TUGAS PENDAHULUAN
MODUL IX
LINKED LIST

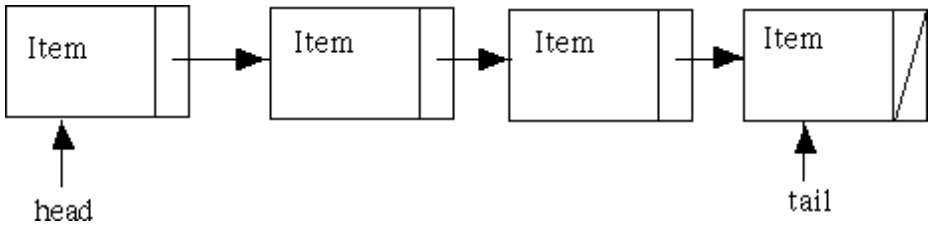
Pengenalan

Linked list

Linked List atau dikenal juga dengan sebutan senarai berantai adalah struktur data yang terdiri dari urutan record data dimana setiap record memiliki field yang menyimpan alamat/referensi dari record selanjutnya (dalam urutan). Elemen data yang dihubungkan dengan link pada Linked List disebut Node. Biasanya didalam suatu linked list, terdapat istilah head dan tail. Head adalah elemen yang berada pada posisi pertama dalam suatu linked list Tail adalah elemen yang berada pada posisi terakhir dalam suatu linked list Ada beberapa macam Linked List, yaitu:

1. Single Linked List
2. Double Linked List
3. Circular Linked List
4. Multiple Linked List.

Single Linked List Single Linked List merupakan suatu linked list yang hanya memiliki satu variabel pointer saja. Dimana pointer tersebut menunjuk ke node selanjutnya. Biasanya field pada tail menunjuk ke NULL. contoh:



Linked list vs Array

Linked List	Array
Koleksi linear dari elemen data	Koleksi linear dari node
Menyimpan data dalam lokasi memori yang berurutan	Tidak menyimpan node dalam lokasi memori yang berurutan
Dapat diakses secara acak menggunakan indeks	Hanya dapat diakses secara berurutan (Dari head ke tail)
Jumlah memory dapat bertambah jika dibutuhkan	Jumlah memory sudah fix sesuai jumlah array yang di inisialisasi di awal

Memory Allocation

Dalam C/C++, alokasi memory dapat dilakukan dengan menggunakan malloc, sedangkan untuk dealokasi dapat menggunakan free. Fungsi free hanya membebaskan memory tetapi tidak menghapus isi dari memory tersebut. contoh penggunaan malloc:

```
int *px = (int *) malloc(sizeof(int));
char *pc = (char *) malloc(sizeof(char));
struct Facebook *curr = (struct Facebook*) malloc(sizeof(struct Facebook));
```

contoh penggunaan free: free(curr);

Alokasi suatu memory biasanya dibutuhkan didalam linked list saat akan menambah node/data baru.

Insert dan Delete Node

dalam Single Linked List Insert (push) dan delete (pop) node pada linked list dapat dilakukan pada posisi depan (head), tengah (mid) dan belakang (tail).

Insert (Push)

Contoh code push depan :

```
void pushDepan(char nama[], int usia){
    struct Facebook *curr = (struct Facebook*) malloc(sizeof(struct Facebook));
    strcpy(curr->nama,nama);
    curr->usia = usia;
    if(head == NULL){
        head = tail = curr;
        tail->next = NULL;
    }else{
        curr->next = head;
        head = curr;
    }
}
```

Contoh code push belakang :

```
void pushBelakang(char nama[], int usia){
    struct Facebook *curr = (struct Facebook*) malloc(sizeof(struct Facebook));
    strcpy(curr->nama,nama);
    curr->usia = usia;
    if(head == NULL){
        head = tail = curr;
        tail->next = NULL;
    }else{
        tail->next = curr;
        tail = curr;
        tail->next = NULL;
    }
}
```

Delete (pop)

Contoh code:

```
void popDepan(){
    if(head == NULL){
        system("color 0A");
        printf("\nThere is no data yet\n");
        getchar();
    }else{
        struct Facebook *curr = head;
        if(head == tail){
            //cuma ada 1 node;
            head = tail = NULL;
            printf("Data %s",curr->nama);
            free(curr);
        }else{
            head = head->next;
            printf("Data %s",curr->nama);
            free(curr);
        }
        printf(" sudah dihapus\n");
        getchar();
    }
}
```

(sumber: http://suciantinovi.blogspot.co.id/, diakses 23 April 2017)

Problem 1: Implementasi Strukutur data Linked List

Definisi Masalah

Buatlah program yang dapat menerima sebuah masukkan bertipe integer ke dalam sebuah element (node) list, yang index dari element (node) tersebut berurutan dimulai dari data terakhir diinput. Banyaknya data yang diinput ditentukan oleh pengguna hingga pengguna selesai memberikan inputan data yang diinginkan. Kemudian data yang sudah diinput ditampilkan ke dalam bentuk data linked list yang telah dibuat.

Note:
Lihat Contoh program → problem.exe
Gunakan Template file "Template_problem.c" yang dapat diunduh pada Google Classroom PMC 2019 untuk mengerjakan problem ini. **TIDAK DIPERBOLEHKAN** menambahkan ataupun memodifikasi parameter pada fungsi/procedure yang diberikan.

Kompilasi kode tersebut dengan GCC lalu jalankan dan lihat hasilnya. Untuk menjalankan *executable file*, gunakan *command prompt* pada Windows lalu berpindah ke direktori tempat *executable file* berada. Kemudian, tulis nama *executable file* tersebut lalu tekan Enter.

Petunjuk Penyerahan Tugas Praktikum Modul IX

Simpan file (`TP_Problem01.c`) dalam satu folder. Gunakan program WinRAR untuk mengkompresi menjadi arsip .Zip. Penamaan file Zip menggunakan format yang telah diberikan. File .Zip ini yang akan di-submit melalui Google Classroom Praktikum PMC 2019. Hanya file kode saja yang dimasukkan ke dalam arsip .Zip. File *executable* tidak perlu dimasukkan.

Selesai