**TUGAS PRAKTIKUM ALGORITMA & STRUKTUR DATA**

**Jilid 4**



**Oleh :**

**Nama : Rosi Arif Mulyadi**

**NRP : 3121522021**

**Prodi : D3 Teknik Informatika PENS PSDKU Sumenep**

**Kelas : 1 ITA D3 Sumenep**

**Dosen :**

**LUSIANA AGUSTIEN M.Kom**

**POLITEKNIK ELEKTRONIKA NEGERI SURABAYA**

**MODUL 2.1**

**LINKED LIST (SINGLE LINKED LIST – Input elemen dari beberapa posisi)**

**B. Kegiatan Praktikum**

1. Implementasikan dan tentukan output percobaan yang ada dalam modul praktikum ini dan lakukan analisa pada tiap fungsi yang dibuat.

Jawab :

Listing Program :

#include <stdio.h>

#include <stdlib.h>

struct node

{

int val;

struct node \*next;

};

void print\_list(struct node \*head)

{

printf("H->");

while(head)

{

printf("%d->", head->val);

head = head->next;

}

printf("|||\n\n");

}

void insert\_front(struct node \*\*head, int value)

{

struct node \* new\_node = NULL;

new\_node = (struct node \*)malloc(sizeof(struct node));

if(new\_node == NULL)

{

printf("tidak bisa menginputkan node baru, memori penuh");

}

new\_node->val = value;

new\_node->next = \*head;

\*head = new\_node;

}

void insert\_end(struct node \*\*head, int value)

{

struct node \* new\_node = NULL;

struct node \* last = NULL;

new\_node = (struct node \*)malloc(sizeof(struct node));

if(new\_node == NULL)

{

printf("tidak bisa menginputkan node baru, memori penuh");

}

new\_node->val = value;

new\_node->next = NULL;

if( \*head == NULL)

{

\*head = new\_node;

return;

}

last = \*head;

while(last->next) last = last->next;

last->next = new\_node;

}

void insert\_after(struct node \*head, int value, int after)

{

struct node \* new\_node = NULL;

struct node \*tmp = head;

while(tmp)

{

if(tmp->val == after)

{

new\_node = (struct node \*)malloc(sizeof(struct node));

if(new\_node == NULL)

{

printf("tidak bisa menginputkan node baru, memori penuh");

}

new\_node->val = value;

new\_node->next = tmp->next;

tmp->next = new\_node;

return;

}

tmp = tmp->next;

}

}

void insert\_before(struct node \*\*head, int value, int before)

{

struct node \* new\_node = NULL;

struct node \* tmp = \*head;

new\_node = (struct node \*)malloc(sizeof(struct node));

if(new\_node == NULL)

{

printf("tidak bisa menginputkan node baru, memori penuh");

return;

}

new\_node->val = value;

if((\*head)->val == before)

{

new\_node->next = \*head;

\*head = new\_node;

return;

}

while(tmp && tmp->next)

{

if(tmp->next->val == before)

{

new\_node->next = tmp->next;

tmp->next = new\_node;

return;

}

tmp = tmp->next;

}

free(new\_node);

}

void main()

{

int count = 0, i, val, after, before;

struct node \* head = NULL;

printf("Inputkan jumlah elemen/node yang akan di inputkan : ");

scanf("%d", &count);

for(i = 0; i < count; i++)

{

printf("Enter %d th elemen/node : ", i);

scanf("%d", &val);

insert\_front(&head, val);

}

printf("Initial List : ");

print\_list(head);

printf("Inputkan nilai untuk di sisipkan pada awal list : ");

scanf("%d", &val);

insert\_front(&head, val);

printf("List setelah input data baru : ");

print\_list(head);

printf("Inputkan nilai untuk di sisipkan pada akhir list : ");

scanf("%d", &val);

insert\_end(&head, val);

printf("List setelah input data baru : ");

print\_list(head);

printf("Inputkan nilai yang akan di sisipkan pada list : ");

scanf("%d", &val);

printf("Di sisipkan setelah ? : ");

scanf("%d", &after);

insert\_after(head, val, after);

printf("List setelah input data baru : ");

print\_list(head);

printf("Inputkan nilai yang akan di sisipkan pada list : ");

scanf("%d", &val);

printf("Di sisipkan sebelum ? : ");

scanf("%d", &before);

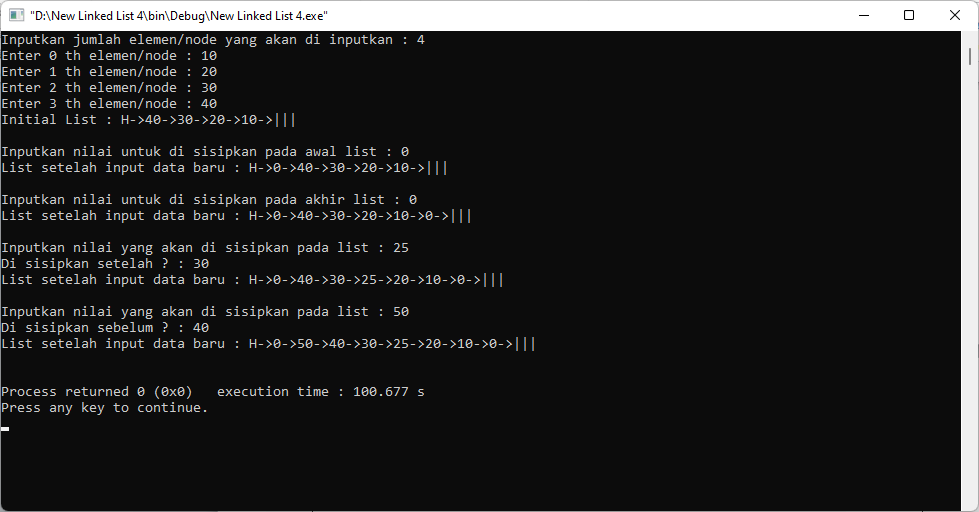
insert\_before(&head, val, before);

printf("List setelah input data baru : ");

print\_list(head);

