

PRAKTIKUM ALGORITMA & STRUKTUR DATA

“PEMROGRAMAN MENU SLL-INT”



DOSEN :

Umi Sa'adah S.Kom, M.Kom (197404162000032003)

PENYUSUN :

Mohammad Ihsan Septiano Firdaus (3125600041)

PROGRAM STUDI SARJANA TERAPAN

TEKNIK INFORMATIKA

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

Program

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

//definisi dan deklarasi
void cls(), cari(), awal(), akhir(), after(), before(), tampil(),
insert(), del(), del_awal(), del_akhir(), del_choice();
typedef struct node
{
    int data;
    struct node *next; } Node;
Node *head = NULL, *p;

//alokasi mem
void cari() {
    p = (Node *) malloc(sizeof(Node));
    if(p == NULL ) {
        printf("gagal alokasi\n");
        return; }
    p->next = NULL;
    printf("data = "); scanf("%d", &p->data); }

//operasi insert
void insert () {
    char y; int ch;
    do {
        printf("OPERASI INSERT\n");
        printf("1. Insert Awal\n2. Insert Akhir\n3. Insert After\n4.
Insert Before\nPilih = ");
        scanf("%d", &ch);
        switch (ch) {
            case 1: awal(); break;
            case 2: akhir(); break;
            case 3: after(); break;
            case 4: before(); break;
            default: printf("Pilihan salah\n"); }
        printf("Insert lagi? "); scanf(" %c", &y);
    } while (y == 'y' || y == 'Y'); }
void awal () {
    cari();
    if (head != NULL)
        p->next = head;
    head = p; }
void akhir () {
    cari();
```

```

    if (head != NULL) {
        Node *tail = head;
        while (tail->next != NULL)
            tail = tail->next;
        tail->next = p; }
    else head = p; }
void after () {
    int key;
    printf("Cari data = "); scanf("%d", &key);
    Node *baca = head;
    while (baca != NULL && baca->data != key)
        baca = baca->next;
    if (baca != NULL) {
        cari();
        p->next = baca->next;
        baca->next = p; }
    else printf("Data tidak ditemukan\n"); }
void before () {
    int key;
    printf("Cari data = "); scanf("%d", &key);
    Node *baca = head, *prev = NULL;
    while (baca != NULL && baca->data != key) {
        prev = baca;
        baca = baca->next; }
    if (baca != NULL) {
        cari();
        p->next = baca;
        if (prev != NULL)
            prev->next = p;
        else head = p; }
    else printf("Data tidak ditemukan\n"); }

//operasi delete
void del () {
    if (head == NULL) {
        printf("Data Kosong\n");
        return;
    }
    char y; int ch;
    do {
        printf("OPERASI DELETE\n");
        printf("1. Delete Awal\n2. Delete Akhir\n3. Delete Tertentu\nPilih = ");
        scanf("%d", &ch);
        switch (ch) {
            case 1: del_awal(); break;
            case 2: del_akhir(); break;
            case 3: del_choice(); break;

```

```

        default: printf("Pilihan salah\n"); }
        printf("Delete Lagi? "); scanf(" %c", &y);
    } while (y == 'y' || y == 'Y'); }
void del_awal () {
    Node *hapus = head;
    head = head->next;
    free(hapus); }
void del_akhir () {
    Node *hapus = head, *phapus = NULL;
    if (head->next == NULL) {
        head = NULL;
        free(hapus); }
    else {
        while (hapus->next != NULL) {
            phapus = hapus;
            hapus = hapus->next; }
        phapus->next = NULL;
        free(hapus); } }
void del_choice () {
    int key; Node *hapus = head, *phapus;
    printf("Cari data = "); scanf("%d", &key);
    if (head->data == key) del_awal();
    else {
        hapus=head;
        while(hapus->data!=key) {
            if (hapus->next == NULL ) {
                printf("data tidak ditemukan\n"); return; }
            phapus = hapus;
            hapus = hapus->next; }
        phapus->next = hapus->next;
        free(hapus);
        hapus =NULL; }}

void tampil() {
    if (head != NULL) {
        Node *baca = head;
        while (baca != NULL) {
            printf("%d ", baca->data);
            baca = baca->next; }
        printf("\n"); }
    else printf("Data kosong\n"); }

int main () {
    printf("SINGLE LINKED LIST\n\n");
    char y; int ch;
    do {
        printf("1. Insert\n2. Delete\n3. Tampilkan\nPilih = ");

```

```
scanf("%d", &ch);
switch (ch) {
    case 1: insert(); break;
    case 2: del(); break;
    case 3: tampil(); break;
    default: printf("Pilihan salah\n"); }
fflush(stdin);
printf("Operasi lain? "); scanf(" %c", &y);
} while (y == 'y' || y == 'Y'); }
```

