



PENGANTAR ALGORITMA DAN PEMROGRAMAN II

Mardiyyah Hasnawi

Outline

➤ Capaian Pembelajaran Lulusan

➤ Review ALPRO

- Penyajian Algoritma
- Dasar Pemrograman C++ part 1

➤ Array Multi Dimensi

- a. Mendeklarasikan Array Multi dimensi
- b. Melewatkan Array Sebagai Argument Fungsi

➤ Tugas 1

Capain Pembelajaran Lulusan

- Mahasiswa menguasai algoritma dan pemrograman
- Mahasiswa mampu mengimplementasikan algoritma ke dalam pemrograman



REVIEW ALPRO I

Struktur Dasar Algoritma

- Algoritma Sekuensial (Sequence Algorithm)
- Algoritma Perulangan (Looping Algorithm)
- Algoritma Percabangan atau Bersyarat (Conditional Algorithm)
- Gabungan ketiganya

Hubungan algoritma, masalah dan solusi



Penyajian Algoritma

- Bentuk tulisan atau Bahasa : pseudocode
- Bentuk gambar/notasi: flowchart

Format Syntax Algoritma

JUDUL ALGORITMA

Nama Algoritma

DEKLARASI/
KAMUS
{Lokal / Umum}
ALGORITMA

Type namatype : type [subrange]
Type namatype : array [min..maks] of type
Namavar : type
Namavar : array [min..maks] of type
Const nama = nilai
Procedure namaproc
(Input/Output[daftar_nama_parameter:type])

TUBUH
ALGORITMA/
DESKRIPSI

Notasi Assignment
Notasi Kondisional/Pemilihan
Notasi Pengulangan
Notasi Pemanggilan

URAIAN
ALGORITMA

Kumpulan algoritma masing-masing
Procedure ataupun Function yang dipanggil
dari Tubuh Algoritma

Struktur Pseudocode

Judul

{Berisi Judul Algoritma}

Deklarasi atau kamus
{Berisi Deklarasi Variabel
atau Konstantan}

Deskripsi / Implementasi
{Berisi Inti Algoritma}

CONTOH - PSEUDOCODE

◦ Menghitung penjumlahan :

1. Mulai
2. baca a, b
3. $c = a + b$
4. Jika ($c > 0$), maka invokasi ke fungsi $\text{sum}(c+2)$
5. Print sum (hasil penjumlahan)
6. Selesai

Contoh lain- Notasi Algoritmik

Algoritma Penjumlahan

{menghitung penjumlahan apabila hasil penjumlahan lebih besar dari nol}

Deklarasi

{Definisi nama tetapan}

integer a=0;

integer b=1;

{definisi nama variabel}

integer c;

