

Struktur Data

Fitri Nuraeni, M.Kom

nufi3@stmik-tasikmalaya.ac.id

<http://latihan.nufi3.com>

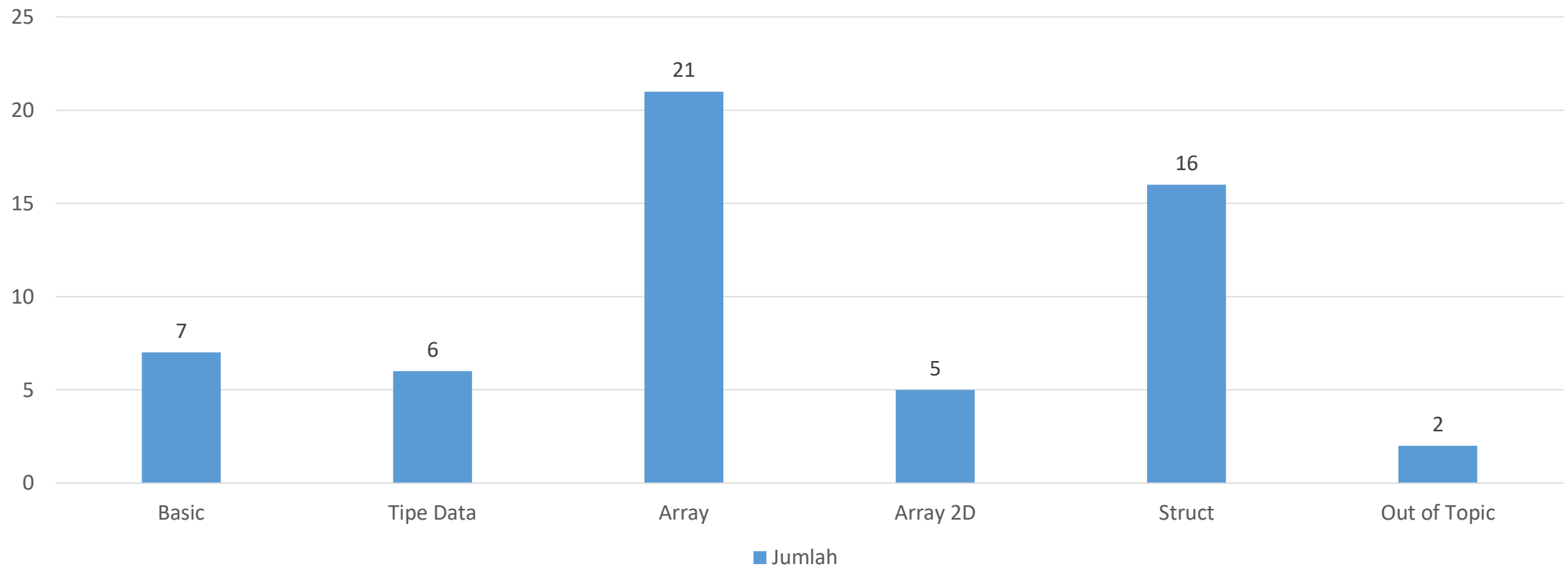
2020

Pembahasan Grup Diskusi

Pertemuan Minggu ke-5

Total Pertanyaan Masuk : 57 buah

Grafik Pertanyaan Yang Masuk Berdasarkan Materi



Struktur Data Vs Pemrograman C++

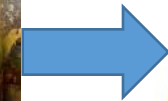
Tema Basic

Apa itu Struktur Data?

- Konsep dalam pemrograman selain algoritma adalah **Struktur Data**.
- Konsep?
 - Bahasan Struktur Data bersifat umum (global), bisa diterapkan pada berbagai bahasa pemrograman yang ada
 - Mendukung proses penyusunan algoritma dan coding aplikasi pada bahasa pemrograman
 - Berguna untuk membuat aplikasi yang dibangun menjadi lebih efektif dan efisien
- Dasar untuk memahami Struktur Data, **memahami Algoritma**.
- Bahasa Pemrograman?
 - Bisa digunakan pada bahasa apapun, jadi sebenarnya tidak hanya tentang C++.

Struktur Data sebagai Konsep

- Saat kita menggunakan suatu VARIABEL anggaplah kita simpan baso 1 porsi di sebuah mangkuk.
- Saat kita punya baso 3 porsi:
 - 3 buah mangkuk
 - 1 rantang susun 3
 - Satu kresek isinya 3 bungkus plastik baso
- Saat kita mau bikin bekal makan siang pake taperwer emak, bisa ada sekat-sekat isi taperwernya:
 - Ada bagian buat nasi
 - Ada bagian buat lauknya
 - Ada bagian buat cuci mulutnya



Struktur Data sebagai bagian Algoritma & Program

- Perhatikan Kasus berikut:
 - Mr X merupakan collector uang koin. Setiap belanja ke pasar, dia selalu menukarkan uang kertasna menjadi uang koin 100, 200, 500 atau 1000. Karena banyak orang yang tidak jujur di pasar, Mr X khawatir dia mendapatkan uang koin yang tidak sesuai jumlahnya. Kemudian, Mr X meminta anda, mahasiswa bidang komputer, untuk dibuatkan program konversi uang kertanya ke uang koin.
 - Program yang diinginkan Mr. X dapat menerima **masukan** berapa besar uang yang akan ditukarkan.
 - Program dapat **menentukan** koin mana saja yang bisa dipakai dan jumlah masing-masing koin dengan besar nominal yang sesuai dengan inputan.
 - Program **menampilkan** informasi koin nominal berapa yg digunakan dan jumlahnya.

Mana Algoritma?

- Perhatikan Kasus berikut:
 - Mr X merupakan collector uang koin. Setiap belanja ke pasar, dia selalu menukarkan uang kertasna menjadi uang koin 100, 200, 500 atau 1000. Karena banyak orang yang tidak jujur di pasar, Mr X khawatir dia mendapatkan uang koin yang tidak sesuai jumlahnya. Kemudian, Mr X meminta anda, mahasiswa bidang komputer, untuk dibuatkan program konversi uang kertasnya ke uang koin.
 - Program yang diinginkan Mr. X dapat menerima **masukan** berapa besar uang yang akan ditukarkan.
 - Program dapat **menentukan** koin mana saja yang bisa dipakai dan jumlah masing-masing koin dengan besar nominal yang sesuai dengan inputan.
 - Program **menampilkan** informasi koin nominal berapa yg digunakan dan jumlahnya.

Algoritma TukarUangKoin

- Input: x (jumlah uang)
- Proses :
$$X = (a * 1000) + (b * 500) + (c * 200) + (d * 100)$$
 - 1) $a \leftarrow X \text{ DIV } 1000$; sisa $\leftarrow X \text{ MOD } 1000$
 - 2) $b \leftarrow \text{sisa} \text{ DIV } 500$; sisa $\leftarrow \text{sisa} \text{ MOD } 500$
 - 3) $c \leftarrow \text{sisa} \text{ DIV } 200$; sisa $\leftarrow \text{sisa} \text{ MOD } 200$
 - 4) $d \leftarrow \text{sisa} \text{ DIV } 100$;
- Output : a, b, c, d

Mana peranan Struktur Data?

