LAPORAN PRAKTIKUM

Algoritma dan Struktur Data

Week 4 - 5 : Single Linked List

MUHAMMAD FARIS ISA

D4LJ – Teknik Informatika

3122640005

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

2022

1. Program Urut Angka

Membuat sebuah program urut angka menggunakan single linked list.

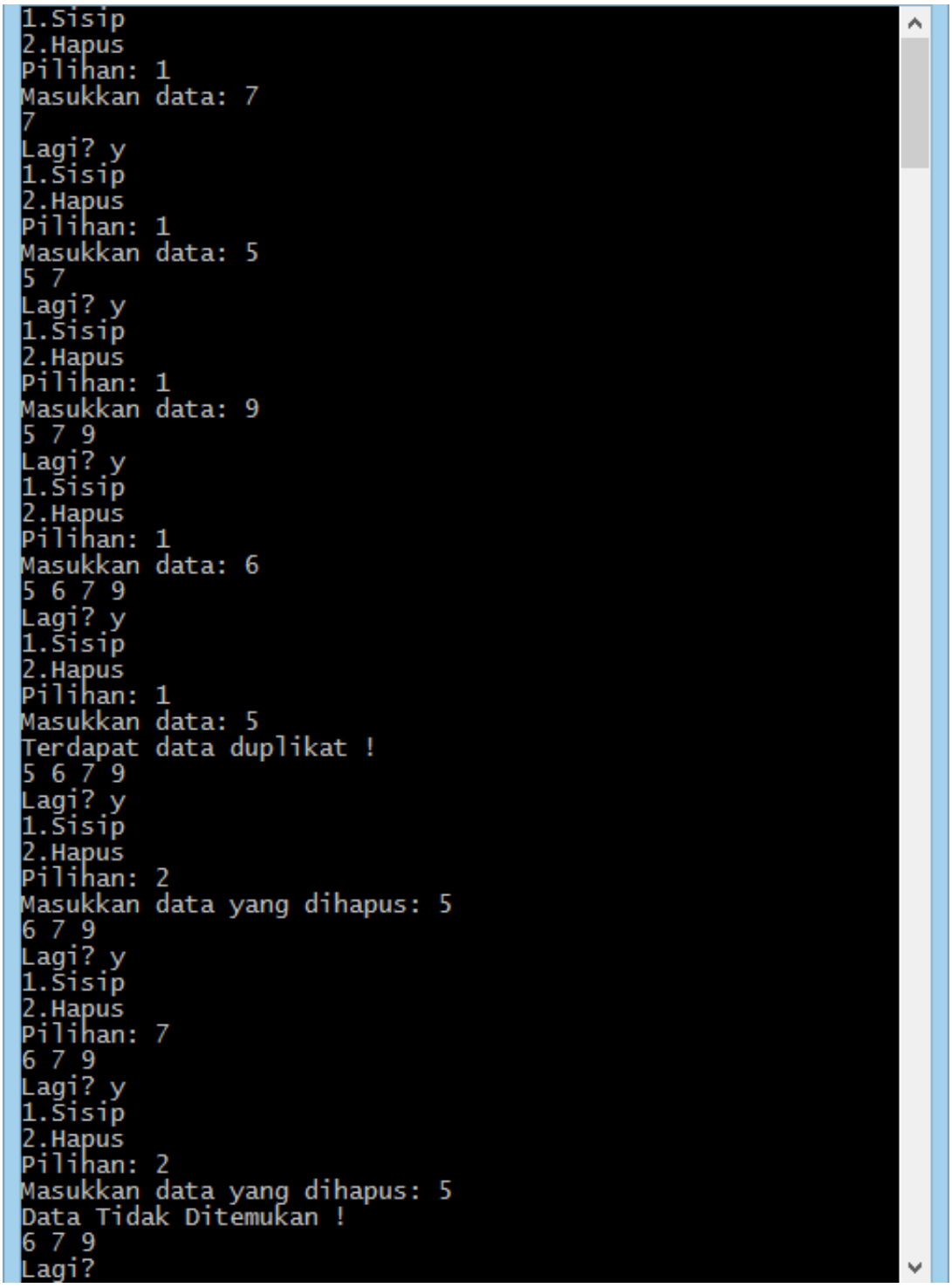
- Source code :

```
#include <stdio.h>
#include <stdlib.h>
typedef struct simpul Node;
struct simpul {
    int data;
    Node *next;
};
Node *head = NULL, *baru;

struct Node* alokasi_simpul(int x){
    baru = (Node *) malloc (sizeof(Node));
    baru->data = x;
    baru->next = NULL;

    return baru;
}

void cetak(Node* p){
    Node *temp = p;
    while ( temp != NULL){
        printf("%d ", temp->data);
        temp = temp->next;
    }
    printf("\n");
}

void sisipUrut(Node** head, Node* baru){
    Node* current;
    if (*head == NULL || (*head)->data > baru->data){
        baru->next = *head;
        *head = baru;
    } else {
        current = *head;
        if (current->data == baru->data){
            return printf("Terdapat data duplikat ! \n");
        } else {
            while (current->next != NULL && current->next->data <= baru-
>data){
                if(current->next->data == baru->data)
                    return printf("Terdapat data duplikat ! \n");
                else
                    current = current->next;
            }
```

```
            baru->next = current->next;
            current->next = baru;
        }
    }
}

void free_Node(Node* p){
    free(p);
    p = NULL;
}

void hapus(int x){
    Node *hapus = head, *before;
    if (hapus->data == x){
        head = hapus->next;
        free_Node(hapus);
    } else {
        while(hapus->data != x){
            if(hapus->next == NULL){
                return printf("Data Tidak Ditemukan ! \n" );
            } else {
                before = hapus;
                hapus = before->next;
            }
        }
        before->next = hapus->next;
        free_Node(hapus);
    }
}

int main(){
    int pilih, dt, x;
    char lagi='y';
    while(lagi=='y'){
        printf("1.Sisip \n");
        printf("2.Hapus\n");
        printf("Pilihan: ");scanf("%d",&pilih);
        switch(pilih){
        case 1: printf("Masukkan data: ");scanf("%d",&dt);
                baru = alokasi_simpul(dt);
                sisipUrut(&head, baru);
                break;
        case 2: printf("Masukkan data yang dihapus: ");scanf("%d",&dt);
                hapus(dt);
                break;
        }
        cetak(head);
        printf("Lagi? ");fflush(stdin);scanf("%c",&lagi);
```

```
    }
}
```

