PRAKTIKUM

STRUKTUR DATA

SEMESTER GENAP TAHUN AKADEMIK 2024/2025

Tanggal

....

Materi

....

PRODI SISTEM INFORMASI FAKULTAS

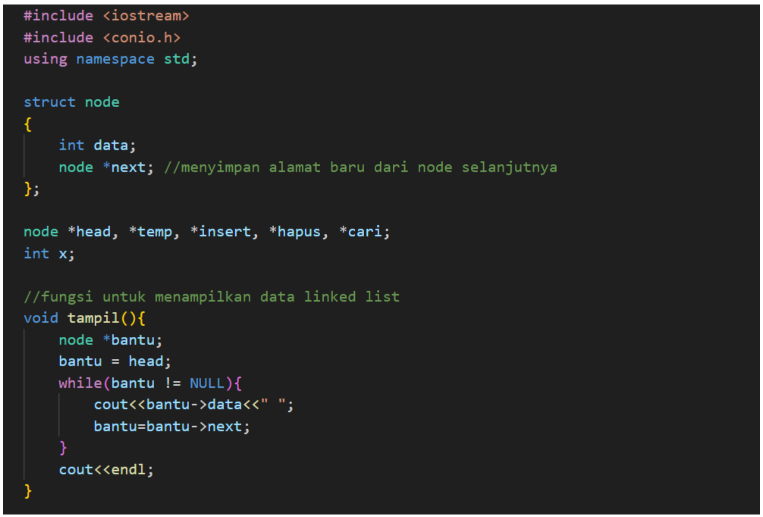
TEKNIK DAN ILMU KOMPUTER

UNIVERSITAS NUSANTARA PGRI KEDIRI 2025

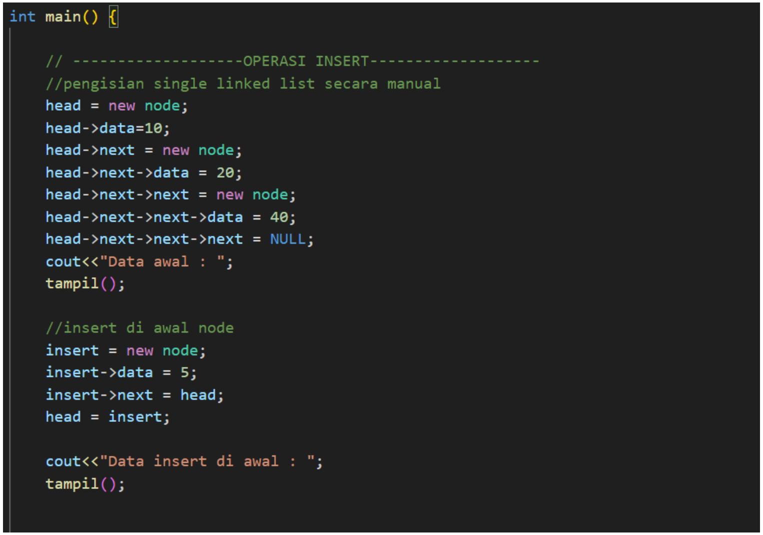
BAB II

PERCOBAAN DAN LATIHAN

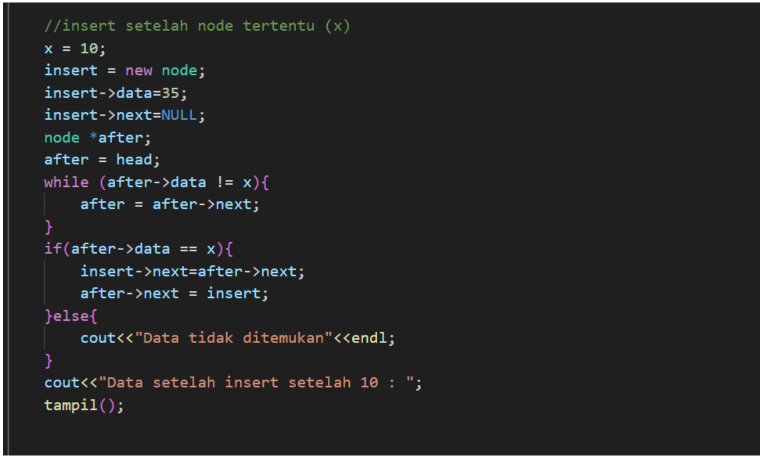
Percobaan 1



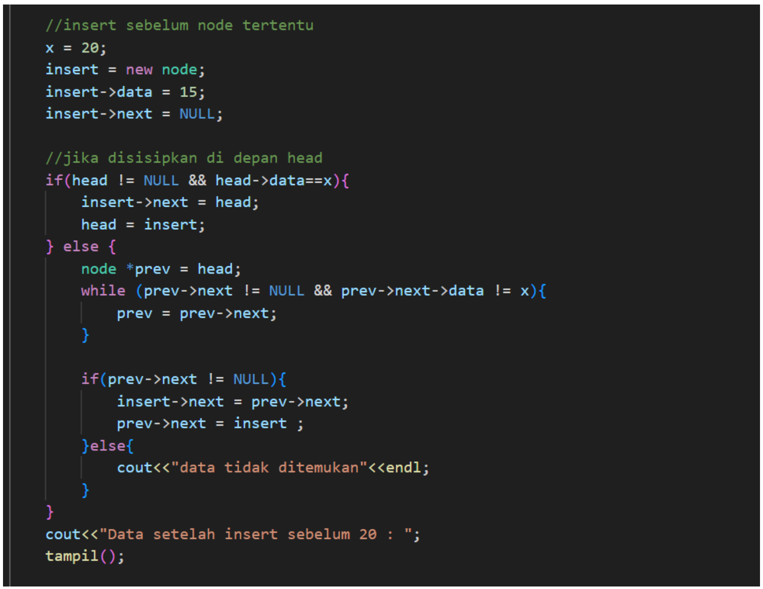
```
#include <iostream>
#include <conio.h>
using namespace std;

struct node
{
    int data;
    node *next; //menyimpan alamat baru dari node selanjutnya
};

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

//fungsi untuk menampilkan data linked list
void tampil(){
    node *bantu;
    bantu = head;
    while(bantu != NULL){
        cout<<bantu->data<<" ";
        bantu=bantu->next;
    }
    cout<<endl;
}
```