function sum(c) → integer

{ c+2; }

Deskripsi

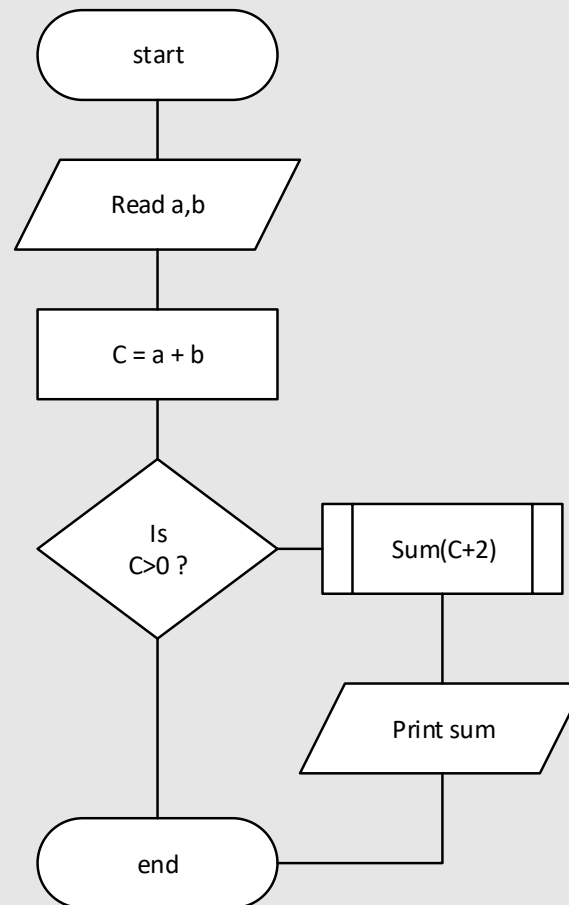
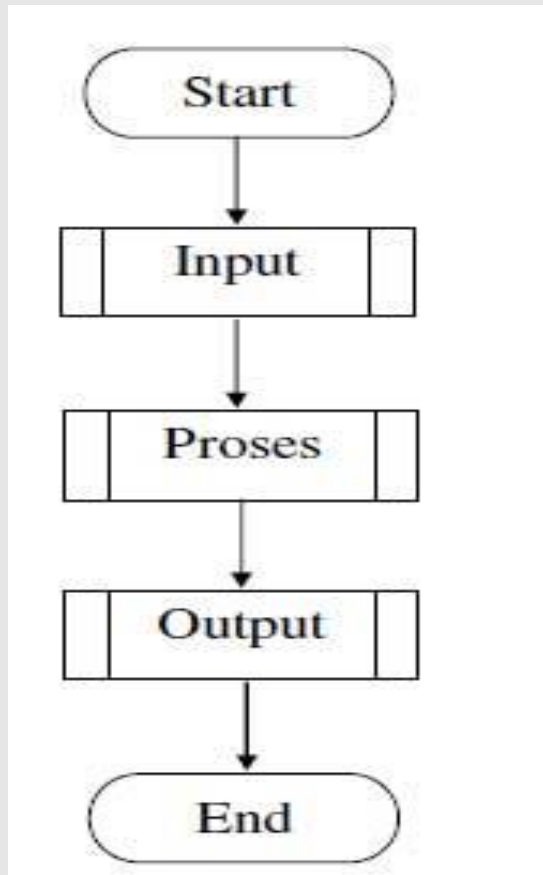
read(a,b);

c ← a+b;

if (c>0), then sum(c)

write(sum);

FLOWCHART DAN CONTOH





DASAR PEMROGRAMAN C++ PART 1

Variabel dan Tipe Data

◦ Bentuk Umum

```
#include <iostream.h>
#include <conio.h>
int a;    //variable global
int main()
{
    int b; //variable local
    return 0;
}
```

tipeData namaVariabel ; <1>

atau

tipeData namaVar1, namaVar2, ; <2>

atau

tipeData namaVariabel = **nilai** ; <3>

atau

tipeData namaVar1 = **nilai**, ... ; <4>

Deklarasi Konstanta Variabel

- Cara 1 : **Deklarasi Konstanta (const)**

- const** tipe_data nama_konstanta = nilai_tetapan;

- Sintak :

- Contoh :

```
const double PI = 3.14;  
const int NILAI_MAX = 100;  
const char MYKARAKTER = 'M';  
const string MYSTRING = "Andi"  
const int WIDTH = 100;  
const char TAB = '\\t';  
const ZIP = 12440;
```

Deklarasi Konstanta Variabel

- Cara 2 : **Konstanta Define (#define)**

- Sintak : `#define nama_konstanta nilai_tetapan`

- Contoh :

```
#define MAX 10
#define PI 3.14
#define MYNAME "MARIMAR"
#define PI 3.14159265
#define NEWLINE '\n'
#define WIDTH 100
```

Tabel Pengelompokan Operator

JENIS OPERATOR	NOTASI		KEGUNAAN
	Algoritma	Bahasa C++	
Negasi	-	-	Mengubah data angka menjadi -/+
Aritmatika	+ - * / div mod ←	+ - * / / % =	Penjumlahan Pengurangan Perkalian Pembagian Pembagian dibulatkan Sisa Pembagian Pemberi nilai
Relasional	< > = <= >= <>	< > == <= >= !=	Kurang dari Lebih dari Sama dengan Kurang dari atau sama dengan Lebih dari atau sama dengan Tidak sama dengan

