

# LEMBAR UJIAN MAHASISWA

Disusun untuk Memenuhi Ujian Mata Kuliah Algoritma dan Pemrograman II

Dosen Pengampu Dhany Indra Gunawan, S.T., M.Kom



Tugas

Disusun Oleh :

Nama	: Noga Restu Aji
NPM	: 21552011092
Kelas / Semester	: TIF RP – 21E   Semester 2
Program Studi Keahlian	: Teknik Informatika

## SOAL

1. Apa yang dimaksud Linked List?
2. Buatlah program bahasa C berupa fungsi rekursif dan pemanggilnya untuk menghitung deret berikut : (pilih 2 soal saja untuk dikerjakan)
  - a.  $\text{sum} = 2 + 4 + 6 + 8 + \dots$
  - b.  $\text{sum} = 1 + 4 + 16 + 64 + 256 + \dots$
  - c.  $\text{sum} = 3 + 6 + 9 + 12 + \dots$
  - d.  $\text{sum} = 1 + 3 + 9 + 27 + 81 + \dots$
3. Buatlah algoritma dan program bahasa C untuk pemrosesan data pasien (kode\_pasien, nama, usia) dalam struktur list berkait. Program harus dibuat menggunakan menu sebagai berikut :
  - a. Input data pasien baru
  - b. Cari data pasien berdasarkan nama
  - c. Hapus data pasien berdasarkan kode\_pasien
  - d. Tampilkan data pasien dengan usia termuda
  - e. Tampilkan data pasien dengan usia tertua
  - f. Tampilkan rata-rata usia pasien
  - g. Tampilkan seluruh data pasien beserta jumlah data yang tersimpan dalam list
  - h. Keluar
4. Buatlah program bahasa C untuk pemrosesan data dosen (kode\_dosen, nama, total\_sks) dalam struktur list berkait. Program harus dibuat menggunakan menu sebagai berikut :
  - a. Input data dosen baru
  - b. Cari data dosen berdasarkan nama
  - c. Hapus data dosen berdasarkan kode\_dosen
  - d. Tampilkan data dosen dengan total\_sks terbanyak
  - e. Tampilkan data dosen dengan total\_sks tersedikit
  - f. Tampilkan rata-rata total sks yang diambil dosen
  - g. Tampilkan seluruh data dosen beserta jumlah data yang tersimpan dalam list
  - h. Keluar

## JAWABAN

1. Linked List merupakan koleksi linear dari data, yang disebut sebagai nodes, dimana setiap node akan menunjuk pada node lain melalui sebuah pointer. Linked List dapat didefinisikan pula sebagai kumpulan nodes yang merepresentasikan sebuah sequence.

2.

a. Coding :

```
1  /*Program Menghitung Deret 2 + 4 + 6 + 8 + ... */
2  #include <stdio.h>
3
4  long int deret(int x);
5  main()
6  {
7      int N;
8      long int hasil;
9      printf("Program Menghitung Deret 2 + 4 + 6 + 8 + ..\n");
10     printf("Masukkan banyaknya suku deret : ");
11     scanf("%d",&N);
12     hasil=deret(N);
13     printf("\n Nilai hasil penjumlahan deret= %d",hasil);
14 }
15
16 long int deret(int x)
17 {
18     int i,a,b,un,n;
19     long int z=0;
20     a=2; // a disini adalah suku pertama deret
21     b=2; // b disini adalah jarak antar suku deret
22     un=a+(x-1)*b;
23     printf("Deret : : ");
24     for (i=a; i<=un; i=i+b)
25     {
26         z+=i;
27         printf("\n %d",i);
28     }
29     return(z);
30 }
31
```