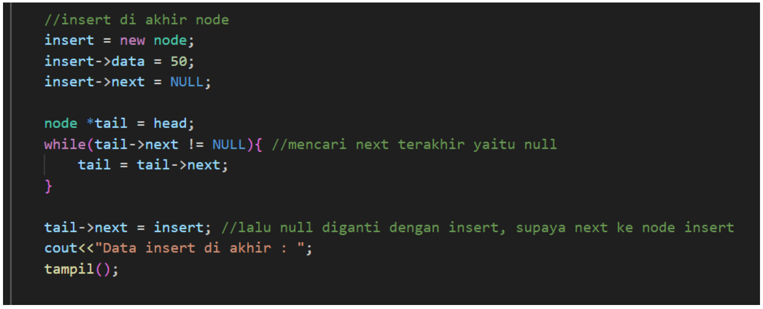
```
int main() {

    // -------------------OPERASI INSERT------------------
    //pengisian single linked list secara manual
    head = new node;
    head->data=10;
    head->next = new node;
    head->next->data = 20;
    head->next->next = new node;
    head->next->next->data = 40;
    head->next->next->next = NULL;
    cout<<"Data awal : ";
    tampil();

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

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



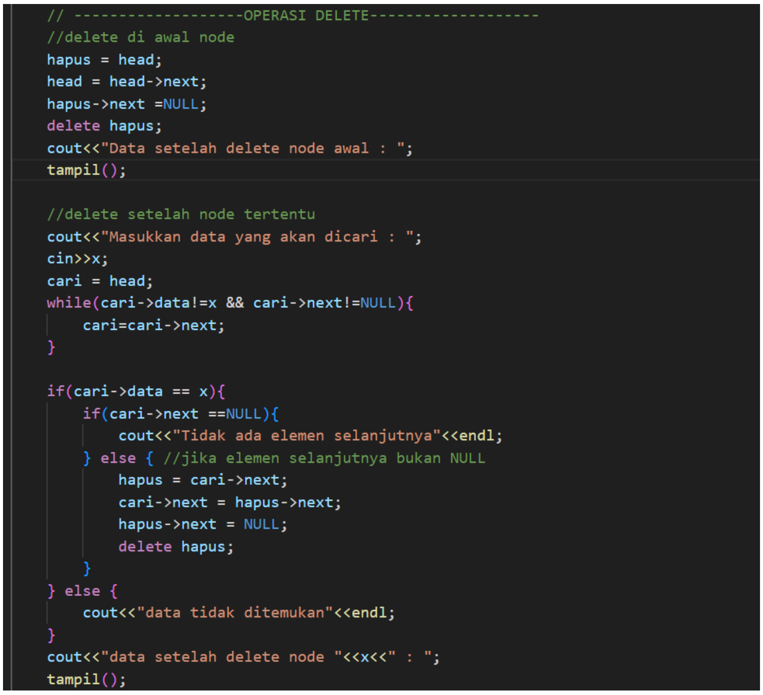
```
    //insert setelah node tertentu (x)
    x = 10;
    insert = new node;
    insert->data=35;
    insert->next=NULL;
    node *after;
    after = head;
    while (after->data != x){
        after = after->next;
    }
    if(after->data == x){
        insert->next=after->next;
        after->next = insert;
    }else{
        cout<<"Data tidak ditemukan"<<endl;
    }
    cout<<"Data setelah insert setelah 10 : ";
