

Laporan Resmi
Praktikum Algoritma dan Struktur Data
Single Linked List Insert (Struct)



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>
#include<string.h>
```

```
typedef struct{
    int id, nilai;
    char nama[25];
}emp;
```

```
typedef struct simpul node;
struct simpul{
    emp employees;
    node *next;
};
node *head=NULL, *p, *tail;
```

```
void input(emp *);
void tampil();
void awal();
void alokasi();
```

```
int main(){
    char jwb;

    puts("Linked List untuk aplikasi INSERT DI AWAL");
    puts("Membentuk linked list dengan Insert di awal");
    do{
        fflush(stdin);
        alokasi();
        awal();
        fflush(stdin);
        printf("Ada data lagi (y/t)? ");
        jwb=getchar();
        puts("");
    }while(jwb=='y');
    tampil();
    return 0;
}
```

```
void alokasi(){
```

```

emp pegawai;

input(&pegawai);
p=(node *)malloc(sizeof(node));
if(p == NULL)
    exit(0);
else{
    p->employees.id = pegawai.id;
    strcpy(p->employees.nama, pegawai.nama);
    p->employees.nilai = pegawai.nilai;
    p->next=NULL;
}
}

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

void tampil(){
    node *baca;

    baca = head;
    puts("Data yang telah diinputkan :");
    puts("No\tNama\tNilai");
    while(baca != NULL){
        printf("%d\t",baca->employees.id);
        printf("%s\t", baca->employees.nama);
        printf("%d\t", baca->employees.nilai);
        puts("");
        baca=baca->next;
    }
}

void input(emp *pegawaii){
    int x, y;

    printf("No\t : ");
    scanf("%d",&pegawaii->id);
    fflush(stdin);

```

```

printf("Nama\t : ");
gets(pegawai->nama);
fflush(stdin);
printf("Nilai\t : ");
scanf("%d",&pegawai->nilai);
}

```

Output :

```

Linked List untuk aplikasi INSERT DI AWAL
Membentuk linked list dengan Insert di awal
No      : 3
Nama    : Ikmal
Nilai   : 90
Ada data lagi (y/t)? y

No      : 2
Nama    : Ling
Nilai   : 92
Ada data lagi (y/t)? y

No      : 1
Nama    : Ali
Nilai   : 87
Ada data lagi (y/t)? t

Data yang telah diinputkan :
No      Nama      Nilai
1       Ali       87
2       Ling      92
3       Ikmal     90

```

2. Insert Akhir

Kode :

```

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

```

```

typedef struct{
    int id, nilai;
    char nama[25];
}emp;

```

```

typedef struct simpul node;
struct simpul{
    emp employees;
    node *next;
};
node *head=NULL, *p, *tail;

```

```

void input(emp *);
void tampil();
void akhir();

```

```
void alokasi();
```

```
int main(){
    char jwb;

    puts("Linked List untuk aplikasi INSERT DI AKHIR");
    puts("Membentuk linked list dengan Insert di akhir");
    do{
        fflush(stdin);
        alokasi();
        akhir();
        fflush(stdin);
        printf("Ada data lagi (y/t)? ");
        jwb=getchar();
        puts("");
    }while(jwb=='y');
    tampil();
    return 0;
}
```

```
void alokasi(){
    emp pegawai;

    input(&pegawai);
    p=(node *)malloc(sizeof(node));
    if(p == NULL)
        exit(0);
    else{
        p->employees.id = pegawai.id;
        strcpy(p->employees.nama, pegawai.nama);
        p->employees.nilai = pegawai.nilai;
        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("Data yang telah diinputkan :");
    puts("No\tNama\tNilai");
    while(baca != NULL){
        printf("%d\t",baca->employees.id);
        printf("%s\t", baca->employees.nama);
        printf("%d\t", baca->employees.nilai);
        puts("");
        baca=baca->next;
    }

}

void input(emp *pegawaii){
    int x, y;

    printf("No\t : ");
    scanf("%d",&pegawaii->id);
    fflush(stdin);
    printf("Nama\t : ");
    gets(pegawaii->nama);
    fflush(stdin);
    printf("Nilai\t : ");
    scanf("%d",&pegawaii->nilai);
}

```

Output :

```
Linked List untuk aplikasi INSERT DI AKHIR
Membentuk linked list dengan Insert di akhir
No      : 1
Nama    : Ling
Nilai   : 89
Ada data lagi (y/t)? y

No      : 2
Nama    : Ikmal
Nilai   : 86
Ada data lagi (y/t)? y

No      : 3
Nama    : Ney
Nilai   : 79
Ada data lagi (y/t)? t

Data yang telah diinputkan :
No      Nama      Nilai
1       Ling      89
2       Ikmal     86
3       Ney       79
```

3. Insert After

Kode :

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