Output :

```
Program Menghitung Deret 2 + 4 + 6 + 8 + ..
Masukkan banyaknya suku deret : 7
Deret : :
2
4
6
8
10
12
14
Nilai hasil penjumlahan deret= 56
-----
```

b. Coding :

```
1  /*Program Menghitung Deret1 + 4 + 16 + 64 + 256 + ... */
2  #include <stdio.h>
3
4  long int deret(int x);
5  main()
6  {
7      int N;
8      long int hasil;
9      printf("Program Menghitung Deret 1 + 4 + 16 + 64 + 256 + ..\n");
10     printf("Masukkan banyaknya suku deret : ");
11     scanf("%d",&N);
12     hasil=deret(N);
13     printf("\n Nilai hasil penjumlahan deret= %d",hasil);
14 }
15
16 long int deret(int x)
17 {
18     int i,a,b,un,n,p;
19     long int z=0;9977
20     a=1; // a disini adalah suku pertama deret
21     b=4; // b disini adalah jarak antar suku deret
22     un=x;
23     printf("Deret : : ");
24     for (i=a; i<=un; i++)
25     {p=i*b;
26     z+=p;
27     printf("\n %d",i*b);
28     }
29     return(z);
30 }
31
```

Output :

```
Program Menghitung Deret 1 + 4 + 16 + 64 + 256 + ..
Masukkan banyaknya suku deret : 7
Deret : :
4
8
16
20
24
28
Nilai hasil penjumlahan deret= 112
```

3. .

- **ALGORITMA :**

a. input data pasien baru

Algoritma input data pasien baru
Procedure insert (input/output datapasien : list, input temp : address) Kamus P=node pasien : address Algoritma if (first(datapasien) = NULL) then insertFirst(data pasien, temp) else pasien ← first(datapasien) while (next(pasien) ≠ NULL) do pasien ← next(pasien) next(pasien) ← temp

b. cari data pasien menggunakan nama

Algoritma cari data pasien menggunakan nama
Function findElm (L : list, X : infotype) → address Kamus P=node Algoritma P ← first(L) while ((P ≠ NULL) and (info(P) ≠ X)) do P ← next(P) if (info(P) = X) then → P else → NULL

- c. Hapus data menggunakan kode pasien

Algoritma Hapus data menggunakan kode pasien
Function delete (L : list, X : infotype) → address Kamus P=node Algoritma P ← first(L)  while ((P ≠ NULL) and (info(P) ≠ X)) do P ← next(P)  if (info(P) = X) then → p(delete) else → NULL

- d. Tampilkan data usia pasien termuda

Algoritma tampilkan data usia termuda
Function termuda Kamus P=node min = INT_MAX data = umur pasien Algoritma P ← first while ((P ≠ NULL) do  if (min > P → data) then min = P → P else P = P → next  Output(min)

- e. Tampilkan data usia pasien tertua

Algoritma tampilkan data usia tertua
Function tertua Kamus P=node max = INT_MIN data = umur pasien Algortima P ← first while ((P ≠ NULL) do if (min>P →data) then min = P→ P else P=P→ next Output(max)

- f. Tampilkan rata-rata usia pasien

Algoritma tampilkan rata-rata usia pasien
Function ratarata Kamus P=node h= jumlah node j = jumlah seluruh usia Algortima P ← first while ((P ≠ NULL) do h++ j+=P →data P=P→ next Rata= j/p Output(rata)

- g. Tampilkan seluruh pasien beserta datanya

Algoritma tampilkan seluruh pasien beserta datanya
Function ratarata Kamus P=node d= data pasien Algortima P ← first while ((P ≠ NULL) do Output(P(d) P=P→ next

- **SOURCE CODE**

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

struct Pasien{
    int kp,umur;
    char nama[25];
};
struct node{
    struct Pasien pasien;
    struct node *next;
};
struct node *head = NULL;
struct node *last = NULL;

bool isEmpty(){
    return head == NULL;
}

char menu(){
    char choice;

    printf("1. Input data pasien baru\n");
    printf("2. Cari data Pasien\n");
    printf("3. Hapus data pasien\n");
    printf("4. Tampilkan data pasien dengan usia termuda\n");
```



