

PRAKTIKUM ALGORITMA & STRUKTUR DATA
“PEMROGRAMAN TAMPILAN MENU SLL INSERT”



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 <stdlib.h>

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

Node *head = NULL, *p;

void isi(), cari();

void cari() {
    fflush(stdin);
    p = (Node *) malloc(sizeof(Node));
    p->next = NULL;
    printf("Data = "); scanf("%d", &p->data); }

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"); }

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 () {
    char y; int ch;
    do {
        fflush(stdin);
        printf("1. SSL Awal\n2. SSL Akhir\n3. SSL After\n4. SSL
Before\n5. Tampil\nPilih = ");
        scanf("%d", &ch);
        switch (ch) {
            case 1: awal(); break;
            case 2: akhir(); break;
            case 3: after(); break;
            case 4: before(); break;
            case 5: tampil(); break;
            default: printf("Pilihan salah\n"); }
        fflush(stdin);
        printf("Lagi? "); scanf(" %c", &y);
    } while (y == 'y' || y == 'Y'); }

```

1. SSL Awal 2. SSL Akhir 3. SSL After 4. SSL Before 5. Tampil Pilih = 1 Data = 4	INSERT AWAL
Lagi? y 1. SSL Awal 2. SSL Akhir 3. SSL After 4. SSL Before 5. Tampil Pilih = 2 Data = 9	INSERT AKHIR
Lagi? y 1. SSL Awal 2. SSL Akhir 3. SSL After 4. SSL Before 5. Tampil Pilih = 4 Cari data = 9 Data = 7	INSERT BEFORE
Lagi? y 1. SSL Awal 2. SSL Akhir 3. SSL After 4. SSL Before 5. Tampil Pilih = 3 Cari data = 2 Data tidak ditemukan	INSERT AFTER (key mismatch)

<pre>Lagi? y 1. SSL Awal 2. SSL Akhir 3. SSL After 4. SSL Before 5. Tampil Pilih = 3 Cari data = 4 Data = 5</pre>	INSERT AWAL
<pre>Lagi? y 1. SSL Awal 2. SSL Akhir 3. SSL After 4. SSL Before 5. Tampil Pilih = 5 4 5 7 9</pre>	PENAMPILAN DATA

Kesimpulan

Program menu insert SLL yang telah dibuat memiliki 6 fungsi user-defined dan 1 fungsi utama. 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 mana yang akan dipanggil.

Pada fungsi user-defined, ada 6 fungsi yang diantaranya fungsi `cari()`, 4 fungsi `insert`, dan fungsi `tampil()`. Fungsi `cari()` digunakan untuk mencari alamat kosong sebesar Node untuk diassign ke variabel `p`, serta pemvalidasiannya. Fungsi `cari()` dibuat karena supaya tidak berulang kali menulis kode pencarian alamat di tiap fungsinya. 4 fungsi `insert`, seperti namanya berarti insert node `p` kedalam SSLnya. Fungsi awal dan akhir sudah diprogram sedemikian supaya bisa menyesuaikan apabila SSL itu kosong/`head == null`. Fungsi `after` dan `before` diprogram supaya kembali apabila SSL itu kosong dan key tidak ditemukan.

PRAKTIKUM ALGORITMA & STRUKTUR DATA
“PEMROGRAMAN MENU SLL INSERT-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>

typedef struct node
{
    int no;
    char nama[50];
    int nilai;
    struct node *next;
} Node;

Node *head = NULL, *p;

void cari(), awal(), akhir(), after(), before(), tampil();

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

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

    printf("Nama \t= "); //mohon izin bu disini saya pakai ai karena
    saya kesulitan untuk melakukan input string karena buffer
    if (fgets(p->nama, sizeof(p->nama), stdin) != NULL) {
        p->nama[strcspn(p->nama, "\n")] = 0;
    }
    printf("Nilai \t= "); scanf("%d", &p->nilai); }

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"); }

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 () {
    char y; int ch;
    do {
        fflush(stdin);
        printf("1. SSL Awal\n2. SSL Akhir\n3. SSL After\n4. SSL
Before\n5. Tampil\nPilih = ");
        scanf("%d", &ch);
        switch (ch) {
            case 1: awal(); break;
            case 2: akhir(); break;
            case 3: after(); break;

```



```

        case 4: before(); break;
        case 5: tampil(); break;
        default: printf("Pilihan salah\n"); }
    fflush(stdin);
    printf("Lagi? "); scanf(" %c", &y);
} while (y == 'y' || y == 'Y'); }

```

```

1. SSL Awal
2. SSL Akhir
3. SSL After
4. SSL Before
5. Tampil
Pilih = 1
No      = 41
Nama    = Ihsan Septiano
Nilai   = 99
Lagi? y
1. SSL Awal
2. SSL Akhir
3. SSL After
4. SSL Before
5. Tampil
Pilih = 4
Cari data = 41
No      = 33
Nama    = Max Verstappen
Nilai   = 97
Lagi? y
1. SSL Awal
2. SSL Akhir
3. SSL After
4. SSL Before
5. Tampil
Pilih = 2
No      = 74
Nama    = Anakin Skywalker
Nilai   = 92
Lagi? y
1. SSL Awal
2. SSL Akhir
3. SSL After
4. SSL Before
5. Tampil
Pilih = 5
No: 33, Nama: Max Verstappen, Nilai: 97
No: 41, Nama: Ihsan Septiano, Nilai: 99
No: 74, Nama: Anakin Skywalker, Nilai: 92

```

Kesimpulan

Program menu insert SLL yang telah dibuat memiliki 6 fungsi user-defined dan 1 fungsi utama. 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 mana yang akan dipanggil.

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.