PRAKTIKUM

STRUKTUR DATA

SEMESTER GENAP TAHUN AKADEMIK 2024/2025

Tanggal

....

Materi

...

PRODI SISTEM INFORMASI FAKULTAS

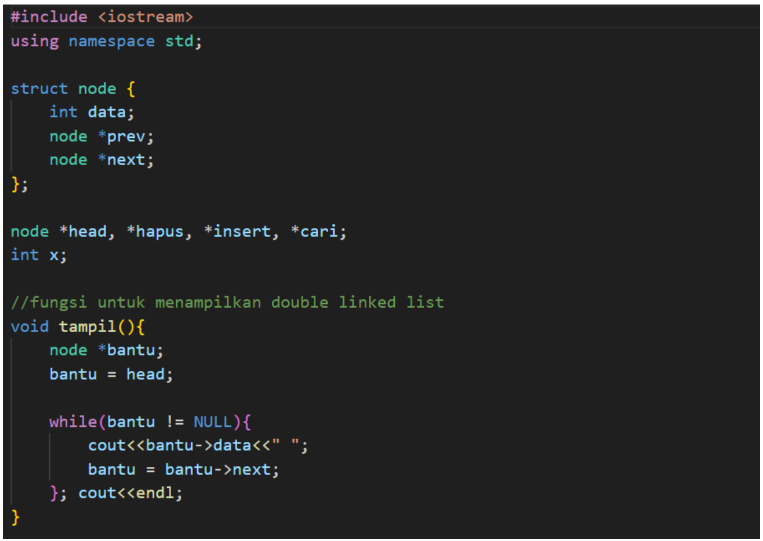
TEKNIK DAN ILMU KOMPUTER

UNIVERSITAS NUSANTARA PGRI KEDIRI 2025

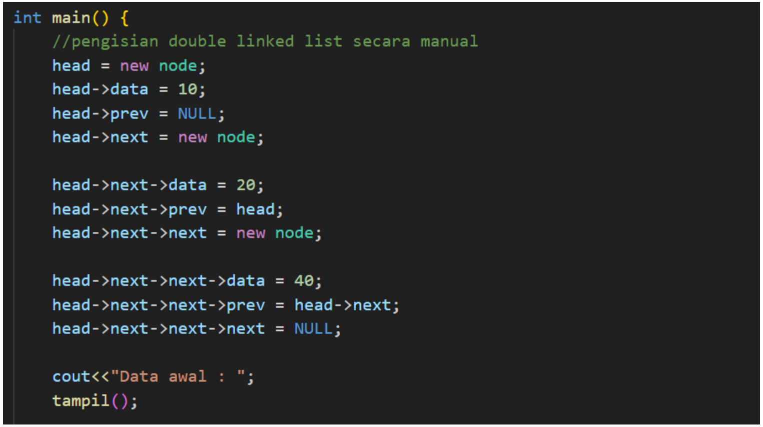
BAB II

PERCOBAAN DAN LATIHAN

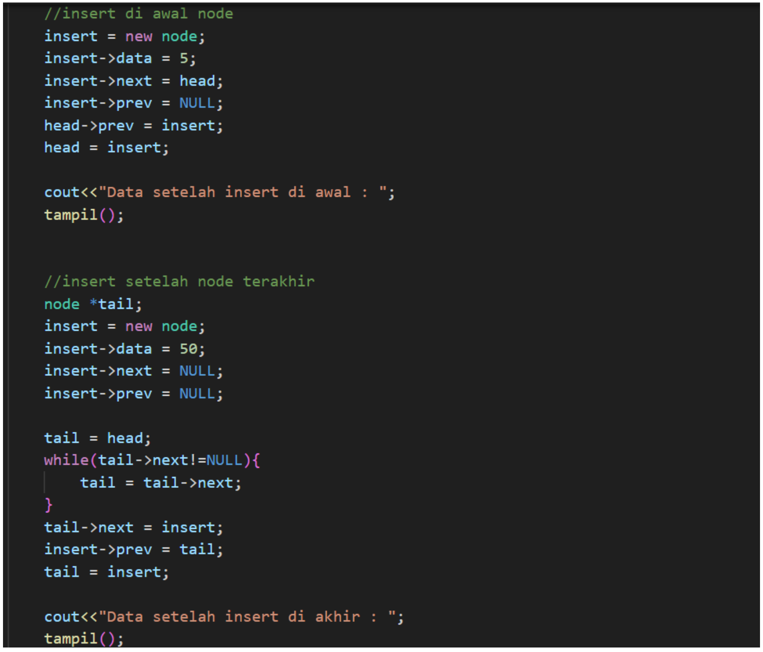
Percobaan 1



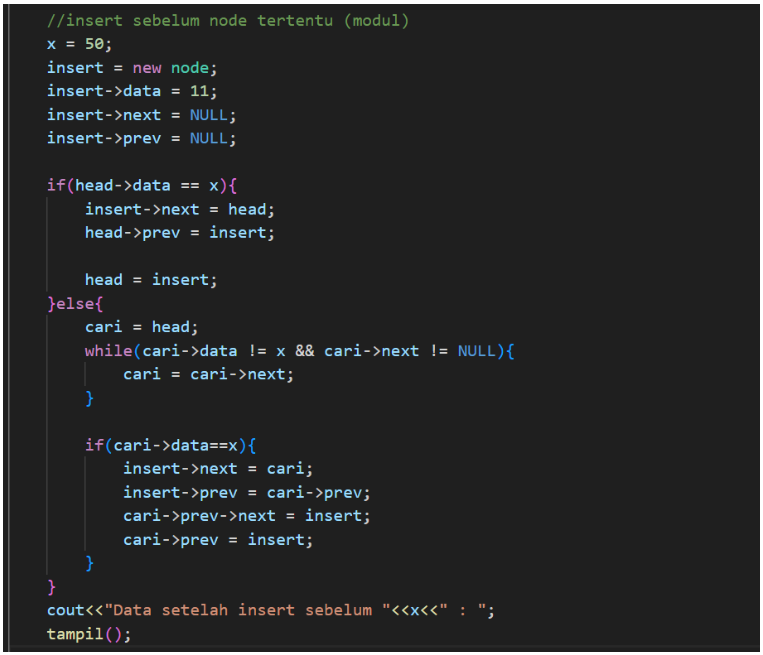
```
#include <iostream>
using namespace std;

struct node {
    int data;
    node *prev;
    node *next;
};

node *head, *hapus, *insert, *cari;
int x;

//fungsi untuk menampilkan double linked list
