

LAPORAN PRAKTIKUM STRUKTUR DATA

LAPORAN AWAL PRAKTIKUM

Pertemuan ke-02

Array



Disusun Oleh:

Nama Lengkap : Nova Ardiansyah

NIM : 211011401309

Kelas : 04-TPLE008

TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PAMULANG

Jl. Raya Puspitek No. 11 Buaran, Serpong Telp. (021) 7412566, Fax. (021) 7412566
Tangerang Selatan - Banten

A. RANGKUMAN MATERI

Pada bahasa pemrograman, kita seringkali membutuhkan sebuah struktur data yang dapat menyimpan sejumlah elemen dengan jenis data yang sama. Salah satu struktur data yang dapat digunakan untuk memenuhi kebutuhan tersebut adalah array. Array merupakan sebuah variabel yang dapat menyimpan beberapa nilai dengan jenis data yang sama dan disimpan secara berurutan di dalam memori komputer.

Array sendiri terdiri dari beberapa elemen yang dapat diakses menggunakan indeks array. Indeks array dimulai dari angka 0 dan terus bertambah sesuai dengan jumlah elemen array. Setiap elemen dalam array dapat diakses secara langsung menggunakan indeksnya. Array sangat berguna dalam pengolahan data, terutama ketika kita perlu memproses sejumlah data yang besar dengan tipe data yang sama.

Perbedaan antara array dengan variabel biasa adalah pada array, kita dapat menyimpan beberapa nilai dengan jenis data yang sama di dalam satu variabel, sedangkan variabel biasa hanya dapat menyimpan satu nilai pada satu waktu.

Array adalah struktur data yang memungkinkan untuk menyimpan sejumlah elemen dengan tipe data yang sama dan diurutkan dalam suatu wadah atau variabel. Array dibedakan menjadi dua jenis, yaitu array satu dimensi dan array dua dimensi.

Array satu dimensi terdiri dari barisan atau deretan elemen dengan tipe data yang sama. Setiap elemen dalam array dapat diakses menggunakan indeks array yang dimulai dari 0 hingga jumlah elemen dikurangi satu

```
int array1D[5] = {1, 2, 3, 4, 5};
```

Sedangkan array dua dimensi terdiri dari elemen-elemen yang disusun dalam bentuk matriks atau tabel berbaris dan berkolom dengan tipe data yang sama. Setiap elemen dalam array dua dimensi dapat diakses menggunakan indeks array yang terdiri dari dua nilai, yaitu indeks baris dan indeks kolom. Contohnya, sebuah array dua dimensi dengan

tiga baris dan tiga kolom yang terdiri dari bilangan bulat dapat dideklarasikan sebagai berikut:

```
int array2D[3][3] = {  
    {1, 2, 3},  
    {4, 5, 6},  
    {7, 8, 9}  
};
```

Perbedaan antara array satu dimensi dan array dua dimensi terletak pada dimensi array tersebut. Array satu dimensi hanya memiliki satu dimensi atau satu baris elemen, sedangkan array dua dimensi memiliki dua dimensi atau baris dan kolom. Array dua dimensi dapat digunakan untuk merepresentasikan data dalam bentuk matriks atau tabel seperti data statistik atau data permainan seperti game tic-tac-toe.

Ketika bekerja dengan array, penting untuk memahami konsep indeks dan batas array untuk mencegah terjadinya kesalahan seperti akses di luar batas array atau indeks yang salah. Array juga dapat digunakan dengan berbagai operasi seperti pengisian nilai, pencarian nilai, pengurutan nilai, dan operasi matematis lainnya.

B. TUGAS PENDAHULUAN

1. Apa yang dimaksud dengan Array!

Jawab :

Array adalah kumpulan elemen yang memiliki tipe data yang sama dan ditempatkan di lokasi penyimpanan yang berurutan. Elemen-elemen ini diakses menggunakan indeks array mulai dari 0.

2. Jelaskan perbedaan Array dengan Variabel biasa!

Jawab :

Perbedaan antara array dan variabel reguler adalah bahwa dalam array kita dapat menyimpan beberapa nilai dari tipe data yang sama dalam satu variabel sedangkan variabel reguler hanya dapat menyimpan satu nilai dalam satu waktu.

3. Jelaskan perbedaan Array Satu Dimensi, Dua Dimensi, dan Tiga Dimensi!

Jawab :

Array satu dimensi hanya memiliki satu baris elemen yang disimpan dalam variabel array, sedangkan array dua dimensi terdiri dari banyak baris dan kolom yang disimpan dalam variabel array yang disusun dalam array. Array tiga dimensi terdiri dari banyak baris, kolom, dan level elemen yang disimpan dalam variabel array dan disusun dalam kubus atau blok.

4. Buatlah contoh program sederhana menggunakan Array Tiga Dimensi!

Jawab :

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int array3D[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}
        }
    };

    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 3; j++) {
            for (int k = 0; k < 4; k++) {
                cout << array3D[i][j][k] << " ";
            }
            cout << endl;
        }
        cout << endl;
    }

    return 0;
}
```

```
Nama      : Nova Ardiansyah  
NIM       : 211011401309  
=====
```

```
1 2 3 4  
5 6 7 8  
9 10 11 12  
  
13 14 15 16  
17 18 19 20  
21 22 23 24
```

C. TUGAS PRAKTIKUM

- Lat2_1.cpp

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

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int Nilai[10] = {8, 12, 20, 15, 7, 5, 6, 4, 32, 3};
    int i, jumlah = 0;
    float Rata_Rata;

    cout << "\nDeretan Bilangan : ";
    // * Menghitung jumlah
    for (i = 0; i < 10; i++)
    {
        jumlah += Nilai[i];
        Rata_Rata = (float)jumlah / 10;

        // * Mencetak Elemnet Array
        cout << Nilai[i] << " ";
    }

    cout << "\nJumlah \t\t: " << jumlah;
    cout << "\nRata-Rata \t: " << Rata_Rata;
}
```

```
Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Deretan Bilangan: 8 12 20 15 7 5 6 4 32 3
Jumlah      : 112
Rata-Rata   : 11.2
```

- Lat2_2.cpp

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

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int Nilai[4];
    int i, Jumlah = 0;
    float Rata_Rata;

    // * Membaca dan Menghitung Jumlah
    for (i = 0; i < 4; i++)
    {
        cout << "Masukkan element ke-" << i << " = ";
    }
```

```

        cin >> Nilai[i];
        Jumlah += Nilai[i];
    }

    // * Menghitung Rata-Rata
    Rata_Rata = (float)Jumlah / 4;

    // * Mencetak Hasil
    cout << "\n\nDeretan Bilangan : ";
    for (i = 0; i < 4; i++)
    {
        cout << Nilai[i] << " ";
    }

    cout << "\nJumlah \t\t: " << Jumlah;
    cout << "\nRata-Rata \t: " << Rata_Rata;
}

```

```

Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Masukkan element ke-0 = 21
Masukkan element ke-1 = 12
Masukkan element ke-2 = 11
Masukkan element ke-3 = 34

Deretan Bilangan: 21 12 11 34
Jumlah          : 78
Rata-Rata       : 19.5

```

- Lat2_3.cpp

```

#include <iostream>
using namespace std;

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int Nilai[5];
    int i, Terbesar;

    for (i = 0; i < 5; i++)
    {
        cout << "Masukkan nilai ke-" << i + 1 << " = ";
        cin >> Nilai[i];
    }

    Terbesar = Nilai[0];
    for (i = 0; i < 5; i++)
    {
        if (Nilai[i] > Terbesar)
        {
            Terbesar = Nilai[i];
        }
    }
}

```

```

}

cout << "\nNilai terbesar adalah " << Terbesar << endl;
}

```

```

Nama      : Nova Ardiansyah
NIM       : 211011401309

```

```

=====

Masukkan nilai ke-1 = 21
Masukkan nilai ke-2 = 3
Masukkan nilai ke-3 = 45
Masukkan nilai ke-4 = 10
Masukkan nilai ke-5 = 7

Nilai terbesar adalah 45

```

- Lat2_4.cpp

```

#include <iostream>
using namespace std;

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int Nilai[5];
    int Posisi[5];
    int i, Bilangan, Banyak;
    bool Ketemu;

    for (i = 0; i < 5; i++)
    {
        cout << "Masukkan nilai ke-" << i + 1 << " = ";
        cin >> Nilai[i];
    }

    cout << "\nDeret Bilangan : ";
    for (i = 0; i < 5; i++)
    {
        cout << Nilai[i] << " ";
    }

    cout << "\n\nMasukkan bilangan yang dicari : ";
    cin >> Bilangan;

    Banyak = 0;
    for (i = 0; i < 5; i++)
    {
        if (Nilai[i] == Bilangan)
        {
            Banyak++;
            Posisi[Banyak] = i;
            Ketemu = true;
        } else {
            Ketemu = false;
        }
    }
}

```



```

}

if (Ketemu) {
    cout << "Bilangan : " << Bilangan << " telah ditemukan sebanyak " << Banyak << "
kali" << endl;
    cout << "Posisi : ";
    for (i = 1; i <= Banyak; i++)
    {
        cout << Posisi[i] << " ";
    }
} else {
    cout << "Bilangan : " << Bilangan << " tidak ditemukan" << endl;
}
}

```

```

Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Masukkan nilai ke-1 = 10
Masukkan nilai ke-2 = 11
Masukkan nilai ke-3 = 12
Masukkan nilai ke-4 = 13
Masukkan nilai ke-5 = 14

Deret Bilangan : 10 11 12 13 14

Masukkan bilangan yang dicari : 15
Bilangan : 15 tidak ditemukan

```

- Lat2_5.cpp

```

#include <iostream>
using namespace std;

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int Nilai[3][4] = {{1, 3, 4, 5}, {2, 4, 6, 8}, {3, 5, 7, 9}};

    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 4; j++)
        {
            cout << Nilai[i][j] << " ";
        }
        cout << endl;
    }
}

```

```

Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

1 3 4 5
2 4 6 8
3 5 7 9

```

LAPORAN PRAKTIKUM STRUKTUR DATA

LAPORAN AKHIR PRAKTIKUM

Pertemuan ke-02



Disusun Oleh:

Nama Lengkap : Nova Ardiansyah

NIM : 211011401309

Kelas : 04-TPLE008

TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PAMULANG

Jl. Raya Puspitek No. 11 Buaran, Serpong Telp. (021) 7412566, Fax. (021) 7412566
Tangerang Selatan - Banten

A. TUGAS AKHIR

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

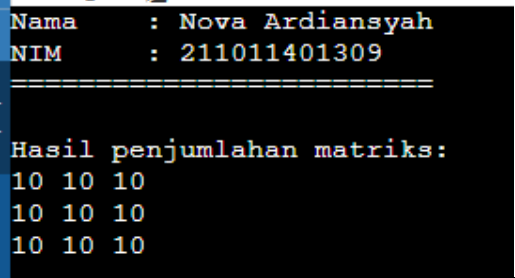
int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int ordo = 3;
    int matriks1[ordo][ordo] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};

    int matriks2[ordo][ordo] = {{9, 8, 7}, {6, 5, 4}, {3, 2, 1}};
    int hasil[ordo][ordo];

    for (int i = 0; i < ordo; i++) {
        for (int j = 0; j < ordo; j++) {
            hasil[i][j] = matriks1[i][j] + matriks2[i][j];
        }
    }

    cout << "Hasil penjumlahan matriks:" << endl;
    for (int i = 0; i < ordo; i++) {
        for (int j = 0; j < ordo; j++) {
            cout << hasil[i][j] << " ";
        }
        cout << endl;
    }
}
```



```
Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Hasil penjumlahan matriks:
10 10 10
10 10 10
10 10 10
```

B. KESIMPULAN

Dalam bahasa pemrograman, array merupakan struktur data yang sangat penting untuk memproses sejumlah data yang besar dengan tipe data yang sama. Terdapat dua jenis array, yaitu array satu dimensi dan array dua dimensi. Array satu dimensi terdiri dari barisan atau deretan elemen dengan tipe data yang sama, sedangkan array dua dimensi terdiri dari elemen-elemen yang disusun dalam bentuk matriks atau tabel berbaris dan berkolom dengan tipe data yang sama.

Kedua jenis array ini memiliki kegunaan dan fungsinya masing-masing. Array satu dimensi sering digunakan untuk menyimpan data dalam jumlah yang besar, seperti data bilangan atau data string. Sedangkan array dua dimensi digunakan untuk merepresentasikan data dalam bentuk matriks atau tabel seperti data statistik atau data permainan.

Penggunaan array dalam bahasa pemrograman C++ memerlukan pemahaman yang baik tentang konsep indeks dan batas array agar dapat mencegah terjadinya kesalahan dalam akses array. Selain itu, array juga dapat digunakan dengan berbagai operasi seperti pengisian nilai, pencarian nilai, pengurutan nilai, dan operasi matematis lainnya. Oleh karena itu, pemahaman tentang array sangat penting bagi setiap programmer dalam mengembangkan program mereka.