Algoritma TukarUangKoin

- Input: x (jumlah uang)
- Proses :
 - $X = (a * 1000) + (b * 500) + (c * 200) + (d * 100)$
 - 1) $a \leftarrow X \text{ DIV } 1000$; $\text{sisal} \leftarrow X \text{ MOD } 1000$
 - 2) $b \leftarrow \text{sisal} \text{ DIV } 500$; $\text{sisal} \leftarrow \text{sisal} \text{ MOD } 500$
 - 3) $c \leftarrow \text{sisal} \text{ DIV } 200$; $\text{sisal} \leftarrow \text{sisal} \text{ MOD } 200$
 - 4) $d \leftarrow \text{sisal} \text{ DIV } 100$;
- Output : a, b, c, d

Variabel:

X = jumlah uang

a = jumlah koin 1000

b = jumlah koin 500

c = jumlah koin 200

d = jumlah koin 100

sisal = jumlah uang sementara

Mana peranan Struktur Data?

Algoritma TukarUangKoin

- Input: x (jumlah uang)
- Proses :
$$X = (a * 1000) + (b * 500) + (c * 200) + (d * 100)$$
 - 1) $a \leftarrow X \text{ DIV } 1000$; $\text{sisal} \leftarrow X \text{ MOD } 1000$
 - 2) $b \leftarrow \text{sisal} \text{ DIV } 500$; $\text{sisal} \leftarrow \text{sisal} \text{ MOD } 500$
 - 3) $c \leftarrow \text{sisal} \text{ DIV } 200$; $\text{sisal} \leftarrow \text{sisal} \text{ MOD } 200$
 - 4) $d \leftarrow \text{sisal} \text{ DIV } 100$;
- Output : a, b, c, d

Tipe Data

X = long integer/ double

a = integer/ long integer

b = integer/ long integer

c = integer/ long integer

d = integer/ long integer

sisal = integer/ long integer

Coding Program : Bahasa C++

```
#include <iostream>
#include <conio.h>

void main() {
    long X;
    int a, b, c, d, sisa;

    cout<<"Masukan jumlah uang yang akan ditukar : ";
    cin>> X;

    //step 1
    a = X / 1000;
    sisa = X % 1000;

    //step 2
    b = sisa / 500;
    sisa = sisa % 500;
```

```
//step 2
b = sisa / 500;
sisa = sisa % 500;
```

```
//step 3
c = sisa / 200;
sisa = sisa % 200;
```

```
//step 4
d = sisa / 100;
```

```
cout<<"Koin 1000 = "<< a<< endl;
cout<<"Koin 500 = "<< b<< endl;
cout<<"Koin 200 = "<< c<< endl;
cout<<"Koin 100 = "<< d<< endl;
```

```
getch();
```

```
}
```

Coding Program : Bahasa Java

```
14 public class TukarKoin {
15     public static void main(String[] args){
16         double X;
17         int a, b, c, d, sisa;
18
19         Scanner input = new Scanner(System.in);
20         System.out.println("Masukan jumlah uang yang akan ditukar : ");
21         X = input.nextDouble();
22
23         a = (int) (X / 1000);
24         sisa = (int) (X % 1000);
25
26         b = sisa / 500;
27         sisa = sisa % 500;
```

```
28
29         c = sisa / 200;
30         sisa = sisa % 200;
31
32         d = sisa / 100;
33
34         System.out.println("Koin 1000 : " + a);
35         System.out.println("Koin 500 : " + b);
36         System.out.println("Koin 200 : " + c);
37         System.out.println("Koin 100 : " + d);
38     }
39 }
40
```

Coding Program : Bahasa Pascal

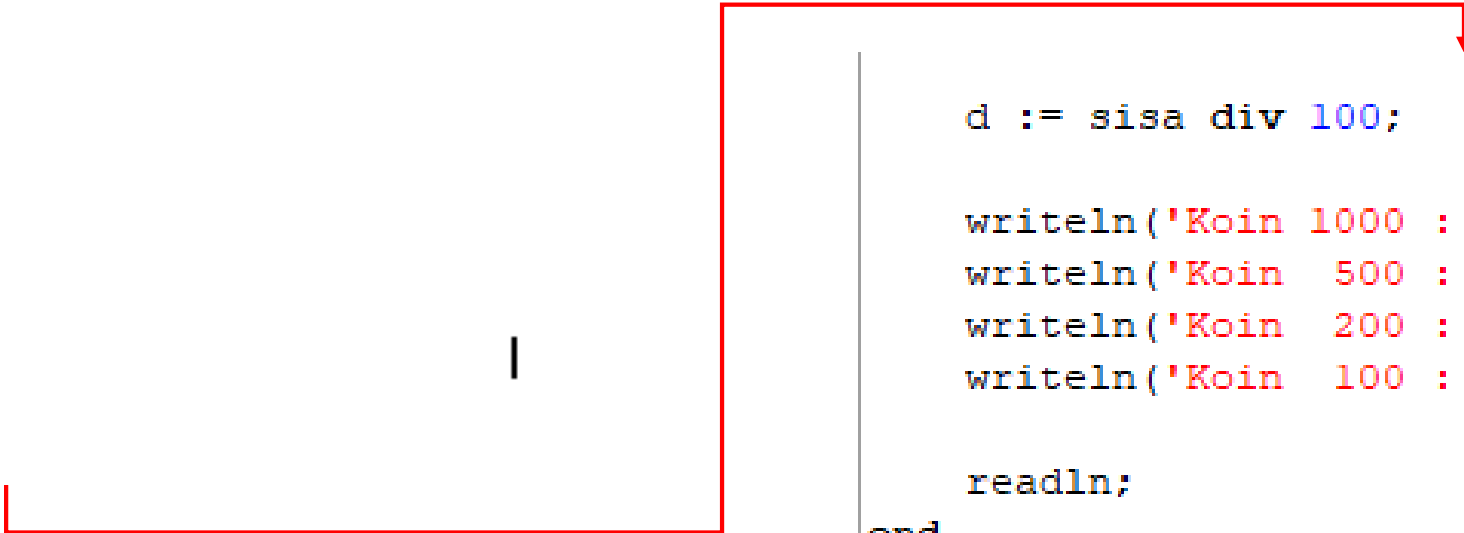
```
program tukarkoin;
uses wincrt;
var
  x : longint;
  a, b, c, d, sisa : integer;

begin
  writeln('Masukan uang yang akan ditukar : ');
  readln(x);

  a := x div 1000;
  sisa := x mod 1000;

  b := sisa div 500;
  sisa := sisa mod 500;

  c := sisa div 200;
  sisa := sisa mod 200;
```



```
  d := sisa div 100;

  writeln('Koin 1000 : ', a);
  writeln('Koin 500 : ', b);
  writeln('Koin 200 : ', c);
  writeln('Koin 100 : ', d);

  readln;
end.
```

Running Program

 E:\TEACHING\STRUKTUR DATA\MINGGUK5\TukarKoin.exe

```
Masukan jumlah uang yang akan ditukar : 8800
Koin 1000 = 8
Koin 500 = 1
Koin 200 = 1
Koin 100 = 1
```

C++

run:

Masukan jumlah uang yang akan ditukar :
8800

Koin 1000 : 8

Koin 500 : 1

Koin 200 : 1

Koin 100 : 1

BUILD SUCCESSFUL (total time: 4 seconds)

Java

 E:\TEACHING\Pascal\tukarkoin.exe

```
Masukan uang yang akan ditukar :
8800
Koin 1000 : 8
Koin 500 : 1
Koin 200 : 1
Koin 100 : 1
```

Pascal

Dasar Perintah Pada C++

Struktur Kode Program C++

```
1 | #Preprocessor Directive
2 | fungsi utama(){
3 |     Definisi yang terdiri dari kumpulan pernyataan;
4 |     //komentar
5 | }
```

Program C maupun C++ selalu tersusun dari 5 (Lima) bagian utama, yaitu :

