

**TUGAS POSTES 1**  
**Algoritma dan Struktur Data**



**Nama : Devafilla Rizqy Santosa**  
**Kelas : 1 D4 Teknik Informatika A**  
**NRP : 3120600026**

## Postes 1.1 Data Int

### 1. A. Listing

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

typedef struct simpul Node;
struct simpul
{
    int data;
    Node *next;
};
Node *head = NULL, *p;

void menu();
void menui(int masukan);
void awal();
void akhir();
void before(int k);
void dawal();
void alokasi();
void urutkan();
void carikan();
void dtertentu();
void hapus();
void tampil();
void rata_rata();

int main()
{
    menu();
}

void menu()
{
    int masukan;

    puts("-----Menu Pilihan User-----");
    puts("1. Penambahan Secara Terurut");
    puts("2. Pencarian Data");
    puts("3. Penghapusan Data");
    puts("4. Keluar");
    printf("Masukan Angka (1-4) ? ");
    scanf("%d", &masukan);
    menui(masukan);
}

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;
    }
}

void menui(int masukan)
{
do
    {
        if(masukan == 1)
            alokasi();
        switch(masukan)
        {
            case 1:
                urutkan();
                break;

            case 2:
                carikan();
                break;

            case 3:
                if(head==NULL)
                    puts("SLL belum diinput");
                else
                    dtertentu();
                break;

            case 4:
                exit(0);

            default:
                puts("Mohon mengulangi pilihan");
                puts("");
                break;
        }
        if (masukan == 1 || masukan == 3)
        {
            tampil();
            printf("\n\n");
        }
        menu();
    }
    while(masukan != 4);

    return 0;
}

void before (int k)
{
    Node *bef, *pbef;
    bef = head;

    if(bef->data == k)
        awal();

    else
    {
        while(bef->data != k)
        {
            pbef = bef;
            bef = bef->next;
        }
        p->next = bef;
        pbef->next = p;
    }
}

```

```

void dawal()
{
    Node *d;
    d = head;
    if (d->next == NULL)
        head = NULL;
    else
    {
        head = d->next;
        hapus(d);
    }
}

void alokasi()
{
    int a;

    printf("Data yang akan disimpan : ");
    scanf("%d", &a);
    fflush(stdin);
    p = (Node *)malloc(sizeof(Node));

    if(p == NULL)
        printf("Alokasi memori gagal");

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

void urutkan()
{
    Node *baca;
    baca = head;

    while(baca != NULL)
    {
        if(baca->data > p->data)
        {
            before(baca->data);
            break;
        }
        baca = baca->next;
    }
    if(baca == NULL)
        akhir();
}

void carikan()
{
    Node *baca;
    int j = 0, k;

    printf("Masukan Data Yang Akan DiCari : ");
    scanf("%d", &k);

    baca = head;
    while(baca != NULL)
    {
        if(baca->data == k)
            j++;
        baca=baca->next;
    }
}

```

```

        if(j > 0)
            printf("data %d yang tersimpan %d kali\n\n", k, j);

        else
            printf("Data %d SLL masih kosong\n\n", k);
    }

void dtentukan()
{
    Node *d, *p_d;
    int k;

    printf("Pilih data yang dihapus : ");
    scanf("%d", &k);
    d = head;
    if (d->data == k)
        daval();
    else
    {
        while (d->data != k)
        {
            if (d->next == NULL)
            {
                printf("%d Belum input dalam SLL\n", k);
                break;
            }
            else
            {
                p_d = d;
                d = d->next;
            }
        }
        if(d->data == k)
        {
            p_d->next = d->next;
            hapus(d);
        }
    }
}

void hapus(Node *a)
{
    free(a);
    a = NULL;
}

void tampil()
{
    Node *baca;
    baca = head;

    printf("\nData Yang DiSimpan : \n");
    while(baca != NULL)
    {
        printf("%d ", baca->data);
        baca = baca->next;
    }
}

```

## B. Output

```
-----Menu Pilihan User-----
1. Penambahan Secara Terurut
2. Pencarian Data
3. Penghapusan Data
4. Keluar
Masukan Angka (1-5) ? 1
Data yang akan disimpan : 2

Data Yang DiSimpan :
2

-----Menu Pilihan User-----
1. Penambahan Secara Terurut
2. Pencarian Data
3. Penghapusan Data
4. Keluar
Masukan Angka (1-5) ? 1
Data yang akan disimpan : 4

Data Yang DiSimpan :
2 4

-----Menu Pilihan User-----
1. Penambahan Secara Terurut
2. Pencarian Data
3. Penghapusan Data
4. Keluar
Masukan Angka (1-5) ? 1
Data yang akan disimpan : 6