Lanjutan Tabel Pengelompokan Operator

JENIS OPERATOR	NOTASI		KEGUNAAN
	Algoritma	Bahasa C++	
Logika	<u>not</u> <u>and</u> <u>or</u> <u>xor</u>	! && tdk ada	Negasi terhadap nilai Boolean Operasi And thd dua nilai Boolean Operasi Or thd dua nilai Boolean Operasi Xor thd dua nilai Boolean
Bit	shl shr sot and or xor	<< >> ~ & ^	Geser satu bit ke kiri Geser satu bit ke kanan Komplemen suatu bit Operasi And terhadap dua bit Operasi Or terhadap dua bit Operasi Xor terhadap dua bit
Address	@	&	Menunjukkan alamat memori suatu variable yang menyatakan nilai yang ditunjuk oleh pointer

Type-casting

- Type-casting → mengubah sebuah nilai dengan tipe data tertentu ke dalam tipe data yang lain.

byte → short, char, int, long, float, double

short → int, long, float, double

char → int, long, float, double

int → long, float, double

long → float, double

float → double

- Type-casting dapat juga dilakukan secara eksplisit

Contoh:

```
int x;
```

```
x = (int) (5.0 / 2);
```

ekspresi **5.0 / 2** menghasilkan nilai bertipe real sedangkan **x** bertipe integer. Agar **x** dapat menyimpan hasil perhitungan, maka nilainya harus di-casting terlebih dahulu menjadi bertipe integer, dengan konsekuensi hilangnya tingkat presisi dari nilai tersebut. Pada akhirnya x akan menyimpan nilai 2.

Pembacaan dan Penulisan Data

- Menggunakan `#include <iostream>`
- Contoh :

```
cin>>Nama_variable;
```

```
cout<<" teksss";  
          atau  
cout<<"      "<< namavariabel;
```

Bentuk lain

- Bahasa C
- Menggunakan `#include <stdio>`
- Contoh:

```
scanf("%d",&nama);
```

```
Printf("%d ", nama);
```

Control Statement

- Selection (c++)

1. statement if
2. statement if... else
3. statement if... else if...
4. switch....case
5. switch....case...default

- Looping (c++)

- **For** - Jelas jumlah perulangannya
- **While** – cek awal,berulang-ulang selama kondisi terpenuhi
- **Do...while** - cek akhir, minimal 1 statement

Function dan Procedure

Struktur Fungsi / Prosedur :

- **Bagian Judul (header).**

Terdiri atas Nama prosedur/fungsi dan deklarasi parameter (jika ada).

- **Bagian deklarasi.**

Untuk mengumumkan nama-nama.

- **Bagian deskripsi.**

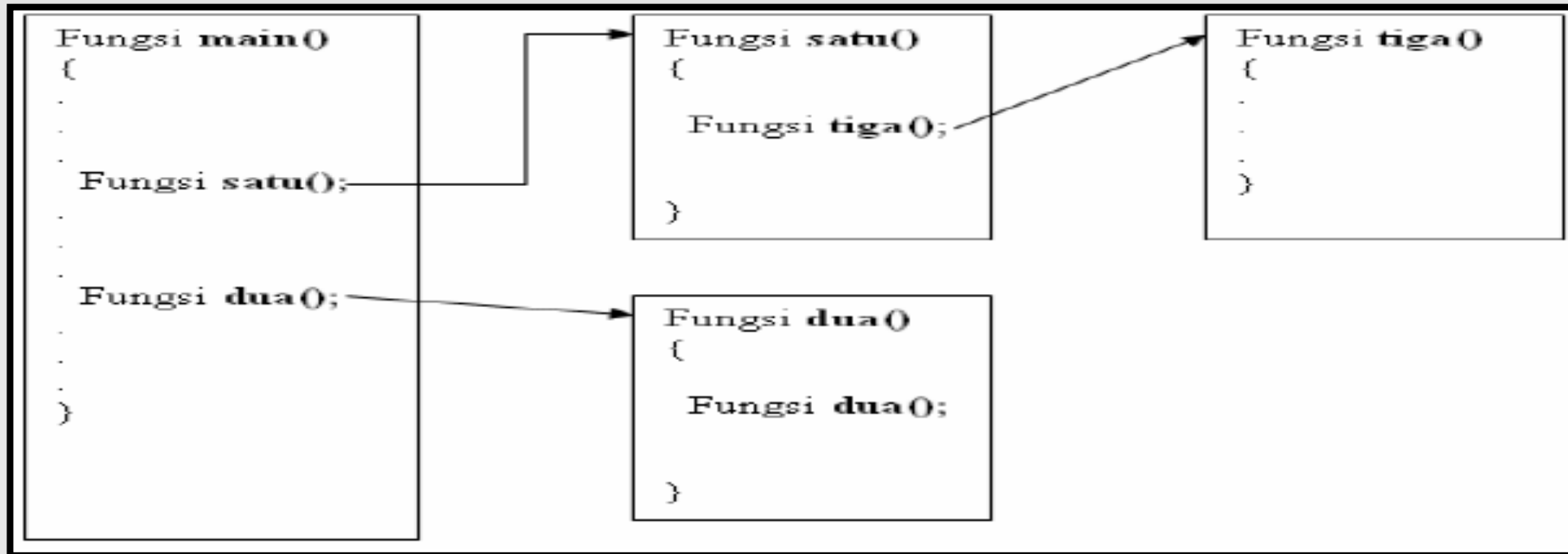
Disebut juga badan prosedur/fungsi.