void tampil(){
    node *bantu;
    bantu = head;

    while(bantu != NULL){
        cout<<bantu->data<<" ";
        bantu = bantu->next;
    }; cout<<endl;
}
```



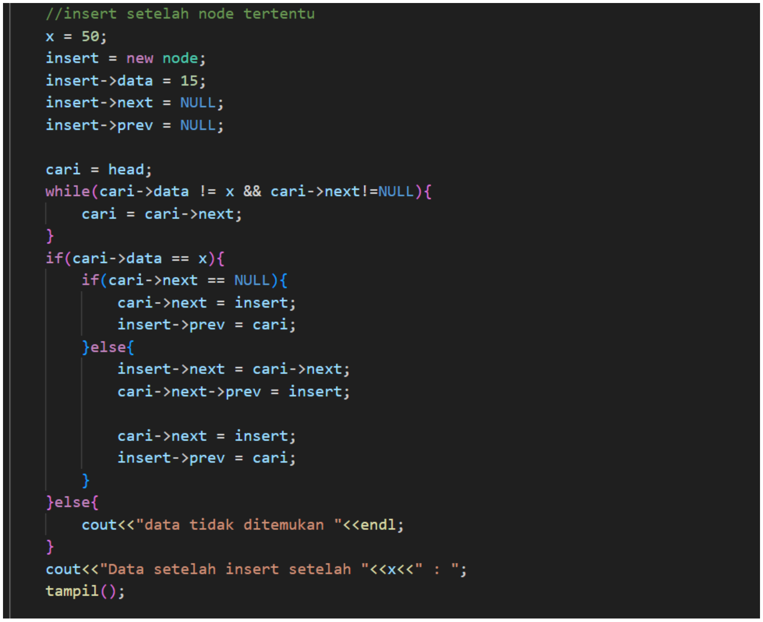
```
int main() {
    //pengisian double linked list secara manual
    head = new node;
    head->data = 10;
    head->prev = NULL;
    head->next = new node;

    head->next->data = 20;
    head->next->prev = head;
    head->next->next = new node;

    head->next->next->data = 40;
    head->next->next->prev = head->next;
    head->next->next->next = NULL;

    cout<<"Data awal : ";
    tampil();
```