```
typedef struct{
    int id, nilai;
    char nama[25];
}emp;
```

```
typedef struct simpul node;
struct simpul{
    emp employees;
    node *next;
};
node *head=NULL, *p, *tail;
```

```
void input(emp *);
void tampil();
void akhir();
void alokasi();
void setelah();
```

```

int main(){
    char jwb;

    puts("Linked List untuk aplikasi INSERT AFTER");
    puts("Membentuk linked list dengan Insert di akhir");
    do{
        fflush(stdin);
        alokasi();
        akhir();
        fflush(stdin);
        printf("Ada data lagi (y/t)? ");
        jwb=getchar();
        puts("");
    }while(jwb=='y');
    fflush(stdin);
    printf("\nIngin memasukkan data setelah ttt (y/t) ? ");
    jwb = getchar();
    fflush(stdin);
    if(jwb == 'y'){
        alokasi();
        setelah();
        puts("");
        tampil();
    }
    return 0;
}

```

```

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

    printf("Data tsb disisipkan setelah nomor berapa : ");
    scanf("%d",&key);
    after = head;
    while(after->employees.id != key){
        if(after->next == NULL){
            printf("Data Tidak Ditemukan");
            exit(0);
        }
        else{
            after = after->next;
        }
    }
}

```



```

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

```

```

void alokasi(){
    emp pegawai;

    input(&pegawai);
    p=(node *)malloc(sizeof(node));
    if(p == NULL)
        exit(0);
    else{
        p->employees.id = pegawai.id;
        strcpy(p->employees.nama, pegawai.nama);
        p->employees.nilai = pegawai.nilai;
        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("Data yang telah diinputkan :");
    puts("No\tNama\tNilai");
    while(baca != NULL){
        printf("%d\t",baca->employees.id);
    }
}

```

```

        printf("%s\t", baca->employees.nama);
        printf("%d\t", baca->employees.nilai);
        puts("");
        baca=baca->next;
    }

}

```

```

void input(emp *pegawaii){
    int x, y;

    printf("No\t : ");
    scanf("%d",&pegawaii->id);
    fflush(stdin);
    printf("Nama\t : ");
    gets(pegawaii->nama);
    fflush(stdin);
    printf("Nilai\t : ");
    scanf("%d",&pegawaii->nilai);
}

```

Output :

Linked List untuk aplikasi INSERT AFTER
Membentuk linked list dengan Insert di akhir

```

No      : 1
Nama    : Ikmal
Nilai   : 90
Ada data lagi (y/t)? y

```

```

No      : 3
Nama    : Ney
Nilai   : 89
Ada data lagi (y/t)? t

```

Ingin memasukkan data setelah ttt (y/t) ? y

```

No      : 2
Nama    : Ling
Nilai   : 98
Data tsb disisipkan setelah nomor berapa : 1

```

Data yang telah diinputkan :

No	Nama	Nilai
1	Ikmal	90
2	Ling	98
3	Ney	89

4. Insert Before

Kode :

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

```
typedef struct{
    int id, nilai;
    char nama[25];
}emp;
```

```
typedef struct simpul node;
struct simpul{
    emp employees;
    node *next;
};
node *head=NULL, *p, *tail;
```

```
void input(emp *);
void tampil();
void awal();
void alokasi();
void sebelum();
```

```
int main(){
    char jwb;

    puts("Linked List untuk aplikasi INSERT BEFORE");
    puts("Membentuk linked list dengan Insert di awal");
    do{
        fflush(stdin);
        alokasi();
        awal();
        fflush(stdin);
        printf("Ada data lagi (y/t)? ");
        jwb=getchar();
        puts("");
    }while(jwb=='y');
    fflush(stdin);
    printf("\nIngin memasukkan data setelah ttt (y/t) ? ");
    jwb = getchar();
    fflush(stdin);
```

```

    if(jwb == 'y'){
        alokasi();
        sebelum();
        puts("");
        tampil();
    }
    return 0;
}

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

    printf("Data tsb disisipkan sebelum nomor berapa : ");
    scanf("%d",&key);
    bef = head;
    while(bef->employees.id != 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(){
    emp pegawai;

    input(&pegawai);
    p=(node *)malloc(sizeof(node));
    if(p == NULL)
        exit(0);
}

```

```

    else{
        p->employees.id = pegawai.id;
        strcpy(p->employees.nama, pegawai.nama);
        p->employees.nilai = pegawai.nilai;
        p->next=NULL;
    }
}

