**Tugas Jobsheet 1 Praktikum Struktur Data**

Review Matakuliah Praktikum Struktur Data :

**1.Pointer, array dan struct**

**Pointer** adalah variabel yang menyimpan suatu alamat memori dari suatu variabel lain sehingga memungkinkan untuk memanipulasi langsung terhadap data di dalam memori serta mempercepat proses akses data tersebut. **Pointer** mampu mengefisiensi proses manipulasi data, memungkinkan pengembangan struktur data dinamis. Akan tetapi, **pointer** memerlukan manajemen memori yang lebih hati-hati untuk menghindari kesalahan alamat memori. **Pointer** dapat digunakan ketika diperlukan kontrol secara langsung terhadap alamat memori seperti dalam alokasi memori dinamis.

**Array** adalah struktur data yang menyimpan elemen-elemen yang bertipe serupa dalam urutan terindeks sehingga memberikan cara yang efisien untuk menyimpan dan mengakses sejumlah besar data dengan indeks. **Array** memiliki ukuran tetap setelah deklarasi. Hal ini menyebabkan terbatasnya fleksibilitas dalam penambahan atau pengurangan elemen. **Array** dapat digunakan ketika jumlah elemen tetap dan akses cepat diperlukan, misalnya untuk menyimpan data seperti “daftar nama”.

**Struct** adalah struktur data yang berisi kumpulan variabel dengan tipe data yang unik atau berbeda beda yang dikelompokkan dalam satu unit sehingga memungkinkan pembuatan struktur data yang kompleks dengan menyatukan data dengan format terstruktur. **Struct** memfasilitasi representasi data yang kompleks sehingga kode menjadi lebih muda dibaca. **Struct** memerlukan lebih banyak memori untuk menyimpan informasi tambahan, terutama jika struct berukuran besar. **Struct** cocok digunakan untuk merepresentasikan objek atau entittas yang memiliki atribut dengan tipe data yang berbeda-beda.

**2.Linked List**

**Linked list** adalah **struktur data linear**, element element yang ada di dalamnya tidak disimpan pada alamat memori yang bersebelahan dan mereka dihubungkan menggunakan **pointers**, **Linked list** membentuk sebuah rangkaian nodes yang terhubung. Setiap node berisi **data** dan **pointer** (alamat memori dari node berikutnya). **Node** pada **linked list** biasanya terdiri dari 2 komponen:

1.**Data**: Berisi data atau nilai yang terkait dengan node.

2.**Next Pointer**: Berisi alamat memori dari node urutan berikutnya.

Linked list diakses melalui **head node**, **head node** menunjuk pada node pertama dalam list node. Node terakhir dalam list node menunjuk pada **NULL** atau **nullptr** yang menandakan akhir dari list.

**3.Double Linked List**

Pada **double linked list**, setiap node berisi referensi/alamat memori dari **node sebelumnya** dan **node sesudahnya**. Hal ini memungkinkan menelusuri node tidak hanya ke arah depan tetapi juga bisa ke arah belakang, sedangkan pada **Single linked list**, penelusuran melalui node hanya bisa ke arah depan dan tidak bisa ke arah belakang.

**4.Circular Linked List**

Pada **Circular linked list**, **node terakhir** menunjuk kembali pada **head node**, membentuk struktur yang melingkar. **Circular linked list** dapat berupa **single linked list** atau **double linked list.**