- Hasil :


1.Sisip
2.Hapus
Pilihan: 1
Masukkan data: 7
7
Lagi? y
1.Sisip
2.Hapus
Pilihan: 1
Masukkan data: 5
5 7
Lagi? y
1.Sisip
2.Hapus
Pilihan: 1
Masukkan data: 9
5 7 9
Lagi? y
1.Sisip
2.Hapus
Pilihan: 1
Masukkan data: 6
5 6 7 9
Lagi? y
1.Sisip
2.Hapus
Pilihan: 1
Masukkan data: 5
Terdapat data duplikat !
5 6 7 9
Lagi? y
1.Sisip
2.Hapus
Pilihan: 2
Masukkan data yang dihapus: 5
6 7 9
Lagi? y
1.Sisip
2.Hapus
Pilihan: 7
6 7 9
Lagi? y
1.Sisip
2.Hapus
Pilihan: 2
Masukkan data yang dihapus: 5
Data Tidak Ditemukan !
6 7 9
Lagi?

- Analisa :

  Program ini merupakan program untuk mengurutkan input angka. Pertama-tama program akan mencetak menu yang bisa dipilih, terdapat menu Sisip dan menu Hapus. Apabila menu sisip dijalankan, program akan meminta pengguna untuk menginputkan angka. Apabila pengguna memilih menu hapus, maka pengguna diminta untuk menginputkan angka yang akan dihapus.

  Saat pengguna memilih menu sisip apabila tidak ada sesuatu didalam list, maka angka tersebut akan langsung diinputkan. Apabila sudah terdapat angka didalam list, maka akan difilter lagi apakah angka tersebut sama dengan yang berada di list, jika sama maka akan langsung direturn print cetak. Apabila tidak ada nilai yang sama, maka akan dicari apakah nilai tersebut lebih dari yang tercatat di list. Jika nilai lebih dari yang terdaftar dilist, maka akan diinputkan ke dalam list sebagai node.

  Pada saat memilih menu hapus program akan mencari angka yang sesuai dengan yang diinput. Jika yang diinput sama dengan yang berada pada head maka akan langsung dihapus. Sedangkan, jika angka yang diinput berbada akan melakukan loop hingga menemukan angka yang sama lalu menghapusnya. Jika program tidak menemukan angka yang sesuai, maka program akan mengembalikan pesan error dan akan mencetak kembali linked listnya.

2. Program Urut Angka Polinom

Membuat sebuah program urut angka dan cetak polinom menggunakan single linked list.

