LAPORAN JOBSHEET 1 PRATIKUM STRUKTUR DATA

NAMA : MUHAMMAD DEVIN RAHADI

NIM : 23343076

PRODI : INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS NEGERI PADANG

KATA PENGANTAR

Mata kuliah Praktek Struktur Data merupakan salah satu mata kuliah yang wajib diambil oleh mahasiswa teknik informatika. Dalam mata kuliah ini, kita akan belajar mengenai konsep-konsep dasar dari struktur data, seperti pointer, struct, dan array. Selain itu, kita juga akan

mempelajari tentang beberapa jenis linked list, yaitu link list, double link list, dancircular link list. Berikut adalah ringkasan mengenai materi-materi tersebut.

Pointer, Struct, dan Array: Pada dasarnya, pointer, struct, dan array adalah konsep-konsep dasar dari bahasa pemrograman C. Pada praktik struktur data, kita akan mempelajari cara menggunakannya dalam membuat struktur data yang lebih kompleks.

SOAL :

Buatlah artikel singkat tentang review matakuliah Prak Struktur Data dengan topik sebagai berikut :

- Pointer, struct dan array
- Link List
- Double Link List
- Circular Link List

JAWABAN :

Berikut dibawah ini saya sajikan artikel singkat mengenai topik pembahasan pointer, struct dan array, linked list, double link list dan circular link list

Review Mata Kuliah Prak Struktur Data: Pointer, Struct, dan Array hingga Circular Linked List

Mata kuliah Praktikum Struktur Data sering menjadi landasan penting bagi mahasiswa jurusan ilmu komputer atau teknik informatika. Melalui mata kuliah ini, mahasiswa diperkenalkan dengan konsep-konsep dasar dalam pengorganisasian dan manipulasi data, yang menjadi fondasi bagi pengembangan program yang efisien dan andal.

Struktur data merupakan sebuah struktur data adalah cara penyimpanan, penyusunan dan pengaturan data di dalam media penyimpanan komputer sehingga data tersebut dapat digunakan secara efisien.Sedangkan Data adalah representasi dari fakta dunia nyata. Pemograman dalam struktur data ada beberapa macam. Salah satunya adalah pemograman C. Dalam pemograman ini biasanya menggunakan variable Array, Struktur dan Linked List.

Pada artikel ini akan dijelaskan hal yang digunakan dalam struktur data. Diantaranya ada pointer struct dan array, lalu ada linked list, double link list juga circular linked list.

I. Pointer, struct dan array

Pointer merupakan tipe data yang berisi Alamat memori dari sebuah variable. Sedangkan array merupakan struktur data yang digunakan menyimpan sekumpulan data pada satu tempat. Selanjutnya ada struct merupakan Kumpulan dari beberapa variable dengan beragam tipe data yang dibungkus dalam satu variable.
Dalam pembuatannya pointer, struct dan array memiliki perbedaan.

Pada pointer dibuat dengan menambahkan symbol * (asterik) di depan Namanya, kemudian diisi dengan Alamat memori yang akan digunakan dalam referensi contoh :

```
int *pointer1 = 00001;
```

Maka *pointer1 akan bisa mengakses data yang ada pada Alamat memori 00001. Dengan kata lain, si *pointer1 akan menggunakan Alamat 00001 sebagai referensinya. Namun terdapat masalah, Dimana kita tidak bisa lihat daftar Alamat memori secara langsung, kita akan kesulitan memberikan referensi Alamat memori untuk pointer. Maka solusinya kita harus mengambil Alamat memori dari variabel yang lain dengan menggunakan symbol “&” contoh :

```
int score = 50;
int hp = 100;
int *ptr_hp = &hp;
```

pada contoh ini, kita membuat pointer dengan nama *ptr_hp dengan isi alamat memori dari variabel hp. Dengan begini pointer *ptr_hp akan bisa mengakses nilai pada memori hp.

Pada struct dapat kita buat dengan kata kunci struct kemudian diikuti dengan nama struct dan isinya. Contoh :

```
struct Mahasiswa
{
    char *name;
    char *address;
    int age;
};
```

Lalu bagaimana cara penggunaanya?
Agar struct dapat digunakan, kita harus membuat variabel untuknya. Contoh :

```
#include <stdio.h>

// membuat struct
struct Mahasiswa {
    char *name;
    char *address;
    int age;
};

void main(){

    // menggunakan struct
    struct Mahasiswa mhs1, mhs2;

    // mengisi nilai ke struct
    mhs1.name = "Dian";
    mhs1.address = "Mataram";
    mhs1.age = 22;

    mhs2.name = "Bambang";
    mhs2.address = "Surabaya";
    mhs2.age = 23;

    // mencetak isi struct
    printf("## Mahasiswa 1 ##\n");
    printf("Nama: %s\n", mhs1.name);
    printf("Alamat: %s\n", mhs1.address);
    printf("Umur: %d\n", mhs1.age);

    printf("## Mahasiswa 2 ##\n");
    printf("Nama: %s\n", mhs2.name);
    printf("Alamat: %s\n", mhs2.address);
    printf("Umur: %d\n", mhs2.age);

}
```