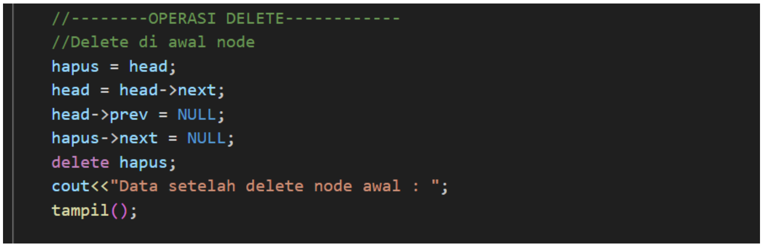
```
    //insert di awal node
    insert = new node;
    insert->data = 5;
    insert->next = head;
    insert->prev = NULL;
    head->prev = insert;
    head = insert;

    cout<<"Data setelah insert di awal : ";
    tampil();


    //insert setelah node terakhir
    node *tail;
    insert = new node;
    insert->data = 50;
    insert->next = NULL;
    insert->prev = NULL;

    tail = head;
    while(tail->next!=NULL){
        tail = tail->next;
    }
    tail->next = insert;
    insert->prev = tail;
    tail = insert;

    cout<<"Data setelah insert di akhir : ";
    tampil();
```



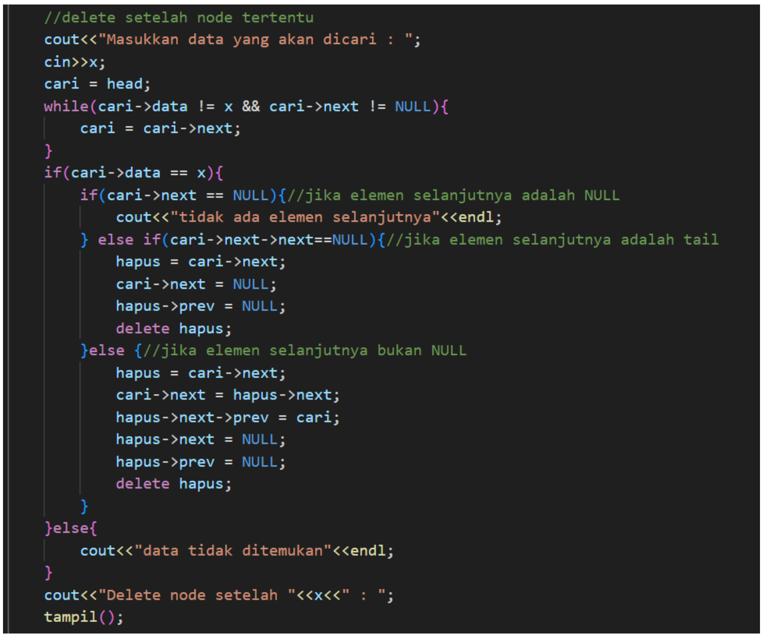
```
//insert sebelum node tertentu (modul)
x = 50;
insert = new node;
insert->data = 11;
insert->next = NULL;
insert->prev = NULL;

if(head->data == x){
    insert->next = head;
    head->prev = insert;

    head = insert;
}else{
    cari = head;
    while(cari->data != x && cari->next != NULL){
        cari = cari->next;
    }

    if(cari->data==x){
        insert->next = cari;
        insert->prev = cari->prev;
        cari->prev->next = insert;
        cari->prev = insert;
    }
}
cout<<"Data setelah insert sebelum "<<x<<" : ";
tampil();
```



