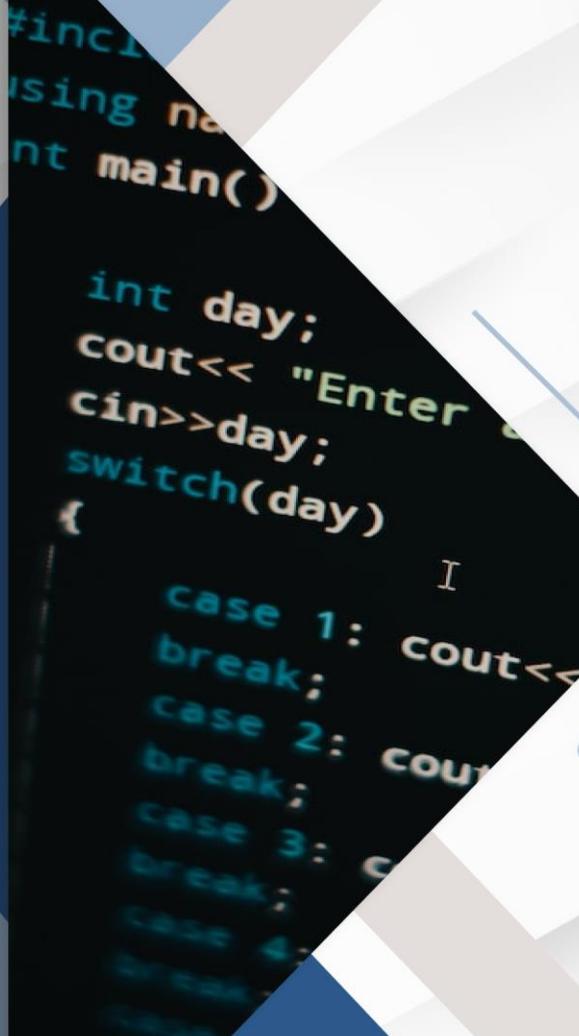


20
23
20
24



```
#include <iostream>
using namespace std;
int main()
{
    int day;
    cout<< "Enter a day: ";
    cin>>day;
    switch(day)
    {
        case 1: cout<< "Sunday";
        break;
        case 2: cout<< "Monday";
        break;
        case 3: cout<< "Tuesday";
        break;
        case 4: cout<< "Wednesday";
        break;
        case 5: cout<< "Thursday";
        break;
        case 6: cout<< "Friday";
        break;
        case 7: cout<< "Saturday";
        break;
        default: cout<< "Unknown day";
    }
}
```

MODUL PRAKTIKUM

ALGORITMA & STRUKTUR DATA

Tim Penyusun

- Ir. St. Hajrah Mansyur, S.Kom., M.Cs., MTA
- Ir. Abdul Rachman Manga, S.Kom., M.T., MTA., MCF
- Ir. Huzain Azis, S.Kom., M.Cs., MTA
- Lutfi Budi Ilmawan, S.Kom., M.Cs., MTA
- Tim Asisten Laboratorium

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Kuasa, yang telah memberikan rahmat-Nya sehingga Modul Praktikum **Algoritma dan Struktur Data** untuk mahasiswa/i Program Studi Sistem Informasi Fakultas Ilmu Komputer Universitas Muslim Indonesia ini dapat diselesaikan dengan sebaik-baiknya.

Modul praktikum ini dibuat sebagai pedoman dalam melakukan kegiatan praktikum **Algoritma dan Struktur Data** yang merupakan kegiatan penunjang mata kuliah pada Program Studi Sistem Informasi. Modul praktikum ini diharapkan dapat membantu mahasiswa/i dalam mempersiapkan dan melaksanakan praktikum dengan lebih baik, terarah, dan terencana. Pada setiap topik telah ditetapkan capaian pembelajaran mata kuliah pelaksanaan praktikum dan semua kegiatan yang harus dilakukan oleh mahasiswa/i serta teori singkat untuk memperdalam pemahaman mahasiswa/i mengenai materi yang dibahas.

Penyusun menyakini bahwa dalam pembuatan Modul Praktikum **Algoritma dan Struktur Data** ini masih jauh dari sempurna. Oleh karena itu penyusun mengharapkan kritik dan saran yang membangun guna penyempurnaan modul praktikum ini dimasa yang akan datang.

Akhir kata, penyusun mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung.

Makassar, Maret 2024

Tim Penyusun

TATA TERTIB PELAKSANAAN PRAKTIKUM

Tata Tertib Pelaksanaan Praktikum pada Laboratorium Terpadu Fakultas Ilmu Komputer UMI adalah sebagai berikut:

1. Seluruh Pengguna laboratorium harus dalam keadaan sehat tidak menunjukkan gejala sakit (batuk, hidung tersumbat, dan suhu badan diatas 37°C).
2. Praktikan hanya diizinkan melaksanakan praktikum apabila :
 - a. Pria
 - Berpakaian rapi memakai kemeja putih polos;
 - Menggunakan celana kain berwarna hitam bukan dari bahan jeans/semi jeans;
 - Rambut rapi dan tidak panjang;
 - b. Wanita
 - Berpakaian rapi memakai kemeja tunik putih polos (tidak transparan)
 - Memakai Jilbab Segitiga Hitam (bukan pasmina) dan menutupi dada.
 - Menggunakan Rok Panjang berwarna hitam yang tidak terbelah dan tidak span serta bukan dari bahan jeans/semi jeans;
 - Memakai kaos kaki dengan tinggi minimal 10 cm di atas mata kaki;
3. Ketika memasuki dan selama berada dalam ruangan, praktikan diwajibkan :
 - Tenang, tertib, dan sopan;
 - Tidak mengganggu praktikan lain yang sedang melaksanakan praktikum;
 - Tidak diperbolehkan merokok, membawa makanan / minuman senjata tajam dan senjata api ke dalam ruangan praktikum;
 - Tidak diperbolehkan membawa *handphone* ke meja praktikum dan *handphone* dalam mode senyap;
 - Tidak diperbolehkan membawa media penyimpanan eksternal atau *flashdisk* ke meja praktikum tanpa seizin Dosen Pengampu atau Asisten;
4. Dilarang membawa, mengambil, serta memindahkan perangkat yang digunakan pada saat praktikum tanpa instruksi dari Dosen Pengampu atau Asisten.
5. Toleransi keterlambatan praktikan maksimal 5 menit.
6. Praktikan berada diarea laboratorium dengan mengikuti jadwal yang telah ditentukan oleh

Kepala Laboratorium.

7. Penggunaan fasilitas Laboratorium menyesuaikan dengan kapasitas ruang Laboratorium.
8. Segala pelanggaran yang dilakukan oleh praktikan akan berakibat pada penutupan dan penghentian penggunaan seluruh fasilitas laboratorium dan ditindak sesuai dengan aturan yang berlaku.

SANKSI-SANKSI

Sanksi terhadap pelanggaran **TATA TERTIB**:

Dosen Pengampu dan Asisten laboratorium berhak menjatuhkan sanksi, sesuai dengan aturan yang berlaku di Laboratorium Terpadu Fakultas Ilmu Komputer UMI apabila :

1. Praktikan merusak peralatan praktikum (*Personal Computer*) secara sengaja, maka praktikan bertanggung jawab untuk mengganti kerusakan tersebut.
2. Praktikan tidak mematuhi dan mentaati aturan praktikum maka tidak diperkenankan mengikuti praktikum.

Pelanggaran point lainnya dikenakan sanksi teguran, dikeluarkan/dicoret namanya dalam kegiatan praktikum (mengulang mata kuliah sesuai dengan semester berjalan) sampai sanksi akademik.