1. Preprocessor Directive (Pengarah Kompilator)
2. Declaration (Deklarasi)
3. Definition (Definisi)
4. Statement atau Expressions (Pernyataan atau Ekspresi)
5. Comments (Komentar)

```
#include <iostream>
#include <string>
#include <conio.h>

void main(){
    int angka;
    std::string kota = "Tasikmalaya";

    cout<<"Masukan sebuah angka : ";
    cin>> angka;

    cout<<"Kota anda akan dicetak "<< angka <<" kali"<<endl;
    for(int i=0; i < angka; i++){
        cout<<kota<<" ke "<<i<<" | ";
    }
    cout<<endl;

    cout<<"Kota anda akan dicetak "<< angka <<" kali"<<endl;
    for(int i=0; i < angka; i++){
        cout<<kota<<" ke "<<i<<endl;
    }

    getch();
}
```

cout : menampilkan ke layar

cin : membaca masukan

endl : pindah baris

void main() vs int main()

```
#include <iostream>
#include <string>
#include <conio.h>

void main(){
    int angka;
    std::string kota = "Tasikmalaya";

    cout<<"Masukan sebuah angka : ";
    cin>> angka;

    cout<<"Kota anda akan dicetak "<< angka <<" kali"<<endl;
    for(int i=0; i < angka; i++){
        cout<<kota<<" ke "<<i<<endl;
    }

    getch();
}
```

C++ shell

```
1 // Example program
2 #include <iostream>
3 #include <string>
4
5 using namespace std;
6
7 int main()
8 {
9     std::string name;
10
11     cout << "What is your name? ";
12     getline (cin, name);
13     cout << "Hello, " << name << "!\n";
14 }
15
```

Get URL

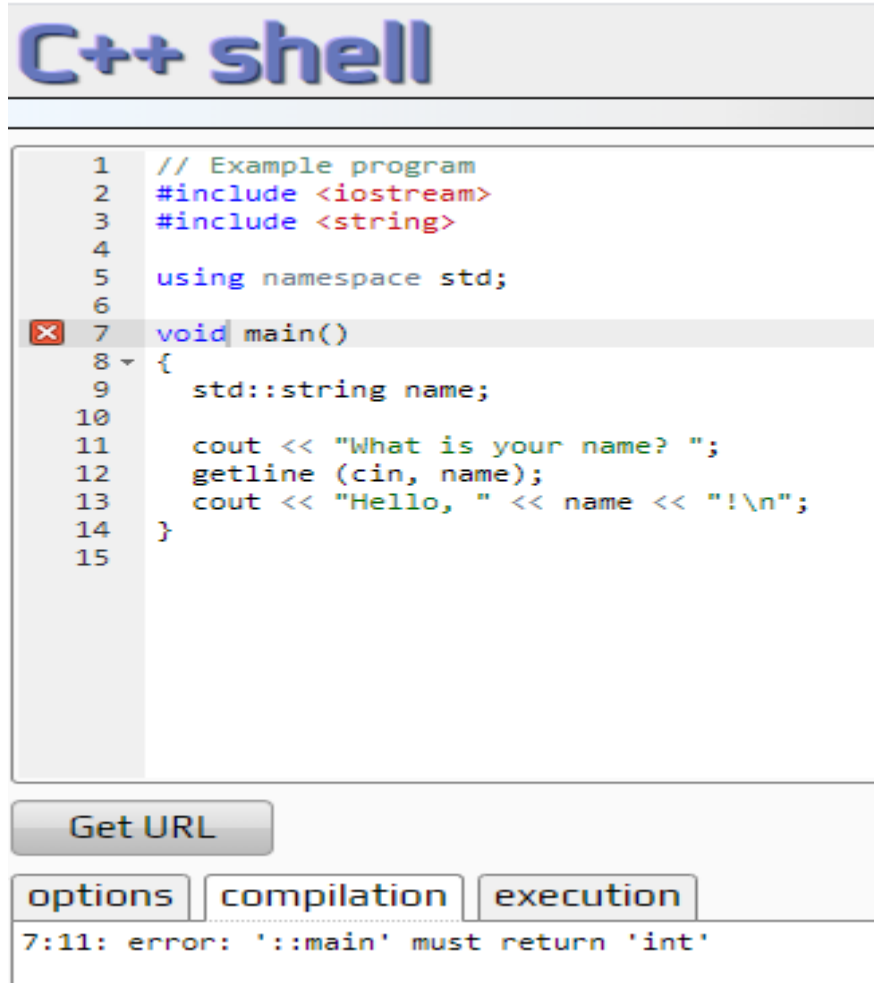
options

compilation

execution

```
What is your name? fitri nuraeni
Hello, fitri nuraeni!
```


Penjelasan tentang Bahasa C++



The screenshot shows a web-based C++ shell interface. At the top, it says "C++ shell". Below that is a text area containing C++ code. The code is as follows:

```
1 // Example program
2 #include <iostream>
3 #include <string>
4
5 using namespace std;
6
7 void main()
8 {
9     std::string name;
10
11     cout << "What is your name? ";
12     getline (cin, name);
13     cout << "Hello, " << name << "!\n";
14 }
15
```

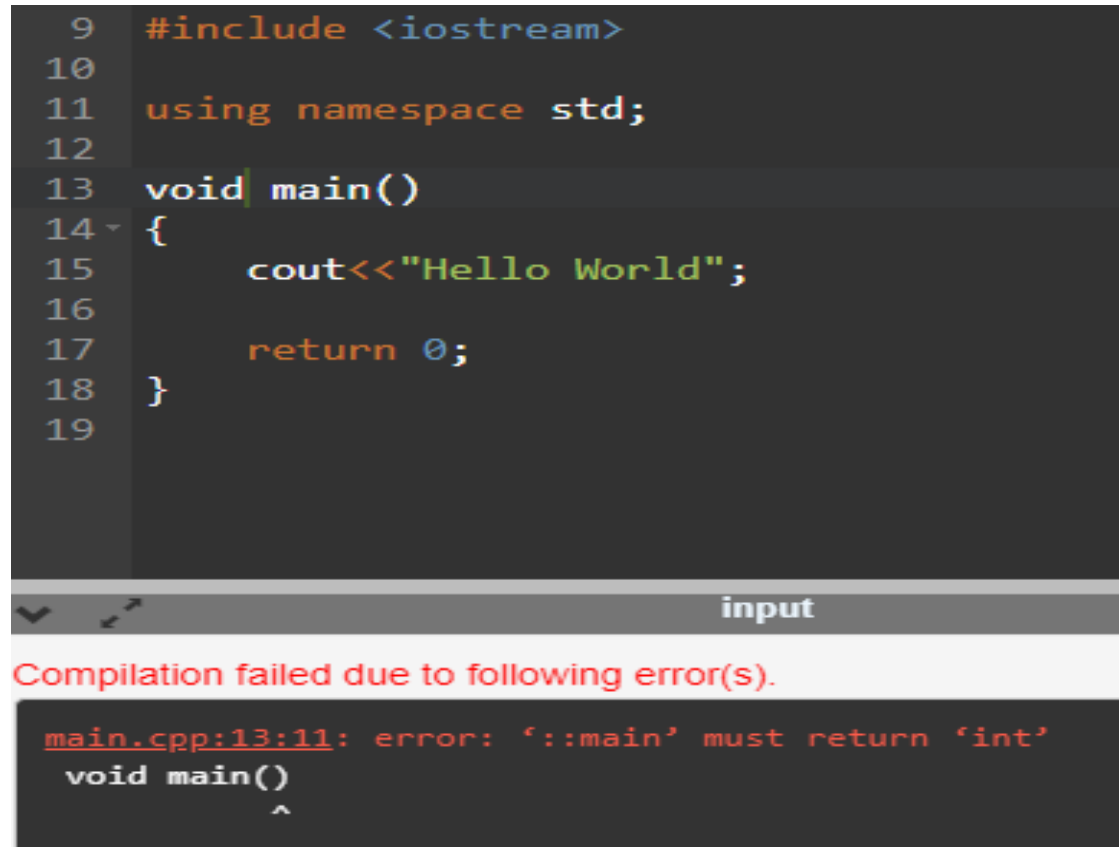
Below the code area is a "Get URL" button. At the bottom, there are three tabs: "options", "compilation", and "execution". The "compilation" tab is selected, and it shows the following error message:

```
7:11: error: '::main' must return 'int'
```

`int main()` , Merupakan fungsi (*function*) utama, fungsi yang akan dibaca oleh kompilator pertama kali secara otomatis, tidak akan ada function yang otomatis dibaca oleh kompilator kecuali fungsi utama `int main()` , fungsi utama bagaikan sebuah kepala dari sebuah program yang mengatur arah kompilator.