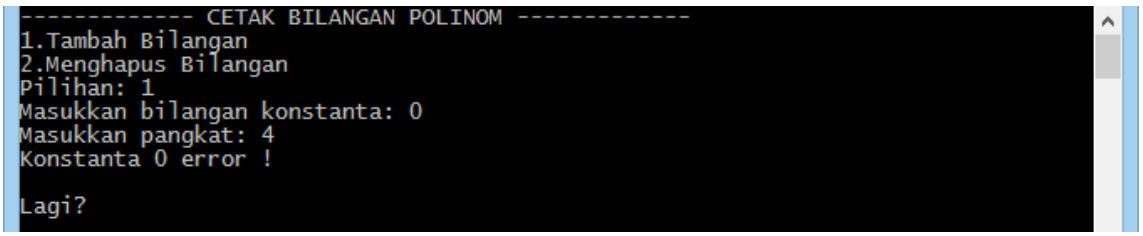
- Source code :

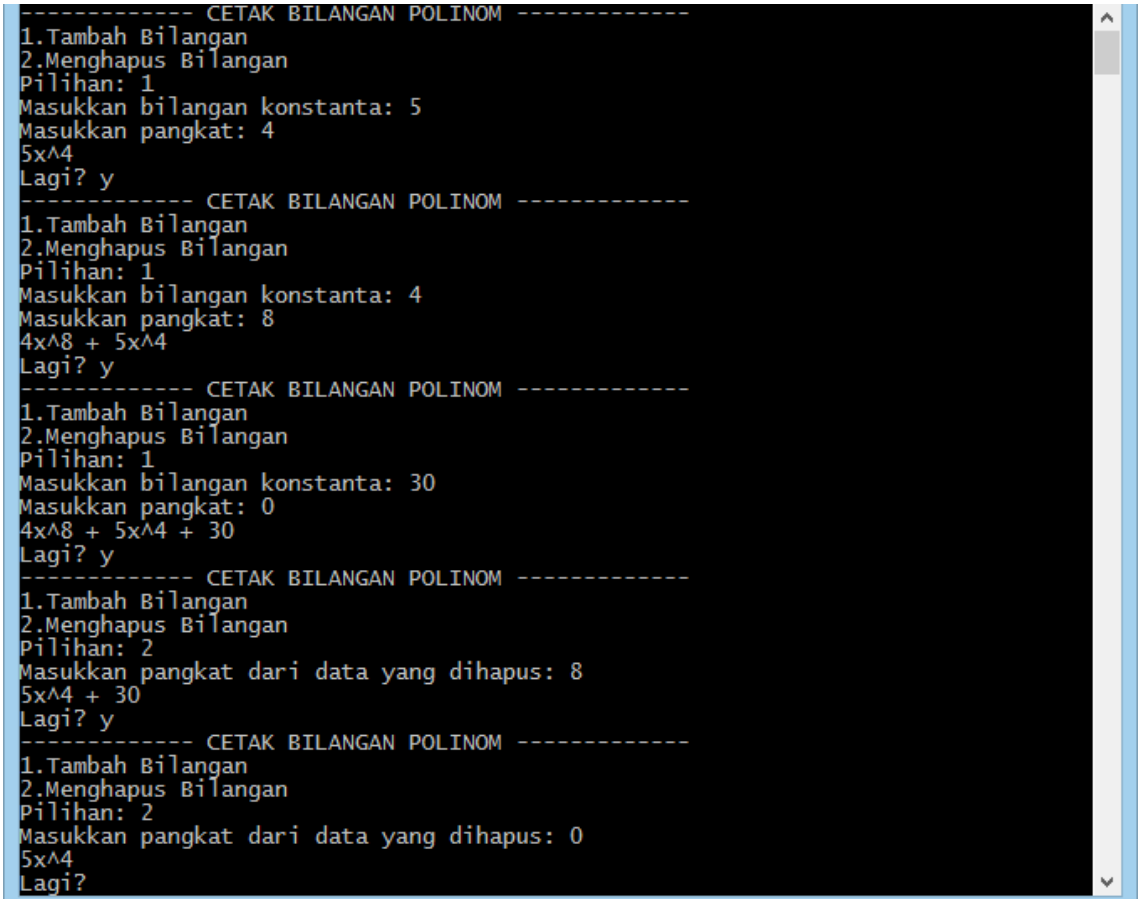
```
#include <stdio.h>
#include <stdlib.h>
typedef struct simpul Node;
struct simpul {
    int konstanta;
    int pangkat;
    Node *next;
};
Node *head = NULL, *baru;

struct Node* alokasi_simpul(int konstanta, int pangkat){
    baru = (Node *) malloc (sizeof(Node));
    baru->konstanta = konstanta;
    baru->pangkat = pangkat;
    baru->next = NULL;

    return baru;
}

void cetak(Node* p){
    Node *temp = p;
    int konstanta;
    while ( temp != NULL){
        if (temp->konstanta > 1)
            printf("%d", temp->konstanta);
        if (temp->pangkat > 1)
            printf("x^%d", temp->pangkat);
        else if (temp->pangkat == 1)
            printf("x");
        if(temp->next != NULL)
            printf(" + ");
        temp = temp->next;
    }
}
```

```
void sisipUrut(Node** head, Node* baru){
    Node* current;
    if (*head == NULL || (*head)->pangkat < baru->pangkat){
        baru->next = *head;
        *head = baru;
    } else {
        current = *head;
        if (current->pangkat == baru->pangkat){
            return printf("Terdapat data duplikat ! \n");
        } else {
            while (current->next != NULL && current->next->pangkat >=
baru->pangkat){
                if(current->next->pangkat == baru->pangkat)
                    return printf("Terdapat data duplikat ! \n");
                else
                    current = current->next;
            }
            baru->next = current->next;
            current->next = baru;
        }
    }
}

void free_Node(Node* p){
    free(p);
    p = NULL;
}

void hapus(int x){
    Node *hapus = head, *before;
    if (hapus->pangkat == x){
        head = hapus->next;
        free_Node(hapus);
    } else {
        while(hapus->pangkat != x){
            if(hapus->next == NULL){
                return printf("Data Tidak Ditemukan ! \n" );
            } else {
                before = hapus;
                hapus = before->next;
            }
        }
        before->next = hapus->next;
        free_Node(hapus);
    }
}
```

```c
int main(){
    int pilih, konstanta, pangkat;
    char lagi='y';
    while(lagi=='y'){
        printf("------------- CETAK BILANGAN POLINOM -------------\n");
        printf("1.Tambah Bilangan \n");
        printf("2.Menghapus Bilangan\n");
        printf("Pilihan: ");scanf("%d",&pilih);
        switch(pilih){
        case 1: printf("Masukkan bilangan konstanta:
");scanf("%d",&konstanta);
                printf("Masukkan pangkat: ");scanf("%d",&pangkat);
                if (konstanta > 0){
                    baru = alokasi_simpul(konstanta, pangkat);
                    sisipUrut(&head, baru);
                } else {
                    printf("Konstanta 0 error ! \n");
                }
                break;
        case 2: printf("Masukkan pangkat dari data yang dihapus:
");scanf("%d",&pangkat);
                hapus(pangkat);
                break;
        }
        cetak(head);printf("\n");
        printf("Lagi? ");fflush(stdin);scanf("%c",&lagi);
    }
}
```

