

## MODUL 6 : ARRAY

### LAB 6.1: TUJUAN PRAKTIKUM

*Learning outcome* untuk praktikum pemrograman dasar pada modul 6 ini adalah bagian dari LO4 khususnya mengenai array statis dan dinamis

LO4 Mahasiswa mampu menggunakan dan menganalisis array, structure, union dan tipe data dinamis dalam pemrograman

### LAB 6.2: DASAR TEORI

#### ARRAY

Array merupakan kumpulan dari nilai-nilai data bertipe sama dalam urutan tertentu yang disimpan menggunakan sebuah nama. Nilai-nilai data yang disimpan dalam suatu array disebut dengan elemen-elemen array. Elemen-elemen dalam suatu array, diorganisir sedemikian rupa dengan menggunakan indeks. Letak urutan dari elemen-elemen array ditunjukkan oleh suatu indeks. Array merupakan struktur data paling sederhana. Array dapat memiliki dimensi satu, dua atau lebih. Array berdimensi satu merepresentasikan bentuk vektor, sedangkan array berdimensi dua merepresentasikan bentuk matriks.

Array dapat bersifat statis atau dinamis. Array statis akan menyimpan jumlah elemen data yang tetap, sesuai pada saat dideklarasikan. Sedangkan array dinamis jumlah elemennya tidak tetap dan dapat berkembang.

#### ARRAY STATIS SATU DIMENSI

Array satu dimensi adalah kumpulan dari nilai-nilai data bertipe sama dalam urutan tertentu yang disimpan menggunakan sebuah nama yang sama dan letak urutan dari elemen-elemen array hanya ditunjukkan oleh satu jenis subscript / indeks saja. Dalam bahasa C++ indeks dimulai dengan 0 (nol). Ukuran dari array satu dimensi statis biasanya sudah ditentukan pada saat deklarasi array. Misal suatu array satu dimensi bernama A memiliki ukuran 5 maka akan ditulis dengan A[5]. Nantinya elemen array tersebut akan dimulai dengan A[0] dan diakhiri dengan A[4]. Nilai elemen array statis akan disimpan dalam lokasi atau alamat memori yang berurutan.

Indeks	Elemen	Alamat (Alamat dalam bentuk bilangan heksadesimal)
0	A[0] = 100	0x7ffdb0f72130
1	A [1] = 101	0x7ffdb0f72134
2	A [2] = 102	0x7ffdb0f72138
3	A [3] = 103	0x7ffdb0f7213c
4	A [4] = 104	0x7ffdb0f72140

### Pendeklarasian array satu dimensi

```
tipe Nama_array[ukuran_array];
```

#### Contoh

```
float Nilai_mahasiswa[5];
```

### Mengisi nilai pada elemen array satu dimensi

Dapat dilakukan ketika deklasi array atau sesudahnya dengan mengisi satu persatu nilai elemen array.

```
float Nilai_mahasiswa[5]= {79.8, 80.5, 94.2, 53.0, 66.6};
```

atau

```
Nilai_mahasiswa[0] = 79.8;  
Nilai_mahasiswa[1] = 80.5;  
Nilai_mahasiswa[2] = 94.2;  
Nilai_mahasiswa[3] = 53.0;  
Nilai_mahasiswa[4] = 66.6;
```

### Menampilkan nilai elemen elemen array satu dimensi :

```
cout <<Nilai_mahasiswa[0]<< endl;  
cout <<Nilai_mahasiswa[1]<< endl;  
cout <<Nilai_mahasiswa[2]<< endl;  
cout <<Nilai_mahasiswa[3]<< endl;  
cout <<Nilai_mahasiswa[4]<< endl;
```

---

## ARRAY STATIS DUA DIMENSI

Array dua dimensi adalah kumpulan dari nilai-nilai data bertipe sama dalam urutan tertentu yang disimpan menggunakan sebuah nama dan letak urutan dari elemen-elemen array ditunjukkan oleh dua jenis subscript / indeks. Kedua indeks tersebut merepresentasikan baris dan kolom pada suatu matriks atau tabel. Ukuran dari array dua dimensi statis biasanya sudah ditentukan pada saat deklarasi array. Misal suatu array dua dimensi bernama A memiliki ukuran baris 5 dan kolom juga 510 maka akan ditulis dengan A[5][5]. Nantinya elemen array tersebut akan dimulai dengan A[0][0] dan diakhiri dengan A[4][4]. Nilai elemen array dua dimensi statis akan disimpan dalam lokasi atau alamat memori yang berurutan,urut per baris.

