

Laporan Resmi
Praktikum Algoritma dan Struktur Data
Single Linked List Insert



Dr. Tita Karlita S.Kom, M.Kom

Nama : Marits Ikmal Yasin

Kelas : 1D4 IT B

NRP : 3121600047

1. Insert Awal

Kode :

```
#include<stdio.h>
#include<stdlib.h>

typedef struct simpul node;
struct simpul{
    int data;
    node *next;
};

node *head = NULL, *p;

void alokasi();
void awal();
void tampil();

int main(){
    char jawab;

    puts("Single Linked List - Insert Awal");
    do{
        fflush(stdin);
        alokasi();
        awal();
        fflush(stdin);
        printf("Lagi (y/t)? ");
        jawab = getchar();
    }while(jawab == 'y' || jawab == 'Y');
    tampil();
    return 0;
}

void alokasi(){
    int bilangan;

    printf("Nilai yang akan disimpan : ");
    scanf("%d",&bilangan);
    p = (node *)malloc(sizeof(node));
    if(p == NULL){
        puts("Gagal memesam alamat");
        exit(0);
    }
}
```

```

    }
    else{
        p->data = bilangan;
        p->next = NULL;
    }
}

void awal(){
    if(head != NULL)
        p->next = head;
    head = p;
}

void tampil(){
    node *baca;

    baca = head;
    puts("");
    puts("Data yang ada dalam SLL");
    while(baca != NULL){
        printf("%d\n", baca->data);
        baca = baca->next;
    }
}

```

Output :

```

Single Linked List - Insert Awal
Nilai yang akan disimpan : 8
Lagi (y/t)? y
Nilai yang akan disimpan : 3
Lagi (y/t)? y
Nilai yang akan disimpan : 2
Lagi (y/t)? t

Data yang ada dalam SLL
2
3
8

```

2. Insert Akhir

Kode :

```
#include<stdio.h>
#include<stdlib.h>

typedef struct simpul node;
struct simpul{
    int data;
    node *next;
};

node *head = NULL, *p;

void alokasi();
void akhir();
void tampil();

int main(){
    char jawab;

    puts("Single Linked List - Insert Akhir");
    do{
        fflush(stdin);
        alokasi();
        akhir();
        fflush(stdin);
        printf("Lagi (y/t)? ");
        jawab = getchar();
    }while(jawab == 'y' || jawab == 'Y');
    tampil();
    return 0;
}

void alokasi(){
    int bilangan;

    printf("Nilai yang akan disimpan : ");
    scanf("%d",&bilangan);
    p = (node *)malloc(sizeof(node));
    if(p == NULL){
        puts("Gagal memesam alamat");
        exit(0);
    }
}
```

```

    else{
        p->data = bilangan;
        p->next = NULL;
    }
}

void akhir(){
    node *tail;

    if(head == NULL)
        head = p;
    else{
        tail = head;
        while(tail->next != NULL)
            tail = tail->next;
        tail->next = p;
        tail = tail->next;
    }
}

void tampil(){
    node *baca;

    baca = head;
    puts("");
    puts("Data yang ada dalam SLL");
    while(baca != NULL){
        printf("%d\n", baca->data);
        baca = baca->next;
    }
}

```

Output :

```

Single Linked List - Insert Akhir
Nilai yang akan disimpan : 2
Lagi (y/t)? y
Nilai yang akan disimpan : 4
Lagi (y/t)? y
Nilai yang akan disimpan : 3
Lagi (y/t)? t

Data yang ada dalam SLL
2
4
3

```

3. Insert After

Kode :

```
#include<stdio.h>
#include<stdlib.h>
```

```
typedef struct simpul node;
struct simpul{
    int data;
    node *next;
};
```

```
node *head = NULL, *p;
```

```
void alokasi();
void akhir();
void tampil();
void setelah();
```

```
int main(){
    char jawab;

    puts("Single Linked List - Insert Akhir");
    do{
        fflush(stdin);
        alokasi();
        akhir();
        fflush(stdin);
        printf("Lagi (y/t)? ");
        jawab = getchar();
    }while(jawab == 'y' || jawab == 'Y');
    tampil();
    alokasi();
    setelah();
    tampil();
    return 0;
}
```

```
void setelah(){
    int key;
    node *after;

    printf("Disisipkan setelah node : ");
    scanf("%d",&key);
```

```

after = head;
while(after->data != key){
    if(after->next == NULL){
        printf("Data Tidak Ditemukan");
        exit(0);
    }
    else{
        after = after->next;
    }
}
p->next = after->next;
after->next = p;
}

```

```

void alokasi(){
    int bilangan;

    printf("Nilai yang akan disimpan : ");
    scanf("%d",&bilangan);
    p = (node *)malloc(sizeof(node));
    if(p == NULL){
        puts("Gagal memesam alamat");
        exit(0);
    }
    else{
        p->data = bilangan;
        p->next = NULL;
    }
}