Kepala Laboratorium Terpadu,

Ir. Abdul Rachman Manga', S.Kom., M.T.,
MTA., MCF

DAFTAR ISI

KATA PENGANTAR	2
TATA TERTIB PELAKSANAAN PRAKTIKUM.....	3
DAFTAR ISI.....	5
CAPAIAN PEMBELAJARAN MATA KULIAH (CPMK)	6
MODUL 1 – Array & Struct.....	7
MODUL 2 – Function.....	13
MODUL 3 – Sorting / Pengurutan.....	19
MODUL 4 – Pointer	25
MODUL 5 – Linked List	29
MODUL 6 – Stack / Tumpukan.....	34
MODUL 7 – Queue / Antrian	38
MODUL 8 – Tree.....	41

CAPAIAN PEMBELAJARAN MATA KULIAH (CPMK)

1. CPMK1 : Mahasiswa mampu memahami struktur data dan algoritma serta implementasinya pada pemrograman berorientasi objek
2. CPMK2 : Mahasiswa mampu memahami algoritma tree serta linked list dan penerapannya pada bahasa program C++
3. CPMK3 : Mahasiswa mampu memahami algoritma pencarian serta pengurutan data dan penerapannya pada bahasa pemrograman C++
4. CPMK4 : Mahasiswa mampu memahami algoritma graph serta penerapannya pada bahasa pemrograman C++

MODUL 1 – Array & Struct

A. Sub Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami array serta penggunaannya serta mampu mengimplementasikannya.
2. Mahasiswa mampu memahami struct serta penggunaannya serta mampu mengimplementasikannya.

B. Instrumen dan Prosedur

1. Instrument

- a) Perangkat komputer / PC / Laptop / Notebook.
- b) Sistem operasi Windows / Linux (optional Mac OS)
- c) Aplikasi Text Editor

2. Prosedur

- a) Baca dan pahami semua tahapan praktikum dengan cermat.
- b) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- c) Rapikan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- d) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain
- e) Pastikan diri anda tidak menyentuh sumber listrik.

C. Teori Dasar

1. Array

Array adalah kumpulan data yang memiliki tipe data yang sama.

Deklarasi Array

```
TipeData NamaArray[PanjangArray];
```

Contoh:

```
int nilai[10]; atau  
int nil[3]={10, 20, 30};  
//penggunaan elemen array dari array nil  
cout<<nil[0];  
//output : 10
```

2. Struct

Struct adalah kumpulan elemen-elemen yang memiliki bentuk dan tipe yang berbeda-beda yang dinyatakan dalam sebuah nama. Elemen struct ini dapat berupa variabel, konstanta, array, atau function.

a) Deklarasi Struct

Ada beberapa cara untuk mendeklarasikan sebuah struct:

Deklarasi Struct

```
struct NamaStruct{
    elemen1; elemen2 ; ... ; elemen n;
};
```

Contoh:

```
struct Nilai{
    int tugas1, tugas2; float mid, uas;
};
```

Deklarasi Struct Menggunakan Typedef

```
typedef struct{
    elemen1; elemen2 ; ... ; elemen n;
} NamaStruct;
```

b) Pembuatan Objek / Variabel dari Struct

Pembuatan objek atau variabel dari struct yang telah dideklarasikan dapat dibuat dalam function(local) maupun diluar function (global).

Deklarasi Objek Struct

```
NamaStruct NamaObjekStruct;
```

c) Pemanggilan / Penggunaan Elemen Struct

Pemanggilan atau penggunaan elemen dari struct yaitu:

Deklarasi Objek Struct

```
NamaObjekStruct.NamaElemen;
```

Contoh:

```
● ● ●

#include <iostream>
using namespace std;

struct Nilai { // deklarasi struct
    int tugas1, tugas2;
};
struct Ujian {
    float mid, uas;
};
int main() {
    Nilai nil; // pembuatan objek
    nil.tugas1 = 88; // penggunaan elemen struct
    nil.tugas2 = 70;

    Ujian ujian;
    ujian = {99, 45};
}
```

3. Struct of Array

Struct of array adalah sebuah struct dimana didalamnya terdapat elemen berupa array.

```
● ● ●

struct Mahasiswa {
    char nim[15]; //struct of array
    char nama[35]; //struct of array
};
```

4. Array of Struct

Array of Struct adalah struct dimana dalam pembuatan objek atau variabelnya dibuat dalam bentuk array.

```
● ● ●  
struct Mahasiswa {  
    string nama, nim;  
};  
int main() {  
    Mahasiswa mhs[20]; // array of struct  
}
```

5. Nested Struct

Nested Struct adalah kondisi dimana didalam sebuah struct terdapat struct lagi

```
● ● ●  
struct Nilai {  
    int tugas1, tugas2;  
    float mid, uas;  
};  
struct Mahasiswa {  
    string nama, nim;  
    Nilai nil;  
};
```

D. Kegiatan Praktikum

1. Studi Kasus

- Buka text editor yang akan digunakan.
- Buat file baru kemudian simpan file tersebut dengan nama “Modul1.cpp”.
- Kerjakan program berikut:

```

● ● ●

#include <iostream>
using namespace std;

struct Kehadiran {
    int izin, alfa;
};

struct Mahasiswa {
    string nim, nama;
    Kehadiran absen;
};

int main() {
    int n = 4;
    Mahasiswa mhs[n];
    mhs[0].nim = "13020190365";
    mhs[0].nama = "Ruzman";
    mhs[0].absen.izin = 14;
    mhs[0].absen.alfa = 2;
    mhs[1].nim = "13020190086";
    mhs[1].nama = "Ical";
    mhs[1].absen.izin = 13;
    mhs[1].absen.alfa = 3;

    mhs[2] = {"13020190363", "Awal", 16, 0};
    mhs[3] = {"13020210134", "Nass", {15, 1}};

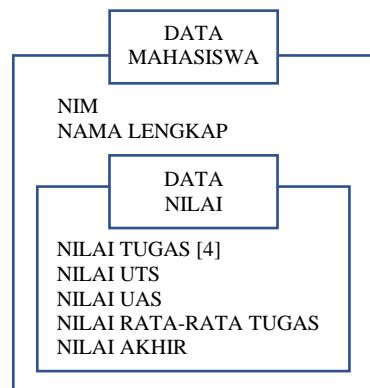
    cout << "Data Dari Struct : " << endl;

    for (int i = 0; i < n; i++) {
        cout << "Nim      : " << mhs[i].nim << endl;
        cout << "Nama     : " << mhs[i].nama << endl;
        cout << "Jumlah Izin : " << mhs[i].absen.izin << endl;
        cout << "Jumlah Alfa : " << mhs[i].absen.alfa << endl;
        cout << "-----" << endl;
    }
}

```

LEMBAR EVALUASI PRAKTIKUM

1. Pak De adalah seorang dosen Algoritma & Struktur Data dan mengampu 20 mahasiswa. Beliau ingin menghitung nilai akhir mahasiswa-mahasiswanya. Setiap mahasiswa memiliki data nim dan nama lengkap. Untuk menghitung nilai Algoritma & Struktur Data Mahasiswanya, pak De membutuhkan 4 nilai tugas, nilai uts dan nilai uas disetiap mahasiswanya. Pak De membutuhkan nilai akhir dalam bentuk nilai angka yang didapatkan dari 30% dari rata-rata nilai tugas, 30% dari nilai uts, dan 40% dari nilai uas. Untuk mengolah data dengan mudah, pak De ingin nilai-nilai tiap siswanya tertampung dalam 1 struktur data seperti gambar disamping. Buatlah program dengan struktur seperti yang disampin dengan menggunakan materi yang telah dipelajari sebelumnya