    tampil();
```



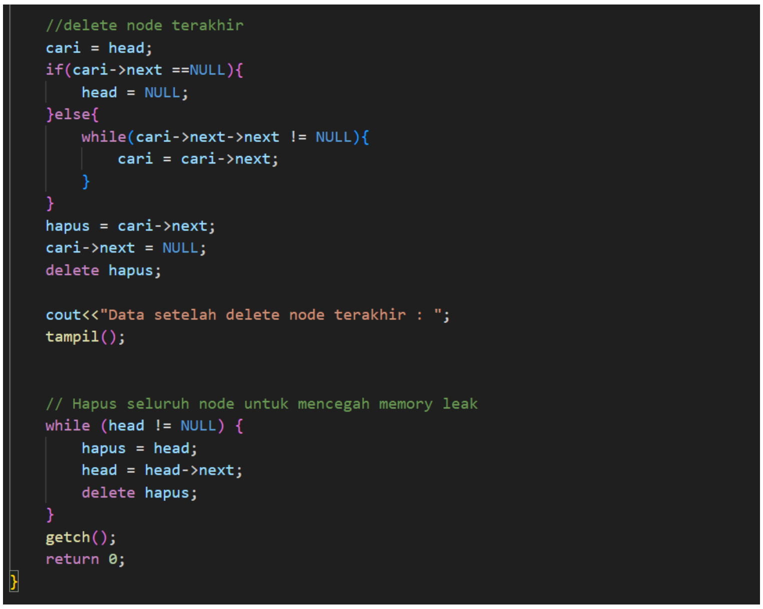
```
//insert sebelum node tertentu
x = 20;
insert = new node;
insert->data = 15;
insert->next = NULL;

//jika disisipkan di depan head
if(head != NULL && head->data==x){
    insert->next = head;
    head = insert;
} else {
    node *prev = head;
    while (prev->next != NULL && prev->next->data != x){
        prev = prev->next;
    }

    if(prev->next != NULL){
        insert->next = prev->next;
        prev->next = insert ;
    }else{
        cout<<"data tidak ditemukan"<<endl;
    }
}
cout<<"Data setelah insert sebelum 20 : ";
tampil();
```



```
//insert di akhir node
insert = new node;
insert->data = 50;
insert->next = NULL;

node *tail = head;
while(tail->next != NULL){ //mencari next terakhir yaitu null
    tail = tail->next;
}

tail->next = insert; //lalu null diganti dengan insert, supaya next ke node insert
cout<<"Data insert di akhir : ";
tampil();
```

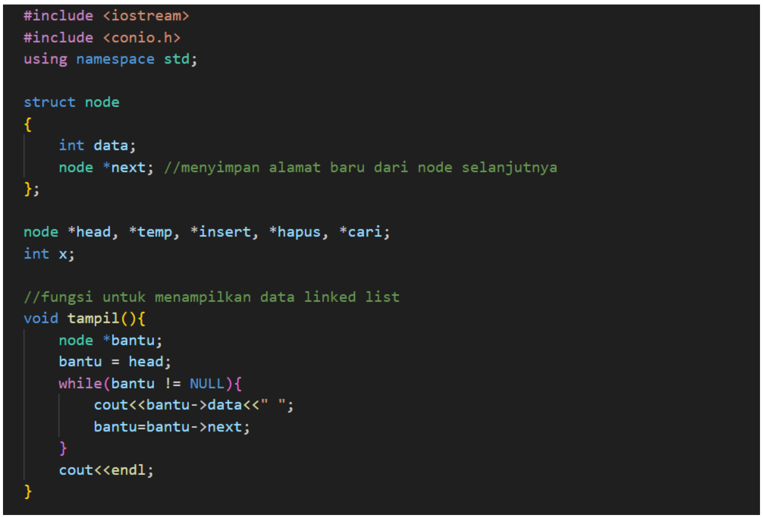