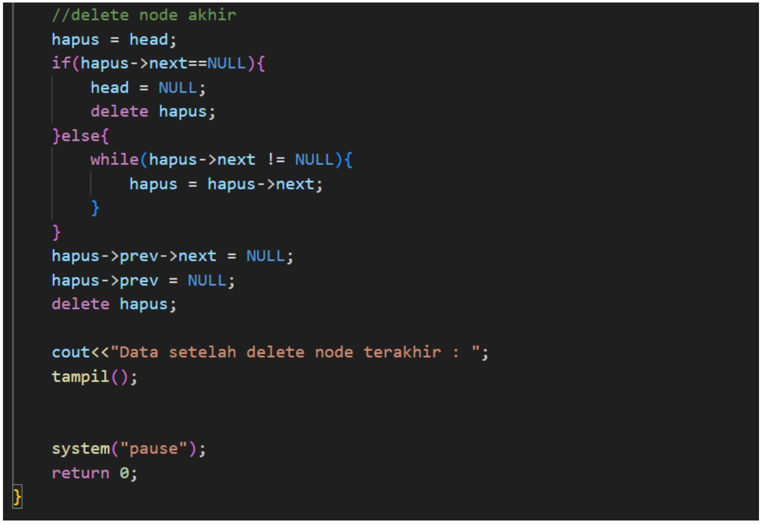
```
//insert setelah node tertentu
x = 50;
insert = new node;
insert->data = 15;
insert->next = NULL;
insert->prev = NULL;

cari = head;
while(cari->data != x && cari->next!=NULL){
    cari = cari->next;
}
if(cari->data == x){
    if(cari->next == NULL){
        cari->next = insert;
        insert->prev = cari;
    }else{
        insert->next = cari->next;
        cari->next->prev = insert;

        cari->next = insert;
        insert->prev = cari;
    }
}else{
    cout<<"data tidak ditemukan "<<endl;
}
cout<<"Data setelah insert setelah "<<x<<" : ";
tampil();
```



```
//--------OPERASI DELETE------------
//Delete di awal node
hapus = head;
head = head->next;
head->prev = NULL;
hapus->next = NULL;
delete hapus;
cout<<"Data setelah delete node awal : ";
tampil();
```



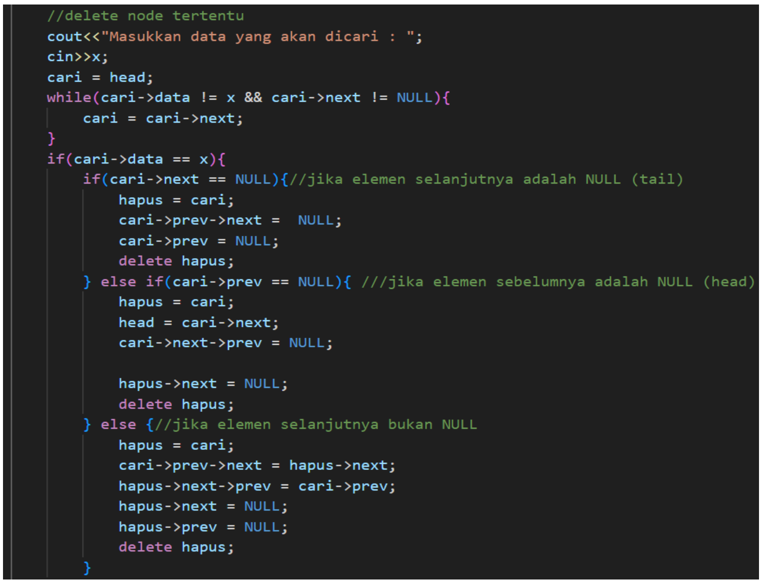
```
//delete setelah node tertentu
cout<<"Masukkan data yang akan dicari : ";
cin>>x;
cari = head;
while(cari->data != x && cari->next != NULL){
    cari = cari->next;
}
if(cari->data == x){
    if(cari->next == NULL){//jika elemen selanjutnya adalah NULL
        cout<<"tidak ada elemen selanjutnya"<<endl;
    } else if(cari->next->next==NULL){//jika elemen selanjutnya adalah tail
        hapus = cari->next;
        cari->next = NULL;
        hapus->prev = NULL;
        delete hapus;
    }else {//jika elemen selanjutnya bukan NULL
        hapus = cari->next;
        cari->next = hapus->next;
        hapus->next->prev = cari;
        hapus->next = NULL;
        hapus->prev = NULL;
        delete hapus;
    }
}else{
    cout<<"data tidak ditemukan"<<endl;
}
cout<<"Delete node setelah "<<x<<" : ";
tampil();
```