```

        printf("5. Tampilkan data pasien dengan usia tertua\n");
        printf("6. Tampilkan rata-rata usia pasien\n");
        printf("7. Tampilkan seluruh data pasien\n");
        printf("8. Exit\n\n");
        printf("Pilihan : ");
        scanf(" %c",&choice);
        return choice;
    }
    void insert(struct Pasien pasien){
        if(isEmpty()){
            struct node *temp= (struct node*) malloc(sizeof(struct node));
            temp->pasien = pasien;
            temp->next = NULL;
            head = temp;
            last = temp;
        }else{
            struct node *temp = (struct node*) malloc(sizeof(struct node));
            temp->pasien = pasien;
            temp->next = NULL;
            last->next = temp;
            last = temp;
        }
    }
    void termuda(){
        int min = INT_MAX;
        int kp;
        struct node *temp = head;
        while(temp != NULL){
            if(min>temp->pasien.umur)
                min = temp->pasien.umur;
            kp = temp ->pasien.kp;
            temp = temp ->next;
        }
        printf("Pasien dengan usia termuda adalah :\n");
        printf("|Kode Pasien| Umur\n");
        printf("-----\n");
        printf("|%7d  |%5d\n", kp, min);
    }
    void tertua(){
        int max = INT_MIN;
        int kp;
        struct node *temp = head;
        while(temp != NULL){

```

```

        if (max < temp->pasien.umur)
            max = temp ->pasien.umur;
            kp = temp ->pasien.kp;
            temp = temp->next;
    }
    printf("Pasien dengan usia tertua adalah :\n");
    printf("|Kode Pasien| Umur\n");
    printf("-----\n");
    printf("|%7d   |%5d\n", kp, max);
}

void displayList(){
    if(isEmpty()){
        printf("Data Kosong");
    }else{
        struct node *temp = head;
        printf("|Kode Pasien|      Nama| Umur\n");
        printf("-----\n");
        while(temp!=NULL){
            printf("|%7d   | %15s|%5d\n", temp->pasien.kp, temp->pasien.nama, temp->pasien.umur);
            temp = temp->next;
        }
    }
}

void hapusData(){
    int tempkp;
    if(head !=NULL){
        printf("kode pasien yang ingin dihapus : ");
        scanf("%d", &tempkp);
        struct node *temp = head;
        if(head->pasien.kp==tempkp){
            head = head->next;
            delete temp;
        }
        else{
            struct node *temp2=NULL;
            while (temp!=NULL && temp->pasien.kp !=tempkp){
                temp2 = temp;
                temp = temp->next;
            }
        }
    }
}

```

```

        if(temp!=NULL){
            printf("Pasien dengan kode %d telah dihapus\n", temp-
>pasien.kp);
            printf("Detail : \n");
            printf("Kode Pasien: %d\n", temp->pasien.kp);
            temp2->next = temp->next;
            delete temp;
        }
    }
}

float ratarata(struct node* head)
{
    if(!head)
        return -1;

    int hitung=0;
    int jumlah=0;
    float rata=0.0;

    struct node* temp=head;
    while(temp!=NULL){
        hitung++;
        jumlah += temp->pasien.umur;
        temp = temp ->next;
    }
    rata =(double)jumlah/hitung;
    return rata;
}

void showTotal(){
    struct node *temp = head;
    int total = 0;
    while(temp!=NULL){
        total += 1;
        temp = temp->next;
    }
    printf("Total Pasien yang ada di dalam rumah sakit adalah %d jiwa",
total);
}

```

```

void searchNama() {
    char f[25];
    struct node* temp, *nn;
    temp = head;

    printf("Masukkan Nama pasien yang ingin dicari: "); scanf("%s",f);
    printf("%s\n", f);
    while (temp != NULL){

        if (strcmp(temp -> pasien.nama, f) == 0) {
            printf ("Pasien ditemukan!\n\n");
            printf("|Kode Pasien|      Nama| Umur\n");

            temp = temp -> next;

        }else{

int main()
{
    char choice;
    do{
        struct Pasien pasien;
        system("cls");
        choice = menu();
        switch(choice){
            case '1':
                printf("Kode Pasien :");
                scanf("%d", &(pasien.kp));
                printf("Nama Pasien :");
                scanf("%s", (pasien.nama));
                printf("Umur Pasien :");
                scanf("%d", &(pasien.umur));
                insert(pasien);
                break;
            case '2':
                searchNama();
                break;
            case '3':
                hapusData();

```