A[Baris][Kolom]	A[0][0]	A[0][1]	A[0][2]	A[0][3]	A[0][4]
	A[1][0]	A[1][1]	A[1][2]	A[1][3]	A[1][4]
	A[2][0]	A[2][1]	A[2][2]	A[2][3]	A[2][4]
	A[3][0]	A[3][1]	A[3][2]	A[3][3]	A[3][4]
	A[4][0]	A[4][1]	A[4][2]	A[4][3]	A[4][4]

#### Pendeklarasian array dua dimensi

```
tipe Nama_array[ukuran_baris_array][ukuran_kolom_array];
```

#### Contoh

```
int Jumlah_sks_mahasiswa[3][3];
```

#### Mengisi nilai pada elemen array dua dimensi

Dapat dilakukan ketika deklasi array atau sesudahnya dengan mengisi satu persatu nilai elemen array.

```
int Jumlah_sks_mahasiswa[3][3]= {19,20,24,38,40,48,76,80,96};
```

atau

```
Jumlah_sks_mahasiswa[0][0] = 19;  
Jumlah_sks_mahasiswa[0][1] = 20;  
Jumlah_sks_mahasiswa[0][2] = 24;  
Jumlah_sks_mahasiswa[1][0] = 38;  
Jumlah_sks_mahasiswa[1][1] = 40;  
Jumlah_sks_mahasiswa[1][2] = 48;  
Jumlah_sks_mahasiswa[2][0] = 76;  
Jumlah_sks_mahasiswa[2][1] = 80;  
Jumlah_sks_mahasiswa[2][2] = 96;
```

#### Menampilkan nilai elemen elemen array dua dimensi :

```
cout <<Jumlah_sks_mahasiswa[0][0]<< endl;  
cout <<Jumlah_sks_mahasiswa[0][1]<< endl;  
cout <<Jumlah_sks_mahasiswa[0][2]<< endl;  
cout <<Jumlah_sks_mahasiswa[1][0]<< endl;
```

```
cout <<Jumlah_sks_mahasiswa[1][1]<< endl;
cout <<Jumlah_sks_mahasiswa[1][2]<< endl;
cout <<Jumlah_sks_mahasiswa[2][0]<< endl;
cout <<Jumlah_sks_mahasiswa[2][1]<< endl;
cout <<Jumlah_sks_mahasiswa[2][2]<< endl;
```

---

## ARRAY DINAMIS

Array dinamis jumlah elemennya tidak tetap dan dapat berkembang. Array dinamis dibangun dengan memanfaatkan data pointer. Tidak seperti array statis, maka ukuran dari array dinamis tidak dibatasi oleh ukuran array di awal pemrograman.

Contoh ;

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

#define N 10

// Dynamically allocate memory for 1D Array in C++
int main()
{
    // dynamically allocate memory of size `N`
    int* A = new int[N];

    // assign values to the allocated memory
    for (int i = 0; i < N; i++) {
        A[i] = i + 1;
    }

    // print the 1D array
    for (int i = 0; i < N; i++) {
        std::cout << A[i] << " ";    // or *(A + i)
    }

    // deallocate memory
    delete[] A;

    return 0;
}
```

Output :

1 2 3 4 5 6 7 8 9 10

---

## STRUCTURE DINAMIS

Pointer adalah suatu variabel khusus yang berisikan alamat memori dari suatu data, tidak berisikan data itu sendiri atau dapat juga dipahami bahwa pointer adalah variabel yang berisikan alamat dari suatu data. Pointer sangat sering digunakan dalam program-program bahasa C++ misalkan untuk membentuk tipe data structure yang dinamis, karena dapat memberikan efisiensi khususnya dalam pemakaian memori dalam program.

Contoh :

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

int main()
{
    struct date
    {
        short int dd, mm, yy;
    }join_date = {25, 04, 2022};

    date *date_ptr;
    date_ptr = &join_date;
    cout<<"Printing the structure elements using the structure
variable\n";
    cout<<"dd = "<<join_date.dd<<", mm = "<<join_date.mm<<", yy =
"<<join_date.yy<<"\n";
    cout<<"\nPrinting the structure elements using the structure
pointer\n";
    cout<<"dd = "<<date_ptr->dd<<", mm = "<<date_ptr->mm<<", yy =
"<<date_ptr->yy<<"\n";

    return 0;
}
```

