[0] = "Yogya";
        kota[1] = "Medan";
        kota[2] = "Bandung";
        kota[3] = "Denpasar";
        kota[4] = "Surabaya";
        kota[5] = "Solo";
        kota[6] = "Mataram";

        //Menampilkan isi array
        for (int i=0; i<7; i++)
            System.out.println(kota[i]);
    }
}
```

Hasil Program :

Yogya
Medan
Bandung
Denpasar
Surabaya
Solo
Mataram

Apakah nilai setiap elemen dalam array tidak boleh ada yang kembar?

- Jawabanya : Boleh.
- Boleh memberikan nilai yang sama pada beberapa eleme. Bahkan, pada aplikasi tertentu. Nilai seluruh elemen array dinolkan (diberi nilai nol) pada keadaan awal.

Bagaimana cara mengakses sebuah elemen array?

- Elemen array diakses dengan notasi:

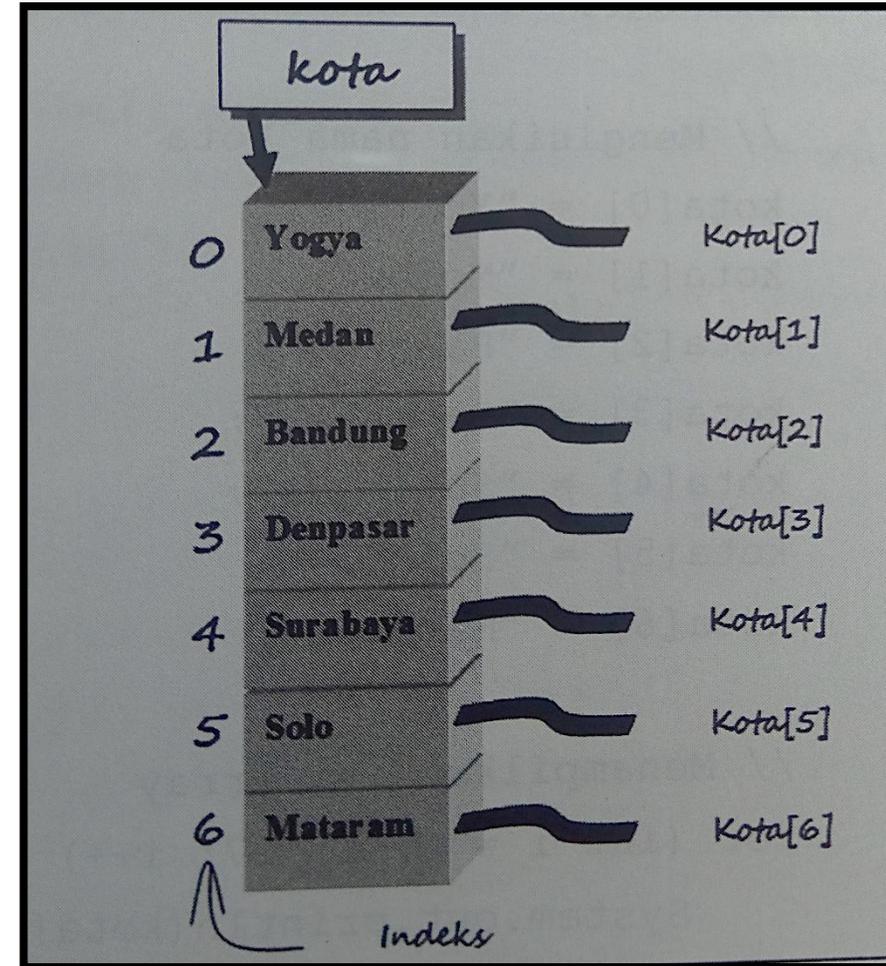
variabel_array[indeks]

Contoh :

Kota [2] menyatakan elemen ketiga dalam array.

Gambar tersebut memberikan notasi akses untuk semua elemen dalam array yang dirujuk oleh kota.

*Jika mengakses elemen yang tidak terdapat pada array Akan membuat runtime error atau kesalahan.



Apakah tipe elemen array bisa berupa char?

Jawabannya: Bisa!

Macam-Macam Array

1. Array Dimensi Satu

- Merupakan kumpulan elemen-elemen identik yang tersusun dalam satu baris. elemen-elemen tersebut memiliki tipe data yang sama, tetapi isi dari elemen tersebut boleh berbeda.

Elemen ke-	0	1	2	3	4	5	6	7
Nilai	25	23	40	75	80	30	10	90

- Bentuk Umum :
 - $\langle \text{tipe data} \rangle \text{NamaArray}[n] = \{\text{elemen0}, \text{elemen1}, \dots, n\};$
 - dimana $n = \text{jumlah elemen}$
 - Contoh : `Int Nilai[8]`