```
                break;
            case '4':
                termuda();
                break;
            case '5':
                tertua();
                break;
            case '6':
                printf("Rata-rata usia Pasien adalah %.2f
tahun\n",ratarata(head));
                break;
            case '7':
                displayList();
                break;
            default:
                printf("salah");
                break;
        }
        getch();
    }while(choice!='8');

return 0;
}
```

- **DEMO PROGRAM**

- a. Input data pasien baru

- a) Pasien ke-1

```
1. Input data pasien baru
2. Cari data Pasien
3. Hapus data pasien
4. Tampilkan data pasien dengan usia termuda
5. Tampilkan data pasien dengan usia tertua
6. Tampilkan rata-rata usia pasien
7. Tampilkan seluruh data pasien
8. Exit

Pilihan : 1
Kode Pasien :001
Nama Pasien :Andi
Umur Pasien :10
```

- b) Pasien ke-2

```
1. Input data pasien baru
2. Cari data Pasien
3. Hapus data pasien
4. Tampilkan data pasien dengan usia termuda
5. Tampilkan data pasien dengan usia tertua
6. Tampilkan rata-rata usia pasien
7. Tampilkan seluruh data pasien
8. Exit

Pilihan : 1
Kode Pasien :002
Nama Pasien :Bagus
Umur Pasien :20
```

- c) Pasien ke-3

```
1. Input data pasien baru
2. Cari data Pasien
3. Hapus data pasien
4. Tampilkan data pasien dengan usia termuda
5. Tampilkan data pasien dengan usia tertua
6. Tampilkan rata-rata usia pasien
7. Tampilkan seluruh data pasien
8. Exit

Pilihan : 1
Kode Pasien :003
Nama Pasien :Cinta
Umur Pasien :18
```

b. Cari data pasien berdasarkan Nama

a) Mencari Pasien atas nama Andi

C:\Users\NOGA RESTU AJI\Documents\SEMESTER 2\C++\linklist fixfix.exe

```
4. Tampilkan data pasien dengan usia termuda
5. Tampilkan data pasien dengan usia tertua
6. Tampilkan rata-rata usia pasien
7. Tampilkan seluruh data pasien
8. Exit
```

Pilihan : 2

Masukkan Nama pasien yang ingin dicari: Andi

Andi

Pasien ditemukan!

Kode Pasien	Nama	Umur
1	Andi	10

Pencarian Selesai

b) Mencari Pasien atas nama Bagus

C:\Users\NOGA RESTU AJI\Documents\SEMESTER 2\C++\linklist fixfix.exe

```
1. Input data pasien baru
2. Cari data Pasien
3. Hapus data pasien
4. Tampilkan data pasien dengan usia termuda
5. Tampilkan data pasien dengan usia tertua
6. Tampilkan rata-rata usia pasien
7. Tampilkan seluruh data pasien
8. Exit
```

Pilihan : 2

Masukkan Nama pasien yang ingin dicari: Bagus

Bagus

Pasien ditemukan!

Kode Pasien	Nama	Umur
2	Bagus	20

Pencarian Selesai

- c. Menghapus data pasien berdasarkan kode

```
C:\Users\NOGA RESTU AJI\Documents\SEMESTER 2\C++\linklist fixfix.exe
1. Input data pasien baru
2. Cari data Pasien
3. Hapus data pasien
4. Tampilkan data pasien dengan usia termuda
5. Tampilkan data pasien dengan usia tertua
6. Tampilkan rata-rata usia pasien
7. Tampilkan seluruh data pasien
8. Exit

Pilihan : 3
kode pasien yang ingin dihapus : 002
Pasien dengan kode 2 telah dihapus
Detail :
|Kode Pasien|          Nama|  Umur
-----
|      2    |      Bagus|   20
```

- d. Mencari data pasien dengan usia termuda

```
1. Input data pasien baru
2. Cari data Pasien
3. Hapus data pasien
4. Tampilkan data pasien dengan usia termuda
5. Tampilkan data pasien dengan usia tertua
6. Tampilkan rata-rata usia pasien
7. Tampilkan seluruh data pasien
8. Exit

Pilihan : 4
Pasien dengan usia termuda adalah :
|Kode Pasien|  Umur
-----
|      3    |   10
```

- e. Mencari data pasien dengan usia tertua

```
Pilihan : 5
Pasien dengan usia tertua adalah :
|Kode Pasien|  Umur
-----
|      3    |   18
```



- f. Mencari rata-rata usia pasien