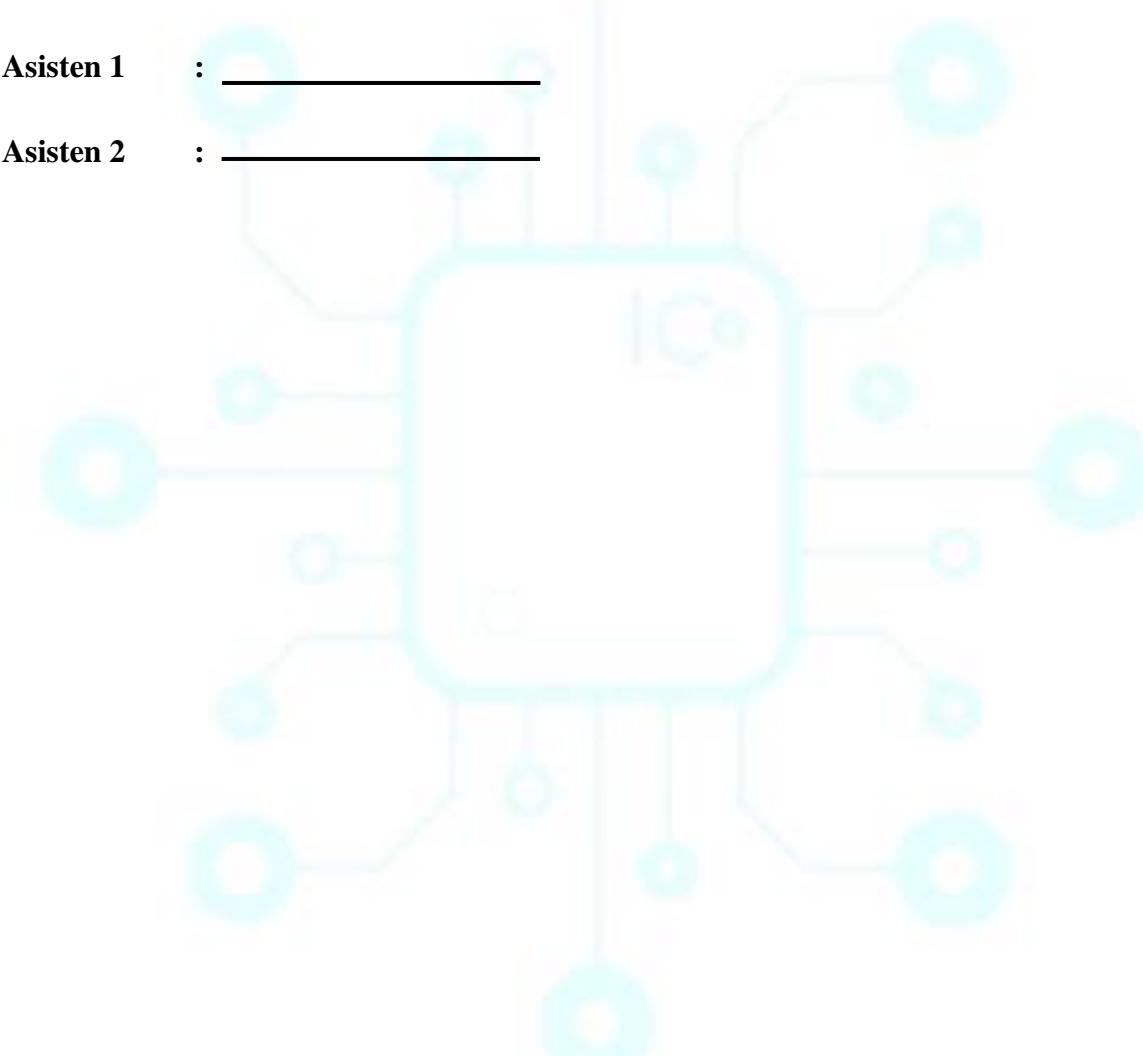
Evaluasi Praktikum 1:

No	Indikator	Skor Penilaian				
		Sangat Kurang (E) <=40	Kurang (D) 41-55	Cukup (C) 55-65	Baik (B) 66-85	Sangat Baik (A) >=86
1.	Ketepatan dalam menguraikan dan mempraktekkan penggunaan array untuk melakukan tambah data					
2.	Ketepatan dalam menguraikan dan mempraktekkan penggunaan array untuk melakukan tampil data					
3.	Ketepatan dalam menguraikan dan mempraktekkan penggunaan array untuk melakukan sisip data					
4.	Ketepatan dalam menguraikan dan mempraktekkan penggunaan array untuk melakukan hapus data					

Catatan Asisten:

Asisten 1 : _____

Asisten 2 : _____



MODUL 2 – Function

A. Sub Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa mampu memahami konsep function serta pengimplementasiannya.

B. Instrumen dan Prosedur

1. Instrument

- a) Perangkat komputer / PC / Laptop / Notebook.
- b) Sistem operasi Windows / Linux (optional Mac OS)
- c) Aplikasi Text Editor

2. Prosedur

- a) Baca dan pahami semua tahapan praktikum dengan cermat.
- b) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- c) Rapikan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- d) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain
- e) Pastikan diri anda tidak menyentuh sumber listrik.

C. Teori Dasar

1. Function

Function adalah sebuah blok berisi kumpulan perintah untuk melakukan proses tertentu.

Deklarasi Function

```
TipeData NamaFunction( ) {  
    // statement  
}
```

Contoh:

```
int ini_function() {  
    return 10;  
}
```

2. Parameter Function

Parameter pada function adalah sebuah nilai awal yang diberikan pada saat sebuah function dipanggil. Parameter function diinisialisasikan pada saat pembuatan function sedangkan pemberian nilai pada parameternya diberikan pada saat function akan dipanggil atau digunakan. Parameter dapat berupa variabel, konstanta, array dan pointer.

Deklarasi Function Berparameter

```
TipeData NamaFunction(parameter1, parameter2, ..., parameter-n) {
    // statement
}
```

Contoh:

```
int tambah(int a, int b) {
    return a + b;
}
```

Secara umum, procedure dan function terbagi menjadi 2 model parameternya, yaitu *passing by value* dan *passing by reference*.

a) Passing by Value

Passing by Value adalah teknik yang digunakan untuk mengirim data ke dalam fungsi dengan menyalin nilai dari argumen yang memanggil ke parameter fungsi tersebut. Dengan kata lain, nilai argumen akan disalin ke dalam parameter fungsi, sehingga perubahan nilai pada parameter tidak akan berpengaruh pada nilai argumen asli yang dikirim ke fungsi.

```
● ● ●

#include <iostream>
using namespace std;

void tambah(int nilai) {
    nilai += 20;
    cout << "Nilai angka dari procedure tambah() nilai += 20 = " << nilai << endl;
}

int main() {
    int angka = 10;
    cout << "Nilai variabel angka = " << angka << endl;
    tambah(angka);
    cout << "Nilai variabel angka = " << angka << endl;
}
```

b) Passing by Reference

Dalam parameter passing by reference, alamat data (reference) dari argumen yang memanggil akan disalin ke parameter function. Sehingga, perubahan pada nilai parameter akan tercermin pada variabel yang diacu oleh parameter tersebut.

```
● ● ●

#include <iostream>
using namespace std;

void tambah(int &nilai) {
    nilai += 20;
    cout << "Nilai angka dari procedure tambah() nilai += 20 = " << nilai << endl;
}

int main() {
    int angka = 10;
    cout << "Nilai variabel angka = " << angka << endl;
    tambah(angka);
    cout << "Nilai variabel angka = " << angka << endl;
}
```

3. Recursive

Recursive adalah sebuah teknik pemrograman dimana sebuah fungsi dapat memanggil dirinya sendiri secara berulang.

```
● ● ●

#include <iostream>
using namespace std;

int faktorial(int nilai) {
    if ((nilai - 1) >= 1)
        nilai = nilai * faktorial(nilai-1);

    return nilai;
}

int main() {
    cout << "Faktorial dari 5 = " << faktorial(5) << endl;

    int nilai;
    cout << "Masukkan nilai : "; cin >> nilai;
    cout << "Faktorial dari " << nilai << " = " << faktorial(nilai) << endl;
}
```

D. Kegiatan Praktikum

1. Studi Kasus

- Buka text editor yang akan digunakan
- Kerjakan program pada contoh program ***Passing by Value, Passing by Reference, dan Recursive.***
- Kerjakan program berikut:

Program 1:

```
● ● ●

#include <iostream>
using namespace std;

float kali10(float a) {
    return a *= 10;
}

float bagi10(float a) {
    return a /= 10;
}

int main() {
    float a;
    cout << "Masukkan Nilai : ";
    cin >> a;

    cout << "Hasil Hitung Dari fungsi kali10 = " << kali10(a) << endl;
    cout << "Hasil Hitung Dari fungsi bagi10 = " << bagi10(a) << endl;
    cout << "-----" << endl;
    cout << "Nilai variabel a : " << a << endl;
}
```

Program 2:

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

void proses1(float *a, float *b) {
    *a *= 100.f;
    *b /= 100.f;
}
void proses2(float &a, float &b) {
    a *= 100.f;
    b /= 100.f;
}
int main() {
    float a, b;
    cout << "Masukkan Nilai : ";
    cin >> a;
    b = a;
    cout << "-----" << endl;
    cout << "Nilai variabel a dan b Sebelum Fungsi Proses Dipanggil : "
<< endl;
    cout << "Variabel a : " << a << endl;
    cout << "Variabel b : " << b << endl;
    cout << "-----" << endl;
    proses1(&a, &b);
    cout << "Nilai variabel a dan b Setelah Fungsi proses1 Dipanggil :
<< endl;
    cout << "Variabel a : " << a << endl;
    cout << "Variabel b : " << b << endl;
    cout << "-----" << endl;
    proses2(a, b);
    cout << "Nilai variabel a dan b Setelah Fungsi proses2 Dipanggil :
<< endl;
    cout << "Variabel a : " << a << endl;
    cout << "Variabel b : " << b << endl;
}
```

- d) Setelah percobaan selesai, tutup semua perangkat lunak yang telah digunakan.
- e) Matikan PC dan rapikan meja praktikum.

LEMBAR EVALUASI PRAKTIKUM

1. Jelaskan Perbedaan Program1 dan Program2 Pada Kegiatan Praktikum.
2. Buatlah sebuah program konversi satuan panjang dari meter ke satuan panjang lainnya dengan memanfaatkan procedure dan function yang berparameter pass by value.
3. Lengkapi program manajemen poin berdasarkan source code dibawah ini. Tidak ada penambahan variabel ataupun procedure dan function.