LAPORAN PRAKTIKUM STRUKTUR DATA

LAPORAN AWAL PRAKTIKUM

Pertemuan ke-03

Structure



Disusun Oleh:

Nama Lengkap : Nova Ardiansyah

NIM : 211011401309

Kelas : 04-TPLE008

TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PAMULANG

Jl. Raya Puspitek No. 11 Buaran, Serpong Telp. (021) 7412566, Fax. (021) 7412566
Tangerang Selatan - Banten

A. RANGKUMAN MATERI

C++ adalah bahasa pemrograman yang kuat dan efisien, dan salah satu fitur utama yang membuatnya begitu berguna adalah struktur. Struktur adalah jenis data yang memungkinkan programmer untuk mengelompokkan beberapa tipe data bersama-sama dalam satu objek. Dalam bahasa C++, struktur terdiri dari beberapa variabel dengan jenis yang berbeda, yang dapat diakses dan dimanipulasi secara bersamaan. Sebagai contoh, struktur bisa digunakan untuk merepresentasikan sebuah buku, dimana struktur bisa memiliki variabel-variabel seperti judul, penulis, dan nomor ISBN.

Dalam C++, struktur didefinisikan menggunakan kata kunci 'struct'. Setiap variabel dalam struktur disebut anggota struktur atau 'member'. Anggota struktur dapat diakses menggunakan operator titik (.) setelah nama variabel struktur. Sebagai contoh, jika Anda memiliki sebuah struktur yang disebut 'Mahasiswa', dan Anda ingin mengakses anggota 'Nama' dari objek mahasiswa tersebut, maka Anda dapat menggunakan sintaks 'mahasiswa.Nama'.

Satu fitur menarik dari struktur dalam C++ adalah kemampuannya untuk disarangkan. Ini berarti bahwa struktur dapat memiliki struktur lain di dalamnya, yang memungkinkan programmer untuk membuat struktur yang kompleks dan terorganisir dengan baik. Misalnya, jika Anda memiliki struktur 'Alamat' dengan anggota seperti nomor jalan, kota, dan kode pos, Anda dapat menyematkan struktur ini ke dalam struktur 'Mahasiswa' untuk merepresentasikan alamat mahasiswa. Ini membuat program lebih mudah dibaca dan dipahami, dan memudahkan pengelolaan data secara umum.

B. TUGAS PENDAHULUAN

1. Apa yang dimaksud dengan Stucture!

Jawab :

Struktur dalam pemrograman adalah jenis data yang memungkinkan pengelompokan beberapa variabel dengan tipe data yang berbeda menjadi satu objek. Struktur memungkinkan programmer untuk mengelola data dengan lebih terstruktur dan terorganisir, sehingga memudahkan pembacaan dan pemeliharaan kode.

2. Jelaskan kelebihan sebuah program menggunakan Structure!

Jawab :

Kelebihan penggunaan struktur dalam sebuah program adalah memungkinkan programmer untuk mengelompokkan data terkait dalam satu objek, sehingga program lebih mudah dipahami dan dikelola. Selain itu, struktur juga dapat disarangkan, sehingga memungkinkan pembuatan struktur yang kompleks dan terorganisir dengan baik. Struktur juga memudahkan programmer untuk memanipulasi data secara bersamaan, serta menghindari kesalahan penulisan variabel yang serupa.

3. Buatlah contoh program sederhana menggunakan Structure!

Jawab :

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

struct Mahasiswa {
    string nama;
    int usia;
    float ipk;
};

int main() {
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    Mahasiswa mhs1;
    mhs1.nama = "Nova Ardiansyah";
    mhs1.usia = 22;
    mhs1.ipk = 3.85;

    cout << "Nama\t: " << mhs1.nama << endl;
    cout << "Usia\t: " << mhs1.usia << endl;
    cout << "IPK\t: " << mhs1.ipk << endl;

    return 0;
}
```

```

Nama      : Nova Ardiansyah
NIM       : 211011401309
=====
Nama      : Nova Ardiansyah
Usia      : 22
IPK       : 3.85

```

4. Buatlah contoh program sederhana kombinasi Array dan Structure!

Jawab :

```

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

struct Mahasiswa {
    string nama;
    int nilai[3];
};

int main() {
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    Mahasiswa mhs[2];

    for(int i = 0; i < 2; i++) {
        cout << "Masukkan nama mahasiswa " << i+1 << ": ";
        cin >> mhs[i].nama;

        for(int j = 0; j < 3; j++) {
            cout << "Masukkan nilai ke-" << j+1 << ": ";
            cin >> mhs[i].nilai[j];
        }
    }

    cout << endl;
    cout << "Data mahasiswa:" << endl;

    for(int i = 0; i < 2; i++) {
        cout << "Mahasiswa " << i+1 << ": " << mhs[i].nama << endl;
        cout << "Nilai: ";
        for(int j = 0; j < 3; j++) {
            cout << mhs[i].nilai[j] << " ";
        }
        cout << endl;
    }

    return 0;
}

```



```
Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Masukkan nama mahasiswa 1: Nova
Masukkan nilai ke-1: 89
Masukkan nilai ke-2: 75
Masukkan nilai ke-3: 80
Masukkan nama mahasiswa 2: Alex
Masukkan nilai ke-1: 80
Masukkan nilai ke-2: 75
Masukkan nilai ke-3: 89

Data mahasiswa:
Mahasiswa 1: Nova
Nilai: 89 75 80
Mahasiswa 2: Alex
Nilai: 80 75 89
```

C. TUGAS PRAKTIKUM

- Lat3_1

```
#include <iostream>
#include <string>

using namespace std;

struct Mahasiswa
{
    char NIM[15];
    char Nama[25];
    char Alamat[40];
    short Umur;
};

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    Mahasiswa Mhs;

    cout << "Masukkan NIM \t: ";
    cin.getline(Mhs.NIM, 15);

    cout << "Masukkan Nama \t: ";
    cin.getline(Mhs.Nama, 25);

    cout << "Masukkan Alamat : ";
    cin.getline(Mhs.Alatmat, 40);

    cout << "Masukkan Umur \t: ";
    cin >> Mhs.Umur;

    cout << "=====\n\n";

    cout << "Berikut adalah data yang telah dimasukkan: \n";
    cout << "NIM \t: " << Mhs.NIM << endl;
    cout << "Nama \t: " << Mhs.Nama << endl;
    cout << "Alamat \t: " << Mhs.Alatmat << endl;
    cout << "Umur \t: " << Mhs.Umur << endl;

    return 0;
}
```

```
Output

/tmp/8kFTszGjmS.o
Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Masukkan NIM      : 211011401309
Masukkan Nama     : Nova Ardiansyah
Masukkan Alamat  : Jl. Jalur Sutera Kav 29A
Masukkan Umur     : 22
=====

Berikut adalah data yang telah dimasukkan:
NIM      : 211011401309
Nama     : Nova Ardiansyah
Alamat  : Jl. Jalur Sutera Kav 29A
Umur    : 22
```

- Lat3_2

```
#include <iostream>
#include <string>

using namespace std;

struct tinggal
{
    char Jalan[40];
    char Kota[40];
    char Pos[5];
};

struct TglLahir
{
    char Tgl[4];
    char Bln[4];
    char Thn[5];
};

struct Mahasiswa
{
    char NIM[15];
    char Nama[25];
};

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    Mahasiswa Mhs;

    cout << "Masukkan NIM \t\t\t: ";
    cin.getline(Mhs.NIM, 15);
```

```

cout << "Masukkan Nama \t\t\t: ";
cin.getline(Mhs>Nama, 25);

TglLahir Mhs_TglLahir;

cout << "Masukkan Tanggal Lahir \t: ";
cin.getline(Mhs_TglLahir.Tgl, 4);

cout << "Masukkan Bulan Lahir \t: ";
cin.getline(Mhs_TglLahir.Bln, 4);

cout << "Masukkan Tahun Lahir \t: ";
cin.getline(Mhs_TglLahir.Thn, 5);

tinggal Mhs_tinggal;

cout << "Masukkan Jalan \t\t\t: ";
cin.getline(Mhs_tinggal.Jalan, 40);

cout << "Masukkan Kota \t\t\t: ";
cin.getline(Mhs_tinggal.Kota, 40);

cout << "Masukkan Kode Pos \t\t: ";
cin.getline(Mhs_tinggal.Pos, 5);

cout << "=====\n\n";

cout << "Berikut adalah data yang telah dimasukkan: \n";
cout << "NIM \t\t\t: " << Mhs.NIM << endl;
cout << "Nama \t\t\t: " << Mhs>Nama << endl;
cout << "Tanggal Lahir \t: " << Mhs_TglLahir.Tgl << "/" << Mhs_TglLahir.Bln << "/" <<
Mhs_TglLahir.Thn << endl;
cout << "Alamat \t\t\t: " << Mhs_tinggal.Jalan << ", " << Mhs_tinggal.Kota << ", " <<
Mhs_tinggal.Pos << "." << endl;

return 0;
}

```

```

/tmp/8kFTszGjmS.o
Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Masukkan NIM      : 211011401309
Masukkan Nama     : Nova Ardiansyah
Masukkan Tanggal Lahir : 11
Masukkan Bulan Lahir   : 11
Masukkan Tahun Lahir  : 2001
Masukkan Jalan     : Jl. Jalur Sutera Kav 29A
Masukkan Kota      : Tangerang Selatan
Masukkan Kode Pos   : 34131
=====

Berikut adalah data yang telah dimasukkan:
NIM      : 211011401309
Nama     : Nova Ardiansyah
Tanggal Lahir : 11/11/2001
Alamat   : Jl. Jalur Sutera Kav 29A, Tangerang Selatan, 3413.
|

```

- Lat3_3

```

#include <iostream>
#include <string>

using namespace std;

struct Mahasiswa
{
    char NIM[30];
    char Nama[25];
    char Alamat[40];
    char Umur[5];
};

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    Mahasiswa Mhs[2];

    for (int i = 0; i < 2; i++)
    {
        cout << "Masukkan NIM \t: ";
        cin.getline(Mhs[i].NIM, 30);

        cout << "Masukkan Nama \t: ";
        cin.getline(Mhs[i].Nama, 25);

        cout << "Masukkan Alamat : ";
        cin.getline(Mhs[i].Alamat, 40);

        cout << "Masukkan Umur \t: ";
        cin.getline(Mhs[i].Umur, 5);

        cout << "=====\n\n";
    }

    cout << "Berikut adalah data yang telah dimasukkan: \n\n";
    for (int i = 0; i < 2; i++)
    {
        cout << "NIM \t: " << Mhs[i].NIM << endl;
        cout << "Nama \t: " << Mhs[i].Nama << endl;
        cout << "Alamat \t: " << Mhs[i].Alamat << endl;
        cout << "Umur \t: " << Mhs[i].Umur << endl;

        cout << "=====\n\n";
    }

    return 0;
}

```

```

/tmp/vKCcou4WJX.o
Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Masukkan NIM      : 211011401309
Masukkan Nama     : Nova Ardiansyah
Masukkan Alamat   : Jl. Jalur Sutra Kav 29A
Masukkan Umur     : 22
=====

Masukkan NIM      : 211011401310
Masukkan Nama     : Nova Ardiansyah 2
Masukkan Alamat   : Jl. Jalur Sutra Kav 29B
Masukkan Umur     : 22
=====

Berikut adalah data yang telah dimasukkan:

NIM      : 211011401309
Nama     : Nova Ardiansyah
Alamat   : Jl. Jalur Sutra Kav 29A
Umur     : 22
=====

NIM      : 211011401310
Nama     : Nova Ardiansyah 2
Alamat   : Jl. Jalur Sutra Kav 29B
Umur     : 22
=====

```

- Lat3_4

```

#include <iostream>
#include <string>

using namespace std;

struct Mahasiswa
{
    char NIM[30];
    char Nama[25];
    char Jalan[40];
    char Kota[40];
    char KodePos[40];
    char TanggalLahir[40];
};

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    Mahasiswa Mhs[2];
}

```

```

for (int i = 0; i < 2; i++)
{
    cout << "Masukkan NIM \t\t: ";
    cin.getline(Mhs[i].NIM, 30);

    cout << "Masukkan Nama \t\t: ";
    cin.getline(Mhs[i].Nama, 25);

    cout << "Masukkan Jalan \t\t: ";
    cin.getline(Mhs[i].Jalan, 40);

    cout << "Masukkan Kota \t\t: ";
    cin.getline(Mhs[i].Kota, 40);

    cout << "Masukkan Kode Pos \t: ";
    cin.getline(Mhs[i].KodePos, 40);

    cout << "Masukkan Tgl Lahir \t: ";
    cin.getline(Mhs[i].TanggalLahir, 40);

    cout << "=====\n\n";
}

cout << "Berikut adalah data yang telah dimasukkan: \n\n";
for (int i = 0; i < 2; i++)
{
    cout << "NIM \t\t\t: " << Mhs[i].NIM << endl;
    cout << "Nama \t\t\t: " << Mhs[i].Nama << endl;
    cout << "Alamat \t\t\t: " << Mhs[i].Jalan << ", " << Mhs[i].Kota << ", " << Mhs[i].KodePos
    << ". " << endl;
    cout << "Tanggal Lahir \t: " << Mhs[i].TanggalLahir << endl;

    cout << "=====\n\n";
}

return 0;
}

```

Berikut adalah data yang telah dimasukkan:

```

NIM          : Nova Ardiansyah
Nama         : 211011401309
Alamat      : Jl. Jalur Sutera Kav 29A, Tangerang Selatan, 15131.
Tanggal Lahir : 11/11/2001
=====

```

```

NIM          : 211011401310
Nama         : Nova Ardiansyah 2
Alamat      : Jl. Jalur Sutera Kav 29B, Tangerang Selatan, 15131.
Tanggal Lahir : 11/11/2001
=====

```

- Lat3_5

```

#include <iostream>
#include <string>

using namespace std;

struct Mahasiswa

```

```

{
    char NIM[30];
    char Nama[25];
    int NilaiTugas;
    int NilaiUTS;
    int NilaiUAS;
    int NilaiAkhir;
    string NilaiHuruf;
};

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    Mahasiswa Mhs;

    cout << "Masukkan NIM \t\t: ";
    cin.getline(Mhs.NIM, 30);

    cout << "Masukkan Nama \t\t: ";
    cin.getline(Mhs.Nama, 25);

    cout << "Masukkan Nilai Tugas \t: ";
    cin >> Mhs.NilaiTugas;

    cout << "Masukkan Nilai UTS \t\t: ";
    cin >> Mhs.NilaiUTS;

    cout << "Masukkan Nilai UAS \t\t: ";
    cin >> Mhs.NilaiUAS;

    cout << "=====\n\n";

    Mhs.NilaiAkhir = (Mhs.NilaiTugas * 0.2) + (Mhs.NilaiUTS * 0.35) + (Mhs.NilaiUAS * 0.45);

    if (Mhs.NilaiAkhir > 85 && Mhs.NilaiAkhir <= 100) {
        Mhs.NilaiHuruf = "A";
    } else if (Mhs.NilaiAkhir > 70 && Mhs.NilaiAkhir <= 85) {
        Mhs.NilaiHuruf = "B";
    } else if (Mhs.NilaiAkhir > 55 && Mhs.NilaiAkhir <= 70) {
        Mhs.NilaiHuruf = "C";
    } else if (Mhs.NilaiAkhir > 40 && Mhs.NilaiAkhir <= 55) {
        Mhs.NilaiHuruf = "D";
    } else if (Mhs.NilaiAkhir <= 40) {
        Mhs.NilaiHuruf = "E";
    }

    cout << "Berikut adalah hasil perhitungan nilai akhir Anda :\n\n";

    cout << "NIM \t\t\t: " << Mhs.NIM << endl;
    cout << "Nama \t\t\t: " << Mhs.Nama << endl;
    cout << "Nilai Tugas \t: " << Mhs.NilaiTugas << endl;
    cout << "Nilai UTS \t\t: " << Mhs.NilaiUTS << endl;
    cout << "Nilai UAS \t\t: " << Mhs.NilaiUAS << endl;
    cout << "Nilai Akhir \t: " << Mhs.NilaiAkhir << endl;
    cout << "Peringkat \t\t: " << Mhs.NilaiHuruf << endl;

    return 0;
}

```



```
}  
  
/tmp/icZaRkdvL4.o  
Nama      : Nova Ardiansyah  
NIM       : 211011401309  
=====
```

Masukkan NIM : 211011401309
Masukkan Nama : Nova Ardiansyah
Masukkan Nilai Tugas : 80
Masukkan Nilai UTS : 85
Masukkan Nilai UAS : 90
=====

Berikut adalah hasil perhitungan nilai akhir Anda :

NIM : 211011401309
Nama : Nova Ardiansyah
Nilai Tugas : 80
Nilai UTS : 85
Nilai UAS : 90
Nilai Akhir : 86
Peringkat : A

LAPORAN PRAKTIKUM STRUKTUR DATA

LAPORAN AKHIR PRAKTIKUM

Pertemuan ke-03

Structure



Disusun Oleh:

Nama Lengkap : Nova Ardiansyah

NIM : 211011401309

Kelas : 04-TPLE008

TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PAMULANG

Jl. Raya Puspitek No. 11 Buaran, Serpong Telp. (021) 7412566, Fax. (021) 7412566
Tangerang Selatan - Banten

A. TUGAS AKHIR

1. Buatlah program untuk membaca dan mencetak biodata pegawai, dengan menggunakan :
 - a) Stuktur BIODATA yang terdiri dari NIP, Nama, Alamat, Jabatan, Agama, Tanggal_Lahir, Tanggal_Mulai_Kerja, Unit_Kerja
 - b) Struktur TINGGAL yang terdiri dari Jalan, Kode_Pos, Kota dan
 - c) Stuktur TANGGAL yang terdiri dari Tanggal, Bulan, Tahun.Gunakan Array of Struct

Jawab :

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

struct TANGGAL {
    int tanggal;
    int bulan;
    int tahun;
};

struct TINGGAL {
    string jalan;
    string kode_pos;
    string kota;
};

struct BIODATA {
    string nip;
    string nama;
    string alamat;
    string jabatan;
    string agama;
    TANGGAL tanggal_lahir;
    TANGGAL tanggal_mulai_kerja;
    TINGGAL unit_kerja;
};

int main() {
    const int MAX_PEGAWAI = 2;
    BIODATA pegawai[MAX_PEGAWAI];

    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int n;
    cout << "Masukkan jumlah pegawai: ";
    cin >> n;

    for (int i = 0; i < n; i++) {
        cout << "\nBiodata Pegawai " << i+1 << ":\n";
```

```

    cout << "NIP: ";
    cin >> pegawai[i].nip;

    cout << "Nama: ";
    cin >> pegawai[i].nama;

    cout << "Alamat: ";
    cin.ignore();
    getline(cin, pegawai[i].alamat);

    cout << "Jabatan: ";
    getline(cin, pegawai[i].jabatan);

    cout << "Agama: ";
    getline(cin, pegawai[i].agama);

    cout << "Tanggal Lahir (dd mm yyyy): ";
    cin >> pegawai[i].tanggal_lahir.tanggal >> pegawai[i].tanggal_lahir.bulan >>
    pegawai[i].tanggal_lahir.tahun;

    cout << "Tanggal Mulai Kerja (dd mm yyyy): ";
    cin >> pegawai[i].tanggal_mulai_kerja.tanggal >> pegawai[i].tanggal_mulai_kerja.bulan
    >> pegawai[i].tanggal_mulai_kerja.tahun;

    cout << "Unit Kerja:\n";
    cout << "Jalan: ";

    cin.ignore();
    getline(cin, pegawai[i].unit_kerja.jalan);

    cout << "Kode Pos: ";
    getline(cin, pegawai[i].unit_kerja.kode_pos);

    cout << "Kota: ";
    getline(cin, pegawai[i].unit_kerja.kota);
}

cout << "\nData Pegawai:\n";
for (int i = 0; i < n; i++) {
    cout << "\nBiodata Pegawai " << i+1 << ":\n";
    cout << "NIP: " << pegawai[i].nip << endl;
    cout << "Nama: " << pegawai[i].nama << endl;
    cout << "Alamat: " << pegawai[i].alamat << endl;
    cout << "Jabatan: " << pegawai[i].jabatan << endl;
    cout << "Agama: " << pegawai[i].agama << endl;
    cout << "Tanggal Lahir: " << pegawai[i].tanggal_lahir.tanggal << "/" <<
    pegawai[i].tanggal_lahir.bulan << "/" << pegawai[i].tanggal_lahir.tahun << endl;
    cout << "Tanggal Mulai Kerja: " << pegawai[i].tanggal_mulai_kerja.tanggal << "/" <<
    pegawai[i].tanggal_mulai_kerja.bulan << "/" << pegawai[i].tanggal_mulai_kerja.tahun <<
    endl;

    cout << "Unit Kerja:\n";
    cout << "Jalan: " << pegawai[i].unit_kerja.jalan << endl;
    cout << "Kode Pos: " << pegawai[i].unit_kerja.kode_pos << endl;
    cout << "Kota: " << pegawai[i].unit_kerja.kota << endl;
}

return 0;
}

```

```
Biodata Pegawai 1:  
NIP: 12345  
Nama: Nova  
Alamat: Lampung  
Jabatan: HRD  
Agama: Islam  
Tanggal Lahir: 11/11/2001  
Tanggal Mulai Kerja: 1/1/2021  
Unit Kerja:  
Jalan: Ahmad Yani  
Kode Pos: 15131  
Kota: Tangerang
```

2. KESIMPULAN

Dalam bahasa pemrograman C++, struktur merupakan tipe data yang memungkinkan programmer untuk mengelompokkan beberapa variabel dengan tipe data yang berbeda menjadi satu objek. Dengan menggunakan struktur, programmer dapat mengelola data dengan lebih terstruktur dan terorganisir, sehingga memudahkan pembacaan dan pemeliharaan kode. Kelebihan penggunaan struktur adalah memungkinkan programmer untuk mengelompokkan data terkait dalam satu objek, sehingga program lebih mudah dipahami dan dikelola. Struktur juga dapat disarangkan, sehingga memungkinkan pembuatan struktur yang kompleks dan terorganisir dengan baik. Struktur juga memudahkan programmer untuk memanipulasi data secara bersamaan, serta menghindari kesalahan penulisan variabel yang serupa.

Selain itu, struktur juga dapat dikombinasikan dengan array untuk membentuk struktur data yang lebih kompleks. Misalnya, struktur Mahasiswa dapat memiliki array nilai dengan beberapa elemen untuk menyimpan nilai-nilai dari beberapa mata kuliah. Program tersebut dapat meminta pengguna untuk memasukkan data mahasiswa, termasuk nama dan nilai-nilai, dan kemudian menampilkan data tersebut dalam bentuk tabel. Dengan menggunakan struktur dan array, programmer dapat membuat program yang lebih terstruktur dan mudah dipelihara. Namun, sebelum menggunakan struktur dalam program, programmer harus memahami konsep struktur dengan baik dan membuat desain struktur yang tepat untuk program yang dibuat.

LAPORAN PRAKTIKUM STRUKTUR DATA

LAPORAN AWAL PRAKTIKUM

Pertemuan ke-04

Pointer



Disusun Oleh:

Nama Lengkap : Nova Ardiansyah

NIM : 211011401309

Kelas : 04-TPLE008

TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PAMULANG

Jl. Raya Puspitek No. 11 Buaran, Serpong Telp. (021) 7412566, Fax. (021) 7412566
Tangerang Selatan - Banten

A. RANGKUMAN MATERI

Pointer adalah salah satu konsep yang penting dalam bahasa pemrograman C++. Pointer memungkinkan kita untuk mengakses dan memodifikasi nilai atau variabel yang disimpan pada alamat memori tertentu. Dalam C++, setiap variabel memiliki alamat memori unik yang dapat diakses melalui operator address-of (&). Variabel pointer dapat digunakan untuk menyimpan alamat memori dari variabel lain, sehingga kita dapat mengakses atau memodifikasi nilai variabel tersebut melalui variabel pointer.

Selain itu, pointer juga memungkinkan pengalokasian memori secara dinamis dengan menggunakan operator new dan delete. Dalam penggunaannya, pointer harus diinisialisasi dengan nilai awal, yaitu alamat memori dari variabel atau objek yang ditunjuk. Kemudian, kita dapat menggunakan operator dereference (*) untuk mengakses nilai yang disimpan pada alamat memori yang ditunjuk oleh variabel pointer. Pemahaman tentang pointer sangat penting bagi para programmer untuk memaksimalkan penggunaan memori pada program dan mengoptimalkan kinerja program.

B. TUGAS PENDAHULUAN

1. Apa yang dimaksud dengan Pointer!

Jawab :

Pointer adalah sebuah variabel yang menyimpan alamat memori dari suatu nilai atau variabel lainnya. Pointer memungkinkan kita untuk mengakses dan memodifikasi nilai atau variabel yang disimpan pada alamat memori tertentu.

2. Apa yang dimaksud dengan Variabel Pointer!

Jawab :

Variabel pointer adalah sebuah variabel yang nilainya adalah alamat memori dari variabel lainnya. Variabel pointer digunakan untuk menyimpan alamat memori dari variabel lain, sehingga kita dapat mengakses atau memodifikasi nilai variabel tersebut melalui variabel pointer.