- Hasil :





- Analisa :

  Program ini merupakan program untuk menginputkan dan mengurutkan bilangan polinom. Pada program ini terdapat dua menu yaitu menginputkan bialngan dan pangkat serta menghapus bilangan berpangkat.

  Saat pengguna memilih menu input, maka pengguna akan diminta menginputkan bilangan konstanta dan juga bilangan berpangkat. Apabila bilangan konstanta yang diinputkan merupakan bilangan 0 maka program akan keluar dan kembali bertanya apakah ingin menginputkan ulang. Hal tersebut dikarenakan apabila konstanta yang diinput 0 maka tidak perlu diinput karena setiap perkalian yang dikali 0 maka akan menghasilkan 0. Selanjutnya jika bilangan lebih dari 0 maka program akan membuat node dengan menggunakan method `alokasi_simpul(konstanta, pangkat);` dan juga method tersebut akan mengembalikan niai Node dan Node yang sebelumnya sudah dibuat dengan variabel baru, akan dimasukkan lagi kedalam method `sisipUrut(&head, baru)`. Didalam method `alokasi_simpul` digunakan untuk membuat node baru dan mereturn nilai dari node tersebut. Setelah itu method `sisipUrut` akan menginputkan urutan dari polinom sesuai dengan yang seharusnya tertulis. Jika pangkat yang berada di head itu lebih kecil dari pada pangkat baru, maka head yang baru nanti akan menjadi nilai head yang baru. Sedangkan, apabila nilai dari pangkat yang dinputkan sudah terdaftar maka akan mereturn cetak error bahwa data sudah terdaftar.

Bila pengguna memilih menu hapus, maka pengguna akan diminta untuk menginputkan nilai dari pangkat mana yang mau pengguna hapus. Apabila pengguna menginputkan nilai yang tidak terdaftar didalam bilangan polinom, maka akan mengembalikan error bahwa nilai tidak terdaftar. Apabila pengguna menginputkan sesuai dengan pangkat yang ada, maka program akan menjalankan eksekusi dengan sesuai seperti yang tertera di gambar.

KESIMPULAN

1. Penggunaan single linked list lebih dinamis dibandingkan menggunakan array yang sedari awal datanya sudah di tentukan berapa panjangnya.
2. Penggunaan single linked list dapat disisipkan nilainya, baik diawal, ditengah ataupun diakhir.
3. Single linked list juga dapat menghapus setiap nilai yang berada diawal, ditengah dan juga diakhir.
4. Single linked list lebih meringankan penggunaan memori dikarenakan setiap node yang dihapus membebaskan penggunaan memori.