Data Yang DiSimpan :
2 4 6

-----Menu Pilihan User-----
1. Penambahan Secara Terurut
2. Pencarian Data
3. Penghapusan Data
4. Keluar
Masukan Angka (1-5) ? 1
Data yang akan disimpan : 1

Data Yang DiSimpan :
1 2 4 6

-----Menu Pilihan User-----
1. Penambahan Secara Terurut
2. Pencarian Data
3. Penghapusan Data
4. Keluar
Masukan Angka (1-5) ? 1
Data yang akan disimpan : 3

Data Yang DiSimpan :
1 2 3 4 6

-----Menu Pilihan User-----
1. Penambahan Secara Terurut
2. Pencarian Data
3. Penghapusan Data
4. Keluar
Masukan Angka (1-5) ? 1
Data yang akan disimpan : 2

Data Yang DiSimpan :
1 1 2 3 4 5 6

-----Menu Pilihan User-----
1. Penambahan Secara Terurut
2. Pencarian Data
3. Penghapusan Data
4. Keluar
Masukan Angka (1-5) ? 1
Data yang akan disimpan : 1

Data Yang DiSimpan :
1 1 2 3 4 5 6

-----Menu Pilihan User-----
1. Penambahan Secara Terurut
2. Pencarian Data
3. Penghapusan Data
4. Keluar
Masukan Angka (1-5) ? 2
Masukan Data Yang Akan DiCari : 1
data 1 yang tersimpan 2 kali

-----Menu Pilihan User-----
1. Penambahan Secara Terurut
2. Pencarian Data
3. Penghapusan Data
4. Keluar
Masukan Angka (1-5) ? 3
Pilih data yang dihapus : 1

Data Yang DiSimpan :
1 2 3 4 5 6

-----Menu Pilihan User-----
1. Penambahan Secara Terurut
2. Pencarian Data
3. Penghapusan Data
4. Keluar
Masukan Angka (1-5) ? 3
Pilih data yang dihapus : 1

Data Yang DiSimpan :
2 3 4 5 6

-----Menu Pilihan User-----
1. Penambahan Secara Terurut
2. Pencarian Data
3. Penghapusan Data
4. Keluar
Masukan Angka (1-5) ? 3
Pilih data yang dihapus : 1

Data Yang DiSimpan :
2 3 4 5 6

-----Menu Pilihan User-----
1. Penambahan Secara Terurut
2. Pencarian Data
3. Penghapusan Data
4. Keluar
Masukan Angka (1-5) ? 4

Process returned 0 (0x0)   execution
Press any key to continue.
■
```

## Postes 1.2 Data Struct

### 2. A. Listing

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

typedef struct mahasiswa Node;
struct mahasiswa
{
    int no;
    char nm[20];
    float nli;
    Node *next;
};
Node *head = NULL, *p;

void menu();
void menui(int masukan);
void awal();
void akhir();
void before(int k);
void dawal();
void alokasi();
void urutkan();
void carikan();
void dtertentu();
void hapus();
void tampil();

int main()
{
    menu();
}

void menu()
{
    int masukan;

    puts("-----Menu Pilihan User-----");
    puts("1. Penambahan Secara Terurut");
    puts("2. Pencarian Data");
    puts("3. Penghapusan Data");
    puts("4. Rata - Rata");
    puts("5. Keluar");
    printf("Masukan Angka (1-5) ? ");
    scanf("%d", &masukan);
    menui(masukan);
}

void menui(int masukan)
{
    do
    {
        if(masukan == 1)
            alokasi();
        switch(masukan)
        {
            case 1:
                urutkan();
                break;

            case 2:
                carikan();
                break;

            case 3:
                hapus();
                break;

            case 4:
                rata();
                break;

            case 5:
                akhir();
                break;

            default:
                printf("Masukan Angka (1-5) ? ");
                scanf("%d", &masukan);
                break;
        }
    } while (masukan != 5);
}
```

```

        case 3:
            if(head==NULL)
                puts("SLL belum diinput");
            else
                dtertentu();
            break;

        case 4:
            rata_rata();
            break;

        case 5:
            exit(0);

        default:
            puts("Mohon mengulangi pilihan");
            puts("");
            break;
    }
    if (masukan == 1 || masukan == 3)
    {
        tampil();
        printf("\n\n");
    }
    menu();
}
while(masukan != 5);

return 0;
}

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;
    }
}

void before (int k)
{
    Node *bef, *pbef;
    bef = head;

    if(bef->no == k)
        awal();

    else
        ,