`int` dalam fungsi utama adalah sebuah *return type integer*. Dalam peraturan C++ fungsi utama memang harus menggunakan *return type integer*.

Dan tepat setelah *identifier* (identitas) `main` terdapat sepasang tanda kurung `()` , itu merupakan tempat dimana kita dapat mengisi *parameter* untuk *function*, tetapi dalam fungsi utama, hal itu tidak dibutuhkan dan dibiarkan dalam keadaan kosong.



The screenshot shows a C++ IDE with a dark theme. The code in the editor is as follows:

```
9 #include <iostream>
10
11 using namespace std;
12
13 void main()
14 {
15     cout<<"Hello World";
16
17     return 0;
18 }
19
```

Below the code area is a tab labeled "input". Below that is a red error message:

```
Compilation failed due to following error(s).
```

Below the error message is a dark box containing the following error message:

```
main.cpp:13:11: error: '::main' must return 'int'
void main()
    ^
```

Tipe Data

Karakter Vs String

```
#include <iostream>
#include <string>
#include <conio.h>

int main() {
    int angka;
    std::string kota;
    char huruf;

    cout<< "Masukan nama kota : "; cin>>kota;
    cout<< "Masukan sebuah angka : "; cin>> angka;

    huruf = kota[(angka-1)];
    cout<<"huruf ke-"<< angka <<" = "<< huruf<<endl;

    huruf = 'a';
    kota = "bekasi";
    cout<<"Kota   : "<< kota <<endl;
    cout<<"Huruf  : "<< huruf <<endl;

    getch();
}
```

- char : tipe data yang dapat menampung 1 byte (8bit) sama dengan 1 karakter.

128	64	32	16	8	4	2	1
0	1	1	0	0	0	0	1
97 (dec)							
'a' (char)							

- Array char = string
char kota[20] = 'bekasi';
- Library string

Input String dengan spasi

cin	cin.getline	Gets
Pada perintah cin, tidak dapat membaca spasi pada karakter data yang kita inputkan.	Pada perintah cin.getline(nama_variabel, ukuran_variabel), dapat membaca spasi pada karakter data yang kita inputkan tetapi kita harus memberikan nama variabel dan ukuran variabel didalam perintah cin.getline	Pada perintah gets juga dapat membaca spasi pada karakter yang kita inputkan namun pada perintah ini kita harus menambahkan Preprocessor #include <stdio.h> dan pada akhir perintah tidak disarankan untuk memasukan ukuran dari variabel

```
#include <iostream>
#include <conio.h>

int main(){
    int angka;
    char kota[50];

    cout<< "Masukan nama kota : ";
    cin.getline(kota, 50);

    cout<< "Masukan sebuah angka : ";
    cin>> angka;

    cout<< "Kota anda akan dicetak "<< angka <<" kali"<<endl;
    for(int i=0; i < angka; i++){
        cout<< kota<<" ke "<<i<<endl;
    }

    getch();
}
```

Borland c++

```
// Contoh program
1 #include <iostream>
2 #include <string>
3
4 using namespace std;
5
6 int main()
7 {
8     int angka;
9     std::string kota;
10
11     cout<< "Masukan nama kota : ";
12     getline (cin, kota);
13
14     cout<< "Masukan sebuah angka : ";
15     cin>> angka;
16
17     cout<< "Kota anda akan dicetak "<< angka <<" kali"<<endl;
18     for(int i=0; i < angka; i++){
19         cout<< kota<<" ke "<<i<<endl;
20     }
21 }
22
```

<http://cpp.sh/>

Menggunakan <stdio.h> gets

```
#include <iostream>
#include <stdio.h>
#include <conio.h>

int main() {
    int angka;
    char kota[50];

    cout<< "Masukan nama kota : ";
    gets(kota);

    cout<< "Masukan sebuah angka : ";
    cin>> angka;

    cout<< "Kota anda akan dicetak "<< angka << " kali"<<endl;
    for(int i=0; i < angka; i++){
        cout<< kota<<" ke "<<i<<endl;
    }

    getch();
}
```

Operator pada C++

- Cek di <http://www.cplusplus.com/doc/tutorial/operators/>
- Jenis operator
 - Assignment operator (=)
 - Arithmetic operators (+, -, *, /, %)
 - Compound assignment (+=, -=, *=, /=, %=, >>=, <<=, &=, ^=, |=)
 - Increment and decrement (++ , --)
 - Relational and comparison operators (==, !=, >, <, >=, <=)
 - Logical operators (!, &&, ||)
 - Conditional ternary operator (?)
 - Comma operator (,)
 - Bitwise operators (&, |, ^, ~, <<, >>)

Integer Vs Float

- Cek : <http://www.cplusplus.com/doc/tutorial/variables/>
- Integer : menyimpan data ukuran 2 bytes (16 bit)

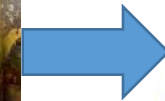
	1 byte								1 byte								decimal
2^x	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	$2^7 + 2^3 + 2^0$
	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	$128 + 8 + 1$
	00000000								10001001								137

- Float : menyimpan data ukuran 4 bytes (32 bit)

Konsep Array

Struktur Data sebagai Konsep

- Saat kita menggunakan suatu VARIABEL anggaplah kita simpan baso 1 porsi di sebuah mangkuk.
- Saat kita punya baso 3 porsi:
 - 3 buah mangkuk
 - 1 rantang susun 3
 - Satu kresek isinya 3 bungkus plastik baso
- Saat kita mau bikin bekal makan siang pake taperwer emak, bisa ada sekat-sekat isi taperwernya:
 - Ada bagian buat nasi
 - Ada bagian buat lauknya
 - Ada bagian buat cuci mulutnya

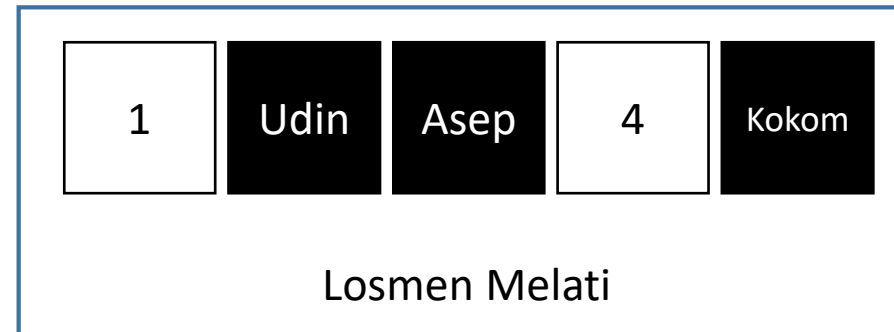


Apa itu Array?