Output:

Printing the structure elements using the structure variable  
dd = 25, mm =04, yy = 2022

Printing the structure elements using the structure pointer  
dd = 25 mm = 04, yy = 2022

## LAB 6.3: LATIHAN

## PERCOBAAN 1 : ARRAY STATIS SATU DIMENSI

Berikut ini adalah sebuah program dalam bahasa C++. Salinlah program berikut menggunakan text editor dan simpan dengan nama larik.cpp. Kemudian *compile* dan *run* (eksekusi) program tersebut.

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

int main()
{
    int larik[5] = {10, 20, 30, 40, 50};
    int i;
    cout<<"Elemen larik\t\t"<<"Nilai elemen larik\n";
    for(i=0; i<5; i++)
    {
        cout<<"larik["<<i<<"]\t\t\t"<<larik[i]<<"\n";
    }
    return 0;
}
```

1. Apakah keluaran dari program di atas tersebut? Jelaskan

## PERCOBAAN 2 : ARRAY STATIS SATU DIMENSI

Perhatikan program di bawah ini, kemudian salin (ketik ulang, jangan copy-paste), *compile*, dan eksekusi program tersebut.

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

int main () {

    int A[5], i, j; // deklarasi array A bertipe integer dan memiliki ukuran 5
    (memiliki 5 elemen)

    // inialisasi indeks dari array A dari indeks 0 nilai array dan indeks
    selanjutnya bertambah dengan 1
    for ( int i = 0; i < 5; i++ ) {

        // mengeset elemen dari array A[0] bernilai 100 dan nilai array
        selanjutnya bertambah dengan 1
        A[i] = i + 100;
    }
    cout << "Indeks\t\t"<< "Nilai Elemen\t" << "Alamat"<<endl;

    // menampilkan indeks dan nilai dari tiap elemen array A serta lokasi
    alamat memorinya
    for ( int j = 0; j < 5; j++ ) {
        cout << j <<"\t\t\t"<<A[ j ]<<"\t\t\t\t"<<&A[j] << endl;
    }
}
```

1. Berapa indeks array A yang paling besar ?
2. Berapa ukuran tipe data dari elemen array A ?
3. Berapa bertambah ukuran memori yang menyimpan tiap nilai elemen dari array A ?

### PERCOBAAN 3 : ARRAY STATIS SATU DIMENSI UNTUK DATA *STRUCTURE*

Perhatikan program di bawah ini, kemudian salin (ketik ulang, jangan copy-paste), *compile*, dan eksekusi program tersebut.

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

typedef struct Person
{
    string name;
    int NIM;
    int age;
}Student;

int main()
{
    int i,n;
    cout<< "Masukkan jumlah data mahasiswa :";
    cin>> n;
    cout<<endl;

    Student s_array[n];
    cout<<"Masukkan Data Mahasiswa \n";
    for(i=0;i<n;i++){
        cout<< "Masukkan Nama mahasiswa :\n";
        cin>>s_array[i].name;
        cout<< "Masukkan NIM mahasiswa :\n";
        cin>>s_array[i].NIM;
        cout<< "Masukkan Umur mahasiswa :\n";
        cin>>s_array[i].age;
    }