```

● ● ●

#include <iostream>
using namespace std;

void tambahPoin(&poinsaya, poin) {
    poinsaya = poinsaya + poin
}
void kurangPoin(int &poinsaya, int poin, int *penggunaanpoin) {
    poinsaya = poinsaya - poin.
    Penggunaanpoin = penggunaanpoin + poin
}
void lihatPoin(...) {
    perintah untuk lihat total poin
}
void lihatPenggunaanPoin(...) {
    perintah untuk penggunaan poin
}
void menu() {
    cout << "Menu Poin" << endl;
    cout << "(1) Tambah Poin " << endl;
    cout << "(2) Kurang Poin " << endl;
    cout << "(3) Cek Poin " << endl;
    cout << "(4) Penggunaan Poin" << endl;
    cout << "Masukkan Pilihan (1, 2, 3 ) : ";
}
int main() {
    int pilih, poinsaya, poin, penggunaanpoin = 0;
    poinsaya = 40;
    panggil procedure untuk melihat total poin.
    while (...) {
        menu();
        cin >> pilih;
        if (pilih == 1) {
            cout << "Masukkan Jumlah Poin Yang Ditambahkan : ";
            cin >> poin;
            tambahPoin(poinsaya, poin);
        }
        else if (pilih == 2) {
            cout << "Masukkan Jumlah Poin Yang DiGunakan : ";
            cin >> poin;
            kurangPoin(poinsaya, poin, penggunaanpoin);
        }
        else if (pilih == 3) {
            lihatPoin(poinsaya);
        }
        else if (pilih == 4) {
            lihatPenggunaanPoin(penggunaanpoin);
        }
        else {
            cout << "PILIHAN SALAH " << endl;
        }
        cout <<
    }
}

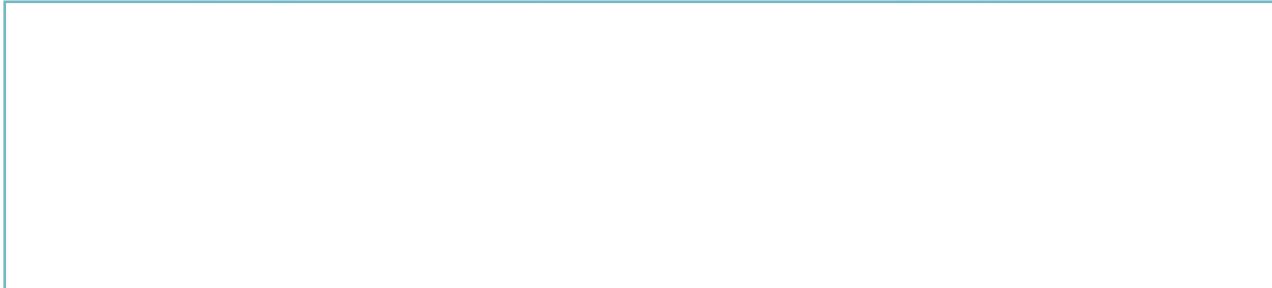
```

4. Buatlah contoh program aritmatika dengan menggunakan procedure dan function dengan parameter passing by reference.
5. Buatlah program perpangkatan dengan menggunakan konsep recursive.

Evaluasi Praktikum 2:

No	Indikator	Skor Penilaian				
		Sangat Kurang (E) <=40	Kurang (D) 41-55	Cukup (C) 55-65	Baik (B) 66-85	Sangat Baik (A) >=86
1.	Ketepatan dalam memahami penggunaan function					
2.	Ketepatan dalam memahami penggunaan procedure					

Catatan Asisten:



Asisten 1 : _____

Asisten 2 : _____

MODUL 3 – Sorting / Pengurutan

A. Sub Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa mampu memahami konsep function serta pengimplementasiannya.

B. Instrumen dan Prosedur

1. Instrument

- a) Perangkat komputer / PC / Laptop / Notebook.
- b) Sistem operasi Windows / Linux (optional Mac OS)
- c) Aplikasi Text Editor

2. Prosedur

- a) Baca dan pahami semua tahapan praktikum dengan cermat.
- b) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- c) Rapikan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- d) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain
- e) Pastikan diri anda tidak menyentuh sumber listrik.

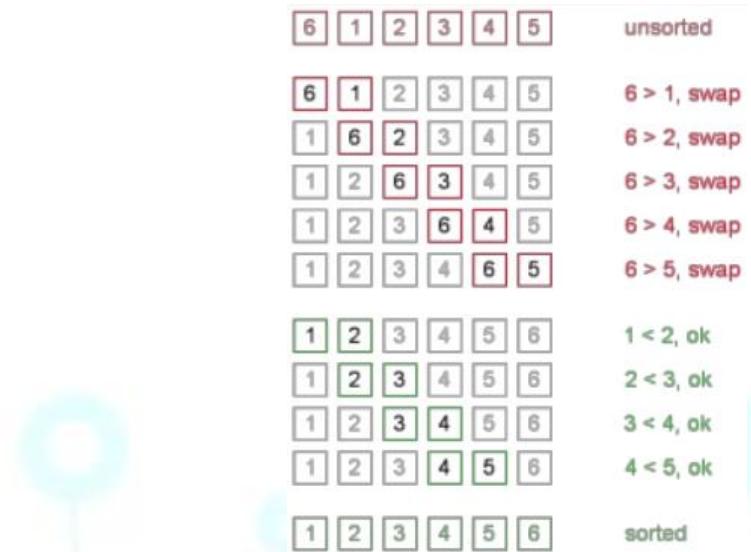
C. Teori Dasar

1. Sorting

Sorting/Pengurutan adalah sebuah proses untuk mengatur beberapa data/objek sesuai aturan tertentu. Pengurutan terdiri dari 2 jenis yaitu ascending (urutan yang menaik) dan descending (urutan yang menurun). Ada beberapa metode dalam pengurutan antara lain bubble sort, selection sort dan merge sort, dll.

a) Bubble Sort

Bubble Sort atau pengurutan gelembung adalah metode pengurutan dengan mengangkat nilai terkecil keatas dan dibawa ke posisi sebelah kiri melalui proses pertukaran.



Program Bubble Sort:

```

● ● ●

#include <iostream>
using namespace std;

int main() {
    int nilai[6] = {6, 1, 2, 3, 4, 5};
    int temp;

    cout << "Data Sebelum diurutkan" << endl;
    for (int i = 0; i <= 5; i++)
        cout << nilai[i] << " ";

    cout << endl << endl;

    for (int i = 5; i >= 0; i--)
        for (int j = 0; j <= i - 1; j++)
            if (nilai[j] > nilai[j + 1]) {
                // proses penukaran posisi elemen
                temp = nilai[j + 1];
                nilai[j + 1] = nilai[j];
                nilai[j] = temp;
            }

    cout << "Data Setelah Diurutkan (Ascending)" << endl;
    for (int i = 0; i <= 5; i++)
        cout << nilai[i] << " ";