3. Sebutkan dan jelaskan jenis-jenis Operator Pointer!

Jawab :

Jenis-jenis operator pointer adalah:

- Operator dereference (*) : digunakan untuk mengakses nilai yang tersimpan pada alamat memori yang ditunjuk oleh pointer.
- Operator address-of (&) : digunakan untuk mendapatkan alamat memori dari suatu variabel.
- Operator assignment (=) : digunakan untuk menetapkan alamat memori ke dalam variabel pointer.
- Operator increment/decrement (++/--): digunakan untuk menggeser alamat yang ditunjuk oleh pointer, sehingga pointer mengarah ke alamat memori yang berbeda.

4. Buatlah contoh program sederhana menggunakan pointer!

Jawab :

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

int main() {
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int x = 10;
    int *ptr;

    ptr = &x;

    cout << "Nilai x = " << x << endl;
    cout << "Alamat memori x = " << &x << endl;
    cout << "Nilai yang ditunjuk oleh ptr = " << *ptr << endl;
    cout << "Alamat memori yang ditunjuk oleh ptr = " << ptr << endl;
    cout << "Alamat memori dari ptr = " << &ptr << endl;

    *ptr = 20;
    cout << "Nilai x sekarang = " << x << endl;

    return 0;
}
```



```
Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Nilai x = 10
Alamat memori x = 0x7ffd9fcaa31c
Nilai yang ditunjuk oleh ptr = 10
Alamat memori yang ditunjuk oleh ptr = 0x7ffd9fcaa31c
Alamat memori dari ptr = 0x7ffd9fcaa320
Nilai x sekarang = 20
```

C. TUGAS PRAKTIKUM

- Lat4_1

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int x = 8, y;
    int *xPtr = &x;

    cout << "Nilai x \t: " << x << endl;
    cout << "Alamat x \t: " << &x << endl;
    cout << "Nilai xPtr \t: " << xPtr << endl;

    cout << "Nilai yang disimpan pada alamat " << xPtr << " adalah " << *xPtr << endl;

    return 0;
}
```

Output

```
/tmp/IxJMt2l2dz.o
Nama    : Nova Ardiansyah
NIM     : 211011401309
=====

Nilai x      : 8
Alamat x     : 0x7ffd085bcc0c
Nilai xPtr   : 0x7ffd085bcc0c
Nilai yang disimpan pada alamat 0x7ffd085bcc0c adalah 8
```

- Lat4_2

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int x = 12, y;
    int *xPtr = &x;

    cout << "Nilai x \t: " << x << endl;
    cout << "Alamat x \t: " << &x << endl;
```

```

cout << "Nilai xPtr \t: " << xPtr << endl;

cout << "Nilai yang disimpan pada alamat " << xPtr << " adalah " << *xPtr << endl;

return 0;
}

```

Output

```

/tmp/IxJMt2l2dz.o
Nama    : Nova Ardiansyah
NIM     : 211011401309
=====

Nilai x      : 12
Alamat x     : 0x7ffc5ddd84fc
Nilai xPtr   : 0x7ffc5ddd84fc
Nilai yang disimpan pada alamat 0x7ffc5ddd84fc adalah 12

```

- Lat4_3

```

#include <iostream>
#include <string>

using namespace std;

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int Nilai[] = { 45, 23, 50, 8, 12, 10, 15 };
    int *NilaiPtr = Nilai;

    int i;
    NilaiPtr = Nilai;

    cout << "Nilai Ptr \t\t: " << NilaiPtr << endl;
    cout << "Alamat Nilai \t: " << &Nilai << endl;
    cout << "Nilai yang disimpan pada alamat " << NilaiPtr << " adalah " << *NilaiPtr << endl;

    cout << "Element array index pertama: " << NilaiPtr[0] << "\n\n";

    for (i = 1; i < 7; i++)
    {
        cout << "Nilai yang disimpan pada alamat " << NilaiPtr << " adalah " << *NilaiPtr << endl;
        NilaiPtr++;
    }

    return 0;
}

```

```
Output

/tmp/IxJMt2l2dz.o
Nama    : Nova Ardiansyah
NIM     : 211011401309
=====

Nilai Ptr      : 0x7ffa9f01630
Alamat Nilai   : 0x7ffa9f01630
Nilai yang disimpan pada alamat 0x7ffa9f01630 adalah 45
Element array index pertama: 45

Nilai yang disimpan pada alamat 0x7ffa9f01630 adalah 45
Nilai yang disimpan pada alamat 0x7ffa9f01634 adalah 23
Nilai yang disimpan pada alamat 0x7ffa9f01638 adalah 50
Nilai yang disimpan pada alamat 0x7ffa9f0163c adalah 8
Nilai yang disimpan pada alamat 0x7ffa9f01640 adalah 12
Nilai yang disimpan pada alamat 0x7ffa9f01644 adalah 10
```

- Lat4_4

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int x = 8;
    int *xPtr1;
    int **xPtr2;

    xPtr1 = &x;
    xPtr2 = &xPtr1;

    cout << "Nilai x \t\t: " << x << endl;
    cout << "Alamat x \t\t: " << &x << endl;
    cout << "Nilai xPtr1 \t: " << xPtr1 << endl;
    cout << "Alamat xPtr1 \t: " << &xPtr1 << endl;
    cout << "Nilai xPtr2 \t: " << xPtr2 << endl;
    cout << "Alamat xPtr2 \t: " << &xPtr2 << endl;

    return 0;
}
```

Output

```
/tmp/IxJMt2l2dz.o
Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Nilai x      : 8
Alamat x     : 0x7ffefba01b94
Nilai xPtr1  : 0x7ffefba01b94
Alamat xPtr1 : 0x7ffefba01b98
Nilai xPtr2  : 0x7ffefba01b98
Alamat xPtr2 : 0x7ffefba01ba0
|
```

- Lat4_5

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int Nilai[] = { 11, 23, 17, 8, 21, 10, 25 };
    int *NilaiPtr = Nilai;

    NilaiPtr = Nilai;

    cout << "Nilai Ptr \t\t: " << NilaiPtr << endl;
    cout << "Alamat Nilai \t: " << &Nilai << endl;
    cout << "Nilai yang disimpan pada alamat " << NilaiPtr << " adalah " << *NilaiPtr << endl;

    cout << "Element array index pertama: " << NilaiPtr[0] << "\n\n";

    while (NilaiPtr < Nilai + 7)
    {
        cout << "Nilai yang disimpan pada alamat " << NilaiPtr << " adalah " << *NilaiPtr << endl;
        NilaiPtr++;
    }

    return 0;
}
```

Output

/tmp/IxJMt2l2dz.o

Nama : Nova Ardiansyah

NIM : 211011401309

=====

Nilai Ptr : 0x7ffd7bcf8210

Alamat Nilai : 0x7ffd7bcf8210

Nilai yang disimpan pada alamat 0x7ffd7bcf8210 adalah 11

Element array index pertama: 11

Nilai yang disimpan pada alamat 0x7ffd7bcf8210 adalah 11

Nilai yang disimpan pada alamat 0x7ffd7bcf8214 adalah 23

Nilai yang disimpan pada alamat 0x7ffd7bcf8218 adalah 17

Nilai yang disimpan pada alamat 0x7ffd7bcf821c adalah 8

Nilai yang disimpan pada alamat 0x7ffd7bcf8220 adalah 21

Nilai yang disimpan pada alamat 0x7ffd7bcf8224 adalah 10

Nilai yang disimpan pada alamat 0x7ffd7bcf8228 adalah 25

LAPORAN PRAKTIKUM STRUKTUR DATA

LAPORAN AKHIR PRAKTIKUM

Pertemuan ke-04

Pointer



Disusun Oleh:

Nama Lengkap : Nova Ardiansyah
NIM : 211011401309
Kelas : 04-TPLE008

TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PAMULANG

Jl. Raya Puspitek No. 11 Buaran, Serpong Telp. (021) 7412566, Fax. (021) 7412566
Tangerang Selatan - Banten

A. TUGAS AKHIR

1. Buatlah program dengan menggunakan kombinasi antara Pointer dan Array Dua Dimensi!

Jawab :

```
#include <iostream>

using namespace std;

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int Nilai[2][3] = { { 11, 23, 17 }, { 8, 21, 10 } };
    cout << "Array : " << "{ { 11, 23, 17 }, { 8, 21, 10 } }\n\n";

    for (int *NilaiPtr = Nilai[0]; NilaiPtr < Nilai[0] + 6; NilaiPtr++)
    {
        cout << "Nilai yang disimpan pada alamat " << NilaiPtr << " adalah " << *NilaiPtr << endl;
    }

    return 0;
}
```

Output

```
/tmp/IxJMt2l2dz.o
Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Array : { { 11, 23, 17 }, { 8, 21, 10 } }

Nilai yang disimpan pada alamat 0x7fffa30b12a0 adalah 11
Nilai yang disimpan pada alamat 0x7fffa30b12a4 adalah 23
Nilai yang disimpan pada alamat 0x7fffa30b12a8 adalah 17
Nilai yang disimpan pada alamat 0x7fffa30b12ac adalah 8
Nilai yang disimpan pada alamat 0x7fffa30b12b0 adalah 21
Nilai yang disimpan pada alamat 0x7fffa30b12b4 adalah 10
```


2. KESIMPULAN

Dalam bahasa pemrograman C++, pointer memungkinkan kita untuk mengakses dan memodifikasi nilai atau variabel yang disimpan pada alamat memori tertentu. Dengan menggunakan pointer, kita dapat mengakses dan memodifikasi variabel melalui alamat memori yang ditunjuk oleh variabel pointer. Oleh karena itu, pemahaman tentang pointer sangat penting bagi para programmer untuk memaksimalkan penggunaan memori pada program dan mengoptimalkan kinerja program.

LAPORAN PRAKTIKUM STRUKTUR DATA

LAPORAN AWAL PRAKTIKUM

Pertemuan ke-05

Function



Disusun Oleh:

Nama Lengkap : Nova Ardiansyah

NIM : 211011401309

Kelas : 04-TPLE008

TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PAMULANG

Jl. Raya Puspitek No. 11 Buaran, Serpong Telp. (021) 7412566, Fax. (021) 7412566
Tangerang Selatan - Banten

A. RANGKUMAN MATERI

Function atau fungsi adalah bagian penting dari bahasa pemrograman C++. Fungsi digunakan untuk memecah program menjadi bagian yang lebih kecil dan lebih mudah dikelola. Fungsi juga memungkinkan pengembang program untuk menulis kode sekali dan menggunakannya berkali-kali dalam program. Fungsi pada C++ juga dapat menerima argumen atau parameter dan mengembalikan nilai. Dalam C++, fungsi didefinisikan dengan menentukan tipe data kembalian, nama fungsi, dan daftar parameter.

C++ juga mendukung fungsi rekursif, yaitu fungsi yang memanggil dirinya sendiri dalam rangka menyelesaikan tugas tertentu. Fungsi rekursif sangat berguna untuk menyelesaikan masalah yang kompleks atau memerlukan pengulangan. Penggunaan fungsi pada C++ dapat meningkatkan efisiensi program dan memungkinkan pengembang program untuk membuat program yang lebih modular dan mudah dikelola. Oleh karena itu, pemahaman yang baik tentang fungsi pada C++ sangat penting bagi pengembang program yang ingin membuat program yang efisien dan mudah dikelola.

B. TUGAS PENDAHULUAN

1. Apa yang dimaksud dengan fungsi!

Jawab :

Fungsi adalah kumpulan instruksi yang dirancang untuk menyelesaikan tugas tertentu dalam program. Fungsi dapat dipanggil dari bagian lain dalam program dan dapat memiliki argumen atau parameter yang dapat diteruskan ke dalamnya. Fungsi juga dapat mengembalikan nilai, sehingga nilai tersebut dapat digunakan dalam operasi selanjutnya dalam program.

2. Jelaskan yang dimaksud pemanggilan secara nilai (call by value) dan pemanggilan secara referensi (call by reference)!

Jawab :

Pemanggilan secara nilai (call by value) dan pemanggilan secara referensi (call by reference) adalah dua cara pemanggilan fungsi dalam C++. Pada pemanggilan secara nilai, salinan nilai dari parameter yang diteruskan ke dalam fungsi dibuat dan digunakan dalam fungsi. Pada pemanggilan secara referensi, parameter diteruskan ke dalam fungsi sebagai referensi, sehingga parameter yang diteruskan dapat diubah

dalam fungsi dan perubahan tersebut akan tercermin pada variabel yang memanggil fungsi.

3. Jelaskan yang dimaksud dengan variabel lokal, variabel eksternal atau global, dan variabel statis!

Jawab :

Variabel lokal adalah variabel yang dideklarasikan dalam fungsi dan hanya dapat diakses di dalam fungsi tersebut. Variabel eksternal atau global adalah variabel yang dideklarasikan di luar fungsi dan dapat diakses di seluruh program. Variabel statis adalah variabel yang dideklarasikan dalam fungsi dan nilainya akan tetap sama selama program berjalan, bahkan setelah fungsi dipanggil beberapa kali.

4. Buatlah contoh program sederhana dengan fungsi!

Jawab :

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

int hitungLuasPersegiPanjang(int panjang, int lebar) {
    int luas = panjang * lebar;
    return luas;
}

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int p, l;

    cout << "Masukkan panjang persegi panjang: ";
    cin >> p;

    cout << "Masukkan lebar persegi panjang: ";
    cin >> l;

    int luas = hitungLuasPersegiPanjang(p, l);
    cout << "Luas persegi panjang adalah: " << luas << endl;

    return 0;
}
```

```
Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Masukkan panjang persegi panjang: 12
Masukkan lebar persegi panjang: 14
Luas persegi panjang adalah: 168
```