lanjut fungsi/prosedur

- Tipedata namafungsi/prosedur (parameter1, ..., parameter n)
- {
 - [Return] //isi
- }

Pemanggilan Fungsi

- Pada dasarnya fungsi dapat memanggil fungsi lain, bahkan fungsi dapat memanggil dirinya sendiri (rekursif)



Deklarasi Array C++

```
tipeData namaArray[jumlah] ;
```

- ***tipeData*** → tipe dari array yang dipesan, dapat berupa tipe data dasar (int, char, float, dll), struktur maupun class.
- ***namaArray*** → nama dari array yang dipesan
- ***[]*** → jumlah pasang kurung-siku menyatakan dimensi dari array
- ***jumlah*** → jumlah elemen dari array, harus disertakan pada saat deklarasi serta harus berupa konstanta bertipe bilangan bulat.



ARRAY MULTI DIMENSI

ARTI ARRAY

suatu tipe data terstruktur yang berupa sejumlah data sejenis (bertipe data sama) yang jumlahnya tetap dan diberi suatu nama tertentu

- Elemen-elemen array tersusun secara berderet dan dapat diakses secara random di dalam memori.
- Array memiliki alamat yang bersebelahan/berdampingan tergantung lebar tipe datanya.
- Array dapat berupa array 1 dimensi, 2 dimensi, bahkan n-dimensi.
- Elemen-elemen array bertipe data sama dan bisa berisi nilai yang sama atau berbeda-beda.



Pengaksesan elemen array

- Secara teknis, data berbentuk array tersimpan di memory komputer dengan lokasi yang berurutan. Array di C++ **bersifat statis**.
- Pengisian dan pengambilan nilai pada indeks tertentu dapat dilakukan dengan mengeset nilai atau menampilkan nilai pada indeks yang dimaksud .



DEKLARASI ARRAY

PSEUDOCODE

Kamus:

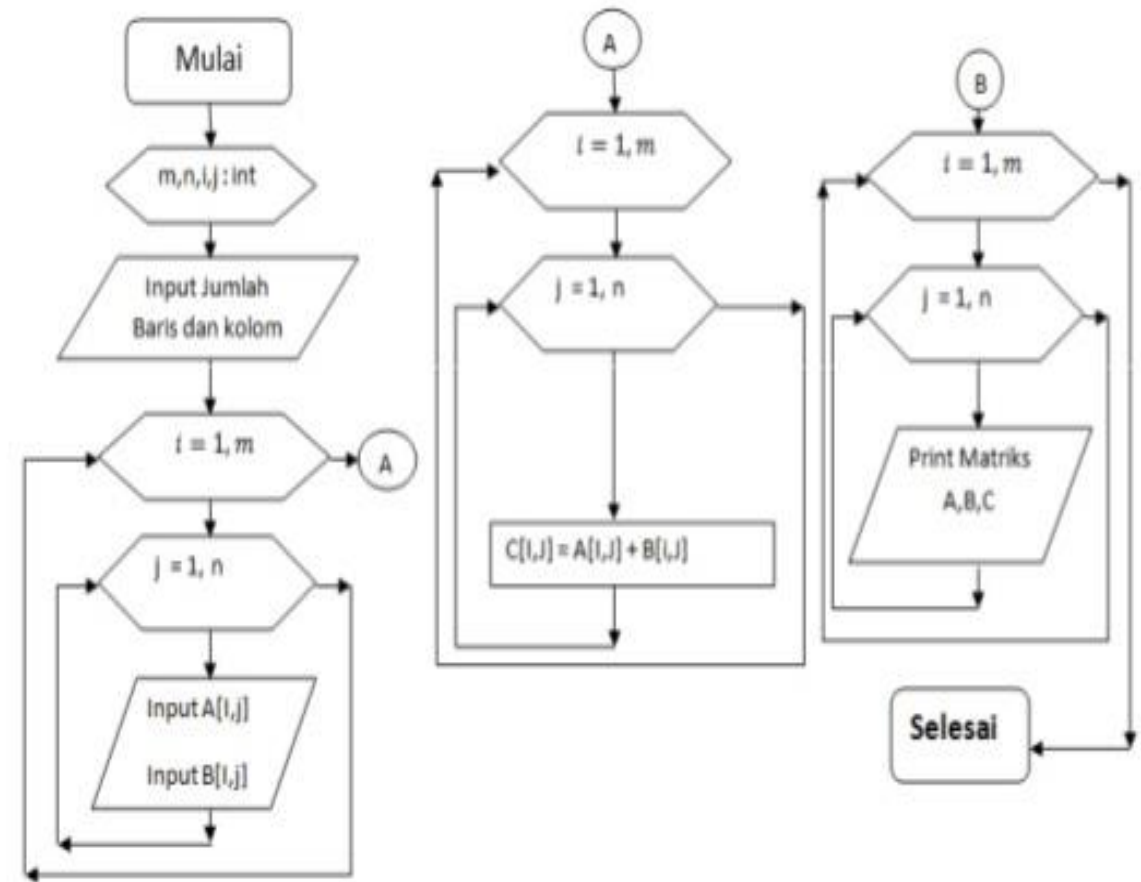
Hasil : array [1..100] of integer
NamaHari : array [1..10] of character
Nilai : array [1..10] of real