    cout<<"\nDaftar Mahasiswa\n";
    cout<<"\nNo.\t\tNama\t\tNIM\t\tUmur\n";
    for(i=0;i<n;i++){
        cout<<i+1<<"\t\t"<<s_array[i].name <<"\t\t\t"<<s_array[i].NIM
<<"\t\t"<<s_array[i].age <<endl;
    }
}
```

1. Apakah keluaran dari program di atas ?
2. Apakah ada kesalahan ? jika ada silahkan diperbaiki sapai menghasilkan keluaran yang sesuai

## PERCOBAAN 4 : ARRAY STATIS DUA DIMENSI

Pada percobaan ini, salinlah program berikut menggunakan *text editor* dan simpan dengan nama tertentu. Kemudian *compile* dan eksekusi program tersebut.

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

int main () {

    int A[5][5], i, j, k,l; // deklarasi array A bertipe integer dan
    memiliki ukuran baris 5 dan kolom 5
    // inisialisasi indeks dari array Adimulai dari indeks [0][0]
    for ( int i = 0; i < 5; i++ )
        {
            for ( int j = 0; j < 5; j++ )

                // set nilai elemen dimulai dari array A[0]0], baris ke 0
                akan diset ke nilai 100 selanjutnya kenaikan baris akan bertambah 1
                A[i][j] = i+100;
            }

        cout << "Indeks\t\t"<< "Nilai Elemen\t" << "Alamat"<<endl;

        // menampilkan indeks dan nilai dari tiap elemen array A serta lokasi
        alamat memorinya
        for ( int k = 0; k < 5; k++ )
            {
                for ( int l = 0; l < 5; l++ )

                    cout << "["<<k<<"]"<< "["<<l<<"]"<< "\t\t"<<A[k][l]<< "\t\t\t\t"<<&A[k][l]
                    << endl;
            }
    }
```

1. Apakah keluaran dari program di atas ?
2. Mengapa diperlukan dua buah perintah “for” untuk iterasinya ?
3. Modifikasi program di atas dengan mengganti perintah iterasi “for” dengan perintah “while”
4. Modifikasi program di atas dengan mengganti perintah iterasi “for” dengan perintah “do while”

## PERCOBAAN 5 : ARRAY STATIS DUA DIMENSI

Salin program di bawah ini, *compile*, dan eksekusi !

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

int main( )
{
    int s[3][3];
    int i, j;
    cout<<"\nMasukkan sks Mahasiswa untuk 3 Angkatan yang berbeda \n";
    for(i=0;i<3;i++)
```



```

{
    cout<<"\n Angkatan 202"<<i<<"\t";
    for(j=0;j<3;j++)
    {
        cout<<"\nsks Mahasiswa ["<<i<<"] ["<<j<<"]=  ";
        cin>>s[i][j];
    }
}
cout<<endl;
cout<<"\nTampilkan data sks Mahasiswa 3 Angkatan\n";
cout<<"\n Angkatan\t\t"<<"sks";
for(i=0;i<3;i++)
{
    cout<<"\n 202"<<i<<"\t";
    for(j=0;j<3;j++)
    {
        cout<<"\t"<<s[i][j];
    }
    cout<<endl;
}
}

```

1. Apakah keluaran dari program di atas ? Jelaskan
2. Modifikasi program di atas sehingga dapat mengeluarkan data sks mahasiswa untuk 6 angkatan dimulai dari angkatan 1990 dan tiap angkatan ada 4 data sks mahasiswa.
3. Gantilah perintah "for" dengan "while" untuk perintah pada poin kedua

## PERCOBAAN 6 : STRUCTURE DINAMIS

Program di bawah ini adalah program structure yang memanfaatkan pointer. Salin, *compile*, dan eksekusi program di berikut ini.

```

#include <iostream>
using namespace std;

struct rectangular {
    int length;
    int width;
};

int main() {
    rectangular *ptr, r;
    int area;
    ptr = &r;

    cout << "Enter length: "<<endl;
    cin >> (*ptr).length;
    cout << "Enter width"<<endl;
    cin >> (*ptr).width;
    area = (*ptr).length * (*ptr).width;
    cout << "Displaying information Rectangular" << endl;
    cout << "Length = " << (*ptr).length <<" \tWidth " << (*ptr).width
    <<endl;
    cout << "Area = " << area<<endl;
}

```

```
    return 0;
}
```

1. Apakah keluaran dari program di atas ?
2. Apa beda antara structure dinamis dan statis ?

---

#### PERCOBAAN 6 : STRUCTURE DINAMIS

Program di bawah ini adalah program array dinamis yang memanfaatkan pointer. Salin, *compile*, dan eksekusi program di berikut ini.

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

int main() {
    int i,n;
    cout<<"Enter total number of elements:"<<"\n";
    cin>>n;
    int *a = new int(n);
    cout<<"Enter " <<n<<" elements"<<endl;
    for(i = 0;i<n;i++) {
        cin>>a[i];
    }
    cout<<"Entered elements are: ";
    for(i = 0;i<n;i++) {
        cout<<a[i]<<" ";
    }
    cout<<endl;
    delete (a);
    return 0;
}
```

1. Jelaskan apa keluaran dari program tersebut
2. Apa perbedaan array dinamis dengan statis ?

---

#### PERCOBAAN 7 : ARRAY DINAMIS

Program di bawah ini adalah program yang memanfaatkan pointer. Salin, *compile*, dan eksekusi program di berikut ini.