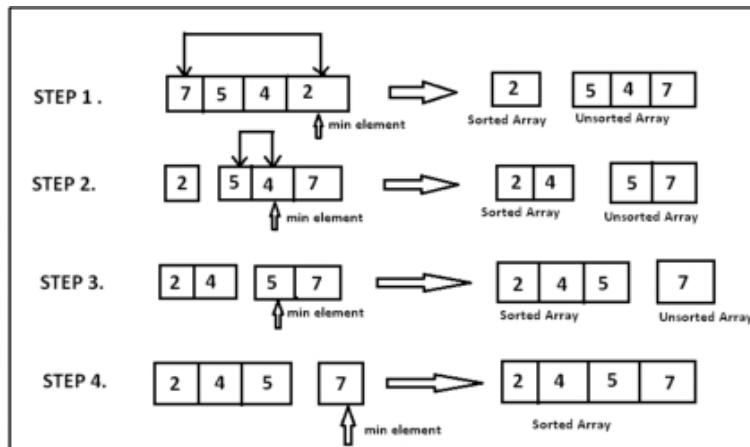
    cout << endl;
    cout << "Data Setelah Diurutkan (Descending)" << endl;
    for (int i = 5; i >= 0; i--)
        cout << nilai[i] << " ";

    return 0;
}

```

b) Selection Sort

Selection Sort adalah metode sorting / pengurutan dengan menemukan data terkecil dengan cara membandingkan elemen sekarang dengan elemen terakhir kemudian menukan posisi elemen terkecil tersebut.



Program Selection Sort:

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

int main() {
    int n = 4;
    int arr[n] = {7, 5, 4, 2};
    int temp;

    cout << "Sebelum Pengurutan :" << endl;
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";
    cout << endl;

    for (int i = 0; i < n; i++)
        for (int j = i + 1; j < n; j++)
            if (arr[i] > arr[j]) {
                temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
    cout << "Setelah Pengurutan : " << endl;
    for (int i = 0; i < n; i++)
        cout << arr[i] << " ";
}

return 0;
}
```

c) Merge Sort

Merge Sort adalah yaitu metode pengurutan dengan memecah kemudian menyelesaikan setiap bagian kemudian menggabungkannya kembali. Pertama data dipecah menjadi 2 bagian dimana bagian pertama merupakan setengah (jika data genap) atau setengah minus satu (jika data ganjil) dari seluruh data, kemudian

dilakukan pemecahan kembali untuk masing-masing blok sampai hanya terdiri dari satu data tiap blok.

```
● ● ●  
  
#include <iostream>  
using namespace std;  
  
void merge(int arr[], int l, int m, int r) {  
    int i, j, k;  
    int n1 = m - l + 1;  
    int n2 = r - m;  
    int L[n1], R[n2];  
  
    for (i = 0; i < n1; i++)  
        L[i] = arr[l + i];  
  
    for (j = 0; j < n2; j++)  
        R[j] = arr[m + 1 + j];  
  
    i = 0;  
    j = 0;  
    k = l;  
  
    while (i < n1 && j < n2) {  
        if (L[i] <= R[j]) {  
            arr[k] = L[i];  
            i++;  
        }  
    }  
}
```



```

        else {
            arr[k] = R[j];
            j++;
        }
        k++;
    }
    while (i < n1) {
        arr[k] = L[i];
        i++;
        k++;
    }
    while (j < n2) {
        arr[k] = R[j];
        j++;
        k++;
    }
}

void mergeSort(int arr[], int l, int r) {
    if (l < r) {
        int m = l + (r - l) / 2;
        mergeSort(arr, l, m);
        mergeSort(arr, m + 1, r);
        merge(arr, l, m, r);
    }
}

void printArray(int arr[], int size) {
    for (int i = 0; i < size; i++)
        cout << arr[i] << " ";
}

int main() {
    int arr[] = {12, 11, 13, 5, 6, 7};
    int arr_size = sizeof(arr) / sizeof(arr[0]);

    cout << "Data Sebelum Pengurutan : " << endl;

    printArray(arr, arr_size);
    mergeSort(arr, 0, arr_size - 1);

    cout << "\nData Setelah Pengurutan : " << endl;

    printArray(arr, arr_size);

    return 0;
}

```

D. Kegiatan Praktikum

1. Studi Kasus

- Buka text editor yang akan digunakan
- Kerjakan contoh program *bubble sort*, *sleection sort*, dan *merge sort* yang ada pada teori dasar.
- Setelah percobaan selesai, tutup semua perangkat lunak yang telah digunakan.
- Matikan PC dan rapihkan meja praktikum.

LEMBAR EVALUASI PRAKTIKUM

- Buatlah program *sorting* huruf yang berfungsi untuk mengurutkan huruf dari nama yang diinputkan pada saat program dijalankan. Gunakan metode *sorting* yang telah dibahas.

Evaluasi Praktikum 3:

No	Indikator	Skor Penilaian				
		Sangat Kurang (E) <=40	Kurang (D) 41-55	Cukup (C) 55-65	Baik (B) 66-85	Sangat Baik (A) >=86
1.	Ketepatan menguraikan teknik pengurutan data					
2.	Ketepatan mempraktekkan algoritma pengurutan data					

Catatan Asisten:

Asisten 1 : _____

Asisten 2 : _____

MODUL 4 – Pointer

A. Sub Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa mampu memahami konsep pointer serta mampu mengimplementasikannya.

B. Instrumen dan Prosedur

1. Instrument

- a) Perangkat komputer / PC / Laptop / Notebook.
- b) Sistem operasi Windows / Linux (optional Mac OS)
- c) Aplikasi Text Editor

2. Prosedur

- a) Baca dan pahami semua tahapan praktikum dengan cermat.
- b) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- c) Rapikan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- d) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain
- e) Pastikan diri anda tidak menyentuh sumber listrik.

C. Teori Dasar

1. Pointer

Pointer adalah variabel yang berisi alamat memori. Dengan kata lain, pointer berisi alamat dari variabel yang mempunyai nilai tertentu.

Deklarasi Pointer

TipeData *nama_variabel;

Contoh:

int *a;

2. Operator Pointer

a) Operator Reference (&)

Operator reference digunakan untuk meghasilkan alamat memori lokasi memori dari sebuah variabel. Operator ini biasa disebut dengan *address of* untuk operator alamat.

b) Operator Deference (*)

Operator deference digunakan untuk mengakses langsung nilai yang terdapat didalam variabel berpointer. Operator ini biasa disebut dengan *value by*.

Contoh :

```
● ● ●

#include <iostream>
using namespace std;

int main() {
    int *nil1;
    int nil2 = 9;
    nil1 = &nil2;

    cout << "Nilai dari nil1 : " << nil1 << endl;
    cout << "Nilai dari *nil1 : " << *nil1 << endl;
    cout << "Nilai dari &nil1 : " << &nil1 << endl;
    cout << "Nilai dari &nil2 : " << &nil2 << endl;

    return 0;
}
```

D. Kegiatan Praktikum

1. Studi Kasus

- Buka text editor yang akan digunakan.
- Buat file baru kemudian simpan file tersebut dengan nama “Pointer.cpp”.
- Kerjakan beberapa praktikan berikut:
 - Program 1 :

```
● ● ●

#include <iostream>
using namespace std;

int main() {
    int *nil1;
    int nil2 = 10;
    nil1 = &nil2;

    cout << "Nilai dari nil2 : " << nil2 << endl;
    cout << "Nilai dari Deference nil2 : " << &nil2 << endl;
    cout << "Nilai dari nil1 : " << nil1 << endl;
    cout << "Nilai dari Reference nil1 : " << *nil1 << endl;
    cout << "Nilai dari Deference nil1 : " << &nil1 << endl;
    return 0;
}
```

b. Program 2 :

```

● ● ●

#include <iostream>
using namespace std;

int main() {
    int *nil1;
    int nil2 = 9;

    nil1 = &nil2;
    cout << "Nilai dari nil1 : " << nil1 << endl;
    cout << "Nilai dari *nil1 : " << *nil1 << endl;

    nil2 = 10;
    cout << "Nilai dari *nil1 : " << *nil1 << endl;

    *nil1 = 8;
    cout << "Nilai dari *nil1 : " << *nil1 << endl;

    nil2 = 19;
    cout << "Nilai dari *nil1 : " << *nil1 << endl;
    return 0;
}

```

- d) Setelah percobaan selesai, tutup semua perangkat lunak yang telah digunakan.
e) Matikan PC dan rapihkan meja praktikum.