```



```

    {
        while(bef-> no != k)
        {
            pbef = bef;
            bef = bef->next;
        }
        p->next = bef;
        pbef->next = p;
    }
}

void dawal()
{
    Node *d;
    d = head;
    if (d->next == NULL)
        head = NULL;
    else
    {
        head = d->next;
        hapus(d);
    }
}

void alokasi()
{
    int no;
    char nm[20];
    float nli;

    printf("\n");
    printf("Data yang akan disimpan : \n");

    printf("%-5s", "No    :");
    scanf("%d", &no);
    fflush(stdin);
    printf("%-5s", "Nama :");
    scanf("%s", &nm);
    fflush(stdin);
    fflush(stdin);
    printf("%-5s", "Nilai :");
    scanf("%f", &nli);
    fflush(stdin);

    p = (Node *) malloc(sizeof(Node));

    if (p == NULL)
        printf("Alokasi memori gagal");

    else
    {
        p->no = no;
        strcpy(p->nm, nm);
        p->nli = nli;
        p->next = NULL;
    }
}

void urutkan()
{
    Node *baca;
    baca = head;

    while(baca != NULL)
    ,

```

```

    {
        if(baca->no > p->no)
        {
            before(baca->no);
            break;
        }
        baca = baca->next;
    }
    if(baca == NULL)
        akhir();
}

void carikan()
{
    Node *baca;
    int j = 0, k;

    printf("Masukan Data Yang Akan DiCari : ");
    scanf("%d", &k);

    baca = head;
    while(baca != NULL)
    {
        if(baca->no == k)
            j++;
        baca=baca->next;
    }
    if(j > 0)
        printf("data %d yang tersimpan %d kali\n\n", k, j);

    else
        printf("Data %d SLL masih kosong\n\n", k);
}

void dtentukan()
{
    Node *d, *p_d;
    int k;
    printf("Pilih data yang dihapus : ");
    scanf("%d", &k);
    d = head;
    if (d->no == k)
        dawal();
    else
    {
        while (d->no != k)
        {
            if (d->next == NULL)
            {
                printf("%d Belum input dalam SLL\n", k);
                break;
            }
            else
            {
                p_d = d;
                d = d->next;
            }
        }
        if(d->no == k)
        {
            p_d->next = d->next;
            hapus(d);
        }
    }
}

```

```

void hapus(Node *a)
{
    free(a);
    a = NULL;
}

void tampil()
{
    Node *baca;
    baca = head;

    printf("Inputan data : \n");
    printf("%-5s", "No"); printf("%-5s", "Nama"); printf("%-5s", "Nilai");
    printf("\n");
    while(baca != NULL)
    {
        printf("%-5d", baca->no); printf("%-5s", baca->nm); printf("%-5.2f", baca->nli);
        printf("\n");
        baca = baca->next;
    }
    printf("\n\n");
}

void rata_rata()
{
    Node *baca;
    baca = head;

    int j = 0;
    float t = 0, h;
    if (head == NULL)
        puts("SLL belum diinput");
    else
    {
        while (baca != NULL)
        {
            t += baca -> nli;
            j++;
            baca = baca -> next;
        }
        h = t/j;
        printf("rata-rata = %g\n", h);
        puts("");
    }
}

```

## B. Output

```
-----Menu Pilihan User-----
1. Penambahan Secara Terurut
2. Pencarian Data
3. Penghapusan Data
4. Rata - Rata
5. Keluar
Masukan Angka (1-5) ? 1

Data yang akan disimpan :
No :1
Nama :deva
Nilai :12
Inputan data :
No Nama Nilai
1 deva 12.00

-----Menu Pilihan User-----
1. Penambahan Secara Terurut
2. Pencarian Data
3. Penghapusan Data
4. Rata - Rata
5. Keluar
Masukan Angka (1-5) ? 2
Masukan Data Yang Akan DiCari : 1
data 1 yang tersimpan 2 kali

-----Menu Pilihan User-----
1. Penambahan Secara Terurut
2. Pencarian Data
3. Penghapusan Data
4. Rata - Rata
5. Keluar
Masukan Angka (1-5) ? 3
Pilih data yang dihapus : 1
Inputan data :
No Nama Nilai
1 dega 45.00
2 deca 23.00
3 dexe 34.00

-----Menu Pilihan User-----
1. Penambahan Secara Terurut
2. Pencarian Data
3. Penghapusan Data
4. Rata - Rata
5. Keluar
Masukan Angka (1-5) ? 4
rata-rata = 34

-----Menu Pilihan User-----
1. Penambahan Secara Terurut
2. Pencarian Data
3. Penghapusan Data
4. Rata - Rata
5. Keluar
Masukan Angka (1-5) ? 6
Mohon mengulangi pilihan

-----Menu Pilihan User-----
1. Penambahan Secara Terurut
2. Pencarian Data
3. Penghapusan Data
4. Rata - Rata
5. Keluar
Masukan Angka (1-5) ? 5

Process returned 0 (0x0) execution
Press any key to continue.
```