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

#define N 10

// Dynamically allocate memory for 1D Array in C++
int main()
{
    // dynamically allocate memory of size `N`
```

```

int* A = new int[N];

// assign values to the allocated memory
for (int i = 0; i < N; i++) {
    A[i] = i + 1;
}

// print the 1D array
for (int i = 0; i < N; i++) {
    std::cout << A[i] << " ";    // or *(A + i)
}

// deallocate memory
delete[] A;

return 0;
}

```

- Jelaskan keluaran program di atas ?
- Apa bedanya dengan array statis ?

#### LAB 6.4: CEK POIN 1

- 1) Perhatikan program di bawah ini, kemudian salin (ketik ulang, jangan copy-paste), *compile*, dan eksekusi program tersebut. Jika terdapat syntax error atau error karena kesalahan dalam hal penggunaan tipe data, operator, ataupun kesalahan logika, perbaiki hingga dapat dikompilasi dan dijalankan sehingga menghasilkan luaran seperti yang diharapkan.

```

#include <iostream>
using namespace std;

int main()
{
    int arr[10];
    int i;
    int sum=0, avg=0;
    cout<<"Masukkan 10 Elemen Array bertipe integer";
    for(i=0; i<10; i++)
    {
        cin>>arr[i];
        sum = sum + arr[i];
    }
    cout<<"\nElemen Array adalah : \n";
    for(i=0; i<10; i++)
    {
        cout<<arr[i]<<" ";
    }
    cout<<"\n\nHasil penjumlahan dari semua elemen array adalah : "<<sum;
    avg = sum/10;
    cout<<"\nRerata elemen array adalah : "<<avg;
    return 0;
}

```

- 2) Perhatikan program di bawah ini, kemudian salin (ketik ulang, jangan copy-paste), *compile*, dan eksekusi program tersebut. Jika terdapat syntax error atau error karena kesalahan dalam hal penggunaan tipe data, operator, ataupun kesalahan logika, perbaiki hingga dapat dikompilasi dan dijalankan sehingga menghasilkan luaran seperti yang diharapkan.

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

const int size=3;

struct employee {
    string name;
    int emp_no;
    float salary;
} ;

int main () {

    emp[size];

    cout<<"Masukkan Data Pegawai"<<endl;

    for(i=0;i<size;i++) {
        cout<<" Pegawai Nomor "<<i+1<<endl;
        cout<<" Nama:"<<endl;
        cin>>emp[i].name;
        cout<<" Kode: "<<endl;
        cin>>emp[i].emp_no; //accessing members
        cout<<" Gaji:"<<endl;
        cin>>emp[i].salary;
    }

    cout<<"\n";
    cout<<"Data Pegawai\n";
    cout<<"No.\tNama\tKode\tGaji\n";
    for(i=0;i<size;i++) {

        cout<<i+1<<"\t"<<emp[i].name<<"\t"<<emp[i].emp_no<<"\t\t"<<emp[i].salary<
        <"\n";
    }
}
```

- 3) Apakah keluaran dari program di bawah ini, jelaskan!

```
#include <iostream>
using namespace std;
typedef int* IntPtr;
int main( )
{
    IntPtr p;
    int a[10];
    int index;
    for (index = 0; index < 10; index++)
        a[index] = index;
    p = a;
```

```
for (index = 0; index < 10; index++)
cout << p[index] << " ";
cout << endl;
for (index = 0; index < 10; index++)
p[index] = p[index] + 1;
for (index = 0; index < 10; index++)
cout << a[index] << " ";
cout << endl;
return 0;
}
```