```
C:\Users\NOGA RESTU AJI\Documents\SEMESTER 2\C++\linklist t
1. Input data pasien baru
2. Cari data Pasien
3. Hapus data pasien
4. Tampilkan data pasien dengan usia termuda
5. Tampilkan data pasien dengan usia tertua
6. Tampilkan rata-rata usia pasien
7. Tampilkan seluruh data pasien
8. Exit

Pilihan : 6
Rata-rata usia Pasien adalah 14.00 tahun
```

- g. Menampilkan seluruh data pasien yang tersimpan

```
C:\Users\NOGA RESTU AJI\Documents\SEMESTER 2\C++\linklist fixfix.exe
1. Input data pasien baru
2. Cari data Pasien
3. Hapus data pasien
4. Tampilkan data pasien dengan usia termuda
5. Tampilkan data pasien dengan usia tertua
6. Tampilkan rata-rata usia pasien
7. Tampilkan seluruh data pasien
8. Exit

Pilihan : 7
|Kode Pasien|          Nama|  Umur
-----
|      1    |      Andi   |   10
|      3    |      Cinta  |   18
```

4.

- **SOURCE CODE**

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

struct Dosen{
    int kp,sks;
    char nama[25];
};
struct node{
    struct Dosen dosen;
    struct node *next;
};
struct node *head = NULL;
struct node *last = NULL;

bool isEmpty(){
    return head == NULL;
}

char menu(){
    char choice;

    printf("1. Input data dosen baru\n");
    printf("2. Cari data dosen\n");
    printf("3. Hapus data dosen\n");
    printf("4. Tampilkan data dosen dengan usia termuda\n");
    printf("5. Tampilkan data dosen dengan usia tertua\n");
    printf("6. Tampilkan rata-rata sks yang diambil dosen\n");
    printf("7. Tampilkan seluruh data dosen\n");
    printf("8. Exit\n\n");
    printf("Pilihan : ");
    scanf(" %c",&choice);
    return choice;
}

void insert(struct Dosen dosen){
    if(isEmpty()){
        struct node *temp= (struct node*) malloc(sizeof(struct node));
        temp->dosen = dosen;
        temp->next = NULL;
```

```

        head = temp;
        last = temp;
    }else{
        struct node *temp = (struct node*) malloc(sizeof(struct node));
        temp->dosen = dosen;
        temp->next = NULL;
        last->next = temp;
        last = temp;
    }
}

void terbanyak(){
    int min = INT_MAX;
    int kp;
    struct node *temp = head;
    while(temp != NULL){
        if(min>temp->dosen.sks)
            min = temp->dosen.sks;
        kp = temp ->dosen.kp;
        temp = temp ->next;
    }
    printf("Pasien dengan usia termuda adalah :\n");

    printf("|Kode Pasien| Umur\n");
    printf("-----\n");
    printf("|%7d  |%5d\n", kp, min);
}

void tersedikit(){
    int max = INT_MIN;
    int kp;
    struct node *temp = head;
    while(temp != NULL){
        if (max<temp->dosen.sks)
            max = temp ->dosen.sks;
        kp = temp ->dosen.kp;
        temp = temp->next;
    }
    printf("Pasien dengan usia tertua adalah :\n");
    printf("|Kode Pasien| Umur\n");
    printf("-----\n");
    printf("|%7d  |%5d\n", kp, max);
}

void displayList(){

```

```

        if(isEmpty()){
            printf("Data Kosong");
        }else{
            struct node *temp = head;
            printf("|Kode Pasien|      Nama|  Umur\n");
            printf("-----\n");
            while(temp!=NULL){
                printf("|%7d   | %15s|%5d\n", temp->dosen.kp, temp-
>dosen.nama, temp->dosen.sks);
                temp = temp->next;
            }
        }
    }

void hapusData(){
    int tempkp;
    if(head !=NULL){
        printf("kode pasien yang ingin dihapus : ");
        scanf("%d", &tempkp);
        struct node *temp = head;
        if(head->dosen.kp==tempkp){
            head = head->next;
            delete temp;
        }
        else{
            struct node *temp2=NULL;
            while (temp!=NULL && temp->dosen.kp !=tempkp){
                temp2 = temp;
                temp = temp->next;
            }
            if(temp!=NULL){
                printf("Pasien dengan kode %d telah dihapus\n", temp-
>dosen.kp);

                printf("Detail : \n");
                printf("|Kode Pasien|      Nama|  Umur\n");
                printf("-----\n");
                printf("|%7d   | %15s|%5d\n\n", temp->dosen.kp, temp-
>dosen.nama, temp->dosen.sks);
                temp2->next = temp->next;
                delete temp;
            }
        }
    }
}