LEMBAR EVALUASI PRAKTIKUM

1. Jelaskan *Source Code* pada kegiatan praktikum
2. Kerjakan source code dibawah ini:

```

● ● ●

#include <iostream>
using namespace std;

int main() {
    Buatlah variabel pointer (*) nil1 bertipe int.
    Buatlah variabel pointer berpointer (**) nil3 bertipe int.
    Buatlah variabel nil2 bertipe int dengan nilai samadengan 15.

    Berikan nilai pada variabel nil1 dengan alamat variabel nil2.
    Berikan nilai pada variabel nil3 dengan alamat variabel nil1.
    Tampilkan nilai dari variabel nil2.
    Tampilkan nilai dari reference nil2.
    cout<<"-----"<<endl;

    Tampilkan nilai dari variabel nil1.
    Tampilkan nilai dari reference nil1.
    Tampilkan nilai dari deference nil1.
    cout<<"-----"<<endl;

    Tampilkan nilai dari variabel nil3.
    Tampilkan nilai dari deference nil3 (*nil3).
    Tampilkan nilai dari deference dari deference nil3 (**nil3).
    Tampilkan nilai dari reference nil3.

    //Tambahkan Soal Nomor 4, tuliskan di bawah ini.
    return 0;
}

```

3. Jelaskan hasil output dari nilai-nilai yang ditampilkan pada program nomor 2

4. Setelah mengerjakan soal nomor 2, berikan nilai pada variabel nil2 dengan nilai 35. Setelah itu tampilkan nilai dari deference nil(**nil3). Jelaskan hasil output program tersebut.
5. Buatlah sebuah contoh program yang mengimplementasikan pointer didalamnya. Buat program tersebut dilengkapi dengan proses aritmatika dan pengimputan dari user.

Evaluasi Praktikum 4:

No	Indikator	Skor Penilaian				
		Sangat Kurang (E) <=40	Kurang (D) 41-55	Cukup (C) 55-65	Baik (B) 66-85	Sangat Baik (A) >=86
1.	Ketepatan menjelaskan fitur-fitur serta dasar pemrograman berorientasi objek pada bahasa C++					

Catatan Asisten:

Asisten 1 : _____

Asisten 2 : _____

MODUL 5 – Linked List

A. Sub Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa mampu memahami konsep Linked list serta mampu mengimplementasikannya.

B. Instrumen dan Prosedur

1. Instrument

- a) Perangkat komputer / PC / Laptop / Notebook.
- b) Sistem operasi Windows / Linux (optional Mac OS)
- c) Aplikasi Text Editor

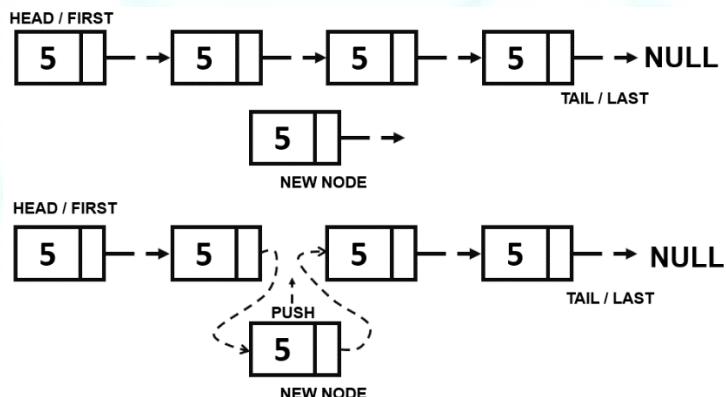
2. Prosedur

- a) Baca dan pahami semua tahapan praktikum dengan cermat.
- b) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- c) Rapikan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- d) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain
- e) Pastikan diri anda tidak menyentuh sumber listrik.

C. Teori Dasar

1. Linked List

LinkedList adalah struktur data yang terdiri dari kumpulan simpul (node) yang saling terhubung satu sama lain dengan menggunakan pointer, dimana setiap simpul menyimpan data serta pointer ke simpul berikutnya.



Contoh Program Linked List:

```
● ● ●  
#include <iostream>  
using namespace std;  
  
struct TNode {  
    int data;  
    TNode *next;  
};  
TNode *head;  
  
void init() {  
    head = NULL;  
}  
  
int isEmpty() {  
    if (head == NULL)  
        return 1;  
    else  
        return 0;  
}
```



```

void insertDepan(int databaru) {
    TNode *baru;
    baru = new TNode;
    baru->data = databaru;
    baru->next = NULL;
    if (isEmpty() == 1) {
        head = baru;
        head->next = NULL;
    }
    else {
        baru->next = head;
        head = baru;
    }
    cout << "data masuk" << endl;
}
void insertBelakang(int databaru) {
    TNode *baru, *bantu;
    baru = new TNode;
    baru->data = databaru;
    baru->next = NULL;
    if (isEmpty() == 1) {
        head = baru;
        head->next = NULL;
    }
    else {
        bantu = head;
        while (bantu->next != NULL) {
            bantu = bantu->next;
        }
        bantu->next = baru;
    }
    cout << "Data masuk" << endl;
}
void tampil() {
    cout << "Isi Data" << endl;

    TNode *bantu;
    bantu = head;
    if (isEmpty() == 0) {
        while (bantu != NULL) {
            cout << bantu->data << " " << endl;
            bantu = bantu->next;
        }
        cout << "" << endl;
    }
    else
        cout << "Masih kosong" << endl;
}
void hapusDepan() {
    TNode *hapus;
    int d;
    if (isEmpty() == 0) {
        if (head->next != NULL) {
            hapus = head;
            d = hapus->data;
            head = head->next;
            delete hapus;
        }
        else {
            d = head->data;
            head = NULL;
        }
        cout << d << " terhapus" << endl;
    }
    else
        cout << "Masih kosong" << endl;
}
void hapusBelakang() {
    TNode *hapus, *bantu;
    int d;
    if (isEmpty() == 0) {
        if (head->next != NULL) {

```

```

        bantu = head;
        while (bantu->next->next != NULL) {
            bantu = bantu->next;
        }
        hapus = bantu->next;
        d = hapus->data;
        bantu->next = NULL;
        delete hapus;
    }
    else {
        d = head->data;
        head = NULL;
    }
    cout << d << " terhapus" << endl;
}
else
    cout << "Masih kosong" << endl;
}
int main() {
    int pil, databaru;
    cout << "-----" << endl;
    cout << "* Single Linked List Non Circular" << endl;
    cout << "-----" << endl;

    while (true) {
        cout << "1. Insert Depan" << endl;
        cout << "2. Insert Belakang" << endl;
        cout << "3. Delete Depan" << endl;
        cout << "4. Delete Belakang" << endl;
        cout << "5. Tampil Data" << endl;
        cout << "6. Keluar" << endl;
        cout << "Masukkan Pilihan Anda : ";
        cin >> pil;

        switch (pil) {
            case 1:
                cout << "Masukkan data = ";
                cin >> databaru;
                insertDepan(databaru);
                break;

            case 2:
                cout << "Masukkan data = ";
                cin >> databaru;
                insertBelakang(databaru);
                break;

            case 3:
                hapusDepan();
                break;

            case 4:
                hapusBelakang();
                break;

            case 5:
                tampil();
                break;

            case 6:
                cout << "Program selesai..." << endl;
                return 0;

            default:
                cout << "Maaf, Tidak ada dalam pilihan" << endl;
        }
        cout << "-----" << endl;
    }
}

```

D. Kegiatan Praktikum

1. Studi Kasus

- a) Buka text editor yang akan digunakan.
- b) Kerjakan contoh program linked list pada teori dasar.
- c) Setelah percobaan selesai, tutup semua perangkat lunak yang telah digunakan.
- d) Matikan PC dan rapihkan meja praktikum.

LEMBAR EVALUASI PRAKTIKUM

Buatlah sebuah program sorting dengan menggunakan metode insertion sort dengan mengimplementasikan linked list.

Evaluasi Praktikum 5:

No	Indikator	Skor Penilaian				
		Sangat Kurang (E) ≤40	Kurang (D) 41-55	Cukup (C) 55-65	Baik (B) 66-85	Sangat Baik (A) ≥86
1.	Ketepatan dalam menguraikan konsep dan bentuk linked list					
2.	Ketepatan dalam mempraktekkan operasi linked list					

Catatan Asisten:

Asisten 1 : _____

Asisten 2 : _____

MODUL 6 – Stack / Tumpukan

A. Sub Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa mampu memahami konsep stack serta mampu mengimplementasikannya.