- Jika kita memiliki data yang tipenya sama tapi jumlahnya banyak dan diproses bersamaan.
- Kasus :
 - Losmen “melati” memiliki 5 kamar yang disewakan. Pegawai losmen harus mencatat kamar mana saja yang sudah disewa dan siapa nama penyewanya.

noKamar	NamaPenyewa
1	Kosong
2	Udin
3	Asep
4	Kosong
5	Kokom



Apa itu Array?

- Jika kita memiliki data yang tipenya sama tapi jumlahnya banyak dan diproses bersamaan.
- Kasus :
 - Losmen “melati” memiliki 5 kamar yang disewakan. Pegawai losmen harus mencatat kamar mana saja yang sudah disewa dan siapa nama penyewanya.
 - Digunakan konsep Array **bukan variabel**
 - Tipe data Array, **string** karena nama penyewa
 - noKamar digunakan sebagai **indeks**

noKamar	NamaPenyewa
1	Kosong
2	Udin
3	Asep
4	Kosong
5	Kokom

Implementasi Program

```
#include <iostream>
#include <string>
#include <conio.h>

int main(){
    std::string penyewa[] = {"kosong", "kosong", "kosong", "kosong", "kosong"};
    int noKamar;
    char lagi = 'Y';

    while(lagi == 'Y'){
        cout<< "Masukan nomor kamar : ";
        cin>> noKamar;

        cout<< "Masukan nama tamu : ";
        cin>> penyewa[(noKamar-1)];

        cout<< "Input lagi (Y/ T) ";
        cin>> lagi;
    }

    cout<< "Daftar Tamu Kamar"<<endl;
    for(int i=0; i < 5; i++){
        cout<<"No Kamar "<< i <<" Tamu : "<< penyewa[i] <<endl;
    }

    getch();
}
```

E:\TEACHING\Struktur Data\mingguK5\basic.exe

```
Masukan nomor kamar : 2
Masukan nama tamu : Udin
Input lagi (Y/ T) Y
Masukan nomor kamar : 3
Masukan nama tamu : Asep
Input lagi (Y/ T) Y
Masukan nomor kamar : 5
Masukan nama tamu : Kokom
Input lagi (Y/ T) T
Daftar Tamu Kamar
No Kamar 0 Tamu : kosong
No Kamar 1 Tamu : Udin
No Kamar 2 Tamu : Asep
No Kamar 3 Tamu : kosong
No Kamar 4 Tamu : Kokom
```

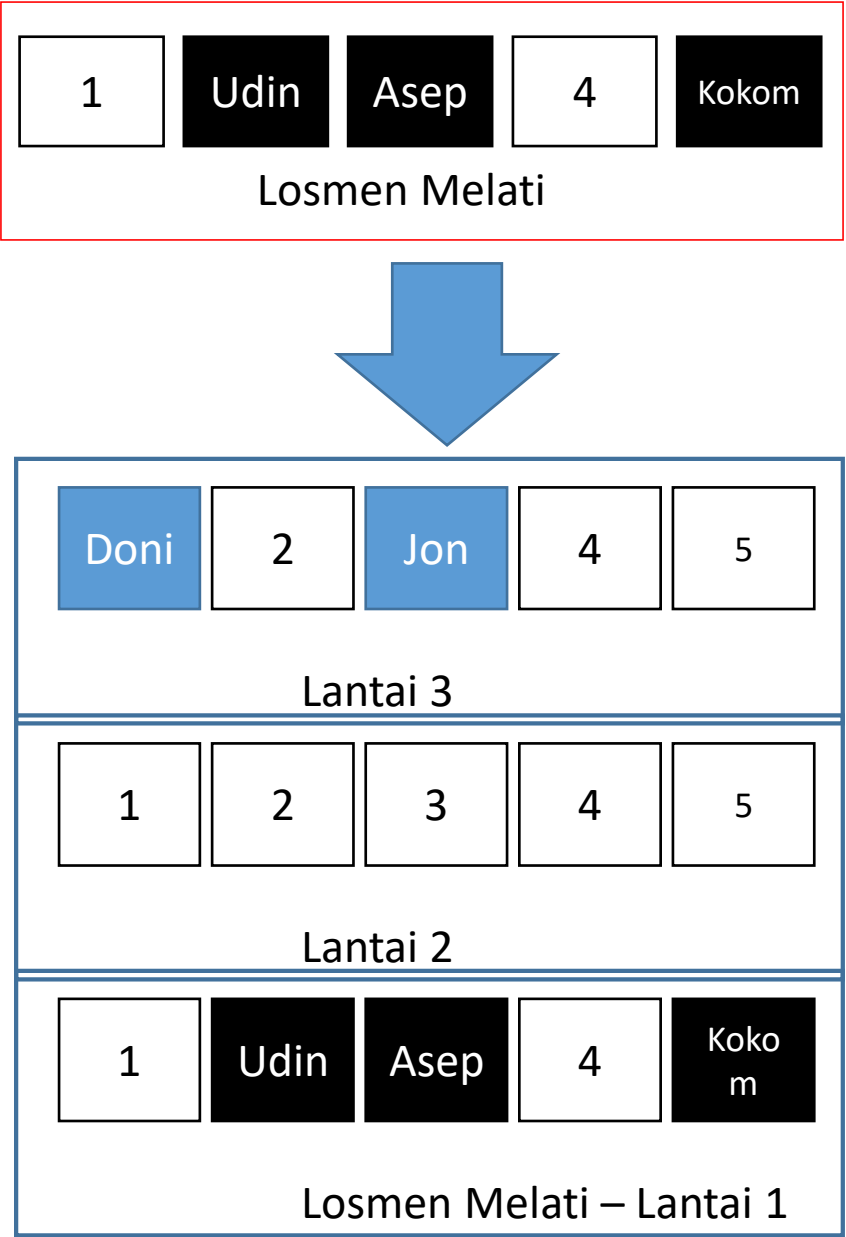
Array 1D vs Array 2D

- Kasus :
 - Losmen “melati” memiliki 5 kamar yang disewakan. Pegawai losmen harus mencatat kamar mana saja yang sudah disewa dan siapa nama penyewanya.
 - Digunakan konsep Array **bukan variabel**
 - Tipe data Array, **string** karena nama penyewa
 - noKamar digunakan sebagai **indeks**

noKamar	NamaPenyewa
1	Kosong
2	Udin
3	Asep
4	Kosong
5	Kokom

- Losmen “melati” direnovasi menjadi gedung 3 lantai, dimana tiap lantai menyediakan 5 kamar untuk disewakan.