```

```

    }
}
float ratarata(struct node* head)
{
    if(!head)
        return -1;

    int hitung=0;
    int jumlah=0;
    float rata=0.0;

    struct node* temp=head;
    while(temp!=NULL){
        hitung++;
        jumlah += temp->dosen.sks;
        temp = temp ->next;
    }
    rata =(double)jumlah/hitung;
    return rata;
}

void showTotal(){
    struct node *temp = head;
    int total = 0;
    while(temp!=NULL){
        total += 1;
        temp = temp->next;
    }
    printf("Total Pasien yang ada di dalam rumah sakit adalah %d jiwa",
total);
}

void searchNama() {
    char f[25];
    struct node* temp, *nn;
    temp = head;

    printf("Masukkan Nama pasien yang ingin dicari: "); scanf("%s",f);
    printf("%s\n", f);
    while (temp != NULL){

```

```

        if (strcmp(temp -> dosen.nama, f) == 0) {
            printf("Pasien ditemukan!\n\n");
            printf("|Kode Pasien|      Nama|  Umur\n");
            printf("-----\n");
            printf("|%7d   | %15s|%5d\n\n", temp->dosen.kp, temp-
>dosen.nama, temp->dosen.sks);
            temp = temp -> next;

        }else{
            temp = temp -> next;

        }
    }printf("Pencarian Selesai");
}

int main()
{
    char choice;
    do{
        struct Dosen dosen;
        system("cls");
        choice = menu();
        switch(choice){
            case '1':
                printf("Kode Dosen :");
                scanf("%d", &(dosen.kp));
                printf("Nama Dosen :");
                scanf("%s", (dosen.nama));
                printf("Jumlah sks :");
                scanf("%d", &(dosen.sks));
                insert(dosen);
                break;
            case '2':
                searchNama();
                break;
            case '3':
                hapusData();
                break;
            case '4':
                tersedikit();
                break;
            case '5':
                terbanyak();

```

```
                break;
            case '6':
                printf("Rata-rata jumlah sks yang diambil adalah
%.2f\n",ratarata(head));
                break;
            case '7':
                displayList();
                break;
            default:
                printf("Terima Kasih, Klik apapun untuk keluar");
                break;
        }
        getch();
    }while(choice!='8');

return 0;
}
```

- **DEMO PROGRAM**

- a. Input data dosen baru

- a) Dosen ke-1

```
C:\Users\NOGA RESTU AJI\Documents\SEMESTER 2\C++\UAS-LINKED LIST DOSEN
1. Input data dosen baru
2. Cari data dosen
3. Hapus data dosen
4. Tampilkan data dosen dengan jumlah sks paling sedikit
5. Tampilkan data dosen dengan jumlah sks paling banyak
6. Tampilkan rata-rata sks yang diambil dosen
7. Tampilkan seluruh data dosen
8. Exit

Pilihan : 1
Kode Dosen :001
Nama Dosen :Noga
Jumlah sks :20
```

- b) Dosen ke-2

```
C:\Users\NOGA RESTU AJI\Documents\SEMESTER 2\C++\UAS-LINKED LIST DOSEN
1. Input data dosen baru
2. Cari data dosen
3. Hapus data dosen
4. Tampilkan data dosen dengan jumlah sks paling sedikit
5. Tampilkan data dosen dengan jumlah sks paling banyak
6. Tampilkan rata-rata sks yang diambil dosen
7. Tampilkan seluruh data dosen
8. Exit

Pilihan : 1
Kode Dosen :002
Nama Dosen :Restu
Jumlah sks :18
```