```
// -------------------OPERASI DELETE-------------------
//delete di awal node
hapus = head;
head = head->next;
hapus->next =NULL;
delete hapus;
cout<<"Data setelah delete node awal : ";
tampil();

//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){
        cout<<"Tidak ada elemen selanjutnya"<<endl;
    } else { //jika elemen selanjutnya bukan NULL
        hapus = cari->next;
        cari->next = hapus->next;
        hapus->next = NULL;
        delete hapus;
    }
} else {
    cout<<"data tidak ditemukan"<<endl;
}
cout<<"data setelah delete node "<<x<<" : ";
tampil();
```

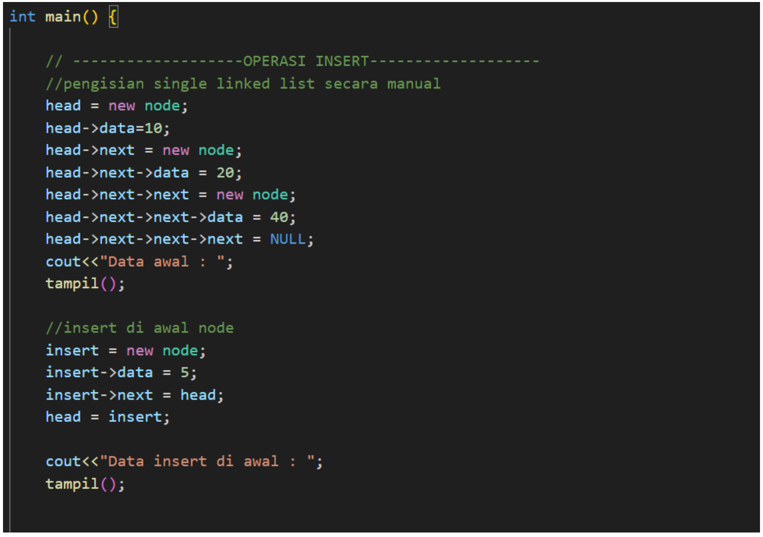


```cpp
    //delete node terakhir
    cari = head;
    if(cari->next ==NULL){
        head = NULL;
    }else{
        while(cari->next->next != NULL){
            cari = cari->next;
        }
    }
    hapus = cari->next;
    cari->next = NULL;
    delete hapus;

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


    // Hapus seluruh node untuk mencegah memory leak
    while (head != NULL) {
        hapus = head;
        head = head->next;
        delete hapus;
    }
    getch();
    return 0;
}
```

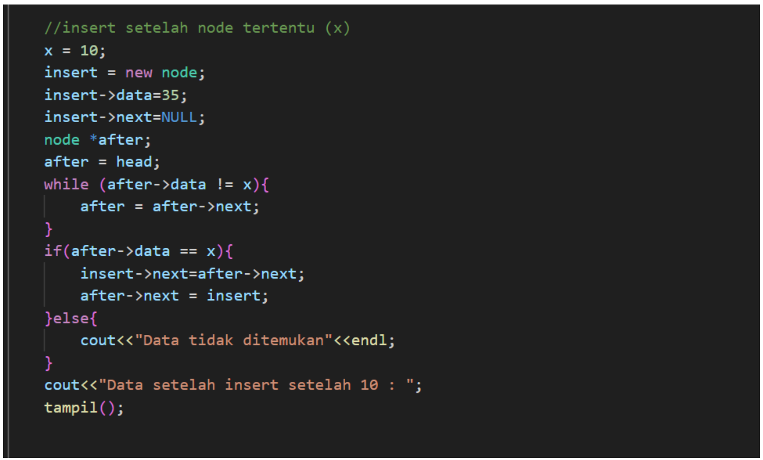
Laihan 1



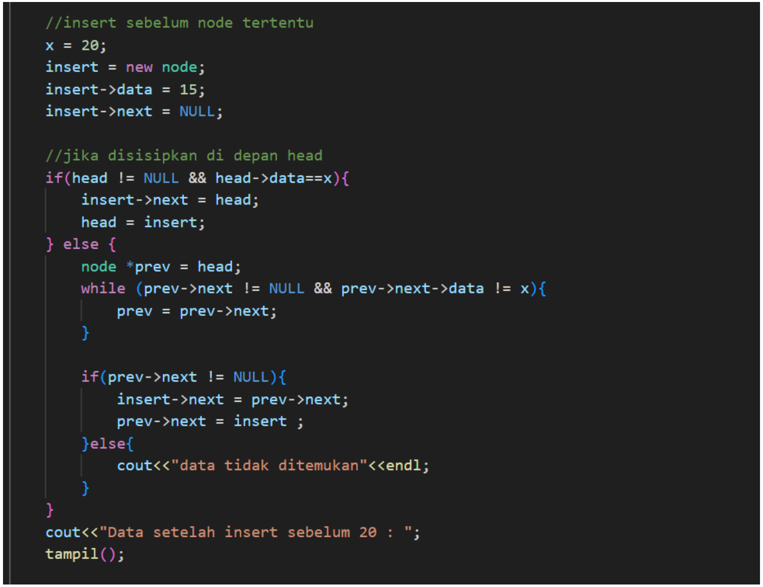
```cpp
#include <iostream>
#include <conio.h>
using namespace std;