NoKamar	Lantai 1	Lantai 2	Lantai 3
1	Kosong	Kosong	Doni
2	Udin	Kosong	Kosong
3	Asep	Kosong	Jon
4	Kosong	Kosong	Kosong
5	Kokom	Kosong	Kosong



Implementasi Program (1)

```
#include <iostream>
#include <string>
#include <conio.h>

int main(){
    std::string penyewa[3][5] = {{ "kosong", "kosong", "kosong", "kosong", "kosong"},
                                  { "kosong", "kosong", "kosong", "kosong", "kosong"},
                                  { "kosong", "kosong", "kosong", "kosong", "kosong"} };

    int noKamar, lantai;
    char lagi = 'Y';

    while(lagi == 'Y'){
        cout<< "Masukan Lantai : ";
        cin>> lantai;

        cout<< "Masukan nomor kamar : ";
        cin>> noKamar;

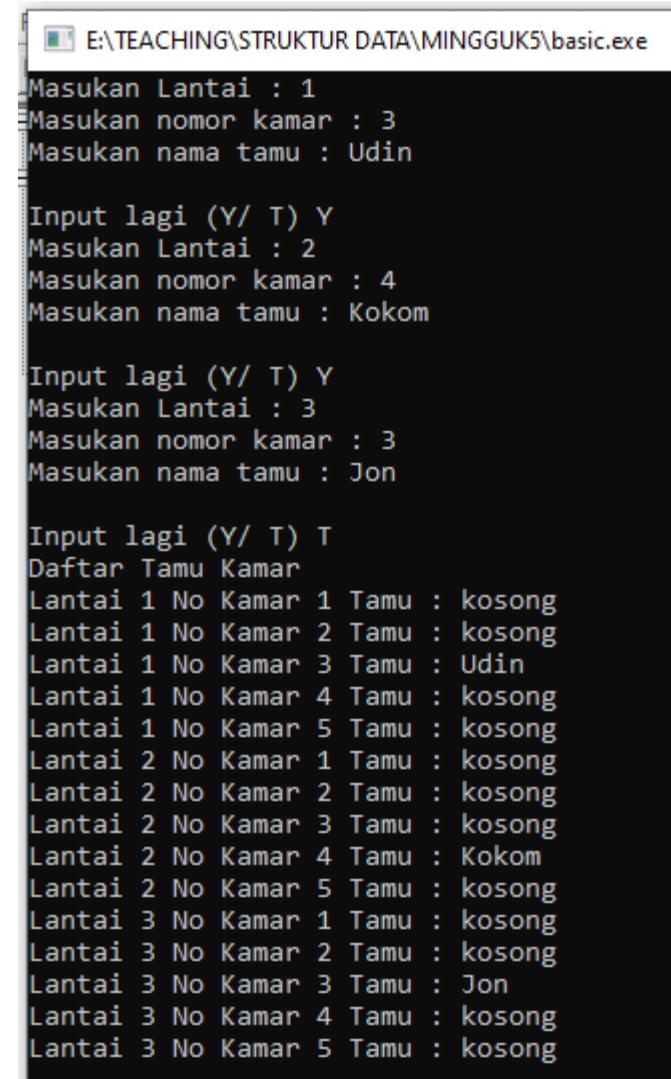
        cout<< "Masukan nama tamu : ";
        cin>> penyewa [(lantai-1)][(noKamar-1)];

        cout<< endl<< "Input lagi (Y/ T) ";
        cin>> lagi;
    }
```

Implementasi Program (2)

```
cout<< "Daftar Tamu Kamar"<<endl;
for(int i=0; i < 3; i++){
    for(int j=0; j < 5; j++){
        cout<<"Lantai " << (i+1) <<" No Kamar " << (j+1) <<" Tamu : " << penyewa[i][j] <<endl;
    }
}

getch();
}
```



The screenshot shows a Windows command prompt window titled "E:\TEACHING\STRUKTUR DATA\MINGGUK5\basic.exe". The program prompts the user to enter floor, room number, and guest name three times. The first input is floor 1, room 3, guest Udin. The second is floor 2, room 4, guest Kokom. The third is floor 3, room 3, guest Jon. After the inputs, the program displays a 3x5 grid of room status. Room 1, room 3, and room 5 on all floors are empty. Room 4 on floor 2 is occupied by Kokom, and room 3 on floor 3 is occupied by Jon.

```
E:\TEACHING\STRUKTUR DATA\MINGGUK5\basic.exe
Masukan Lantai : 1
Masukan nomor kamar : 3
Masukan nama tamu : Udin

Input lagi (Y/ T) Y
Masukan Lantai : 2
Masukan nomor kamar : 4
Masukan nama tamu : Kokom

Input lagi (Y/ T) Y
Masukan Lantai : 3
Masukan nomor kamar : 3
Masukan nama tamu : Jon

Input lagi (Y/ T) T
Daftar Tamu Kamar
Lantai 1 No Kamar 1 Tamu : kosong
Lantai 1 No Kamar 2 Tamu : kosong
Lantai 1 No Kamar 3 Tamu : Udin
Lantai 1 No Kamar 4 Tamu : kosong
Lantai 1 No Kamar 5 Tamu : kosong
Lantai 2 No Kamar 1 Tamu : kosong
Lantai 2 No Kamar 2 Tamu : kosong
Lantai 2 No Kamar 3 Tamu : kosong
Lantai 2 No Kamar 4 Tamu : Kokom
Lantai 2 No Kamar 5 Tamu : kosong
Lantai 3 No Kamar 1 Tamu : kosong
Lantai 3 No Kamar 2 Tamu : kosong
Lantai 3 No Kamar 3 Tamu : Jon
Lantai 3 No Kamar 4 Tamu : kosong
Lantai 3 No Kamar 5 Tamu : kosong
```

Deklarasi Array

<type data> *<identifier>* [**<konstanta jumlah elemen>**];



```
char alfabet [26];
```

Deklarasi Array sekaligus inisialisasi elemennya

```
char alfabet[26] = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h',  
                    'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p',  
                    'q', 'r', 's', 't', 'u', 'v', 'w', 'x',  
                    'y', 'z'};
```

```
char alfabet[] = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h',  
                  'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p',  
                  'q', 'r', 's', 't', 'u', 'v', 'w', 'x',  
                  'y', 'z', 'a'};
```



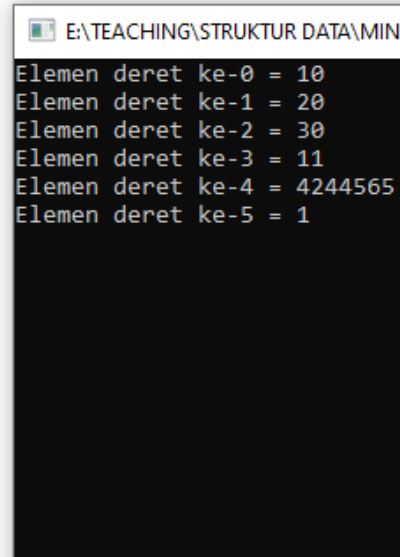
```
#include <iostream>
#include <conio.h>

int main(){
    int deret[3];

    deret[0] = 10;
    deret[1] = 20;
    deret[2] = 30;
    deret[3] = 11;

    for(int i = 0; i <= 5; i++){
        cout<<"Elemen deret ke-"<< i <<" = ";
        cout<< deret[i] << endl;
    }

    getch();
}
```



```
E:\TEACHING\STRUKTUR DATA\MIN
Elemen deret ke-0 = 10
Elemen deret ke-1 = 20
Elemen deret ke-2 = 30
Elemen deret ke-3 = 11
Elemen deret ke-4 = 4244565
Elemen deret ke-5 = 1
```

Kenapa tidak error?

- Secara konsep, array memiliki jumlah elemen yang tetap.
- Namun, Bahasa Pemrograman C/ C++ tidak memberikan spesifikasi apa pun yang berhubungan dengan masalah mengakses indeks yang tidak valid. Sesuai standar ISO C itu disebut **Undefined Behavior**.
- Perilaku tidak terdefinisi (**Undefined Behavior**) adalah hasil dari mengeksekusi kode komputer yang perilakunya tidak ditentukan oleh spesifikasi bahasa yang dapat dipatuhi kode tersebut, untuk kondisi program saat ini.