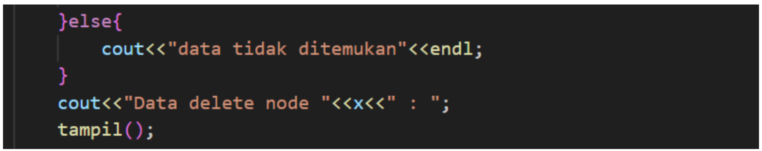
```
    //delete node akhir
    hapus = head;
    if(hapus->next==NULL){
        head = NULL;
        delete hapus;
    }else{
        while(hapus->next != NULL){
            hapus = hapus->next;
        }
    }
    hapus->prev->next = NULL;
    hapus->prev = NULL;
    delete hapus;

    cout<<"Data setelah delete node terakhir : ";
    tampil();


    system("pause");
    return 0;
}
```

Latihan 1

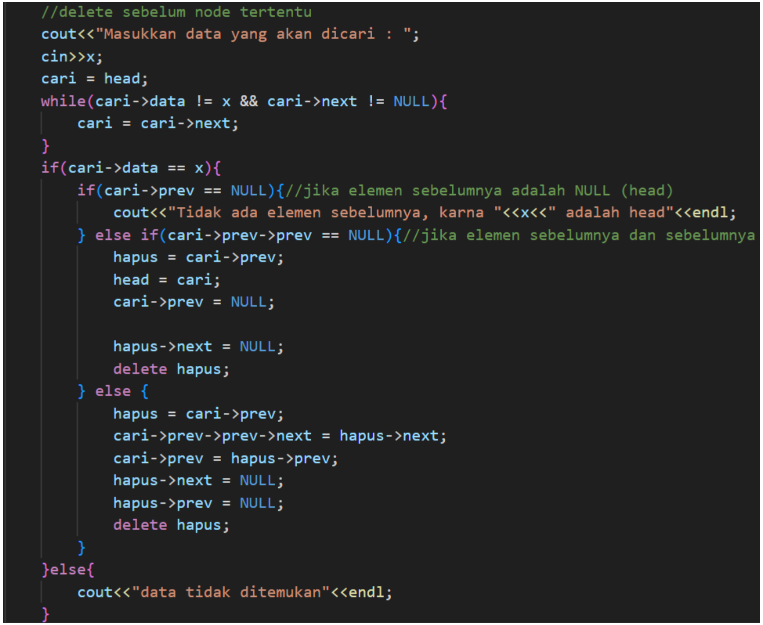
Lanjutan dari Program percobaan 1, penambahan operasi delete node tertentu dan delete sebelum node tertentu



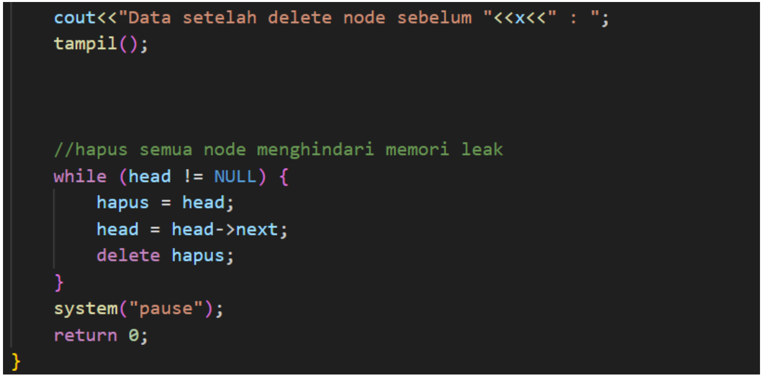
```
//delete node tertentu
cout<<"Masukkan data yang akan dicari : ";
cin>>x;
cari = head;
while(cari->data != x && cari->next != NULL){
    cari = cari->next;
}
if(cari->data == x){
    if(cari->next == NULL){//jika elemen selanjutnya adalah NULL (tail)
        hapus = cari;
        cari->prev->next =  NULL;
        cari->prev = NULL;
        delete hapus;
    } else if(cari->prev == NULL){ ///jika elemen sebelumnya adalah NULL (head)
        hapus = cari;
        head = cari->next;
        cari->next->prev = NULL;

        hapus->next = NULL;
        delete hapus;
    } else {//jika elemen selanjutnya bukan NULL
        hapus = cari;
        cari->prev->next = hapus->next;
        hapus->next->prev = cari->prev;
        hapus->next = NULL;
        hapus->prev = NULL;
        delete hapus;
    }
```



```
    }else{
        cout<<"data tidak ditemukan"<<endl;
    }
    cout<<"Data delete node "<<x<<" : ";
    tampil();
```



```
    //delete sebelum node tertentu
    cout<<"Masukkan data yang akan dicari : ";
    cin>>x;
    cari = head;
    while(cari->data != x && cari->next != NULL){
        cari = cari->next;
    }
    if(cari->data == x){
        if(cari->prev == NULL){//jika elemen sebelumnya adalah NULL (head)
            cout<<"Tidak ada elemen sebelumnya, karna "<<x<<" adalah head"<<endl;
        } else if(cari->prev->prev == NULL){//jika elemen sebelumnya dan sebelumnya
            hapus = cari->prev;
            head = cari;
            cari->prev = NULL;

            hapus->next = NULL;
            delete hapus;
        } else {
            hapus = cari->prev;
            cari->prev->prev->next = hapus->next;
            cari->prev = hapus->prev;
            hapus->next = NULL;
            hapus->prev = NULL;
            delete hapus;
        }
    }else{
        cout<<"data tidak ditemukan"<<endl;
    }
```