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

void tampil(){
    node *baca;

    baca = head;
    puts("Data yang telah diinputkan :");
    puts("No\tNama\tNilai");
    while(baca != NULL){
        printf("%d\t",baca->employees.id);
        printf("%s\t", baca->employees.nama);
        printf("%d\t", baca->employees.nilai);
        puts("");
        baca=baca->next;
    }
}

void input(emp *pegawaii){
    int x, y;

    printf("No\t : ");
    scanf("%d",&pegawaii->id);
    fflush(stdin);
    printf("Nama\t : ");
    gets(pegawaii->nama);
    fflush(stdin);
    printf("Nilai\t : ");
    scanf("%d",&pegawaii->nilai);
}

```

Output :

```
Linked List untuk aplikasi INSERT BEFORE
Membentuk linked list dengan Insert di awal
No      : 5
Nama    : Ikmal
Nilai   : 89
Ada data lagi (y/t)? y

No      : 3
Nama    : Ling
Nilai   : 78
Ada data lagi (y/t)? y

No      : 1
Nama    : Low
Nilai   : 86
Ada data lagi (y/t)? t

Ingin memasukkan data setelah ttt (y/t) ? y
No      : 4
Nama    : Law
Nilai   : 85
Data tsb disisipkan sebelum nomor berapa : 5

Data yang telah diinputkan :
No      Nama      Nilai
1       Low       86
3       Ling      78
4       Law       85
5       Ikmal     89
```

5. Menu Insert

Kode :

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

```
typedef struct{
    int id, nilai;
    char nama[25];
}emp;
```

```
typedef struct simpul node;
struct simpul{
    emp employees;
    node *next;
};
node *head=NULL, *p, *tail;
```

```
void input(emp *);
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(){
    emp pegawai;

    input(&pegawai);

```



```

p=(node *)malloc(sizeof(node));
if(p == NULL)
    exit(0);
else{
    p->employees.id = pegawai.id;
    strcpy(p->employees.nama, pegawai.nama);
    p->employees.nilai = pegawai.nilai;
    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("Data tsb disisipkan setelah nomor berapa : ");
    scanf("%d",&key);
    after = head;
    while(after->employees.id != 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("Data tsb disisipkan sebelum nomor berapa : ");
    scanf("%d",&key);
    bef = head;
    while(bef->employees.id != 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("\nData yang telah diinputkan :");
    puts("No\tNama\tNilai");
}

```

```

while(baca != NULL){
    printf("%d\t", baca->employees.id);
    printf("%s\t", baca->employees.nama);
    printf("%d\t", baca->employees.nilai);
    puts("");
    baca=baca->next;
}

}

```

```

void input(emp *pegawai){
    int x, y;

    printf("\nNo\t : ");
    scanf("%d",&pegawai->id);
    fflush(stdin);
    printf("Nama\t : ");
    gets(pegawai->nama);
    fflush(stdin);
    printf("Nilai\t : ");
    scanf("%d",&pegawai->nilai);
}

```

Output :

Single Linked List - Insert Menu

Menu Insert

1. Awal
2. Akhir
3. After
4. Before
5. Keluar

Masukkan Pilihan Anda : 1

No : 2
 Nama : Ikmal
 Nilai : 89

Data yang telah diinputkan :

No	Nama	Nilai
2	Ikmal	89

Menu Insert

1. Awal
2. Akhir
3. After
4. Before
5. Keluar

Masukkan Pilihan Anda : 2

No : 4
 Nama : Ling
 Nilai : 90

Data yang telah diinputkan :

No	Nama	Nilai
2	Ikmal	89
4	Ling	90

Menu Insert

1. Awal
2. Akhir
3. After
4. Before
5. Keluar

Masukkan Pilihan Anda : 3

No : 5
 Nama : Law
 Nilai : 87

Data tsb disisipkan setelah nomor berapa : 4

Data yang telah diinputkan :

No	Nama	Nilai
2	Ikmal	89
4	Ling	90
5	Law	87

Menu Insert

1. Awal
2. Akhir
3. After
4. Before
5. Keluar

Masukkan Pilihan Anda : 4

No : 3
 Nama : Low
 Nilai : 88

Data tsb disisipkan sebelum nomor berapa : 4

Data yang telah diinputkan :

No	Nama	Nilai
2	Ikmal	89
3	Low	88
4	Ling	90
5	Law	87

Menu Insert

1. Awal
2. Akhir
3. After
4. Before
5. Keluar

Masukkan Pilihan Anda : 5

Kesimpulan :

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. Praktikum ini kurang lebih sama seperti sama sebelumnya. Bedanya hanya berada pada tipe data nya saja. Jadi tipe data node memiliki tipe data struct yang isinya merupakan nomor, nama, dan nilai.