1. **Pointer**

Pointer adalah variabel yang menyimpan alamat memori dari variabel lainnya. Dengan menggunakan pointer, kita dapat mengakses dan memanipulasi data secara langsung melalui alamat memori yang tersimpan. Konsep pointer sangat penting dalam alokasi memori dinamis, di mana kita dapat mengalokasikan memori secara fleksibel selama runtime. Misalnya, dalam pengelolaan memori, penggunaan pointer memungkinkan kita untuk mengakses dan mengubah nilai variabel secara efisien, serta membuat struktur data yang lebih kompleks seperti linked list, tree, dan graph. Selain itu, pointer juga berguna dalam parameter fungsi, di mana kita dapat meneruskan argumen dengan referensi daripada nilai.

1. **Struct**

Struct adalah kumpulan variabel dengan tipe data yang berbeda yang dibungkus dalam satu kesatuan. Dengan menggunakan struct, kita dapat membuat tipe data baru yang memiliki struktur yang kompleks. Contohnya, dalam aplikasi manajemen data, kita dapat menggunakan struct untuk merepresentasikan entitas seperti mahasiswa, pegawai, atau barang dengan menyimpan informasi yang terkait dalam satu objek. Struktur data ini memungkinkan pengelompokan data terkait dalam satu unit yang terorganisir. Mahasiswa akan mempelajari cara mendefinisikan, menginisialisasi, dan mengakses elemen-elemen dalam struct, serta bagaimana menggunakannya dalam implementasi algoritma dan aplikasi.

1. **Array**

Array adalah kumpulan elemen data dengan tipe data yang sama yang disimpan dalam lokasi memori yang berurutan. Array memungkinkan kita untuk menyimpan dan mengakses sekumpulan data dengan cara yang terstruktur. Dalam matakuliah ini, mahasiswa mempelajari cara mendeklarasikan, menginisialisasi, dan mengakses elemen-elemen dalam array. Mereka juga belajar tentang operasi-operasi dasar seperti pencarian, pengurutan, dan manipulasi data dalam array. Array memiliki keunggulan dalam akses elemen secara langsung melalui indeksnya dan merupakan struktur data yang efisien untuk menyimpan data dalam jumlah besar.

1. **Link List**

Link List adalah struktur data dinamis yang terdiri dari serangkaian node, di mana setiap node terdiri dari dua bagian: data itu sendiri dan pointer yang menunjuk ke node berikutnya dalam urutan. Link List memungkinkan penyimpanan dan pengaksesan data secara fleksibel, tanpa memerlukan alokasi memori statis seperti yang terjadi pada array. Dalam matakuliah ini, mahasiswa mempelajari cara membuat, mengakses, dan menghapus elemen dalam Link List, serta operasi-operasi dasar seperti pencarian dan pengurutan.

1. **Double Link List**

Double Link List adalah variasi dari Link List di mana setiap node memiliki dua pointer: satu untuk menunjuk ke node sebelumnya dan satu lagi untuk menunjuk ke node berikutnya. Hal ini memungkinkan traversal maju dan mundur dalam struktur data tersebut. Mahasiswa belajar tentang implementasi dan manfaat dari Double Link List, serta cara menggunakannya dalam pemecahan masalah pemrograman yang lebih kompleks.

1. **Circular Link List**

Circular Link List adalah bentuk khusus dari Link List di mana node terakhir terhubung kembali ke node pertama, membentuk lingkaran. Konsep ini memiliki aplikasi yang bermanfaat dalam implementasi antrian dan buffer sirkular. Dalam matakuliah ini, mahasiswa belajar tentang pembuatan dan pemanfaatan Circular Link List, serta cara mengimplementasikan operasi-operasi dasar seperti penambahan, penghapusan, dan traversal.