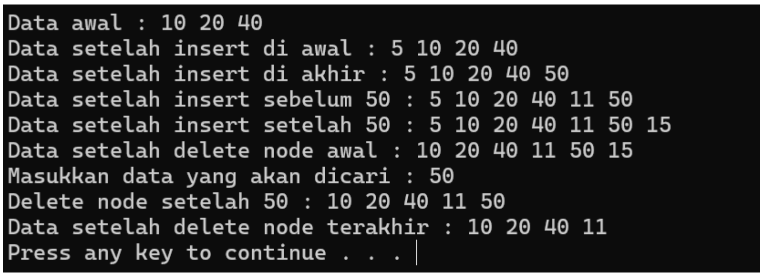


```
    cout<<"Data setelah delete node sebelum "<<x<<" : ";
    tampil();



    //hapus semua node menghindari memori leak
    while (head != NULL) {
        hapus = head;
        head = head->next;
        delete hapus;
    }
    system("pause");
    return 0;
}
```

BAB IV

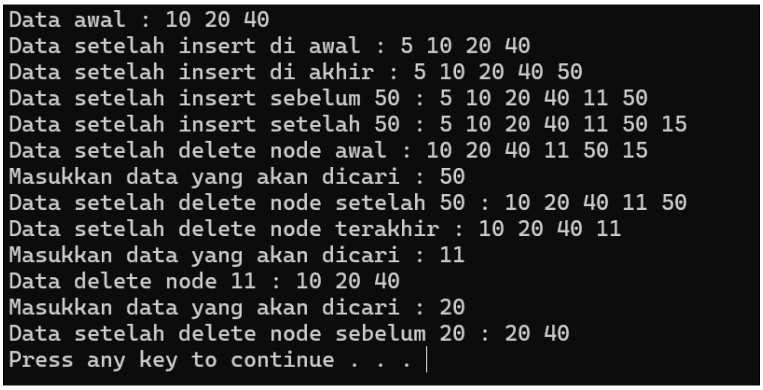
TAMPILAN PROGRAM

Percobaan 1



```
Data awal : 10 20 40
Data setelah insert di awal : 5 10 20 40
Data setelah insert di akhir : 5 10 20 40 50
Data setelah insert sebelum 50 : 5 10 20 40 11 50
Data setelah insert setelah 50 : 5 10 20 40 11 50 15
Data setelah delete node awal : 10 20 40 11 50 15
Masukkan data yang akan dicari : 50
Delete node setelah 50 : 10 20 40 11 50
Data setelah delete node terakhir : 10 20 40 11
Press any key to continue . . .
```

Latihan 1



```
Data awal : 10 20 40
Data setelah insert di awal : 5 10 20 40
Data setelah insert di akhir : 5 10 20 40 50
Data setelah insert sebelum 50 : 5 10 20 40 11 50
Data setelah insert setelah 50 : 5 10 20 40 11 50 15
Data setelah delete node awal : 10 20 40 11 50 15
Masukkan data yang akan dicari : 50
Data setelah delete node setelah 50 : 10 20 40 11 50
Data setelah delete node terakhir : 10 20 40 11
Masukkan data yang akan dicari : 11
Data delete node 11 : 10 20 40
Masukkan data yang akan dicari : 20
Data setelah delete node sebelum 20 : 20 40
Press any key to continue . . .
```