C++

```
int Hasil[100];  
char NamaHari[10];  
float Nilai[10];
```

CONTOH

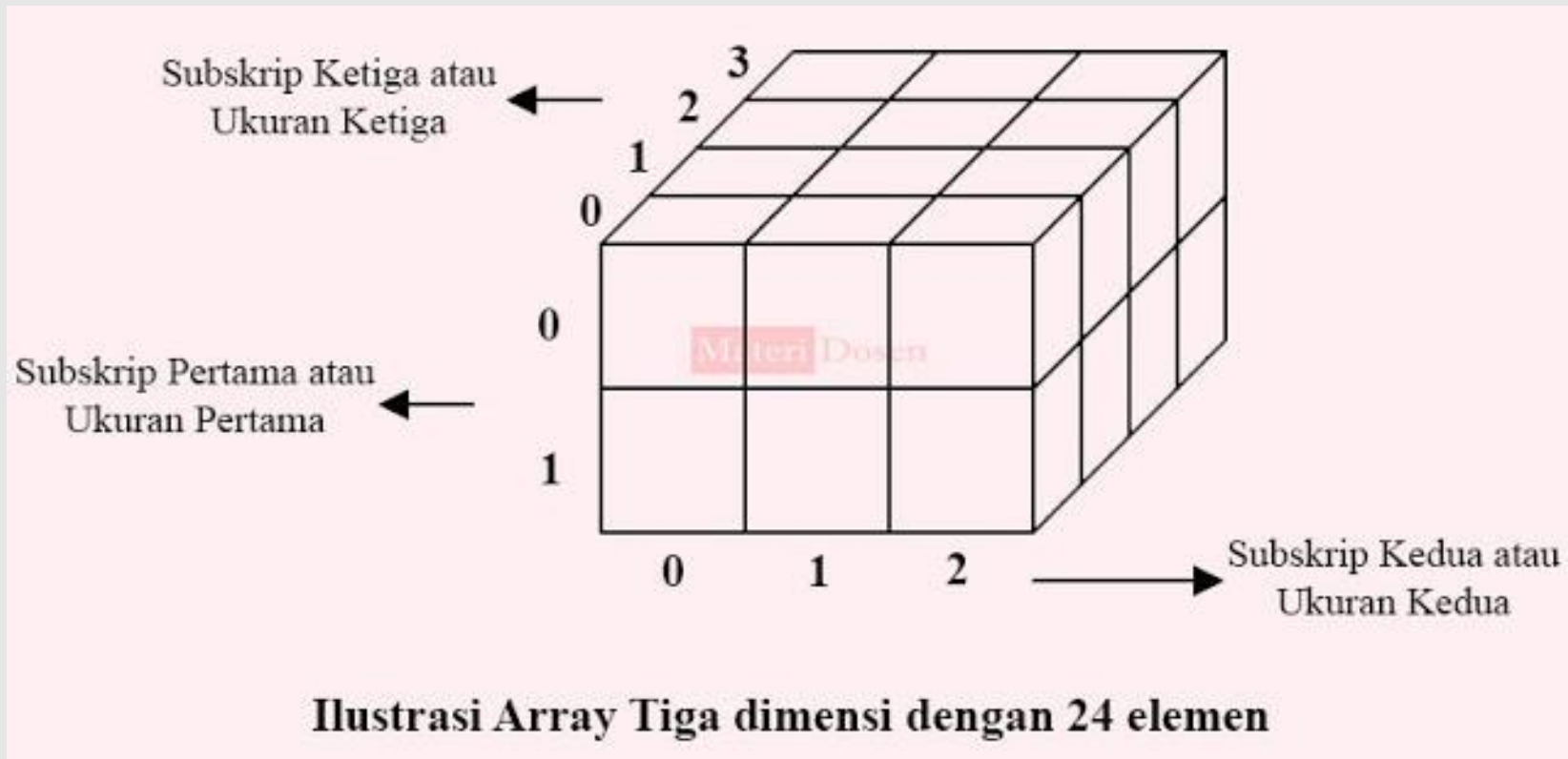
FLOWCHART KASUS
PENGUNAAN ARRAY PADA
MATRIKS



Ilustrasi array 1 dimensi

0	1	2	3	4	5	6	7	indeks
10	44	2	76	0	56	70	7	value
1d2	1d4	1d6	1d8	1da	2dc	2de	1e0	alamat

Ilustrasi array multi dimensi



Bentuk / Cara Pendeklarasian Array Multidimensi

- Bentuk Umum :

tipe_data nama_array [ukuran 1][ukuran 2] ... [ukuran N];

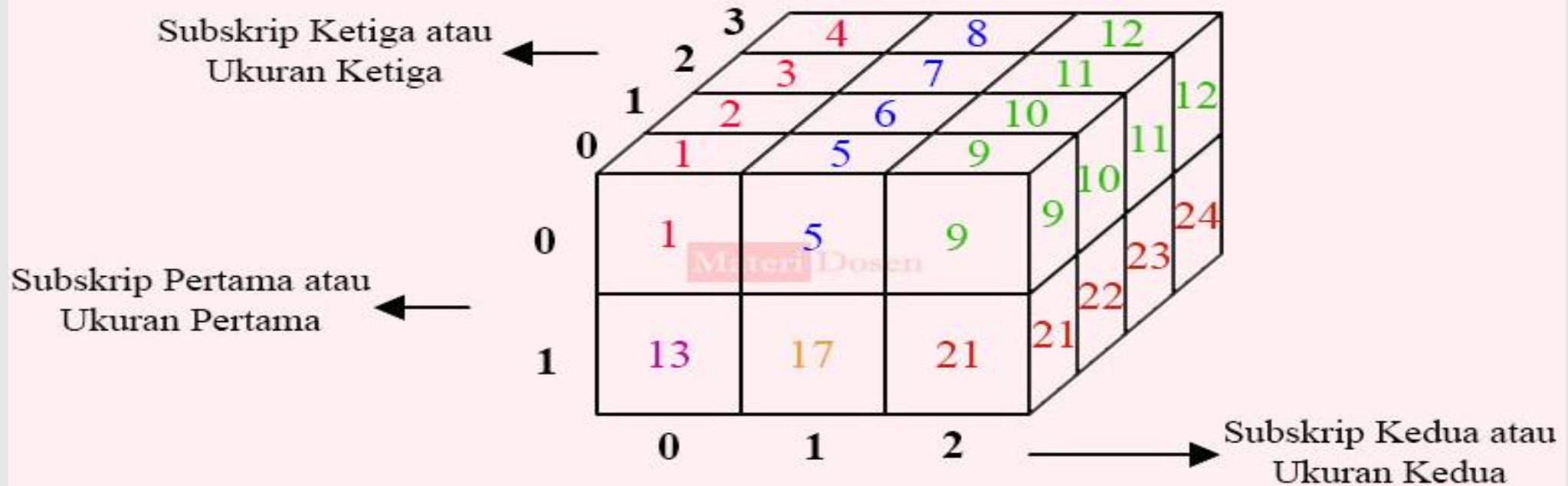
- Contoh :