```

```

void akhir(){
    node *tail;

    if(head == NULL)
        head = p;
    else{
        tail = head;
        while(tail->next != NULL)
            tail = tail->next;
        tail->next = p;
        tail = tail->next;
    }
}

```

```

void tampil(){
    node *baca;

    baca = head;
    puts("");
    puts("Data yang ada dalam SLL");
    while(baca != NULL){
        printf("%d\n", baca->data);
        baca = baca->next;
    }
}

```

Output :

```

Single Linked List - Insert Akhir
Nilai yang akan disimpan : 2
Lagi (y/t)? y
Nilai yang akan disimpan : 4
Lagi (y/t)? y
Nilai yang akan disimpan : 3
Lagi (y/t)? t

Data yang ada dalam SLL
2
4
3
Nilai yang akan disimpan : 1
Disisipkan setelah node : 4

Data yang ada dalam SLL
2
4
1
3

```

4. Insert Before

Kode :

```

#include<stdio.h>
#include<stdlib.h>

```

```

typedef struct simpul node;
struct simpul{
    int data;
    node *next;
};

```



```

node *head = NULL, *p;

void alokasi();
void akhir();
void tampil();
void sebelum();

int main(){
    char jawab;

    puts("Single Linked List - Insert Akhir");
    do{
        fflush(stdin);
        alokasi();
        akhir();
        fflush(stdin);
        printf("Lagi (y/t)? ");
        jawab = getchar();
    }while(jawab == 'y' || jawab == 'Y');
    tampil();
    alokasi();
    sebelum();
    tampil();
    return 0;
}

void sebelum(){
    int key;
    node *bef, *pbef;

    printf("Disisipkan sebelum node : ");
    scanf("%d",&key);
    bef = head;
    while(bef->data != key){
        if(bef->next == NULL){
            printf("Data Tidak Ditemukan");
            exit(0);
        }
        else{
            pbef = bef;
            bef = bef->next;
        }
    }
}

```

```

    }
    if(bef == head){
        p->next = bef;
        head = p;
    }
    else{
        p->next = bef;
        pbef->next = p;
    }
}

```

```

void alokasi(){
    int bilangan;

    printf("Nilai yang akan disimpan : ");
    scanf("%d",&bilangan);
    p = (node *)malloc(sizeof(node));
    if(p == NULL){
        puts("Gagal memesam alamat");
        exit(0);
    }
    else{
        p->data = bilangan;
        p->next = NULL;
    }
}

```

```

void akhir(){
    node *tail;

    if(head == NULL)
        head = p;
    else{
        tail = head;
        while(tail->next != NULL)
            tail = tail->next;
        tail->next = p;
        tail = tail->next;
    }
}

```

```

void tampil(){
    node *baca;

```

```

    baca = head;
    puts("");
    puts("Data yang ada dalam SLL");
    while(baca != NULL){
        printf("%d\n", baca->data);
        baca = baca->next;
    }
}

```

Output :

```

Single Linked List - Insert Akhir
Nilai yang akan disimpan : 1
Lagi (y/t)? y
Nilai yang akan disimpan : 2
Lagi (y/t)? y
Nilai yang akan disimpan : 4
Lagi (y/t)? t

Data yang ada dalam SLL
1
2
4
Nilai yang akan disimpan : 3
Disisipkan sebelum node : 4

Data yang ada dalam SLL
1
2
3
4

```

5. Menu insert

Kode :

```

#include<stdio.h>
#include<stdlib.h>

```

```

typedef struct simpul node;
struct simpul{
    int data;
    node *next;
};

```

```

node *head = NULL, *p;

```

```

int menu(int);

```

```

void alokasi();
void awal();
void akhir();
void setelah();
void sebelum();
void tampil();

int i;
int main(){
    int jawab;

    puts("Single Linked List - Insert Menu");
    while(jawab != 5){
        jawab = menu(jawab);
        if(jawab > 0 && jawab <= 5){
            alokasi();
            switch(jawab){
                case 1 :
                    awal();
                    break;
                case 2 :
                    akhir();
                    break;
                case 3 :
                    setelah();
                    break;
                case 4 :
                    sebelum();
                    break;
            }
            tampil();
            puts("");
            i++;
        }
        else if(jawab == 6)
            puts("SLL masih kosong, tidak bisa insert after\n");
        else if(jawab == 7)
            puts("SLL masih kosong, tidak bisa insert before\n");
        else
            puts("Pilih Yang Benar SHEESSHH");
        puts("");
    }
    return 0;
}