C. TUGAS PRAKTIKUM

- Lat5_1

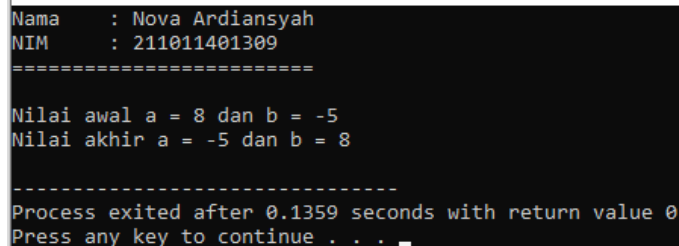
```
#include <iostream>
#include <string>

using namespace std;

void tukar(int &x, int &y)
{
    int temp;
    temp = x;
    x = y;
    y = temp;
}

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int a = 8, b = -5;
    cout << "Nilai awal a = " << a << " dan b = " << b << "\n";
    tukar(a, b);
    cout << "Nilai akhir a = " << a << " dan b = " << b << "\n";
}
```

A screenshot of a terminal window showing the output of the C++ program. The output is as follows:

```
Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Nilai awal a = 8 dan b = -5
Nilai akhir a = -5 dan b = 8

-----
Process exited after 0.1359 seconds with return value 0
Press any key to continue . . .
```

- Lat5_2

```
#include <iostream>
#include <string>

using namespace std;

void tukar(int &x, int &y)
{
    int temp;
    temp = x;
    x = y;
    y = temp;
}

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";
```

```

int a = 8, b = -5;
cout << "Nilai awal a = " << a << " dan b = " << b << "\n";
tukar(a, b);
cout << "Nilai akhir a = " << a << " dan b = " << b << "\n";
}

```

```

Nama    : Nova Ardiansyah
NIM     : 211011401309
=====

Nilai awal a = 8 dan b = -5
Nilai akhir a = -5 dan b = 8

-----
Process exited after 0.1359 seconds with return value 0
Press any key to continue . . .

```

- Lat5_3

```

#include <iostream>
#include <string>

using namespace std;

int Maksimum(int a, int b, int c)
{
    int maks;
    maks = a;
    if (b > maks)
        maks = b;
    if (c > maks)
        maks = c;
    return maks;
}

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int a = 8, b = 12, c = -5;
    cout << "Nilai maksimum dari " << a << ", " << b << " dan " << c << " adalah " <<
    Maksimum(a, b, c) << endl;
}

```

```

Nama    : Nova Ardiansyah
NIM     : 211011401309
=====

Nilai maksimum dari 8, 12 dan -5 adalah 12

-----
Process exited after 0.1232 seconds with return value 0
Press any key to continue . . .

```

- Lat5_4

```

#include <iostream>
#include <string>

using namespace std;

```

```

void Tukat (int *a, int *b)
{
    int temp;
    temp = *a;
    *a = *b;
    *b = temp;
}

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int a = 8, b = -5;
    cout << "Nilai awal a = " << a << " dan b = " << b << "\n";
    Tukat(&a, &b);
    cout << "Nilai akhir a = " << a << " dan b = " << b << "\n";
}

```

```

Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Nilai awal a = 8 dan b = -5
Nilai akhir a = -5 dan b = 8

-----
Process exited after 0.1037 seconds with return value 0
Press any key to continue . . .

```

- Lat5_5

```

#include <iostream>

const int N = 10;

void Baca(int Nilai[], int M);
void Cetak(int Nilai[], int M);
void Jumlah(int Nilai[], int M, int &jml, float &rata);

using namespace std;

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int Nilai[N], M, jml;
    float rata;

    cout << "Masukkan banyaknya data: ";
    cin >> M;

    Baca(Nilai, M);
    Cetak(Nilai, M);
    Jumlah(Nilai, M, jml, rata);

    return 0;
}

```



```

void Baca(int Nilai[], int M) {
    cout << "Masukkan " << M << " elemen array: ";
    for (int i = 0; i < M; i++) {
        cin >> Nilai[i];
    }
}

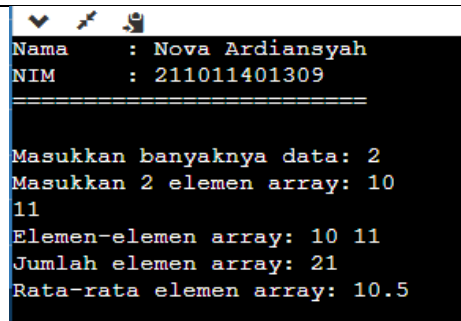
void Cetak(int Nilai[], int M) {
    cout << "Elemen-elemen array: ";
    for (int i = 0; i < M; i++) {
        cout << Nilai[i] << " ";
    }

    cout << endl;
}

void Jumlah(int Nilai[], int M, int &jml, float &rata) {
    jml = 0;
    for (int i = 0; i < M; i++) {
        jml = jml + Nilai[i];
    }

    rata = static_cast<float>(jml) / M;
    cout << "Jumlah elemen array: " << jml << endl;
    cout << "Rata-rata elemen array: " << rata << endl;
}

```



```

Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Masukkan banyaknya data: 2
Masukkan 2 elemen array: 10
11
Elemen-elemen array: 10 11
Jumlah elemen array: 21
Rata-rata elemen array: 10.5

```

- Lat5_6

```

#include <iostream>
#include <string>

using namespace std;

struct Mahasiswa
{
    char NIM[20];
    char Nama[40];
    char Alamat[40];
    short Umur;
};

void Baca(Mahasiswa &Mhs);
void Cetak(Mahasiswa Mhs);

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
}

```

```

cout << "NIM \t: 211011401309\n";
cout << "=====\n\n";

Mahasiswa Mhs;
cout << "Masukkan data mahasiswa: " << endl;
Baca(Mhs);

cout << "Data mahasiswa: " << endl;
Cetak(Mhs);
}

void Baca(Mahasiswa &Mhs)
{
    cout << "NIM \t: ";
    cin.getline(Mhs.NIM, 20);

    cout << "Nama \t: ";
    cin.getline(Mhs>Nama, 40);

    cout << "Alamat \t: ";
    cin.getline(Mhs.Alat, 40);

    cout << "Umur \t: ";
    cin >> Mhs.Umur;
}

void Cetak(Mahasiswa Mhs)
{
    cout << "NIM \t: " << Mhs.NIM << endl;
    cout << "Nama \t: " << Mhs>Nama << endl;
    cout << "Alamat \t: " << Mhs.Alat << endl;
    cout << "Umur \t: " << Mhs.Umur << endl;
}

```

```

Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Masukkan data mahasiswa:
NIM       : 211011401309
Nama      : Nova Ardiansyah
Alamat    : Tangerang
Umur      : 22
Data mahasiswa:
NIM       : 211011401309
Nama      : Nova Ardiansyah
Alamat    : Tangerang
Umur      : 22

```

- Lat5_7

```

#include <iostream>
#include <string>

using namespace std;

struct Mahasiswa
{
    char NIM[20];
    char Nama[40];
    char Alamat[40];
    short Umur;
}

```

```

};

void Baca(Mahasiswa *Mhs);
void Cetak(Mahasiswa *Mhs);

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    Mahasiswa Mhs;
    Mahasiswa *pMhs = &Mhs;
    cout << "Masukkan data mahasiswa: " << endl;
    Baca(pMhs);

    cout << "Data mahasiswa: " << endl;
    Cetak(pMhs);
}

void Baca(Mahasiswa *Mhs)
{
    cout << "NIM \t: ";
    cin.getline(Mhs->NIM, 20);

    cout << "Nama \t: ";
    cin.getline(Mhs->Nama, 40);

    cout << "Alamat \t: ";
    cin.getline(Mhs->Alamat, 40);

    cout << "Umur \t: ";
    cin >> Mhs->Umur;
}

void Cetak(Mahasiswa *Mhs)
{
    cout << "NIM \t: " << Mhs->NIM << endl;
    cout << "Nama \t: " << Mhs->Nama << endl;
    cout << "Alamat \t: " << Mhs->Alamat << endl;
    cout << "Umur \t: " << Mhs->Umur << endl;
}

```

```

Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Masukkan data mahasiswa:
NIM       : 211011401309
Nama      : Nova Ardiansyah
Alamat    : Tangerang
Umur      : 22
Data mahasiswa:
NIM       : 211011401309
Nama      : Nova Ardiansyah
Alamat    : Tangerang
Umur      : 22

```

LAPORAN PRAKTIKUM STRUKTUR DATA

LAPORAN AKHIR PRAKTIKUM

Pertemuan ke-05

Function



Disusun Oleh:

Nama Lengkap : Nova Ardiansyah
NIM : 211011401309
Kelas : 04-TPLE008

TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PAMULANG

Jl. Raya Puspitek No. 11 Buaran, Serpong Telp. (021) 7412566, Fax. (021) 7412566
Tangerang Selatan - Banten

A. TUGAS AKHIR

1. Buatlah program dengan menggunakan kombinasi antara Fungsi dan Pointer!

Jawab :

```
#include <iostream>
#include <string>

using namespace std;
void cariNilaiTerkecil(int *arr, int n);

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int arr[] = {5, 8, 1, 10, 3, 7, 4};
    int n = sizeof(arr) / sizeof(arr[0]);

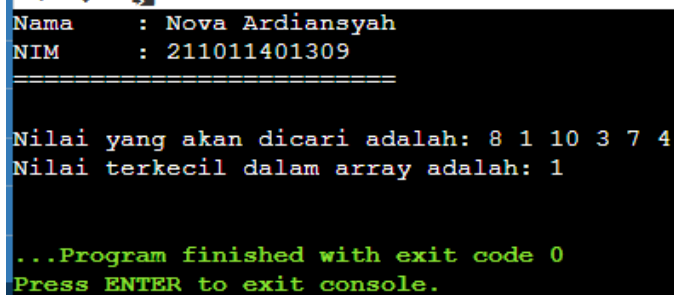
    cariNilaiTerkecil(arr, n);

    return 0;
}

void cariNilaiTerkecil(int *arr, int n)
{
    int nilaiTerkecil = *arr;

    cout << "Nilai yang akan dicari adalah: ";
    for (int i = 1; i < n; i++)
    {
        cout << *(arr + i) << " ";
        if (*(arr + i) < nilaiTerkecil)
        {
            nilaiTerkecil = *(arr + i);
        }
    }

    cout << endl;
    cout << "Nilai terkecil dalam array adalah: ";
    cout << nilaiTerkecil << endl;
}
```



```
Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Nilai yang akan dicari adalah: 8 1 10 3 7 4
Nilai terkecil dalam array adalah: 1

...Program finished with exit code 0
Press ENTER to exit console.
```

2. KESIMPULAN

Kesimpulan dari materi function pada C++ adalah bahwa fungsi sangat penting dalam pengembangan program. Dengan menggunakan fungsi, program dapat dibagi menjadi beberapa bagian yang lebih kecil dan lebih mudah dikelola. Fungsi juga dapat meningkatkan kegunaan dan fleksibilitas program, serta menghindari duplikasi kode. Selain itu, fungsi juga memungkinkan kode program yang digunakan berkali-kali dapat digunakan dalam berbagai bagian program yang berbeda. Oleh karena itu, pemahaman yang baik tentang fungsi pada C++ sangat diperlukan bagi pengembang program yang ingin membuat program yang efisien dan mudah dikelola.

LAPORAN PRAKTIKUM STRUKTUR DATA

LAPORAN AWAL PRAKTIKUM

Pertemuan ke-06

Searching



Disusun Oleh:

Nama Lengkap : Nova Ardiansyah

NIM : 211011401309

Kelas : 04-TPLE008

TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PAMULANG

Jl. Raya Puspitek No. 11 Buaran, Serpong Telp. (021) 7412566, Fax. (021) 7412566
Tangerang Selatan - Banten

A. RANGKUMAN MATERI

Searching atau pencarian adalah proses mencari nilai tertentu pada struktur data seperti array, list, maupun tree. Pada program C++, terdapat beberapa metode searching yang dapat digunakan seperti sequential search, binary search, dan interpolation search. Sequential search adalah metode searching sederhana dimana program mencari nilai dengan cara membandingkan setiap nilai pada struktur data satu persatu. Jika nilai ditemukan, maka program akan memberikan output indeks dari nilai tersebut. Namun, jika nilai tidak ditemukan, program akan memberikan output bahwa nilai tersebut tidak ditemukan.