}

Output :



Penjelasan :

* Fungsi print\_list digunakan untuk mengisi inputan linked list pada program.
* Fungsi insert\_front digunakan untuk mengisi inputan pada linked list di awal inputan.
* Fungsi insert\_end digunakan untuk mengisi inputan pada linked list di akhir inputan.
* Fungsi insert\_after digunakan untuk mengisi inputan yang disisipkan pada salah satu inputan tapi diinput setelah inputan tersebut.
* Fungsi insert\_before digunakan untuk mengisi inputan yang disisipkan pada salah satu inputan tapi diinput sebelum inputan tersebut.

**C. Tugas Praktikum**

1. Implementasikan 3 algoritma penambahan node baru (end, after, dan Before) pada program percobaan yang ada pada modul sebelumnya (modul 2).

Jawab :

#include <stdio.h>

#include <stdlib.h>

struct node

{

char nama[50];

int val;

struct node \*next;

};

void print\_list(struct node \*head)

{

printf("H->");

while(head)

{

printf("%s %i ->",head->nama, head->val);

head = head->next;

}

printf("|||\n\n");

}

void insert\_front(struct node \*\*head, char nama[],int value)

{

struct node \* new\_node = NULL;

new\_node = (struct node \*)malloc(sizeof(struct node));

if(new\_node == NULL)

{

printf("tidak bisa menginputkan node baru, memori penuh");

}

strcpy(new\_node->nama,nama);

new\_node->val = value;

new\_node->next = \*head;

\*head = new\_node;

}

void insert\_end(struct node \*\*head, char nama[], int value)

{

struct node \* new\_node = NULL;

struct node \* last = NULL;

new\_node = (struct node \*)malloc(sizeof(struct node));

if(new\_node == NULL)

{

printf("tidak bisa menginputkan node baru, memori penuh");

}

strcpy(new\_node->nama,nama);

new\_node->val = value;

new\_node->next = NULL;

if( \*head == NULL)

{

\*head = new\_node;

return;

}

last = \*head;

while(last->next) last = last->next;

last->next = new\_node;

}

void insert\_after(struct node \*head, char nama[],int value, int after)

{

struct node \* new\_node = NULL;

struct node \*tmp = head;

while(tmp)

{

if(tmp->val == after)

{

new\_node = (struct node \*)malloc(sizeof(struct node));

if(new\_node == NULL)

{

printf("tidak bisa menginputkan node baru, memori penuh");

}

strcpy(new\_node->nama,nama);

new\_node->val = value;

new\_node->next = tmp->next;

tmp->next = new\_node;

return;

}

tmp = tmp->next;

}

}

void insert\_before(struct node \*\*head, char nama[], int value, int before)

{

struct node \* new\_node = NULL;

struct node \* tmp = \*head;

new\_node = (struct node \*)malloc(sizeof(struct node));

if(new\_node == NULL)

{

printf("tidak bisa menginputkan node baru, memori penuh");

return;

}

strcpy(new\_node->nama,nama);

new\_node->val = value;

if((\*head)->val == before)

{

new\_node->next = \*head;

\*head = new\_node;

return;

}

while(tmp && tmp->next)

{

if(tmp->next->val == before)

{

new\_node->next = tmp->next;

tmp->next = new\_node;

return;

}

tmp = tmp->next;

}

free(new\_node);

}

void main()

{

int count = 0, i, val, after, before;

char nama[10];

struct node \* head = NULL;

printf("Inputkan jumlah elemen/node yang akan di inputkan : ");

scanf("%d", &count);

fflush(stdin);

for(i = 0; i < count; i++)

{

printf("Enter %d th elemen/node : ", i);

scanf("%s", &nama);

fflush(stdin);

printf("Enter %d th elemen/node : ", i);

scanf("%d", &val);

fflush(stdin);

insert\_front(&head, nama, val);

}

printf("Initial List : ");

print\_list(head);

printf("Inputkan nama untuk di sisipkan pada awal list : ");

scanf("%s", &nama);

fflush(stdin);

printf("Inputkan nilai untuk di sisipkan pada awal list : ");

scanf("%d", &val);

fflush(stdin);

insert\_front(&head, nama, val);

printf("List setelah input data baru : ");

print\_list(head);

printf("Inputkan nama untuk di sisipkan pada awal list : ");

scanf("%s", &nama);

fflush(stdin);

printf("Inputkan nilai untuk di sisipkan pada akhir list : ");

scanf("%d", &val);

fflush(stdin);

insert\_end(&head, nama, val);

printf("List setelah input data baru : ");

print\_list(head);

printf("Inputkan nama untuk di sisipkan pada awal list : ");

scanf("%s", &nama);

fflush(stdin);

printf("Inputkan nilai yang akan di sisipkan pada list : ");

scanf("%d", &val);

fflush(stdin);

printf("Di sisipkan setelah ? : ");

scanf("%d", &after);

fflush(stdin);

insert\_after(head, nama, val, after);

printf("List setelah input data baru : ");

print\_list(head);

printf("Inputkan nama untuk di sisipkan pada awal list : ");

scanf("%s", &nama);

fflush(stdin);

printf("Inputkan nilai yang akan di sisipkan pada list : ");

scanf("%d", &val);

fflush(stdin);

printf("Di sisipkan sebelum ? : ");

scanf("%d", &before);

fflush(stdin);

insert\_before(&head, nama, val, before);

printf("List setelah input data baru : ");

print\_list(head);

}

Output :