```
OPERASI INSERT
1. Insert Awal
2. Insert Akhir
3. Insert After
4. Insert Before
Pilih = 2
data = 6
Insert lagi? y
OPERASI INSERT
1. Insert Awal
2. Insert Akhir
3. Insert After
4. Insert Before
Pilih = 1
data = 7
Insert lagi? y
OPERASI INSERT
1. Insert Awal
2. Insert Akhir
3. Insert After
4. Insert Before
Pilih = 4
Cari data = 6
data = 4
Insert lagi? t
Operasi lain? y
1. Insert
2. Delete
3. Tampilkan
Pilih = 3
7 1 4 6
Operasi lain? |

/ 1 4 6
Operasi lain? y
1. Insert
2. Delete
3. Tampilkan
Pilih = 2
OPERASI DELETE
1. Delete Awal
2. Delete Akhir
3. Delete Tertentu
Pilih = 1
Delete Lagi? y
OPERASI DELETE
1. Delete Awal
2. Delete Akhir
3. Delete Tertentu
Pilih = 3
Cari data = 4
Delete Lagi? t
Operasi lain? y
1. Insert
2. Delete
3. Tampilkan
Pilih = 3
1 6
```

Kesimpulan

Program menu insert SLL yang telah dibuat memiliki 3 fungsi hero yang berada pada fungsi utama, 3 fungsi delete, dan 4 fungsi insert. Fungsi utama, yaitu int main(), sesuai namanya berfungsi sebagai fungsi utama yang menjadi wadah saat program dijalankan, yang juga memanggil fungsi-fungsi lain. Untuk pemanggilan fungsi lain, digunakan switch case dari input user untuk memilih fungsi hero mana yang akan dipanggil, nantinya, fungsi hero insert akan memanggil 4 fungsi yang memasukkan data, fungsi hero delete akan memanggil 3 fungsi membebaskan data, dan fungsi hero tampil akan menampilkan datanya.

Untuk program yang ini, memasukkan data dilakukan pada saat fungsi cari() dipanggil, yang akan mencari alamat kosong dan inputkan data ke dalamnya. Di program ini fungsi input dan delete disatukan menjadi satu codingan.

PRAKTIKUM ALGORITMA & STRUKTUR DATA

“PEMROGRAMAN MENU SLL-STRUCT”



DOSEN :

Umi Sa'adah S.Kom, M.Kom (197404162000032003)

PENYUSUN :

Mohammad Ihsan Septiano Firdaus (3125600041)

PROGRAM STUDI SARJANA TERAPAN

TEKNIK INFORMATIKA

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

Program

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

//definisi dan deklarasi
void cls(), cari(), awal(), akhir(), after(), before(), tampil(),
insert(), del(), del_awal(), del_akhir(), del_choice();
typedef struct node
{
    int no;
    char nama[20];
    int nilai;
    struct node *next; } Node;
Node *head = NULL, *p;

//alokasi mem
void cari() {
    p = (Node *) malloc(sizeof(Node));
    p->next = NULL;

    printf("No \t= "); scanf("%d", &p->no);
    while (getchar() != '\n');

    printf("Nama \t= ");
    if (fgets(p->nama, sizeof(p->nama), stdin) != NULL) {
        p->nama[strcspn(p->nama, "\n")] = 0;
    }
    printf("Nilai \t= "); scanf("%d", &p->nilai); }

//operasi insert
void insert () {
    char y; int ch;
    do {
        printf("OPERASI INSERT\n");
        printf("1. Insert Awal\n2. Insert Akhir\n3. Insert After\n4.
Insert Before\nPilih = ");
        scanf("%d", &ch);
        switch (ch) {
            case 1: awal(); break;
            case 2: akhir(); break;
            case 3: after(); break;
            case 4: before(); break;
            default: printf("Pilihan salah\n"); }
        printf("Insert lagi? "); scanf(" %c", &y);
    } while (y == 'y' || y == 'Y'); }
```

```

void awal () {
    cari();
    if (head != NULL)
        p->next = head;
    head = p; }
void akhir () {
    cari();
    if (head != NULL) {
        Node *tail = head;
        while (tail->next != NULL)
            tail = tail->next;
        tail->next = p; }
    else head = p; }
void after () {
    int key;
    printf("Cari data = "); scanf("%d", &key);
    Node *baca = head;
    while (baca != NULL && baca->no != key)
        baca = baca->next;
    if (baca != NULL) {
        cari();
        p->next = baca->next;
        baca->next = p; }
    else printf("Data tidak ditemukan\n"); }
void before () {
    int key;
    printf("Cari data = "); scanf("%d", &key);
    Node *baca = head, *prev = NULL;
    while (baca != NULL && baca->no != key) {
        prev = baca;
        baca = baca->next; }
    if (baca != NULL) {
        cari();
        p->next = baca;
        if (prev != NULL)
            prev->next = p;
        else head = p; }
    else printf("Data tidak ditemukan\n"); }

//operasi delete
void del () {
    if (head == NULL) {
        printf("Data Kosong\n");
        return;
    }
    char y; int ch;
    do {
        printf("OPERASI DELETE\n");