Metode searching yang lebih efektif adalah binary search. Binary search bekerja dengan mengurutkan struktur data terlebih dahulu, kemudian mencari nilai dengan cara membagi struktur data menjadi dua bagian secara terus menerus sampai nilai yang dicari ditemukan. Binary search lebih efisien daripada sequential search, terutama pada struktur data yang besar. Namun, syarat untuk menggunakan binary search adalah struktur data harus sudah terurut terlebih dahulu.

Selain itu, terdapat metode searching lain yang bisa digunakan pada data yang memiliki distribusi yang merata yaitu interpolation search. Interpolation search mencari nilai dengan mengestimasi lokasi nilai tersebut berada di dalam struktur data. Metode ini menghitung nilai perkiraan lokasi nilai yang akan dicari, dan mencari nilai tersebut secara berulang-ulang hingga ditemukan. Interpolation search lebih cepat dibandingkan sequential search dan bisa digunakan pada data yang besar dan memiliki distribusi yang merata.

B. TUGAS PENDAHULUAN

1. Apa yang dimaksud dengan Searching!

Jawab :

Searching atau pencarian adalah proses mencari nilai tertentu pada suatu struktur data, seperti array, list, maupun tree. Tujuan pencarian adalah untuk menemukan nilai yang dicari, atau memberikan pesan bahwa nilai tersebut tidak ada dalam struktur data.

2. Jelaskan perbedaan dari Sequential Search, Binary Search, dan Interpolation Search sebagai metode-metode searching!

Jawab :

Sequential search adalah metode pencarian sederhana yang dilakukan dengan membandingkan setiap elemen pada struktur data satu per satu. Jika nilai ditemukan, maka program akan memberikan output indeks dari nilai tersebut. Namun, jika nilai tidak ditemukan, program akan memberikan output bahwa nilai tersebut tidak ditemukan.

Binary search bekerja dengan mengurutkan struktur data terlebih dahulu, kemudian mencari nilai dengan cara membagi struktur data menjadi dua bagian secara terus menerus sampai nilai yang dicari ditemukan. Binary search lebih efisien daripada sequential search, terutama pada struktur data yang besar. Namun, syarat untuk menggunakan binary search adalah struktur data harus sudah terurut terlebih dahulu.

Interpolation search mencari nilai dengan mengestimasi lokasi nilai tersebut berada di dalam struktur data. Metode ini menghitung nilai perkiraan lokasi nilai yang akan dicari, dan mencari nilai tersebut secara berulang-ulang hingga ditemukan. Interpolation search lebih cepat dibandingkan sequential search dan bisa digunakan pada data yang besar dan memiliki distribusi yang merata.

3. Jelaskan apa saja yang mempengaruhi kecepatan proses pencarian data didalam penyimpanan data!

Jawab :

Kecepatan proses pencarian data dalam penyimpanan data dipengaruhi oleh beberapa faktor, seperti ukuran data, metode pencarian yang digunakan, jenis penyimpanan data, dan kemampuan mesin untuk memproses data. Semakin besar ukuran data, semakin lama proses pencarian data akan memakan waktu. Selain itu, metode pencarian juga mempengaruhi kecepatan proses pencarian. Binary search dan interpolation search lebih cepat daripada sequential search karena keduanya memiliki kemampuan membagi data menjadi dua bagian atau memperkirakan letak data yang

akan dicari. Jenis penyimpanan data juga mempengaruhi kecepatan proses pencarian. Penyimpanan data pada media yang lebih cepat seperti SSD akan lebih cepat dalam proses pencarian data dibandingkan media penyimpanan yang lebih lambat seperti HDD. Kemampuan mesin untuk memproses data juga mempengaruhi kecepatan proses pencarian data.

4. Buatlah contoh program sederhana menggunakan Searching!

Jawab :

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

int binarySearch(int arr[], int n, int x) {
    int l = 0, r = n-1;
    while (l <= r) {
        int mid = l + (r - l) / 2;
        if (arr[mid] == x)
            return mid;
        if (arr[mid] < x)
            l = mid + 1;
        else
            r = mid - 1;
    }
    return -1;
}

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int arr[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    int n = sizeof(arr) / sizeof(arr[0]);
    int x;

    cout << "Deretan angka: ";
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }

    cout << "\n\nMasukkan angka yang ingin dicari: ";
    cin >> x;

    int result = binarySearch(arr, n, x);

    if (result == -1) {
        cout << "Nilai tidak ditemukan." << endl;
    } else {
        cout << "Nilai ditemukan pada indeks ke-" << result << endl;
    }

    return 0;
}
```

```
Nama      : Nova Ardiansyah  
NIM       : 211011401309  
=====
```

Deretan angka: 1 2 3 4 5 6 7 8 9 10

Masukkan angka yang ingin dicari: 5
Nilai ditemukan pada indeks ke-4

C. TUGAS PRAKTIKUM

- Lat6_1

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

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int Nilai[15];
    int i, N, angka, Bilangan;

    cout << "Masukkan banyaknya data : ";
    cin >> N;

    for (i = 0; i < N; i++) {
        cout << "Masukkan data ke-" << i + 1 << " : ";
        cin >> Nilai[i];
    }

    cout << "Data yang telah dimasukkan adalah : ";
    for (i = 0; i < N; i++) {
        cout << Nilai[i] << " ";
    }

    cout << "\nMasukkan nilai yang akan dicari : ";
    cin >> Bilangan;

    i = 0;
    do {
        if (Nilai[i] == Bilangan) angka = i;
        i++;
    } while (i < N);

    if (Nilai[angka] == Bilangan) {
        cout << "Data ditemukan pada indeks ke-" << angka << endl;
    } else {
        cout << "Data tidak ditemukan" << endl;
    }
}
```

```
Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Masukkan banyaknya data : 5
Masukkan data ke-1 : 10
Masukkan data ke-2 : 20
Masukkan data ke-3 : 30
Masukkan data ke-4 : 40
Masukkan data ke-5 : 50
Data yang telah dimasukkan adalah : 10 20 30 40 50
Masukkan nilai yang akan dicari : 40
Data ditemukan pada indeks ke-3
```

- Lat6_2

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

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int Nilai[15];
    int Posisi[15];
    int i, N, angka, Bilangan, Banyak = 0;
    bool ketemu;

    cout << "Masukkan banyaknya data : ";
    cin >> N;

    for (i = 0; i < N; i++) {
        cout << "Masukkan data ke-" << i + 1 << " : ";
        cin >> Nilai[i];
    }

    cout << "Data yang telah dimasukkan adalah : ";
    for (i = 0; i < N; i++) {
        cout << Nilai[i] << " ";
    }

    cout << "\nMasukkan nilai yang akan dicari : ";
    cin >> Bilangan;

    for (i = 0; i < N; i++) {
        if (Nilai[i] == Bilangan) {
            ketemu = true;
            Posisi[Banyak] = i;
            Banyak++;
        }
    }

    if (ketemu) {
        cout << "Data ditemukan pada indeks ";
        for (i = 0; i < Banyak; i++) {
            cout << "ke-" << Posisi[i] << " ";
        }
        cout << "\nBanyaknya data yang ditemukan: " << Banyak << endl;
    } else {
        cout << "Data tidak ditemukan" << endl;
    }
}
```

```

Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Masukkan banyaknya data : 5
Masukkan data ke-1 : 11
Masukkan data ke-2 : 12
Masukkan data ke-3 : 11
Masukkan data ke-4 : 13
Masukkan data ke-5 : 14
Data yang telah dimasukkan adalah : 11 12 11 13 14
Masukkan nilai yang akan dicari : 11
Data ditemukan pada indeks ke-0 ke-2
Banyaknya data yang ditemukan: 2

```

- Lat6_3

```

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

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int Nilai[15];
    int i, j, N;
    int temp, awal, akhir, tengah, Bilangan;

    cout << "Masukkan banyaknya data : ";
    cin >> N;

    for (i = 0; i < N; i++) {
        cout << "Masukkan data ke-" << i + 1 << " : ";
        cin >> Nilai[i];
    }

    cout << "Data sebelum diurutkan : ";
    for (i = 0; i < N; i++) {
        cout << Nilai[i] << " ";
    }

    for (i = 0; i < N - 1; i++) {
        for (j = i + 1; j < N; j++) {
            if (Nilai[i] > Nilai[j]) {
                temp = Nilai[i];
                Nilai[i] = Nilai[j];
                Nilai[j] = temp;
            }
        }
    }

    cout << "\nData setelah diurutkan : ";
    for (i = 0; i < N; i++) {
        cout << Nilai[i] << " ";
    }
}

```

```

}

cout << "\nMasukkan nilai yang akan dicari : ";
cin >> Bilangan;

awal = 0;
akhir = N - 1;

do {
    tengah = (awal + akhir) / 2;
    if (Nilai[tengah] == Bilangan) {
        cout << "Data ditemukan pada indeks ke-" << tengah << endl;
        break;
    } else if (Nilai[tengah] < Bilangan) {
        awal = tengah + 1;
    } else {
        akhir = tengah - 1;
    }
} while (awal <= akhir);
}

```

```

Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Masukkan banyaknya data : 5
Masukkan data ke-1 : 15
Masukkan data ke-2 : 11
Masukkan data ke-3 : 17
Masukkan data ke-4 : 12
Masukkan data ke-5 : 10
Data sebelum diurutkan : 15 11 17 12 10
Data setelah diurutkan : 10 11 12 15 17
Masukkan nilai yang akan dicari : 12
Data ditemukan pada indeks ke-2

```

- Lat6_4

```

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

// * Struct Declaration
struct Pegawai {
    string NIP;
    string Nama;
    string Alamat;
    string Agama;
    string Jabatan;
};

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    const int MAX_PEGAWAI = 10;
}

```

```

Pegawai daftar_pegawai[MAX_PEGAWAI]; // * Array of struct

int n;
string nip_cari;
bool found = false;

cout << "Masukkan jumlah pegawai: ";
cin >> n;

for (int i = 0; i < n; i++) {
    cout << "Pegawai ke-" << i+1 << endl;
    cout << "NIP\t: ";
    cin >> daftar_pegawai[i].NIP;

    cout << "Nama\t: ";
    cin.ignore();
    getline(cin, daftar_pegawai[i].Nama);

    cout << "Alamat\t: ";
    getline(cin, daftar_pegawai[i].Alamat);

    cout << "Agama\t: ";
    getline(cin, daftar_pegawai[i].Agama);

    cout << "Jabatan\t: ";
    getline(cin, daftar_pegawai[i].Jabatan);

    cout << endl;
}

// * Sequential search
cout << "Masukkan NIP pegawai yang ingin dicari: ";
cin >> nip_cari;

for (int i = 0; i < n; i++) {
    if (daftar_pegawai[i].NIP == nip_cari) {
        found = true;
        cout << "Biodata Pegawai dengan NIP " << nip_cari << endl;
        cout << "Nama\t: " << daftar_pegawai[i].Nama << endl;
        cout << "Alamat\t: " << daftar_pegawai[i].Alamat << endl;
        cout << "Agama\t: " << daftar_pegawai[i].Agama << endl;
        cout << "Jabatan\t: " << daftar_pegawai[i].Jabatan << endl;
        break;
    }
}

if (!found) {
    cout << "Pegawai dengan NIP " << nip_cari << " tidak ditemukan" << endl;
}

return 0;
}

```



```
Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Masukkan jumlah pegawai: 2
Pegawai ke-1
NIP       : 211011
Nama      : Nova
Alamat    : Tangerang
Agama    : Islam
Jabatan   : Developer

Pegawai ke-2
NIP       : 211012
Nama      : RA
Alamat    : Serang
Agama    : Islam
Jabatan   : Administrator