- *int larik [2][3][4];*

- *int larik[2][3][4] = {{{1,2,3,4},{5,6,7,8},{9,10,11,12}} ,
{{13,14,15,16},{17,18,19,20},{21,22,23,24}}};*

Ilustrasi

`Int LARIK [2][3][4] = {{{1,2,3,4},{5,6,7,8},{9,10,11,12}}, {{13,14,15,16},{17,18,19,20},{21,22,23,24}}};`



Ilustrasi Array Tiga dimensi dengan 24 elemen

Cara Akses Array Multidimensi

- Cara yang digunakan untuk mengakses elemen array multidimensi (dimisalkan array 3 dimensi) adalah dengan menuliskan indeks ukuran pertama / subskrip pertama, indeks ukuran kedua dan indeks ukuran ketiga
- Contoh : nilai = larik[0][0][0] → nilai = 1

Contoh program- array -Multidimensi

```
#include <iostream>

using namespace std;

int main(){

    // Deklarasi array tiga dimensi dengan nama "angka"
    // Dengan jumlah ukuran pertama / subskrip pertama = 2
    // Jumlah Ukuran kedua = 3 & jumlah ukuran ketiga = 4
    int angka [2][3][4];

    // Mendeklarasi variabel untuk indeks perulangan
    int i,j,k;

    cout<<"\t=====";
    cout<<"\n\t== Contoh Array Tiga Dimensi ==\n";
    cout<<"\t=====\\n\\n";
```

Contoh program → lanjut

```
// Mengisi nilai kedalam elemen-elemen array angka
cout<<"== Masukkan elemen-elemen array angka ==\n";
for(i=0;i<2;i++){
    for(j=0;j<3;j++){
        for(k=0;k<4;k++){
            cout<<"angka indeks ke ["<<i<<"] ["<<j<<"] ["<<k<<"] "<<" = ";
            cin>>angka[i][j][k];
        }
    }
}

cout<<"\n\n=====\n";
cout<<"== Tampil nilai elemen Array ==\n";
cout<<"=====\n";
```

Contoh program → lanjut

```
//menampilkan nilai dari setiap elemen array angka
for(i=0;i<2;i++){
    for(j=0;j<3;j++){
        for(k=0;k<4;k++){
            cout<<"angka indeks ke ["<<i<<"]["<<j<<"]["<<k<<"]"<<" = "<<angka[i][j][k]<<endl;
        }
    }
}

return 0;
}
```

```

=====
== Contoh Array Tiga Dimensi ==
=====

== Masukkan elemen-elemen array angka ==
angka indeks ke [0][0][0] = 1
angka indeks ke [0][0][1] = 2
angka indeks ke [0][0][2] = 3
angka indeks ke [0][0][3] = 4
angka indeks ke [0][1][0] = 5
angka indeks ke [0][1][1] = 6
angka indeks ke [0][1][2] = 7
angka indeks ke [0][1][3] = 8
angka indeks ke [0][2][0] = 9
angka indeks ke [0][2][1] = 10
angka indeks ke [0][2][2] = 11
angka indeks ke [0][2][3] = 12
angka indeks ke [1][0][0] = 13
angka indeks ke [1][0][1] = 14
angka indeks ke [1][0][2] = 15
angka indeks ke [1][0][3] = 16
angka indeks ke [1][1][0] = 17
angka indeks ke [1][1][1] = 18
angka indeks ke [1][1][2] = 19
angka indeks ke [1][1][3] = 20
angka indeks ke [1][2][0] = 21
angka indeks ke [1][2][1] = 22
angka indeks ke [1][2][2] = 23
angka indeks ke [1][2][3] = 24

=====
== Tampil nilai elemen Array ==
=====
angka indeks ke [0][0][0] = 1
angka indeks ke [0][0][1] = 2
angka indeks ke [0][0][2] = 3
angka indeks ke [0][0][3] = 4
angka indeks ke [0][1][0] = 5
angka indeks ke [0][1][1] = 6
angka indeks ke [0][1][2] = 7
angka indeks ke [0][1][3] = 8
angka indeks ke [0][2][0] = 9
angka indeks ke [0][2][1] = 10