Bahasa Pascal : *Warning range check error while evaluating constant*

```
10 program ArrayOutOfBound;
11 uses crt;
12 var
13     deret : array[1..3] of integer;
14     i : integer;
15 begin
16     deret[1] := 10;
17     deret[2] := 20;
18     deret[3] := 30;
19     deret[4] := 11;
20     for i:=1 to 6 do
21     begin
22         writeln ('Deret ke-', i, ' = ', deret[i]);
23     end;
24 end.
25
```

input

```
main.pas(19,11) Warning: range check error while evaluating constants
Linking a.out
25 lines compiled, 0.2 sec
1 warning(s) issued
/usr/bin/ld.bfd: warning: link.res contains output sections; did you forget -T?
Deret ke-1 = 10
Deret ke-2 = 20
Deret ke-3 = 30
Deret ke-4 = 4
Deret ke-5 = 0
```

Bahasa Java : tidak dapat mengakses diluar batas elemen array

```
11 public class ArrayJava {  
12     public static void main(String[] args){  
13         int[] deret = new int[3];  
14  
15         deret[0] = 10;  
16         deret[1] = 20;  
17         deret[2] = 30;  
18         deret[3] = 11;  
19  
20         for(int i = 0; i <= 5; i++){  
21             System.out.print("Elemen deret ke-" + i + " = ");  
22             System.out.println( deret[i] );  
23         }  
24     }  
25 }  
26
```

Output - StrukDatP5 (run) ×

run:

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 3
at ArrayJava.main(ArrayJava.java:18)

Pada C++, bounds-checking bias dilakukan pada tipe class

As for *why* there is no bounds checking, there are a couple aspects to the answer:

- An array is a leftover from C. C arrays are about as primitive as you can get. Just a sequence of elements with contiguous addresses. There is no bounds checking because it is simply exposing raw memory. Implementing a robust bounds-checking mechanism would have been almost impossible in C.
- In C++, bounds-checking is possible on class types. But an array is still the plain old C-compatible one. It is not a class. Further, C++ is also built on another rule which makes bounds-checking non-ideal. The C++ guiding principle is "you don't pay for what you don't use". If your code is correct, you don't need bounds-checking, and you shouldn't be forced to pay for the overhead of runtime bounds-checking.
- So C++ offers the `std::vector` class template, which allows both. `operator[]` is designed to be efficient. The language standard does not require that it performs bounds checking (although it does not forbid it either). A vector also has the `at()` member function which is *guaranteed* to perform bounds-checking. So in C++, you get the best of both worlds if you use a vector. You get array-like performance without bounds-checking, *and* you get the ability to use bounds-checked access when you want it.

Class Array & Class Vector

- **bounds-checking**

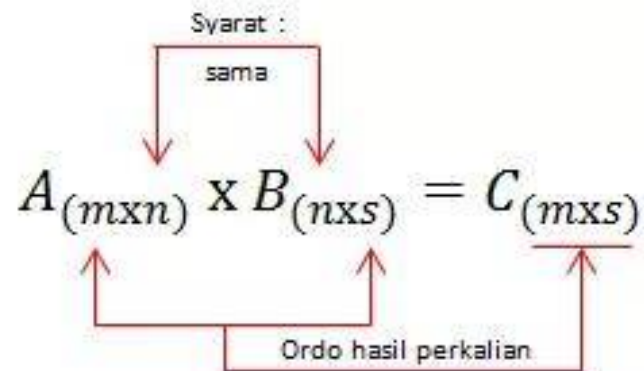
```
9  #include <iostream>
10 #include <array>
11
12 using namespace std;
13
14 int
15 main ()
16 {
17     char alfabet[3] = {'a','b','c'};
18     std::array<char, 3> angka = {'1','2','3'};
19
20     alfabet[4] = 'z';
21     for(int id=0; id < 5; id++){
22         cout<< alfabet[id] <<" ";
23     }
24     cout<<endl;
25
26     cout << "Ukuran array angka : " << angka.size() << endl;
27     for(int i=0; i < angka.size(); i++){
28         cout<< angka[i] <<" ";
29     }
30     cout<<endl;
31
32     return 0;
33 }
```

```
9  #include <iostream>
10 #include <vector>
11
12 using namespace std;
13
14 int
15 main ()
16 {
17     vector<int> g1;
18
19     for (int i = 1; i <= 5; i++)
20         g1.push_back(i);
21
22     cout << "Isi vektor g1 : ";
23     for (int id = 0; id < g1.size(); id++)
24         cout << g1[id] << " ";
25     cout << endl;
26
27     cout << "Output of begin and end: ";
28     for (auto i = g1.begin(); i != g1.end(); ++i)
29         cout << *i << " ";
30
31     return 0;
32 }
```

Array 2D : Perkalian Matriks (teori matematika)

- Perkalian antara dua matriks yaitu matriks A dan B, dapat dilakukan jika **jumlah kolom A sama dengan jumlah baris B**.
- Perkalian tersebut menghasilkan suatu matriks dengan jumlah baris sama dengan matriks A dan jumlah saman dengan matriks B, sehingga:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \times \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} a.e + b.g & a.f + b.h \\ c.e + d.g & c.f + d.h \end{pmatrix}$$



```

#include <iostream>
#include <conio.h>

int main () {
    int M1 [3][4] = {{1,2,3,4}, {1,2,3,4}, {1,2,3,4}};
    int M2 [4][2] = {{1,2}, {3,4}, {5,6}, {7,8}};
    int M3 [3][2];
    int bar, kol;

    cout<<"Matrik M1 "<< endl;
    for(bar=0; bar<3; bar++){
        for(kol=0; kol<4; kol++){
            cout<< M1[bar][kol]<<" ";
        }
        cout<<endl;
    }

    cout<<endl;
    cout<<"Matrik M2 "<< endl;
    for(bar=0; bar<4; bar++){
        for(kol=0; kol<2; kol++){
            cout<< M2[bar][kol]<<" ";
        }
        cout<<endl;
    }
}

```

```

    cout<<endl;
    cout<<"Matrik M3 = M1 * M2 "<< endl;
    //perhitungan tiap elemen M3
    for(bar=0; bar<3; bar++){
        for(kol=0; kol<2; kol++){
            M3[bar][kol] = 0;
            for(int i=0; i<4; i++){
                M3[bar][kol] += (M1[bar][i] * M2[i][kol]);
            }
        }
    }

    //menampilkan M3
    for(bar=0; bar<3; bar++){
        for(kol=0; kol<2; kol++){
            cout<< M3[bar][kol]<<" ";
        }
        cout<<endl;
    }

    getch();
}

```

Matrik M1

1 2 3 4

1 2 3 4

1 2 3 4

Matrik M2

1 2

3 4

5 6

7 8

Matrik M3 = M1 * M2

M3 [0][0] = (M1[0][0] * M2[0][0]) + (M1[0][1] * M2[1][0]) + (M1[0][2] * M2[2][0]) + (M1[0][3] * M2[3][0])

M3 [0][1] = (M1[0][0] * M2[0][1]) + (M1[0][1] * M2[1][1]) + (M1[0][2] * M2[2][1]) + (M1[0][3] * M2[3][1])

M3 [1][0] = (M1[1][0] * M2[0][0]) + (M1[1][1] * M2[1][0]) + (M1[1][2] * M2[2][0]) + (M1[1][3] * M2[3][0])

M3 [1][1] = (M1[1][0] * M2[0][1]) + (M1[1][1] * M2[1][1]) + (M1[1][2] * M2[2][1]) + (M1[1][3] * M2[3][1])

M3 [2][0] = (M1[2][0] * M2[0][0]) + (M1[2][1] * M2[1][0]) + (M1[2][2] * M2[2][0]) + (M1[2][3] * M2[3][0])