Masukkan NIP pegawai yang ingin dicari: 211011
Biodata Pegawai dengan NIP 211011
Nama      : Nova
Alamat    : Tangerang
Agama    : Islam
Jabatan   : Developer
```

LAPORAN PRAKTIKUM STRUKTUR DATA

LAPORAN AKHIR PRAKTIKUM

Pertemuan ke-06

Searching



Disusun Oleh:

Nama Lengkap : Nova Ardiansyah

NIM : 211011401309

Kelas : 04-TPLE008

TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PAMULANG

Jl. Raya Puspitek No. 11 Buaran, Serpong Telp. (021) 7412566, Fax. (021) 7412566
Tangerang Selatan - Banten

A. TUGAS AKHIR

1. Buatlah program untuk melakukan pencarian terhadap data 12, 15, dan 37, dari sederetan data 34, 8, 50, 74, 87, 90, 12, 25, 20, 30, 35, 45, 40, 22, 29, 72, 60, 55, 53, 12, 32, 33, 12, 41, 12, Jika data yang dicari terdapat lebih dari satu, tentukan banyaknya dan sebutkan berada pada posisi berapa saja data yang dicari berada pada sederetan data.

Jawab :

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

int main() {
    int data[] = {34, 8, 50, 74, 87, 90, 12, 25, 20, 30, 35, 45, 40, 22, 29, 72, 60, 55, 53, 12, 32,
33, 12, 12, 41, 12};
    int n = sizeof(data) / sizeof(data[0]); // * Menghitung ukuran array
    int count_12 = 0, count_15 = 0, count_37 = 0;
    int posisi_12[10], posisi_15[10], posisi_37[10];

    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    cout << "Data yang ada pada array: " << endl;
    for (int i = 0; i < n; i++) {
        cout << data[i] << " ";
    }
    cout << endl << endl;

    for (int i = 0; i < n; i++) {
        if (data[i] == 12) {
            count_12++;
            cout << "Data 12 ditemukan pada posisi ke-" << i << endl;
        } else if (data[i] == 15) {
            count_15++;
            cout << "Data 15 ditemukan pada posisi ke-" << i << endl;
        } else if (data[i] == 37) {
            count_37++;
            cout << "Data 37 ditemukan pada posisi ke-" << i << endl;
        }
    }

    cout << endl;
    cout << "Banyaknya data 12 yang ditemukan: " << count_12 << endl;
    cout << "Banyaknya data 15 yang ditemukan: " << count_15 << endl;
    cout << "Banyaknya data 37 yang ditemukan: " << count_37 << endl;

    return 0;
}
```

```
Nama      : Nova Ardiansyah
NIM       : 211011401309
=====
```

```
Data yang ada pada array:
```

```
34 8 50 74 87 90 12 25 20 30 35 45 40 22 29 72 60 55 53 12 32 33 12 12 4
```

```
Data 12 ditemukan pada posisi ke-6
```

```
Data 12 ditemukan pada posisi ke-19
```

```
Data 12 ditemukan pada posisi ke-22
```

```
Data 12 ditemukan pada posisi ke-23
```

```
Data 12 ditemukan pada posisi ke-25
```

```
Banyaknya data 12 yang ditemukan: 5
```

```
Banyaknya data 15 yang ditemukan: 0
```

```
Banyaknya data 37 yang ditemukan: 0
```

2. KESIMPULAN

Searching atau pencarian adalah proses untuk mencari nilai tertentu pada suatu struktur data. Metode-metode searching yang dapat digunakan antara lain sequential search, binary search, dan interpolation search. Sequential search adalah metode sederhana yang dilakukan dengan membandingkan setiap elemen pada struktur data satu per satu, sedangkan binary search bekerja dengan mengurutkan struktur data terlebih dahulu dan membaginya menjadi dua bagian secara terus menerus sampai nilai yang dicari ditemukan. Interpolation search mencari nilai dengan mengestimasi lokasi nilai tersebut berada di dalam struktur data.

Kecepatan proses pencarian data dipengaruhi oleh beberapa faktor, seperti ukuran data, metode pencarian yang digunakan, jenis penyimpanan data, dan kemampuan mesin untuk memproses data. Binary search dan interpolation search lebih cepat dibandingkan sequential search karena keduanya memiliki kemampuan membagi data menjadi dua bagian atau memperkirakan letak data yang akan dicari. Jenis penyimpanan data juga mempengaruhi kecepatan proses pencarian.

LAPORAN PRAKTIKUM STRUKTUR DATA

LAPORAN AWAL PRAKTIKUM

Pertemuan ke-07

Sorting



Disusun Oleh:

Nama Lengkap : Nova Ardiansyah

NIM : 211011401309

Kelas : 04-TPLE008

TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PAMULANG

Jl. Raya Puspitek No. 11 Buaran, Serpong Telp. (021) 7412566, Fax. (021) 7412566
Tangerang Selatan - Banten

A. RANGKUMAN MATERI

Sorting adalah proses mengatur atau menyusun elemen-elemen dalam satu kumpulan data dengan aturan tertentu. Dalam program C++, terdapat beberapa algoritma sorting yang digunakan untuk mengurutkan array atau struktur data lainnya, seperti Bubble Sort, Selection Sort, dan Insertion Sort. Keuntungan menggunakan algoritma sorting yaitu menghemat waktu dan memudahkan dalam mencari nilai tertentu pada suatu array yang sudah terurut.

Algoritma Bubble Sort merupakan salah satu metode pengurutan sederhana dalam pemrograman C++. Pengurutan dilakukan dengan cara membandingkan 2 angka dalam sebuah array, jika angka pertama lebih besar dari angka kedua, maka mereka akan ditukar posisinya. Proses ini berlangsung terus-menerus hingga tidak ada nilai yang ditukar lagi. Algoritma Bubble Sort sangat mudah dipahami dan diimplementasikan oleh pemula, tetapi memiliki kompleksitas waktu yang tinggi saat jumlah data semakin banyak.

Selain itu, algoritma Selection Sort juga sering digunakan dalam pemrograman C++ untuk mengurutkan data. Cara kerja Selection Sort yaitu mencari angka terkecil dalam array dan menukar posisi angka tersebut dengan angka pertama dalam array. Kemudian proses diulang untuk subarray yang belum terurut sampai seluruh array tersusun dengan benar. Meskipun lebih cepat daripada Bubble Sort, Selection Sort masih sulit diimplementasikan untuk array yang sangat besar. Oleh karena itu, untuk data yang bersifat dinamis dan kompleks, diperlukan algoritma sorting yang lebih efisien seperti Merge Sort atau Quick Sort.

B. TUGAS PENDAHULUAN

1. Apa yang dimaksud dengan Sorting!

Jawab :

Sorting atau pengurutan merupakan proses mengatur atau menyusun elemen-elemen dalam suatu kumpulan data dengan aturan tertentu. Dalam konteks pemrograman, sorting umumnya digunakan untuk mengurutkan elemen-elemen dalam array atau struktur data lainnya sehingga memudahkan dalam mencari nilai tertentu atau membuat operasi lain pada data yang sudah terurut. Ada beberapa algoritma

sorting yang dapat digunakan, seperti Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, dan Quick Sort, masing-masing dengan kelebihan dan kekurangan yang berbeda-beda. Pemilihan algoritma sorting harus disesuaikan dengan kebutuhan dan karakteristik dari data yang akan diurutkan agar dapat menghasilkan waktu eksekusi yang optimal.

2. Jelaskan Perbedaan dari pengurutan Internal dan Pengurutan Eksternal!

Jawab :

Pengurutan internal adalah pengurutan elemen-elemen dalam struktur data di dalam memori komputer secara keseluruhan. Sedangkan pengurutan eksternal adalah pengurutan elemen-elemen dalam struktur data yang lebih besar dari kapasitas memori komputer, dengan melibatkan operasi baca-tulis pada media penyimpanan eksternal seperti hard disk atau CD-ROM.

Perbedaan mendasar antara keduanya adalah ketergantungan terhadap media penyimpanan data, dimana pengurutan internal hanya menggunakan memori komputer dan lebih cepat, sedangkan pengurutan eksternal melibatkan media penyimpanan eksternal dan lebih lambat tetapi cocok untuk dataset yang sangat besar.

3. Jelaskan perbedaan metode-metode Sorting seperti: Bubble Sort, Quick Sort, Selection Sort, Merge Sort, Tree Sort, Maximum Sort, Insertion Sort!

Jawab :

- **Bubble Sort:** Metode sorting sederhana di mana angka dibandingkan secara berpasangan dan kemudian ditukar jika perlu. Element-elemen besar akan bergeser ke akhir array, sehingga digunakan untuk pengurutan data terbatas.

- **Quick Sort:** Algoritma divide and conquer di mana elemen terakhir digunakan sebagai pivot point untuk membagi array menjadi dua bagian, satu dengan nilai lebih rendah dari pivot dan satunya lagi dengan nilainya lebih tinggi. Kemudian kedua bagian juga dicacah lagi menjadi bagian yang lebih kecil dan diurutkan secara rekursif.

- **Selection Sort:** Algoritma sorting sederhana yang bekerja dengan memilih elemen terkecil dari array dan menukar posisinya dengan elemen pertama. Selanjutnya, elemen kedua terkecil ditemukan di sisa array, dan ditukar dengan elemen kedua, dan seterusnya hingga seluruh array terurut.

- **Merge Sort:** Algoritma sorting dengan strategi divide and conquer di mana array dibagi menjadi dua bagian dan diurutkan secara independen, kemudian dua bagian tersebut digabungkan kembali menjadi satu array yang terurut.

- **Tree Sort:** Algoritma sorting mirip dengan binary search tree yaitu menyimpan setiap elemen sebagai simpul pada pohon biner, kemudian mengurutkan elemen-elemen tersebut dengan menjelajahi pohon secara in-order traversal.

- **Maximum Sort:** Metode sorting dalam pengembangan game di mana elemen-elemen diurutkan dengan cara memperbesar elemen-elemen terakhir sambil memindahkan elemen-elemen ke atas.

- **Insertion Sort:** Algoritma sorting sederhana yang mengambil element dari array dan memasukkannya pada tempat yang sudah terurut sebelumnya. Array dibagi menjadi dua bagian, satu bagian yang sudah diurutkan dan satunya yang belum diurutkan, lalu elemen-elemen baru ditempatkan pada posisi yang tepat di dalam array yang telah diurutkan secara increment.

Perbedaan utama antara metode-metode tersebut adalah pada strategi pengurutan yang digunakan, efisiensi waktu dan ruang, serta kompleksitas algoritmanya. Pemilihan metode sorting harus disesuaikan dengan karakteristik data yang akan diurutkan agar dapat menghasilkan waktu eksekusi yang optimal.

4. Buatlah contoh program sederhana menggunakan Sorting!

Jawab :

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

void bubbleSort(int arr[], int n)
```



```

{
    for(int i = 0; i < n-1; i++)
    {
        for(int j = 0; j < n-i-1; j++)
        {
            if(arr[j] > arr[j+1])
            {
                int temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }
        }
    }
}

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int arr[] = {64, 25, 12, 22, 11};
    int n = sizeof(arr) / sizeof(arr[0]);

    cout << "Nilai sebelum diurutkan: ";
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }

    bubbleSort(arr, n);
    cout << "\nNilai setelah diurutkan: ";
    for(int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }

    return 0;
}

```

```

Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Nilai sebelum diurutkan: 64 25 12 22 11
Nilai setelah diurutkan: 11 12 22 25 64

```

C. TUGAS PRAKTIKUM

- Lat7_1

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

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int Nilai[20];
    int i, j, k, N;
    int temp;
    bool tukar;

    cout << "Masukkan banyak data : ";
    cin >> N;

    for (i = 0; i < N; i++) {
        cout << "Masukkan data ke-" << i + 1 << " : ";
        cin >> Nilai[i];
    }

    cout << endl;
    cout << "Data sebelum diurutkan : ";
    for (i = 0; i < N; i++)
    {
        cout << Nilai[i] << " ";
    }

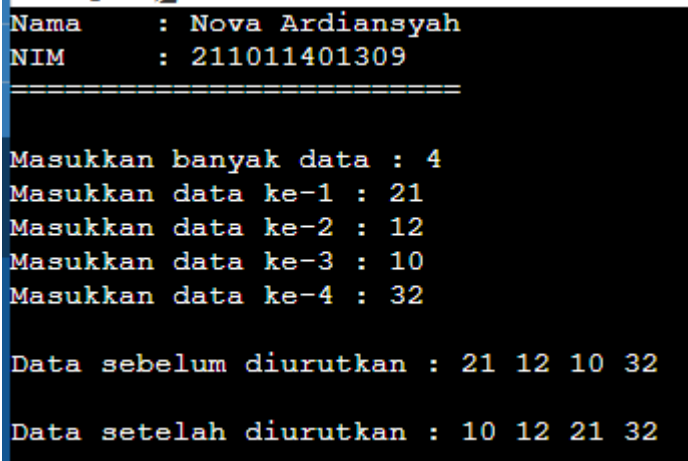
    i = 0;
    for (j = N - 1; j >= 0; j--)
    {
        tukar = false;
        for (k = 0; k < j; k++) {
            if (Nilai[k] > Nilai[k + 1]) {
                temp = Nilai[k];
                Nilai[k] = Nilai[k + 1];
                Nilai[k + 1] = temp;
                tukar = true;
            }
        }
    }

    if (tukar == false) {
        break;
    }

    cout << endl;
    i++;
}

cout << "Data setelah diurutkan : ";
for (i = 0; i < N; i++)
{
    cout << Nilai[i] << " ";
}
```

```
return 0;
}
```