```

```
}
```

```
int menu(int pilih){
    puts("Menu Insert");
    puts("1. Awal");
    puts("2. Akhir");
    puts("3. After");
    puts("4. Before");
    puts("5. Keluar");
    printf("Masukkan Pilihan Anda : ");
    scanf("%d",&pilih);
    if(i == 0 && pilih == 3)
        return 6;
    else if(i == 0 && pilih == 4)
        return 7;
    else{
        switch(pilih){
            case 1:
                return 1;
                break;
            case 2:
                return 2;
                break;
            case 3:
                return 3;
                break;
            case 4:
                return 4;
                break;
            case 5:
                exit(0);
                break;
            default:
                return 8;
        }
    }
}

void alokasi(){
    int bilangan;

    printf("Nilai yang akan disimpan : ");
```

```

scanf("%d",&bilangan);
p = (node *)malloc(sizeof(node));
if(p == NULL){
    puts("Gagal memesam alamat");
    exit(0);
}
else{
    p->data = bilangan;
    p->next = NULL;
}
}

void awal(){
    if(head != NULL)
        p->next = head;
    head = p;
}

void akhir(){
    node *tail;

    if(head == NULL)
        head = p;
    else{
        tail = head;
        while(tail->next != NULL)
            tail = tail->next;
        tail->next = p;
        tail = tail->next;
    }
}

void setelah(){
    int key;
    node *after;

    printf("Disisipkan setelah data berapa ? ");
    scanf("%d",&key);
    after = head;
    while(after->data != key){
        if(after->next == NULL){
            printf("Data Tidak Ditemukan");
            exit(0);

```

```

    }
    else{
        after = after->next;
    }
}
p->next = after->next;
after->next = p;
}

```

```

void sebelum(){
    int key;
    node *bef, *pbef;

    printf("Disisipkan sebelum data berapa ? ");
    scanf("%d",&key);
    bef = head;
    while(bef->data != key){
        if(bef->next == NULL){
            printf("Data Tidak Ditemukan");
            exit(0);
        }
        else{
            pbef = bef;
            bef = bef->next;
        }
    }
    if(bef == head){
        p->next = bef;
        head = p;
    }
    else{
        p->next = bef;
        pbef->next = p;
    }
}

```

```

void tampil(){
    node *baca;

    baca = head;
    puts("");
    puts("Data yang ada dalam SLL");
    while(baca != NULL){

```

```

        printf("%d\n", baca->data);
        baca = baca->next;
    }
}

```

Output :

```

Single Linked List - Insert Menu
Menu Insert
1. Awal
2. Akhir
3. After
4. Before
5. Keluar
Masukkan Pilihan Anda : 3
SLL masih kosong, tidak bisa insert after

Menu Insert
1. Awal
2. Akhir
3. After
4. Before
5. Keluar
Masukkan Pilihan Anda : 4
SLL masih kosong, tidak bisa insert before

Menu Insert
1. Awal
2. Akhir
3. After
4. Before
5. Keluar
Masukkan Pilihan Anda : 1
Nilai yang akan disimpan : 5

Data yang ada dalam SLL
5

```

```

Menu Insert
1. Awal
2. Akhir
3. After
4. Before
5. Keluar
Masukkan Pilihan Anda : 2
Nilai yang akan disimpan : 9

Data yang ada dalam SLL
5
9

Menu Insert
1. Awal
2. Akhir
3. After
4. Before
5. Keluar
Masukkan Pilihan Anda : 3
Nilai yang akan disimpan : 7
Disisipkan setelah data berapa ? 5

Data yang ada dalam SLL
5
7
9

Menu Insert
1. Awal
2. Akhir
3. After
4. Before
5. Keluar
Masukkan Pilihan Anda : 4
Nilai yang akan disimpan : 8
Disisipkan sebelum data berapa ? 9

```

```

Data yang ada dalam SLL
5
7
8
9

```

```

Menu Insert
1. Awal
2. Akhir
3. After
4. Before
5. Keluar
Masukkan Pilihan Anda : 5

```

Analisis :

Dari Praktikum ini, kita mempelajari bagaimana single linked list itu. Mulai dari linked list di awal, linked list di akhir, linked list after, dan linked list before. Linked list di awal yaitu apabila kita membuat data baru maka data itu akan diletakkan di awal. Linked list di akhir yaitu apabila kita membuat data baru maka data itu akan diletakkan di akhir. Linked list after yaitu apabila kita ingin menyisipkan suatu data, data itu akan kita sisipkan setelah data yang kita inputkan. Linked list before yaitu apabila kita ingin menyisipkan suatu data, data itu akan kita sisipkan sebelum data yang kita inputkan. Untuk linked list awal, pembuatannya cukup sederhana. Apabila linked list di akhir kita masih memerlukan Bantuan dari sebuah pointer to node tail. Fungsi dari tail tersebut kurang lebih untuk mencari data mana yang berada di akhir. Apabila linked list after kita perlu menginputkan key tetapi tidak perlu menggunakan pointer Bantuan. Beda halnya dengan linked list before yang masih memerlukan 2 pointer Bantuan yaitu bef dan pbef. Untuk menampilkan data-data yang tadi kita memerlukan Bantuan dari sebuah pointer Bantuan yaitu baca. Fungsinya untuk melakukan printing sebelum value dari baca bernilai NULL. Ada 3 algoritma yang harus diingat untuk linked list, yaitu siapkan alamat dan datanya, pilih ingin diletakkan di mana, setelah itu hubungkan data-data tersebut agar saling berkaitan / node tidak terputus.