### LAB 1.5: CEK POIN 2

1. Berikut adalah program untuk apa ? Jelaskan. Modifikasilah untuk masukan atau input array yang bisa diisi oleh user secara interaktif..

```
#include<iostream>
using namespace std;
int main ()
{
    int myarray[5] = {12,4,3,1,15};
    cout<<"\nInput list is \n";
    for(int i=0;i<5;i++)
    {
        cout <<myarray[i]<<"\t";
    }
    for(int k=1; k<5; k++)
    {
        int temp = myarray[k];
        int j= k-1;
        while(j>=0 && temp <= myarray[j])
        {
            myarray[j+1] = myarray[j];
            j = j-1;
        }
        myarray[j+1] = temp;
    }
    cout<<"\nOutput list is \n";
    for(int i=0;i<5;i++)
    {
        cout <<myarray[i]<<"\t";
    }
}
```

2. Berikut adalah program untuk mengisi dan menampilkan array structure data student, juga menampilkan data umur yang tertua. Jika akan ditambahkan untuk menampilkan nama student yang memiliki umur tertua tersebut, bagaimana caranya ? Modifikasi program tersebut supaya dapat menampilkan hal tersebut dengan benar.

```

#include<iostream>
using namespace std;
// struct declaration outside main()
struct
{
    int roll,age;
    string name;
}vr[4];

int main()
{
    int i,j,k, max;
    for(i=0; i<4; i++)
    {
        cout<<"Enter "<<i+1<<" Student Record\n";
        cout<<"Enter Roll no: "<<endl;
        cin>>vr[i].roll;
        cout<<"Enter age : "<<endl;
        cin>>vr[i].age;
        cout<<"Enter Name : "<<endl;
        cin>>vr[i].name;

    }
    cout<<endl;
    for(j=0; j<4; j++)
    {
        cout<<"\nDisplay "<<j+1<<" Student Record";
        cout<<"\nRoll no.: "<<vr[j].roll;
        cout<<"\nAge : "<<vr[j].age;
        cout<<"\nName : "<<vr[j].name<<endl;

    }
    max = vr[0].age;
    for (k = 1; k < 4; k++)

    {
        if (max<vr[k].age)
            max=vr[k].age;
    }
    cout << "\nThe oldest age's student is : " <<max;
    return 0;
}

```

3. Berikut adalah program penghitung jumlah dan rerata dari 3 buah masukan bilangan bertipe double. Program mengimplementasikan suatu program modular yang terdiri dari 2 fungsi user definition function, yaitu fungsi add dan avg. Modifikasilah program tersebut sehingga masukannya menjadi array bertipe double.

```

#include<iostream>
using namespace std;

// declare functions
double add(double num1, double num2, double num3);

```

```
double avg(double sum, int n);

// main function
int main()
{
    // declare variables
    double num1, num2, num3;
    double sum, average;

    // take input from end-user
    cout << "Enter three Numbers :: ";
    cin >> num1 >> num2 >> num3;

    // calculate sum value
    sum = add(num1 , num2, num3);

    // calculate average value
    average = avg(sum, 3);

    // display result
    cout << "Sum = " << sum << endl;
    cout << "Average = " << average << endl;

    return 0;
}

// function to add three numbers
double add(double num1, double num2, double num3)
{
    return num1+num2+num3;
}

// function to calculate average of n number
double avg(double sum, int n)
{
    return sum / n;
}
```

4. Berikut adalah program penghitung jumlah dan rerata dari 3 buah masukan bilangan bertipe double. Program mengimplementasikan suatu program modular yang terdiri dari 2 *user defined function*, yaitu fungsi add dan avg. Modifikasilah program tersebut sehingga masukannya menjadi array bertipe double.

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

// declare functions
double add(double num1, double num2, double num3);
double avg(double sum, int n);

// main function
int main()
{
```

```
// declare variables
double num1, num2, num3;
double sum, average;

// take input from end-user
cout << "Enter three Numbers :: ";
cin >> num1 >> num2 >> num3;

// calculate sum value
sum = add(num1 , num2, num3);

// calculate average value
average = avg(sum, 3);

// display result
cout << "Sum = " << sum << endl;
cout << "Average = " << average << endl;

return 0;
}

// function to add three numbers
double add(double num1, double num2, double num3)
{
    return num1+num2+num3;
}