struct node
{
    int data;
    node *next; //menyimpan alamat baru dari node selanjutnya
};

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

//fungsi untuk menampilkan data linked list
void tampil(){
    node *bantu;
    bantu = head;
    while(bantu != NULL){
        cout<<bantu->data<<" ";
        bantu=bantu->next;
    }
    cout<<endl;
}
```



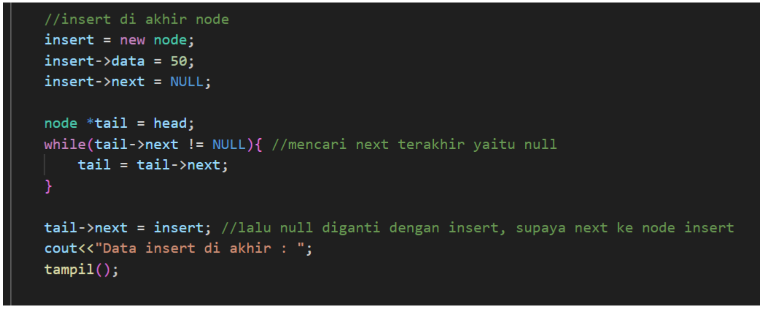
```cpp
int main() {

    // -------------------OPERASI INSERT-------------------
    //pengisian single linked list secara manual
    head = new node;
    head->data=10;
    head->next = new node;
    head->next->data = 20;
    head->next->next = new node;
    head->next->next->data = 40;
    head->next->next->next = NULL;
    cout<<"Data awal : ";
    tampil();

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

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



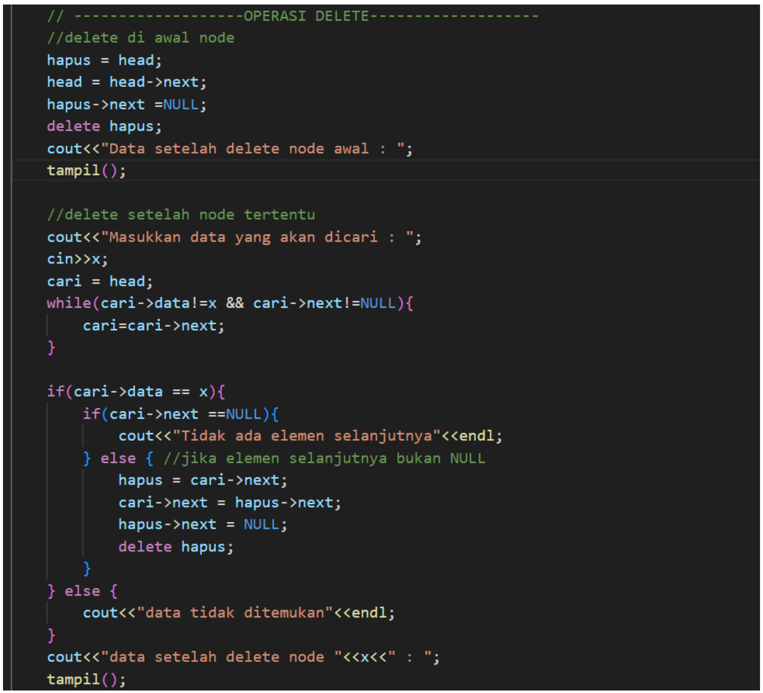
```
//insert setelah node tertentu (x)
x = 10;
insert = new node;
insert->data=35;
insert->next=NULL;
node *after;
after = head;
while (after->data != x){
    after = after->next;
}
if(after->data == x){
    insert->next=after->next;
    after->next = insert;
}else{
    cout<<"Data tidak ditemukan"<<endl;
}
cout<<"Data setelah insert setelah 10 : ";
tampil();
```