```
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
    float suhu[5];
    cout<<"Masukan 5 data suhu"<<endl;
    for (int i= 0; i< 5; i++)
    {
        cout<<"Suhu ["<<i+1<<"]:";
        cin>>suhu[i];
    }
}
```

Hasil program :

```
Masukan 5 data suhu
```

```
Suhu [1]:10
```

```
Suhu [2]:20
```

```
Suhu [3]:70
```

```
Suhu [4]:100
```

```
Suhu [5]:80
```

```
Process exited after 22.32 seconds with return v
Press any key to continue . . .
```

2. Array Dimensi Dua

array dua dimensi sering digambarkan sebagai sebuah matriks, Jika ***array satu dimensi hanya terdiri dari sebuah baris*** dan beberapa kolom elemen,

maka ***array dua dimensi terdiri dari beberapa baris*** dan **beberapa kolom elemen yang bertipe sama.**

Gambar array berdimensi (baris x kolom = 3x4):

	0	1	2	3
0	5	20	1	11
1	4	7	67	-9
2	9	0	45	3

▪ **Bentuk Umum dari Array Dua Dimensi**

<tipe data> NamaArray [m][n];

atau

<tipe data> NamaArray [m][n] = { {a,b,..z},{1,2,...,n-1} };

Contoh :

- double matrix[4][4];
- bool papan[2][2] = { {true,false},{true,false} };

Jurusan	2001	2002	2003	2004
Teknik Industri	53	70	65	60
Teknik Planalogi	10	13	21	15
Teknik Informatika	86	132	242	195

Contoh di atas merupakan matrik L berukuran 3 X 4 dapat dideklarasikan dalam C++ sebagai berikut: {[53,70,65,60],[10,13,21,15],[86,132,242,195]} dan definisi variabel untuk setiap elemen tersebut adalah:

	0	1	2	3
0	L[0,0]	L[0,1]	L[0,2]	L[0,3]
1	L[1,0]	L[1,1]	L[1,2]	L[1,3]
2	L[2,0]	L[2,1]	L[2,2]	L[2,3]

```
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
    int matrix[2][3]={{2,4,5},{7,5,3}};
    for(int i=0;i<2;i++)
    {
        for(int j=0;j<3;j++)
        {
            cout<<matrix[i][j]<<" ";
        }
        cout<<endl;
    }
    // getch();

    return 0;
}
```

```
2 4 5
7 5 3
-----
Process exited after 4.545 seconds
Press any key to continue . . . _
```

Apakah Array berdimensi tiga juga ada? Ya, ada.

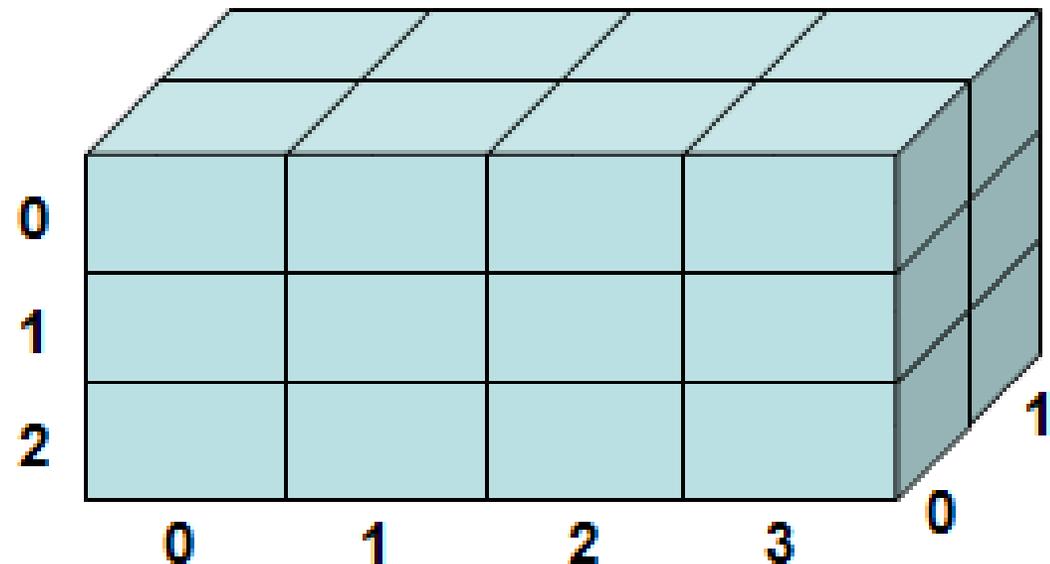
- “Digunakan untuk mengelola data dalam bentuk 3 dimensi atau tiga sisi.”

- Deklarasi :

Type_Data Nama_Variabel [index1] [index2] [index3];

- Misal : int A [3][4][2];

- Penggambaran secara Logika:



```

int main(){
    //Mendeklarasikan array
    int angka[2][5][2]={{ {10,2}, {8,20}, {2,79}, {4,67}, {8,21}},
                        {{10,27}, {27,48}, {83,28}, {2,73}, {4,98}}};

    //Menampilkan array
    for(int i = 0; i<=1; i++){
        cout<<i+1<<"\n";
        for(int j = 0; j<=4; j++){
            for(int k = 0; k<=1; k++){
                cout<<angka[i][j][k]<<"\t";
            }
            cout<<endl;
        }
        cout<<endl;
    }
}

```

```

1.
10      2
8       20
2       79
4       67
8       21

2.
10      27
27      48
83      28
2       73
4       98

```

Bisakah Array dijadikan argumen fungsi?

- Ya, Array bisa dijadikan argumen fungsi.
- **Apa itu Fungsi?**

Fungsi adalah kumpulan statmen yang dikelompokan menjadi satu bagian kode ataublok program untuk menyelesaikan tugas-tugas spesifik tertentu.

Bentuk umum dari fungsi :

1. Deklarasi : **tipe_kembalian nama_fungsi (daftar_parameter)**

- ✓ fungsi bisa memiliki satau ataua lebih parameter,
- ✓ Parameter itu sendiri merupakan variabel yang berisi nilai yang akan disertakan kedalam proses yang dilakukan dalam badan fungsi,
- ✓ daftar_parameter merupakan opsional (tidak harus)

2. Badan Fungsi (diapit menggunakan tanda “{” dan “}”)

- Badan fungsi merupakan kode atau statmen yang akan dilaksanakan didalam fungsi untuk menyelesaikan suatu permasalahan spesifik
- Dalam C++ **tipe_kembalian /tipe fungsi** ada 2 bagian :
 - a) Tidak memiliki kembalian
 - b) Memiliki kembalian sesuai dengan tipe_data

- Pertama, Tipe_kembalian yang tidak memiliki Kembalian (fungsi void)
Tipe_kembalian yang tidak memiliki Kembalian dalam C++ yaitu tipenya adalah void :

Cara 1

```
1 #include <iostream>
2 using namespace std;
3
4 void cetak5();
5
6 int main()
7 {
8     cetak5();
9 }
10
11 void cetak5() {
12     for(int i=1;i<=5;i++){
13         cout << i << " ";
14     }
15 }
```

Cara 2

```
1 #include <iostream>
2 using namespace std;
3
4 void cetak5() {
5     for(int i=1;i<=5;i++){
6         cout << i << " ";
7     }
8 }
9
10 int main()
11 {
12     cetak5();
13 }
```

- Bagaimana jika menggunakan parameter

```
1 #include <iostream>
2 using namespace std;
3
4 void cetak(int a) {
5     for(int i=1;i<=a;i++){
6         cout << i << " ";
7     }
8 }
9
10 int main()
11 {
12     cetak(100);
13 }
```

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77
78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
-----
Process exited after 0.03006 seconds with return value 0
Press any key to continue . . .
```

- Kedua, Tipe_kembalian/tipe fungsi yang memiliki kembalian

```
1 #include <iostream>
2 using namespace std;
3
4 tipe_data nama_fungsi(daf_param){
5     badan_fungsi
6
7     return
8 }
9
10 int main()
11 {
12
13 }
```

- Mempunyai sintak **return** untuk mengembalikan **nilai** sesuai **tipe_data**

Contoh :

```
1 #include <iostream>
2 using namespace std;
3
4 int maksimal(int a,int b){
5     if(a>b){
6         return a;
7     } else {
8         return b;
9     }
10 }
11
12 int main()
13 {
14     cout << maksimal(10,20);
15 }
```

20

Process exited after 0.01175 seconds with return value 0
Press any key to continue . . .

Terimakasih