The screenshot shows the output of a C++ program. It starts with the user's name and NIM being printed. Then, it prompts the user to enter the number of data points, which is 4. The user then enters four data points: 21, 12, 10, and 32. The program then displays the data before sorting (21 12 10 32) and after sorting (10 12 21 32).

```
Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Masukkan banyak data : 4
Masukkan data ke-1 : 21
Masukkan data ke-2 : 12
Masukkan data ke-3 : 10
Masukkan data ke-4 : 32

Data sebelum diurutkan : 21 12 10 32
Data setelah diurutkan : 10 12 21 32
```

- Lat7_2

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

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int Nilai[20];
    int i, j, k, N;
    int temp;
    bool tukar;

    cout << "Masukkan banyak data : ";
    cin >> N;

    for (i = 0; i < N; i++) {
        cout << "Masukkan data ke-" << i + 1 << " : ";
        cin >> Nilai[i];
    }

    cout << endl;
    cout << "Data sebelum diurutkan : ";
    for (i = 0; i < N; i++)
    {
        cout << Nilai[i] << " ";
    }

    i = 0;
    for (j = N - 1; j >= 0; j--)
    {
        tukar = false;
        for (k = 0; k < j; k++) {
            if (Nilai[k] < Nilai[k + 1]) {
                temp = Nilai[k];
                Nilai[k] = Nilai[k + 1];
                Nilai[k + 1] = temp;
            }
        }
    }
}
```

```

        Nilai[k + 1] = temp;
        tukar = true;
    }
}

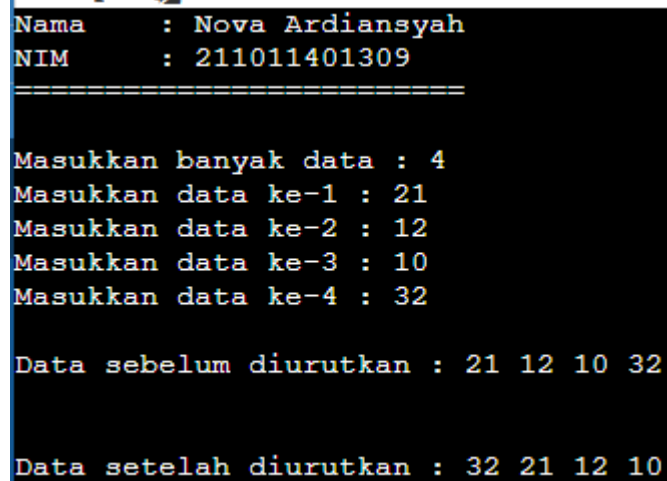
if (tukar == false) {
    break;
}

cout << endl;
i++;
}

cout << "Data setelah diurutkan : ";
for (i = 0; i < N; i++) {
    cout << Nilai[i] << " ";
}

return 0;
}

```



```

Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Masukkan banyak data : 4
Masukkan data ke-1 : 21
Masukkan data ke-2 : 12
Masukkan data ke-3 : 10
Masukkan data ke-4 : 32

Data sebelum diurutkan : 21 12 10 32

Data setelah diurutkan : 32 21 12 10

```

- Lat7_3

```

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

void Cetak(int data[], int n)
{
    int i = 0;
    for (i = 0; i < n; i++)
    {
        cout << data[i] << " ";
    }
}

int partisi(int data[], int p, int r)
{
    int x, i, j, temp;
    x = data[p];
    i = p;
    j = r;

```

```

while (true)
{
    while (data[j] > x) {
        j--;
    }

    while (data[i] < x) {
        i++;
    }

    if (i < j) {
        temp = data[i];
        data[i] = data[j];
        data[j] = temp;
    } else {
        return j;
    }
}

void QuickSort(int data[], int p, int r)
{
    int q;
    if (p < r) {
        q = partisi(data, p, r);
        QuickSort(data, p, q);
        QuickSort(data, q + 1, r);
    }
}

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int Nilai[20];
    int i, N;

    cout << "Masukkan banyak data : ";
    cin >> N;

    for (i = 0; i < N; i++) {
        cout << "Masukkan data ke-" << i + 1 << " : ";
        cin >> Nilai[i];
    }

    cout << endl;
    cout << "Data sebelum diurutkan : ";
    for (i = 0; i < N; i++) {
        cout << Nilai[i] << " ";
    }

    QuickSort(Nilai, 0, N - 1);

    cout << "\nData setelah diurutkan : ";
    Cetak(Nilai, N);

    return 0;
}

```

```
}  
Nama      : Nova Ardiansyah  
NIM       : 211011401309  
=====
```

```
Masukkan banyak data : 4  
Masukkan data ke-1 : 21  
Masukkan data ke-2 : 12  
Masukkan data ke-3 : 10  
Masukkan data ke-4 : 32
```

```
Data sebelum diurutkan : 21 12 10 32  
Data setelah diurutkan : 10 12 21 32
```

- Lat7_4

```
#include <iostream>  
#include <string>  
using namespace std;  
  
void Cetak(int data[], int n)  
{  
    int i = 0;  
    for (i = 0; i < n; i++)  
    {  
        cout << data[i] << " ";  
    }  
}  
  
int partisi(int data[], int p, int r)  
{  
    int x, i, j, temp;  
    x = data[p];  
    i = p;  
    j = r;  
  
    while (true)  
    {  
        while (data[j] > x) {  
            j--;  
        }  
  
        while (data[i] < x) {  
            i++;  
        }  
  
        if (i < j) {  
            temp = data[i];  
            data[i] = data[j];  
            data[j] = temp;  
        } else {  
            return j;  
        }  
    }  
}  
  
void QuickSort(int data[], int p, int r)
```

```

{
    int q;
    if (p < r) {
        q = partisi(data, p, r);
        QuickSort(data, p, q);
        QuickSort(data, q + 1, r);
    }
}

void pilihan1()
{
    int Nilai[20];
    int i, j, k, N;
    int temp;
    bool tukar;

    cout << "Masukkan banyak data : ";
    cin >> N;

    for (i = 0; i < N; i++) {
        cout << "Masukkan data ke-" << i + 1 << " : ";
        cin >> Nilai[i];
    }

    cout << endl;
    cout << "Data sebelum diurutkan : ";
    for (i = 0; i < N; i++)
    {
        cout << Nilai[i] << " ";
    }

    i = 0;
    for (j = N - 1; j >= 0; j--)
    {
        tukar = false;
        for (k = 0; k < j; k++) {
            if (Nilai[k] > Nilai[k + 1]) {
                temp = Nilai[k];
                Nilai[k] = Nilai[k + 1];
                Nilai[k + 1] = temp;
                tukar = true;
            }
        }
    }

    if (tukar == false) {
        break;
    }

    cout << endl;
    i++;
}

cout << "Data setelah diurutkan : ";
for (i = 0; i < N; i++)
{
    cout << Nilai[i] << " ";
}
}

void pilihan2()

```

```

{
    int Nilai[20];
    int i, j, k, N;
    int temp;
    bool tukar;

    cout << "Masukkan banyak data : ";
    cin >> N;

    for (i = 0; i < N; i++) {
        cout << "Masukkan data ke-" << i + 1 << " : ";
        cin >> Nilai[i];
    }

    cout << endl;
    cout << "Data sebelum diurutkan : ";
    for (i = 0; i < N; i++)
    {
        cout << Nilai[i] << " ";
    }

    i = 0;
    for (j = N - 1; j >= 0; j--)
    {
        tukar = false;
        for (k = 0; k < j; k++) {
            if (Nilai[k] < Nilai[k + 1]) {
                temp = Nilai[k];
                Nilai[k] = Nilai[k + 1];
                Nilai[k + 1] = temp;
                tukar = true;
            }
        }
    }

    if (tukar == false) {
        break;
    }

    i++;
}

cout << "\nData setelah diurutkan : ";
for (i = 0; i < N; i++) {
    cout << Nilai[i] << " ";
}
}

void pilihan3()
{
    int Nilai[20];
    int i, N;

    cout << "Masukkan banyak data : ";
    cin >> N;

    for (i = 0; i < N; i++) {
        cout << "Masukkan data ke-" << i + 1 << " : ";
        cin >> Nilai[i];
    }
}

```



```

    cout << endl;
    cout << "Data sebelum diurutkan : ";
    for (i = 0; i < N; i++) {
        cout << Nilai[i] << " ";
    }

    QuickSort(Nilai, 0, N - 1);

    cout << "\nData setelah diurutkan : ";
    Cetak(Nilai, N);
}

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int pilihan;

    cout << "Pilih metode pengurutan data: \n";
    cout << "1. Bubble Sort Menaik\n";
    cout << "2. Bubble Sort Menurun\n";
    cout << "3. Quick Sort Menaik\n";

    cout << "Pilihan: ";
    cin >> pilihan;

    switch (pilihan)
    {
    case 1:
        cout << "\n===== 1. Bubble Sort Menaik =====\n\n";
        pilihan1();
        break;
    case 2:
        cout << "\n===== 2. Bubble Sort Menurun =====\n\n";
        pilihan2();
        break;
    case 3:
        cout << "\n===== 3. Quick Sort Menaik =====\n\n";
        pilihan3();
        break;
    default:
        cout << "\n===== Pilihan tidak tersedia =====\n\n";
        break;
    }

    return 0;
}

```

```
Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Pilih metode pengurutan data:
1. Bubble Sort Menaik
2. Bubble Sort Menurun
3. Quick Sort Menaik
Pilihan: 2

===== 2. Bubble Sort Menurun =====

Masukkan banyak data : 4
Masukkan data ke-1 : 21
Masukkan data ke-2 : 12
Masukkan data ke-3 : 10
Masukkan data ke-4 : 32

Data sebelum diurutkan : 21 12 10 32
Data setelah diurutkan : 32 21 12 10
```

LAPORAN PRAKTIKUM STRUKTUR DATA

LAPORAN AKHIR PRAKTIKUM

Pertemuan ke-07

Sorting



Disusun Oleh:

Nama Lengkap : Nova Ardiansyah
NIM : 211011401309
Kelas : 04-TPLE008

TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PAMULANG

Jl. Raya Puspitek No. 11 Buaran, Serpong Telp. (021) 7412566, Fax. (021) 7412566
Tangerang Selatan - Banten

A. TUGAS AKHIR

1. Buatlah program untuk mengurutkan sederetan data: 34, 12, 56, 78, 6, 43, 32, 29, 90, 50, 55, 75, 85, 95, 25!

Jawab :

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

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

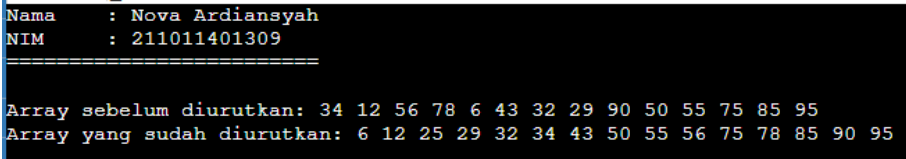
    int arr[] = {34, 12, 56, 78, 6, 43, 32, 29, 90, 50, 55, 75, 85, 95, 25};
    int n = sizeof(arr) / sizeof(arr[0]);

    cout << "Array sebelum diurutkan: ";
    for (int i = 0; i < n - 1; i++) {
        cout << arr[i] << " ";
    }

    // * Bubble Sort
    for (int i = 0; i < n - 1; i++)
    {
        for (int j = 0; j < n - i - 1; j++)
        {
            if (arr[j] > arr[j + 1])
            {
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }

    cout << "\nArray yang sudah diurutkan: ";
    for (int i = 0; i < n; i++)
    {
        cout << arr[i] << " ";
    }
    cout << endl;

    return 0;
}
```



The screenshot shows the output of the C++ program. It displays the user's name and NIM, followed by a separator line. Then, it shows the array before sorting and the array after sorting using bubble sort. The arrays are: [34, 12, 56, 78, 6, 43, 32, 29, 90, 50, 55, 75, 85, 95, 25] and [6, 12, 25, 29, 32, 34, 43, 50, 55, 56, 75, 78, 85, 90, 95].

```
Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Array sebelum diurutkan: 34 12 56 78 6 43 32 29 90 50 55 75 85 95
Array yang sudah diurutkan: 6 12 25 29 32 34 43 50 55 56 75 78 85 90 95
```

2. KESIMPULAN

Dalam pemrograman C++, sorting digunakan untuk mengurutkan elemen-elemen dalam array atau struktur data lainnya. Terdapat beberapa algoritma sorting yang dapat digunakan seperti Bubble Sort, Selection Sort, Merge Sort, dan Quick Sort. Meskipun algoritma sorting yang sederhana seperti Bubble Sort dan Selection Sort mudah dipahami dan diimplementasikan oleh pemula, tetapi memiliki kompleksitas waktu yang tinggi saat jumlah data semakin banyak. Sedangkan algoritma sorting yang lebih efisien seperti Merge Sort atau Quick Sort dapat digunakan untuk data yang bersifat dinamis dan kompleks. Oleh karena itu, penggunaan algoritma sorting harus disesuaikan dengan kebutuhan dan karakteristik dari data yang akan diurutkan.