

TUGAS PRAKTIKUM ALGORITMA & STRUKTUR DATA

Jilid 6



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 3.1

LINKED LIST (Reverse Linked List)

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");
```

```
}
```

```
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("Failed to insert element. Out of memory");
```

```
    }
```

```
    new_node->val = value;
```

```

    new_node->next = *head;
    *head = new_node;
}

void reverse_linked_list(struct node **head)
{
    struct node * new_head = NULL;
    struct node * tmp = NULL;
    while(*head)
    {
        tmp = *head;
        *head = (*head)->next;
        tmp->next = new_head;
        new_head = tmp;
    }
    *head = new_head;
}

void main()
{
    int count = 0, i, val;
    struct node * head = NULL;
    printf("Enter number of elements : ");
    scanf("%d", &count);
    for(i=0; i<count; i++)
    {
        printf("Enter %d th element : ", i);
        scanf("%d", &val);
        insert_front(&head, val);
    }
    printf("Initial List : ");
    print_list(head);
    reverse_linked_list(&head);
    printf("Reverse List : ");
    print_list(head);
}

```

Output :

```
"D:\Linked List (Reverse) 1\bin\Debug\Linked List (Reverse) 1.exe"
Enter number of elements : 4
Enter 0 th element : 10
Enter 1 th element : 20
Enter 2 th element : 30
Enter 3 th element : 40
Initial List : H->40->30->20->10->|||
Reverse List : H->10->20->30->40->|||

Process returned 0 (0x0)   execution time : 8.516 s
Press any key to continue.
```

Analisa :

- Reverse digunakan untuk mengubah dan mengurutkan value pada percobaan.
- Dapat mengurutkan value sesuai dengan element yang dimasukkan / diinputkan.

C. Tugas Praktikum

1. Implementasikan algoritma reverse pada program percobaan yang ada pada modul 2.

Jawab :

Algoritma :

1. Keluarkan simpul pertama dari daftar, satu pointer baru akan menunjuk ke node pertama dan head akan berpindah ke node baru yang di inputkan pada list baru
2. Ulangi langkah sebelumnya sampai semua node pindah ke list baru.
3. Arahkan head ke node pertama dari list baru tersebut.

Listing Program (Reverse) :

```
#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");
}
```

```
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("Failed to insert element. Out of memory");
    }
    strcpy(new_node->nama,nama);
    new_node->val = value;
    new_node->next = *head;
    *head = new_node;
}
```

```
void reverse_linked_list(struct node **head, char nama[],int value)
```

```
{
    struct node * new_head = NULL;
    struct node * tmp = NULL;
    while(*head)
    {
        tmp = *head;
        *head = (*head)->next;
        tmp->next = new_head;
        new_head = tmp;
    }
    *head = new_head;
```

```
}
```

```
void main()
```

```
{
```

```
    int count = 0, i, val;
```

```
    char nama[10];
```

```
    struct node * head = NULL;
```

```
    printf("Enter number of elements : ");
```

```
    scanf("%d", &count);
```

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

```
    {
```

```
        printf("Enter %d th element : ", i);
```

```
        scanf("%s", &nama);
```

```
        fflush(stdin);
```

```
        printf("Enter %d th element : ", i);
```

```
        scanf("%d", &val);
```

```
        fflush(stdin);
```

```
        insert_front(&head, nama, val);
```

```
    }
```

```
    printf("Initial List : ");
```

```
    print_list(head);
```

```
    reverse_linked_list(&head, nama, val);
```

```
    printf("Reverse List : ");
```

```
    print_list(head);
```

```
}
```

Output :

```
"D:\Linked List (Reverse) 1\bin\Debug\Linked List (Reverse) 1.exe"
Enter number of elements : 4
Enter 0 th element : Agus
Enter 0 th element : Agus
Enter 1 th element : Bayu
Enter 1 th element : Bayu
Enter 2 th element : Rina
Enter 2 th element : Rina
Enter 3 th element : Melly
Enter 3 th element : Melly
Initial List : H->Melly 0 ->Rina 0 ->Bayu 0 ->Agus 0 ->|||
Reverse List : H->Agus 0 ->Bayu 0 ->Rina 0 ->Melly 0 ->|||

Process returned 0 (0x0)   execution time : 18.559 s
Press any key to continue.
```