```

Output-
program array
multidimensi

Melewatkan Array Sebagai Argument Fungsi

- Array dapat dikirim dan dikembalikan oleh fungsi.
- Pada saat array dikirim ke dalam fungsi, nilai aktualnya dapat dimanipulasi
- Contoh

Contoh

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

void ubah(int x[2][3][4])
{
    x[1][2][3]=100;
}

int main()
{
    int nilai[2][3][4] =
    {{{1,2,3,4},{5,6,7,8},{9,10,11,12}} ,
    {{13,14,15,16},{17,18,19,20},{21,22,23,24}}};
    ubah(nilai);
    cout<<nilai[1][2][3];

    return 0;
}
```



TUGAS 1

Soal 1

- Buatlah suatu program untuk menjumlahkan bilangan bulat yang dimasukkan melalui keyboard, batas awal (i) bertipe integer, batas akhir(n) bertipe integer dan jumlah (sum) bertipe integer untuk menyimpan total bilangan bulat
- Masalah - Diketahui:
 - Input nilai melalui keyboard
 - Batas awal (i) bertipe integer
 - Batas Akhir (n) bertipe integer
 - Hasil Penjumlahan disimpan dalam variable jumlah bertipe integer
- Ditanyakan : Algoritma : Flowchart, Pseudocode, Notasi Algoritmik, dan Program Penjumlahan Bilangan Bulat

Ilustrasi – **Output yang diharapkan**

- Jumlah bilangan : 5
- **2 3 10 7 3**
- Hasil Penjumlahan : 25

Solusi – Analisis Algoritma

- Pseudocode

Solusi – Analisis Algoritma - Alternatif

- **Notasi Algoritmik**

Solusi – Analisis Algoritma - Alternatif

- **Flowchart**

Penerapan Algoritma ke Pemrograman

lanjutan

Soal 2

- Buat program untuk menyusun nomor dus air 2,4,6,...18 ke dalam rang kontainer yang memuat ukuran 3 x 3, setiap dus akan diberi nomor sebanyak jumlah dus dan ukuran kontainer.
- Masalah :
 - Penomoran genap pada setiap dus
 - Jumlah nomor sebanyak 9 dengan penomoran mulai dari 2 – 18
 - Kontainer dapat memuat Ukuran 3 x 3
- Ditanyakan : Algoritma : Flowchart, Pseudocode, Notasi Algoritmik, dan program

Ilustrasi – Output yang diharapkan

- Jumlah Dus : 9
- Nomor dus = 2 4 6 8 10 12 14 16 18
- Nomor dus 1 : 2
- Nomor dus 2 : 4
- Nomor dus 3 : 6
- Nomor dus 4 : 8
- Nomor dus 5 : 10
- Nomor dus 6 : 12
- Nomor dus 7 : 14
- Nomor dus 8 : 16
- Nomor dus 9 : 18

Solusi – Analisis Algoritma

- Pseudocode

Solusi – Analisis Algoritma - Alternatif

- **Notasi Algoritmik**

Solusi – Analisis Algoritma - Alternatif

- **Flowchart**

Penerapan Algoritma ke Pemrograman



TUGAS 2

Soal

- Buatlah suatu program untuk **mengalikan bilangan bulat yang dimasukkan melalui keyboard**, **batas awal (i)** bertipe integer, **batas akhir(n)** bertipe integer dan **hasil** bertipe **integer** untuk menyimpan hasil perkalian bilangan bulat, disertai Algoritma : Flowchart, Pseudocode, Notasi Algoritmik,

Referensi



[HTTP://WWW.CPLUSPLUS.COM/DOC/TUTORIAL/](http://www.cplusplus.com/doc/tutorial/)



MODUL PRAKTIKUM ALPRO



ABDUL KADIR, ALGORITMA DAN
PEMROGRAMAN