B. Instrumen dan Prosedur

1. Instrument

- a) Perangkat komputer / PC / Laptop / Notebook.
- b) Sistem operasi Windows / Linux (optional Mac OS)
- c) Aplikasi Text Editor

2. Prosedur

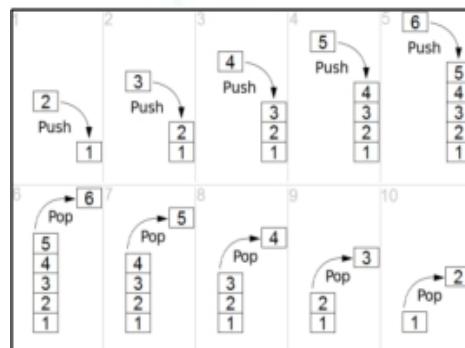
- a) Baca dan pahami semua tahapan praktikum dengan cermat.
- b) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- c) Rapikan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- d) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain
- e) Pastikan diri anda tidak menyentuh sumber listrik.

C. Teori Dasar

1. Stack

Stack berarti tumpukan yang menerapkan konsep LIFO (*Last In First Out*) yang artinya data yang terakhir masuk akan menjadi data yang pertama keluar. Ada beberapa operasi-operasi pada stack yaitu:

- Push : Penambahan elemen pada stack
- Pop : Pengambilan/pengeluaran elemen dari stack
- IsEmpty : Pengecekan apakah stack telah terisi atau tidak
- IsFull : Pengecekan apakah stack telah penuh atau tidak
- ShowElements : Menampilkan element pada stack
- Clear : Pengosongan elemen dari stack



D. Kegiatan Praktikum

1. Studi Kasus

- Buka text editor yang akan digunakan.
- Buat file baru kemudian simpan file tersebut dengan nama “Stack.cpp”
- Kerjakan program berikut:

```

● ● ●

#include <iostream>
using namespace std;

int top = -1;
int data[5];
void Push(int input) {
    top = top + 1;
    data[top] = input;
}
void Pop() {
    top = top - 1;
}
void isiStack() {
    for (int i = top; i >= 0; i--) {
        cout << "| " << data[i] << " | " << endl;
    }
}
int main() {
    int pilih, input;
    while (true) {
        cout << "Menu" << endl;
        cout << "1. Push Stack" << endl;
        cout << "2. Pop Stack" << endl;
        cout << "3. Isi Stack" << endl;
        cout << "Masukkan Pilihan : ";
        cin >> pilih;

        if (pilih == 1) {
            cout << "Masukkan Data : ";
            cin >> input;
            Push(input);
            cout << "Push Berhasil !" << endl;
        }
        else if (pilih == 2) {
            Pop();
            cout << "Pop Berhasil" << endl;
        }
        else if (pilih == 3)
            isiStack();
        else
            cout << "Tidak Ada Pilihan !" << endl;
    }
}

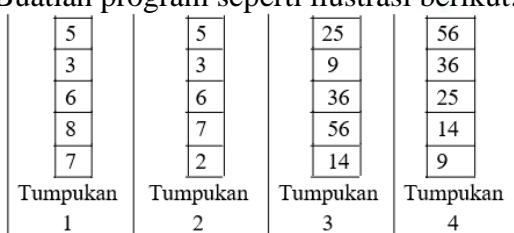
```

- Setelah percobaan selesai, tutup semua perangkat lunak yang telah digunakan
- Matikan PC dan rapihkan meja praktikum.

LEMBAR EVALUASI PRAKTIKUM

1. Lengkapilah program pada kegiatan praktikum dengan operasi:
 - a. Clear untuk mengosongkan stack.
 - b. IsEmpty untuk memeriksa stack dalam kondisi kosong
 - c. IsFull untuk memeriksa stack dalam kondisi full
 - d. Tambahkan validasi untuk tidak dapat menambahkan data jika stack penuh dan validasi untuk tidak dapat menghapus data jika stack kosong.

2. Buatlah program seperti ilustrasi berikut:



Tumpukan 3 : berisi hasil kali dari tumpukan 1 dan tumpukan 2

Tumpukan 4 : berisi hasil pengurutan tumpukan 3 menggunakan bubble sort

Evaluasi Praktikum 6:

No	Indikator	Skor Penilaian				
		Sangat Kurang (E) <=40	Kurang (D) 41-55	Cukup (C) 55-65	Baik (B) 66-85	Sangat Baik (A) >=86
1.	Ketepatan dalam menguraikan konsep dan bentuk stack					
2.	Ketepatan dalam mempraktekkan operasi stack					

Catatan Asisten:

Asisten 1 : _____

Asisten 2 : _____



MODUL 7 – Queue / Antrian

A. Sub Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa mampu memahami konsep Queue serta mampu mengimplementasikannya.

B. Instrumen dan Prosedur

1. Instrument

- a) Perangkat komputer / PC / Laptop / Notebook.
- b) Sistem operasi Windows / Linux (optional Mac OS)
- c) Aplikasi Text Editor

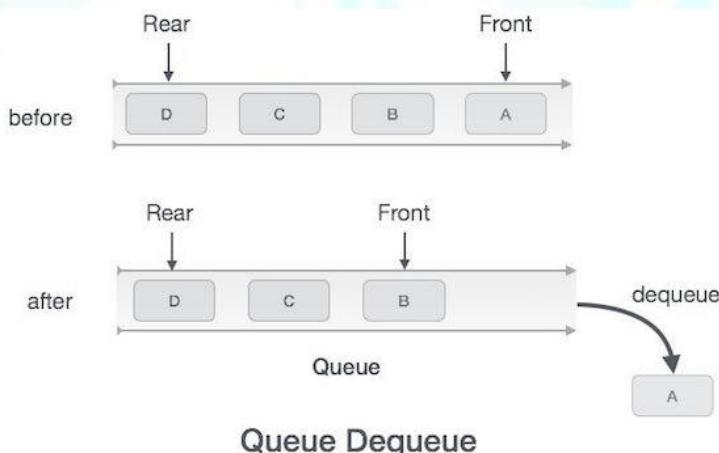
2. Prosedur

- a) Baca dan pahami semua tahapan praktikum dengan cermat.
- b) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- c) Rapikan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- d) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain
- e) Pastikan diri anda tidak menyentuh sumber listrik.

C. Teori Dasar

Queue berarti antrian yang menerapkan konsep FIFO (*First In First Out*) yang artinya data yang pertama masuk akan menjadi pertama keluar. Ada beberapa operasi-operasi pada antrian yaitu:

- Enqueue : Penambahan elemen pada queue
- Dequeue : Pengambilan/pengeluaran elemen dari queue
- IsEmpty : Pengecekan apakah queue telah terisi atau tidak
- IsFull : Pengecekan apakah queue telah penuh atau tidak
- ShowElements : Menampilkan element pada queue
- Clear : Pengosongan elemen dari queue



D. Kegiatan Praktikum

1. Studi Kasus

- a) Buka text editor yang akan digunakan.
- b) Buat file baru kemudian simpan file tersebut dengan nama “Queue.cpp”
- c) Kerjakan program berikut:

Andi ingin membeli tiket konser dari band favoritnya yang akan digelar dalam waktu dekat. Untuk itu, Andi harus mengantri di loket tiket yang sudah disediakan oleh panitia konser. Karena konser tersebut sangat populer, antrian untuk membeli tiket pun sangat panjang.

Andi ingin membuat program sederhana menggunakan C++ untuk memudahkan dirinya dalam memantau antrian tersebut. Buatlah program untuk menambahkan dan menghapus data dari antrian menggunakan konsep queue dan dengan menerapkan operasi-operasi pada antrian seperti, dequeue, enqueue, isEmpty, isFull, showElements, dan clear

- d) Setelah percobaan selesai, tutup semua perangkat lunak yang telah digunakan
- e) Matikan PC dan rapihkan meja praktikum.

LEMBAR EVALUASI PRAKTIKUM

1. Buatlah program antrian sesuai ilustrasi berikut ini:

Pengguna melakukan Enqueue sebanyak 10 kali. Kemudian data tersimpan pada **Antrian1**.