Kita juga dapat membuat struct dengan menggunakan typedef pada struct. Kata kunci typedef adalah kata kunci untuk mendefinisikan tipe data baru. Kita bisa menggunkan kata kunci ini di depan struct untuk menyatakannya sebagai tipe data baru. Contoh :

***tanpa typedef**

```
// membuat struct
struct Distance{
    int feet;
    float inch;
};

void main() {
    // menggunakan struct
    struct Distance d1, d2;
}
```

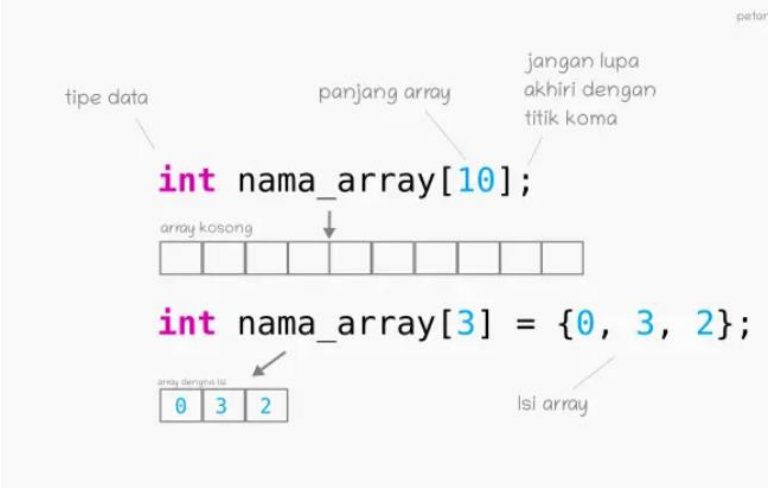
***pakai typedef**

```c
// membuat struct dengan typedef
typedef struct Distance{
    int feet;
    float inch;
} distances;

void main() {
    // menggunakan struct
    distances dist1, dist2, sum;
}
```

Lalu bagaimana dengan array. Pada pemograman C, array dapat kita buat dengan cara seperti ini

```c
// membuat array kosong dengan tipe data integer dan panjang 10
int nama_array[10];

// membuat array dengan langsung diisi
int nama_arr[3] = {0, 3, 2}
```

Cara membuat array hamper sama seperti cara membuat variabel biasa. Bedanya pada array kita harus menentukan panjangnya



Array akan menyimpan sekumpulan data dan memberinya nomer indeks agar mudah diakses. Indeks array selalu dimulai dari 0, misalkan kita punya araay seperti ini:

```c
char huruf[5] = {'a', 'b', 'c', 'd', 'e'};
```

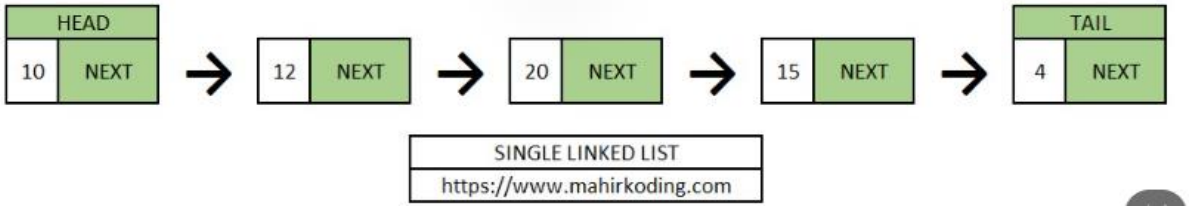
Missal kita ingin mengambil huruf c maka penulisannya adalah :

```c
huruf[2];
```

Kenapa bukan huruf[3]; ?. hal ini karena indeks array selalu dimulai dari nol(0), sehingga untuk mendapatkan nilai c nilai array nya adalah huruf[2];

II. LINKED LIST

Linked List adalah salah satu bentuk implementasi dari struktur data yang paling sederhana. Single linked list bisa kita analogikan sebuah balok data dalam memory yang saling terhubung satu sama lain. Satu blok data dengan blok data lainnya dihubungkan melalui penanda berupa pointer (pointer bertugas menyimpan address blok data selanjutnya).

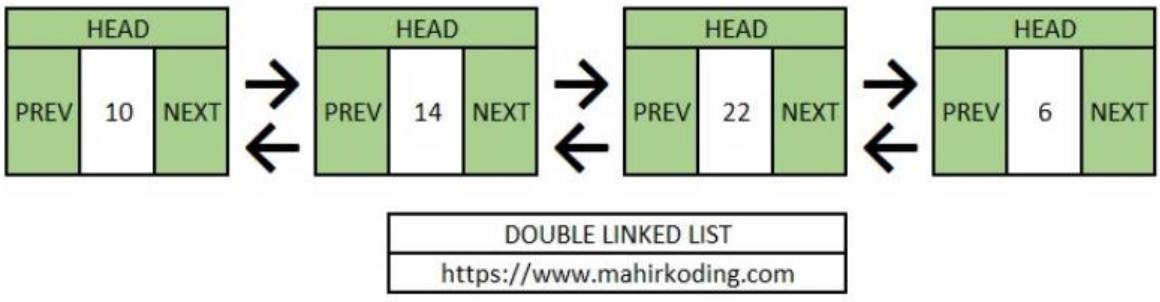


Dalam pembelajaran struktur data, kita akan lebih sering mengenal dengan istilah :