// function to calculate average of n number
double avg(double sum, int n)
{
    return sum / n;
}
```

5. Buatlah program dengan menggunakan prinsip array dua dimensi, untuk menampilkan matriks diagonal dan matriks segitiga bawah yang isinya huruf "X" apa ?

### LAB 1.6: CEK POIN 3

1. Jelaskan maksud dan keluaran program di bawah. Bagaimana cara menyempurnakan program supaya ketika masukan pada penggalan program ini tidak ditemukan, maka tidak terjadi error ?  
`cout<<"Accessing element at index. Enter the index : "; cin>>l;`  
 Modifikasilah menjadi program modular dengan 3 *user defined function*. Masing-masing fungsi digunakan untuk memecah tugas yang ada diprogram aslinya. Tetap gunakan tipe data array seperti pada program aslinya.

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



```

int n,k,l,ans=-1;
cout<<"Enter size of array"<<endl;
cin>>n;
int arr[n];
cout<<"Enter elements of array"<<endl;
for(int i=0;i<n;i++)
{
    cin>>arr[i];
}

cout<<"Accessing element at index. Enter the index : ";
cin>>l;
cout<<endl;
cout<<"\nElement present at index "<<l<<" is "<<arr[l]<<endl;

cout<<"Enter element to be searched : "<<endl;
cin>>k;
for(int i=0;i<n;i++)
{
    if(arr[i]==k)
    {
        ans=i;
        break;
    }
}
if(ans!=-1)
cout<<"The element "<<k<<" is present at index "<<ans;
else
cout<<"The element "<<k<<" is not there in the array";
return 0;
}

```

2. Modifikasi program di bawah menjadi program modular yang memiliki 2 3 *user defined function*. Fungsi yang pertama untuk meng-inputkan array satu dimensi, sedangkan fungsi yang kedua untuk mengeluarkan output seperti pada program aslinya.

```

#include<iostream>
using namespace std;
int main ()
{
    int myarray[5] = {12,4,3,1,15};
    cout<<"\nInput list is \n";
    for(int i=0;i<5;i++)
    {
        cout <<myarray[i]<<"\t";
    }
    for(int k=1; k<5; k++)
    {
        int temp = myarray[k];
        int j= k-1;
        while(j>=0 && temp <= myarray[j])
        {
            myarray[j+1] = myarray[j];
            j = j-1;
        }
        myarray[j+1] = temp;
    }
    cout<<"\nOutput list is \n";
}

```

```
for(int i=0;i<5;i++)
{
    cout <<myarray[i]<<"\t";
}
}
```

3. Buatlah program untuk menampilkan elemen suatu matriks (array dua dimensi bertipe integer) dan juga untuk mencari dan menampilkan nilai terkecil dari matriks tersebut. Tampilkan juga nilai terkecil tersebut terletak pada indeks berapa ?
4. Apakah keluaran program di bawah ini ? Program ini menggunakan jenis tipe data apa ? Bagaimana cara menambah beberapa data di program ini ?

```
#include <iostream>#include <iostream>
using namespace std;
struct Employee
{
    string Name;
    int Age;
};
int main()
{
    Employee* DynArray;
    DynArray = new (nothrow) Employee[3];
    DynArray[0].Name = "Agus";
    DynArray[0].Age = 33;
    DynArray[1].Name = "Sigit";
    DynArray[1].Age = 26;
    DynArray[2].Name = "Sulis";
    DynArray[2].Age = 52;
    cout << "Displaying the Array Content" << endl;
    for (int i = 0; i < 3; i++)
    {
        cout << "Name: " << DynArray[i].Name <<
            "\tAge: " << DynArray[i].Age << endl;
    }
    delete[] DynArray;
    return 0;
}
```

### LAB 1.7: REFERENSI

Di bawah ini adalah referensi online dari Modul 1. Apabila mengalami kesulitan dalam memahami materi, silahkan buka referensi di bawah ini :

1. [https://www.tutorialspoint.com/cplusplus/cpp\\_arrays.htm](https://www.tutorialspoint.com/cplusplus/cpp_arrays.htm)

### LAB 1.8: SEKILAS MATERI UNIT BERIKUTNYA

Materi praktikum pemrograman dasar unit selanjutnya adalah responsi. Akan diadakan evaluasi untuk materi materi yang sudah diberikan.