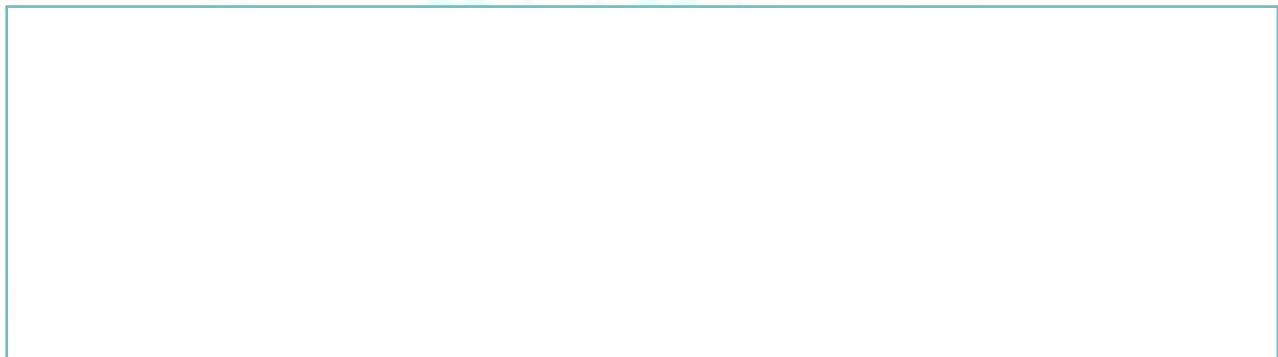
Ketika Pengguna melakukan Dequeue, data yang dikeluarkan akan masuk ke antrian selanjutnya. Data dengan angka yang lebih besar dari 20 akan masuk ke **Antrian2** sedangkan data dengan angka yang kurang dari 20 akan masuk ke **Antrian3**.

Pengguna akan dapat melihat isi dari ketiga Antrian tersebut.

Evaluasi Praktikum 7:

No	Indikator	Skor Penilaian				
		Sangat Kurang (E) <=40	Kurang (D) 41-55	Cukup (C) 55-65	Baik (B) 66-85	Sangat Baik (A) >=86
1.	Ketepatan dalam menguraikan konsep dan bentuk queue					
2.	Ketepatan dalam mempraktekkan operasi queue					

Catatan Asisten:



Asisten 1 : _____

Asisten 2 : _____

MODUL 8 – Tree

A. Sub Capaian Pembelajaran Mata Kuliah (CPMK)

Mahasiswa mampu memahami konsep Tree serta mampu mengimplementasikannya.

B. Instrumen dan Prosedur

1. Instrument

- a) Perangkat komputer / PC / Laptop / Notebook.
- b) Sistem operasi Windows / Linux (optional Mac OS)
- c) Aplikasi Text Editor

2. Prosedur

- a) Baca dan pahami semua tahapan praktikum dengan cermat.
- b) Gunakan fasilitas yang disediakan dengan penuh rasa tanggung jawab.
- c) Rapikan kembali setelah menggunakan komputer (mouse, keyboard, kursi, dll)
- d) Perhatikan sikap anda untuk tidak mengganggu rekan praktikan lain
- e) Pastikan diri anda tidak menyentuh sumber listrik.

C. Teori Dasar

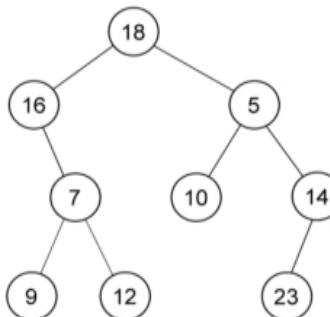
1. Tree

Merupakan salah satu bentuk struktur data tidak linear yang menggambarkan hubungan yang bersifat hirarkis (hubungan one to many) antara elemen-elemen. Tree bisa didefinisikan sebagai kumpulan simpul/node dengan satu elemen khusus yang disebut Root dan node lainnya terbagi menjadi himpunan-himpunan yang saling tak berhubungan satu sama lainnya (disebut subtree). Untuk jelasnya, di bawah akan diuraikan istilah-istilah umum dalam tree :

- Predecessor : node yang berada diatas node tertentu.
- Successor : node yang berada di bawah node tertentu.
- Parent : predecessor satu level di atas suatu node.
- Child : successor satu level di bawah suatu node.
- Sibling : node-node yang memiliki parent yang sama dengan suatu node.
- Subtree : bagian dari tree yang berupa suatu node beserta descendantnya dan memiliki semua karakteristik dari tree tersebut.
- Size : banyaknya node dalam suatu tree.
- Height : banyaknya tingkatan/level dalam suatu tree.
- Root : satu-satunya node khusus dalam tree yang tak punya predecessor.
- Leaf : node-node dalam tree yang tak memiliki successor.
- Degree : banyaknya child yang dimiliki suatu node

2. Binary Tree

Binary Tree adalah tree dengan syarat bahwa tiap node hanya boleh memiliki maksimal dua subtree dan kedua subtree tersebut harus terpisah. Sesuai dengan definisi tersebut, maka tiap node dalam binary tree hanya boleh memiliki paling banyak dua child.



Contoh program Binary Tree:

```

● ● ●

#include <iostream>
using namespace std;

struct Node {
    int data;
    Node *kiri;
    Node *kanan;
};

// FUNGSI UNTUK MENAMBAHKAN DATA
void tambah(Node **root, int databaru) {
    if ((*root) == NULL) { // jika pohon/subpohon masih kosong
        Node *baru; // node baru dibentuk
        baru = new Node;
        baru->data=databaru;
        baru->kiri=NULL;
        baru->kanan=NULL;
        *root = baru;
        cout << "DATA BERTAMBAH !" << endl;
    }
    else if (databaru < (*root)->data)
        tambah(&(*root)->kiri, databaru);
    else if (databaru > (*root)->data)
        tambah(&(*root)->kanan, databaru);
    else if (databaru == (*root)->data)
        cout << "DATA SUDAH ADA !";
}
// FUNGSI PRE-ORDER itu data ditampilkan dari node induk, node anak
// kiri, lalu node anak kanan
void preOrder(Node *root) {
    if (root != NULL) {
        cout << root->data << ", ";
        preOrder(root->kiri);
        preOrder(root->kanan);
    }
}
// FUNGSI IN-ORDER itu data ditampilkan dari node anak kiri, node
// induk, node anak kanan
void inOrder(Node *root) {
    if (root != NULL) {
        inOrder(root->kiri);
        cout << root->data << ", ";
        inOrder(root->kanan);
    }
}
// FUNGSI POST-ORDER itu data ditampilkan dari node anak kiri, node
// anak kanan, lalu node induk
void postOrder(Node *root) {
    if (root != NULL) {
        postOrder(root->kiri);
        postOrder(root->kanan);
        cout << root->data << ", ";
    }
}

```

```
int main() {
    int pil;
    Node *pohon;
    pohon = NULL;
    do {
        int data;
        cout << "MENU" << endl;
        cout << "(1) TAMBAH" << endl;
        cout << "(2) LIHAT PRE-ORDER" << endl;
        cout << "(3) LIHAT IN-ORDER" << endl;
        cout << "(4) LIHAT POST-ORDER" << endl;
        cout << "(5) KELUAR" << endl;
        cout << "Pilihan : ";
        cin >> pil;

        switch (pil) {
            case 1:
                cout << "Data Baru : ";
                cin >> data;
                tambah(&pohon, data);
                break;
            case 2:
                if (pohon != NULL)
                    preOrder(pohon);
                else
                    cout << "MASIH KOSONG ! ";
                break;
            case 3:
                if (pohon != NULL)
                    inOrder(pohon);
                else
                    cout << "MASIH KOSONG ! ";
                break;
            case 4:
                if (pohon != NULL)
                    postOrder(pohon);
                else
                    cout << "MASIH KOSONG ! ";
                break;
            }
        cout << endl;
        cout << "===== " << endl;
    } while (pil != 5);
    return 0;
}
```

D. Kegiatan Praktikum

1. Studi Kasus

- Buka text editor yang akan digunakan.
- Kerjakan contoh program linked list pada teori dasar.
- Setelah percobaan selesai, tutup semua perangkat lunak yang telah digunakan
- Matikan PC dan rapihkan meja praktikum.

LEMBAR EVALUASI PRAKTIKUM

Buatlah sebuah program searching dengan menggunakan binary tree (Binary Tree Searching)!

Evaluasi Praktikum 8:

No	Indikator	Skor Penilaian				
		Sangat Kurang (E) <=40	Kurang (D) 41-55	Cukup (C) 55-65	Baik (B) 66-85	Sangat Baik (A) >=86
1.	Ketepatan dalam menguraikan konsep dan bentuk tree					
2.	Ketepatan dalam mempraktekkan operasi tree					

Catatan Asisten:

Asisten 1 : _____

Asisten 2 : _____