```

```

        printf("1. Delete Awal\n2. Delete Akhir\n3. Delete Tertentu\nPilih = ");
        scanf("%d", &ch);
        switch (ch) {
            case 1: del_awal(); break;
            case 2: del_akhir(); break;
            case 3: del_choice(); break;
            default: printf("Pilihan salah\n"); }
        printf("Delete Lagi? "); scanf(" %c", &y);
    } while (y == 'y' || y == 'Y'); }
void del_awal () {
    Node *hapus = head;
    head = head->next;
    free(hapus); }
void del_akhir () {
    Node *hapus = head, *phapus = NULL;
    if (head->next == NULL) {
        head = NULL;
        free(hapus); }
    else {
        while (hapus->next != NULL) {
            phapus = hapus;
            hapus = hapus->next; }
        phapus->next = NULL;
        free(hapus); } }
void del_choice () {
    int key; Node *hapus = head, *phapus;
    printf("Cari data = "); scanf("%d", &key);
    if (head->no == key) del_awal();
    else {
        hapus=head;
        while(hapus->no!=key) {
            if (hapus->next == NULL ) {
                printf("data tidak ditemukan\n"); return; }
            phapus = hapus;
            hapus = hapus->next; }
        phapus->next = hapus->next;
        free(hapus);
        hapus =NULL; }}

void tampil() {
    if (head != NULL) {
        Node *baca = head;
        while (baca != NULL) {
            printf("No: %d, Nama: %s, Nilai: %d\n", baca->no, baca->nama, baca->nilai);
            baca = baca->next; }
    }
}

```

```

        printf("\n"); }
    else printf("Data kosong\n"); }

int main () {
    printf("SINGLE LINKED LIST\n\n");
    char y; int ch;
    do {
        printf("1. Insert\n2. Delete\n3. Tampilkan\nPilih = ");
        scanf("%d", &ch);
        switch (ch) {
            case 1: insert(); break;
            case 2: del(); break;
            case 3: tampil(); break;
            default: printf("Pilihan salah\n"); }
        fflush(stdin);
        printf("Operasi lain? "); scanf(" %c", &y);
    } while (y == 'y' || y == 'Y'); }

```

SINGLE LINKED LIST

```

1. Insert
2. Delete
3. Tampilkan
Pilih = 1
OPERASI INSERT
1. Insert Awal
2. Insert Akhir
3. Insert After
4. Insert Before
Pilih = 1
No = 13
Nama = Ihsan Sept
Nilai = 99
Insert lagi? y
OPERASI INSERT
1. Insert Awal
2. Insert Akhir
3. Insert After
4. Insert Before
Pilih = 2
No = 69
Nama = Hdy Cinta
Nilai = 92
Insert lagi? y
OPERASI INSERT
1. Insert Awal
2. Insert Akhir
3. Insert After
4. Insert Before
Pilih = 4
Cari data = 69
No = 44
Nama = Putri Syailah
Nilai = 94
Insert lagi? t
Operasi lain? y
1. Insert
2. Delete
3. Tampilkan
Pilih = 3
No: 13, Nama: Ihsan Sept, Nilai: 99
No: 44, Nama: Putri Syailah, Nilai: 94
No: 69, Nama: Hdy Cinta, Nilai: 92

Operasi lain? y
1. Insert
2. Delete
3. Tampilkan
Pilih = 2
OPERASI DELETE
1. Delete Awal
2. Delete Akhir
3. Delete Tertentu
Pilih = 3
Cari data = 69
Delete Lagi? y
OPERASI DELETE
1. Delete Awal
2. Delete Akhir
3. Delete Tertentu
Pilih = 2
Delete Lagi? t
Operasi lain? y
1. Insert
2. Delete
3. Tampilkan
Pilih = 3
No: 13, Nama: Ihsan Sept, Nilai: 99

```

Kesimpulan

Program menu insert SLL yang telah dibuat memiliki 3 fungsi hero yang berada pada fungsi utama, 3 fungsi delete, dan 4 fungsi insert. Fungsi utama, yaitu `int main()`, sesuai namanya berfungsi sebagai fungsi utama yang menjadi wadah saat program dijalankan, yang juga memanggil fungsi-fungsi lain. Untuk pemanggilan fungsi lain, digunakan `switch case` dari input user untuk memilih fungsi hero mana yang akan dipanggil, nantinya, fungsi hero insert akan memanggil 4 fungsi yang memasukkan data, fungsi hero delete akan memanggil 3 fungsi membebaskan data, dan fungsi hero tampil akan menampilkan datanya.

Untuk program yang ini, perbedaan dengan modul yang pertama hanyalah pada defining struct node serta fungsi `cari()` dan `tampil()`, karena di modul 1 hanya berupa data dan next, sedangkan di modul ini berupa no, nama, nilai, dan next. Perubahan pada defining struct node adalah karena kita butuh 4 (`no(int)`, `nama(string)`, `nilai(int)`, dan `next(Node *)`) data field instead of 2 (`data(int)`, `next(Node *)`). Perubahan ini juga berpengaruh pada fungsi `cari()` yang sekarang membutuhkan input no, nama, dan nilai, serta fungsi `tampil()` yang sekarang menampilkan 3 variabel tersebut.