- c) Dosen ke-3

```
1. Input data dosen baru
2. Cari data dosen
3. Hapus data dosen
4. Tampilkan data dosen dengan jumlah sks paling sedikit
5. Tampilkan data dosen dengan jumlah sks paling banyak
6. Tampilkan rata-rata sks yang diambil dosen
7. Tampilkan seluruh data dosen
8. Exit

Pilihan : 1
Kode Dosen :003
Nama Dosen :Aji
Jumlah sks :24
```



b. Cari data dosen berdasarkan Nama

a) Dosen atas nama Noga

```
C:\Users\NOGA RESTU AJI\Documents\SEMESTER 2\C++\UAS-LINKED LIST DOSEN.exe
1. Input data dosen baru
2. Cari data dosen
3. Hapus data dosen
4. Tampilkan data dosen dengan jumlah sks paling sedikit
5. Tampilkan data dosen dengan jumlah sks paling banyak
6. Tampilkan rata-rata sks yang diambil dosen
7. Tampilkan seluruh data dosen
8. Exit

Pilihan : 2
Masukkan Nama Dosen yang ingin dicari: Noga
Noga
Dosen ditemukan!

|Kode Dosen |          Nama|  Sks
-----
|      1    |          Noga|   20

Pencarian Selesai
```

b) Dosen atas nama Restu

```
1. Input data dosen baru
2. Cari data dosen
3. Hapus data dosen
4. Tampilkan data dosen dengan jumlah sks paling sedikit
5. Tampilkan data dosen dengan jumlah sks paling banyak
6. Tampilkan rata-rata sks yang diambil dosen
7. Tampilkan seluruh data dosen
8. Exit

Pilihan : 2
Masukkan Nama Dosen yang ingin dicari: Restu
Restu
Dosen ditemukan!

|Kode Dosen |          Nama|  Sks
-----
|      2    |          Restu|   18

Pencarian Selesai
```

c. Hapus data dosen berdasarkan kodenya

```
1. Input data dosen baru
2. Cari data dosen
3. Hapus data dosen
4. Tampilkan data dosen dengan jumlah sks paling sedikit
5. Tampilkan data dosen dengan jumlah sks paling banyak
6. Tampilkan rata-rata sks yang diambil dosen
7. Tampilkan seluruh data dosen
8. Exit
```

Pilihan : 3

Kode Dosen yang ingin dihapus : 002

Dosen dengan kode 2 telah dihapus

Detail :

Kode Dosen	Nama	Sks
2	Restu	18

d. Menampilkan data dosen dengan jumlah sks paling sedikit

```
1. Input data dosen baru
2. Cari data dosen
3. Hapus data dosen
4. Tampilkan data dosen dengan jumlah sks paling sedikit
5. Tampilkan data dosen dengan jumlah sks paling banyak
6. Tampilkan rata-rata sks yang diambil dosen
7. Tampilkan seluruh data dosen
8. Exit
```

Pilihan : 4

Dosen dengan jumlah SKS tersedikit adalah :

Kode Pasien	Sks
3	20

e. Menampilkaan data dosen dengan jumlah sks terbanyak

```
3. Hapus data dosen
4. Tampilkan data dosen dengan jumlah sks paling sedikit
5. Tampilkan data dosen dengan jumlah sks paling banyak
6. Tampilkan rata-rata sks yang diambil dosen
7. Tampilkan seluruh data dosen
8. Exit
```

Pilihan : 5

Dosen dengan jumlah SKS terbanyak adalah :

Kode Pasien	Sks
3	24

- f. Tampilkan seluruh data dosen yang ada

```
C:\Users\NOGA RESTU AJI\Documents\SEMESTER 2\C++\UAS-LINKED LIST DOSEN.exe
1. Input data dosen baru
2. Cari data dosen
3. Hapus data dosen
4. Tampilkan data dosen dengan jumlah sks paling sedikit
5. Tampilkan data dosen dengan jumlah sks paling banyak
6. Tampilkan rata-rata sks yang diambil dosen
7. Tampilkan seluruh data dosen
8. Exit

Pilihan : 7
|Kode Dosen |          Nama|   Sks
-----
|      1     |      Noga   |    20
|      3     |      Aji    |    24
```