M3 [2][1] = (M1[2][0] * M2[0][1]) + (M1[2][1] * M2[1][1]) + (M1[2][2] * M2[2][1]) + (M1[2][3] * M2[3][1])

M3 [0][0] = (1 * 1) + (2 * 3) + (3 * 5) + (4 * 7) +

M3 [0][1] = (1 * 2) + (2 * 4) + (3 * 6) + (4 * 8) +

M3 [1][0] = (1 * 1) + (2 * 3) + (3 * 5) + (4 * 7) +

M3 [1][1] = (1 * 2) + (2 * 4) + (3 * 6) + (4 * 8) +

M3 [2][0] = (1 * 1) + (2 * 3) + (3 * 5) + (4 * 7) +

M3 [2][1] = (1 * 2) + (2 * 4) + (3 * 6) + (4 * 8) +

M3[0][0] = 50 M3[0][1] = 60

M3[1][0] = 50 M3[1][1] = 60

M3[2][0] = 50 M3[2][1] = 60

Hasil Running Aplikasi

 E:\TEACHING\STRUKTUR DATA\MINGGUK5\ar

Matrik M1

1 2 3 4

1 2 3 4

1 2 3 4

Matrik M2

1 2

3 4

5 6

7 8

Matrik M3 = M1 * M2

50 60

50 60

50 60

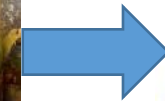
Tentang Array

- Bagaimana cara menggunakan array dengan tipe data lebih dari 100
 - Maksimal Elemen Array? The maximum size of an array is determined by the amount of memory that a program can access.
- Apakah Array mempunyai batasan Dimensi??
 - Secara konsep, array multidimensi bisa banyak, tapi secara umum implementasinya hanya array 1D dan 2D.
- Konsep array itu hanya untuk program c++ atau bisa untuk program yg lain
 - Konsep Array itu ada pada tiap bahasa pemrograman, hanya cara deklarasinya saja yang berbeda-beda tergantung aturan sintaks masing-masing bahasa.
- "sizeof" itu fungsi nya untuk apa
 - Sizeof untuk mengetahui ukuran suatu objek

Struct

Struktur Data sebagai Konsep

- Saat kita menggunakan suatu VARIABEL anggaplah kita simpan baso 1 porsi di sebuah mangkuk.
- Saat kita punya baso 3 porsi:
 - 3 buah mangkuk
 - 1 rantang susun 3
 - Satu kresek isinya 3 bungkus plastik baso
- Saat kita mau bikin bekal makan siang pake taperwer emak, bisa ada sekat-sekat isi taperwernya:
 - Ada bagian buat nasi
 - Ada bagian buat lauknya
 - Ada bagian buat cuci mulutnya



Apa itu **Struct** ?

- Saat kita mau bikin bekal makan siang pake taperwer emak, bisa ada sekat-sekat isi taperwernya:
 - Ada bagian buat nasi
 - Ada bagian buat lauknya
 - Ada bagian buat cuci mulutnya

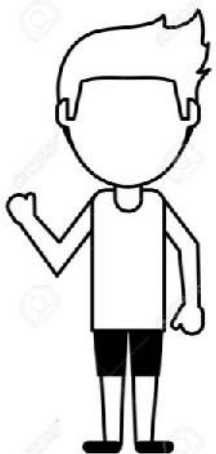


Struct adalah sekelompok elemen data yang dikelompokkan bersama dalam satu nama.

Elemen data ini, yang dikenal sebagai anggota/ field, dapat memiliki tipe dan panjang yang **berbeda**

Kapan menggunakan Struct?

- Saat kita harus mengolah data yang banyak namun data-data tersebut merupakan satu kesatuan.
- Contoh:
 - Orang memiliki nama, jenis kelamin, usia.



Orang

Nama : string
jnsKel : char
Usia : int

Deklarasi



Dosen : Orang

Nama = "asep"
jnsKel = 'L'
Usia = 48

Implementasi Variabel



Mahasiswa[5] : Orang

[1] Nama = "diki"
jnsKel = 'L'
Usia = 19

[2] Nama = "asti"
jnsKel = 'P'
Usia = 18

[3] Nama = "deden"
jnsKel = 'L'
Usia = 20

Dst..

Implementasi Array

Cara Deklarasi

- Kata kunci **typedef** memungkinkan pemrogram untuk membuat nama-nama baru untuk tipe bentukan - itu secara harfiah berarti “*type definition*” atau “definisi tipe”.

- Cara deklarasi:

```
struct {  
    int nilaiAngka;  
    char nilaiHuruf;  
} Sname, Ename;
```

```
struct namaType {  
    int nilaiAngka;  
    char nilaiHuruf;  
};  
struct namaType Sname, Ename;
```

```
typedef struct {  
    int nilaiAngka;  
    char nilaiHuruf;  
} namaType;  
namaType Sname, Ename;
```

```

#include <iostream>
#include <conio.h>
#include <string>

int main () {
    struct orang{
        std::string nama;
        char jnsKel;
        int usia;
    };

```

Tipe **Orang** digunakan pada
variabel dosen

```

    struct orang dosen;

```

```

    //input data ke Variabel dosen

```

```

    dosen.nama = "Asep";

```

```

    dosen.jnsKel = 'L';

```

```

    dosen.usia = 48;

```

```

    //output data ke Variabel dosen

```

```

    cout<<"Data Dosen : "<<endl;

```

```

    cout<<" Nama      : "<< dosen.nama<<endl;

```

```

    cout<<" Kelamin   : "<< dosen.jnsKel<<endl;

```

```

    cout<<" Usia      : "<< dosen.usia<<endl;

```

```

    getch();

```

```

}

```



E:\TEACHING\STRUKTUR I

Data Dosen :

Nama : Asep

Kelamin : L

Usia : 48


```

#include <iostream>
#include <conio.h>
#include <string>

int main () {
    struct orang{
        std::string nama;
        char jnsKel;
        int usia;
    };

    struct orang mahasiswa[3];

    //input data ke array mahasiswa
    mahasiswa[0].nama = "Diki";
    mahasiswa[0].jnsKel = 'L';
    mahasiswa[0].usia = 19;

    mahasiswa[1].nama = "Asti";
    mahasiswa[1].jnsKel = 'P';
    mahasiswa[1].usia = 18;

```

```

    mahasiswa[2].nama = "Deden";
    mahasiswa[2].jnsKel = 'L';
    mahasiswa[2].usia = 20;

    //output data ke Variabel dosen
    cout<<"Data Mahasiswa "<<endl;
    for(int i=0; i < 3; i++){
        cout<< (i+1)<<". "<< mahasiswa[i].nama<<" (";
        cout<< mahasiswa[i].jnsKel<<") ";
        cout<< mahasiswa[i].usia<<" tahun"<< endl;
    }
    getch();
}

```

 E:\TEACHING\STRUKTUR DATA\MINGGUK5\array_out

```

Data Mahasiswa
1. Diki (L) 19 tahun
2. Asti (P) 18 tahun
3. Deden (L) 20 tahun

```

Alternatif Alat Praktikum

- Editor C++ Online

- https://www.onlinegdb.com/online_c++_compiler
- <http://cpp.sh/>
- <https://repl.it/languages/cpp>
- <https://www.jdoodle.com/online-compiler-c++/>

- Aplikasi Editor C++ di Android

- CppDroid
 - <https://play.google.com/store/apps/details?id=name.antonsmirnov.android.cppdroid&hl=in>
- AnWriter Free
 - <https://play.google.com/store/apps/details?id=com.anism.anwriter&hl=en>

Sorting in Array

Next Meeting