- **Push** untuk menambah data.
  - **PushHead** – Menambah data ke barisan paling awal
  - **PushTail** – Menambah data ke barisan paling akhir
  - **PushMid** – Menambah data ke barisan di tengah (sorting)
- **Pop** untuk menghapus data.
  - **PopHead** – Menghapus data paling awal
  - **PopTail** – Menghapus data paling akhir
  - **PopMid** – Menghapus data ditengah (sesuai parameter value)

III. DOUBLE LINKED LIST

Double artinya blok data yang kita miliki akan memiliki **2 penunjuk kiri dan kanan** untuk menentukan *data sebelum/sesudahnya*. Berbeda dengan single linked list yang hanya mempunyai satu penunjuk, **double linked list** mempunyai 2 penunjuk kiri dan kanan untuk menentukan *urutan datanya*. Agar lebih paham, bisa perhatikan gambar di bawah ini :



Beberapa operasi yang biasanya ada di dalam sebuah *doubly linked list* pada dasarnya sama dengan yang ada di dalam *single linked list,* yakni:

1. Push

Push merupakan sebuah operasi *insert* dimana di dalam linked list terdapat 2 kemungkinan insert, yaitu insert melalui depan (pushDepan) ataupun belakang (pushBelakang). Operasi pushDepan berarti data yang paling baru dimasukkan akan

berada di depan data lainnya, dan sebaliknya pushBelakang berarti data yang paling baru akan berada di belakang data lainnya.
Representasinya adalah sebagai berikut:

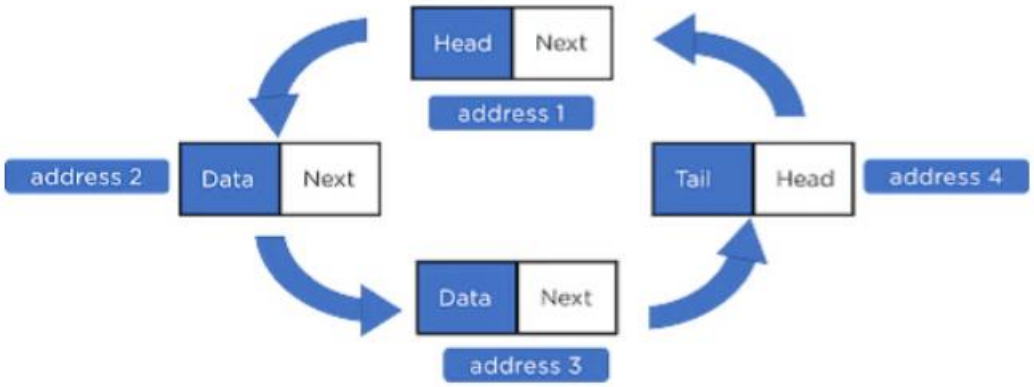
- pushDepan: 5, 3, 7, 9 maka hasilnya adalah berturut-turut: 9, 7, 3, 5
- pushBelakang: 5, 3, 7, 9 maka hasilnya adalah 5, 3, 7, 9

2. Pop

Pop, kebalikan dari push, merupakan operasi *delete*, dimana di dalam *linked list* memiliki 2 kemungkinan *delete*, yaitu melalui depan (popDepan) dan melalui belakang (popBelakang). PopDepan berarti data yang akan dihapus adalah data paling depan, dan popBelakang berarti data yang akan dihapus adalah data paling belakang (akhir).

IV. CIRCULAR LINKED LIST

Circular linked list adalah suatu linked list yang tidak memiliki nilai nil/NULL untuk medan sambungannya di mana node terakhir terhubung ke node pertama. Circular linked list akan dilakukan sampai kita mencapai simpul yang sama dengan nilai awal yang kita buat.



Jadi pertama-tama, mari kita pertimbangkan empat elemen untuk dimasukkan ke dalam daftar. Untuk itu buatlah empat node, dimana setiap node terdiri dari bagian data dan alamat yang disimpan di beberapa alamat acak. Dalam single linked list, simpul terakhir penunjuk berikutnya menunjuk ke Null. Namun ketika kita membandingkan circular single linked list dengan single linked list, kita tidak menemukan nol.

Dalam circular linked list, penunjuk berikutnya dari simpul terakhir menyimpan alamat simpul pertama. Dengan kata lain, kita dapat mengatakan bahwa “Alamat simpul Ekor mengarah ke simpul Kepala dari daftar tertaut.”

Setiap node dalam daftar tertaut terhubung satu sama lain melalui penunjuk yang menunjuk ke alamat node berikutnya, dan panah dalam diagram ini mewakili alamat tersebut.