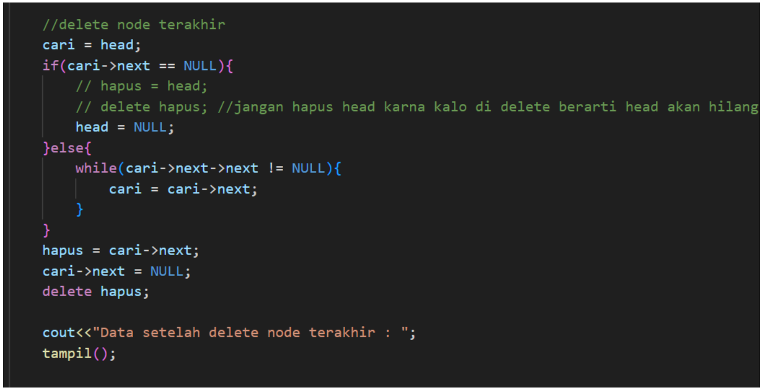
```
//insert sebelum node tertentu
x = 20;
insert = new node;
insert->data = 15;
insert->next = NULL;

//jika disisipkan di depan head
if(head != NULL && head->data==x){
    insert->next = head;
    head = insert;
} else {
    node *prev = head;
    while (prev->next != NULL && prev->next->data != x){
        prev = prev->next;
    }

    if(prev->next != NULL){
        insert->next = prev->next;
        prev->next = insert ;
    }else{
        cout<<"data tidak ditemukan"<<endl;
    }
}
cout<<"Data setelah insert sebelum 20 : ";
tampil();
```



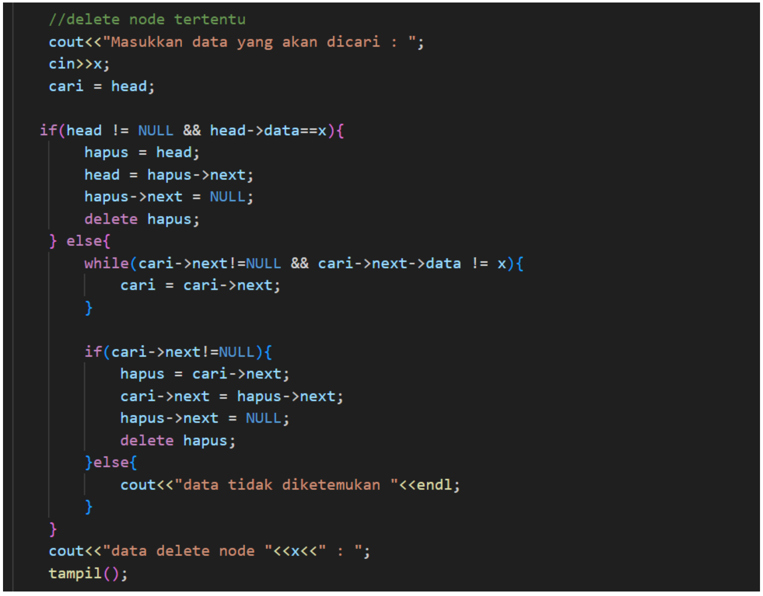
```
//insert di akhir node
insert = new node;
insert->data = 50;
insert->next = NULL;

node *tail = head;
while(tail->next != NULL){ //mencari next terakhir yaitu null
    tail = tail->next;
}

tail->next = insert; //lalu null diganti dengan insert, supaya next ke node insert
cout<<"Data insert di akhir : ";
tampil();
```



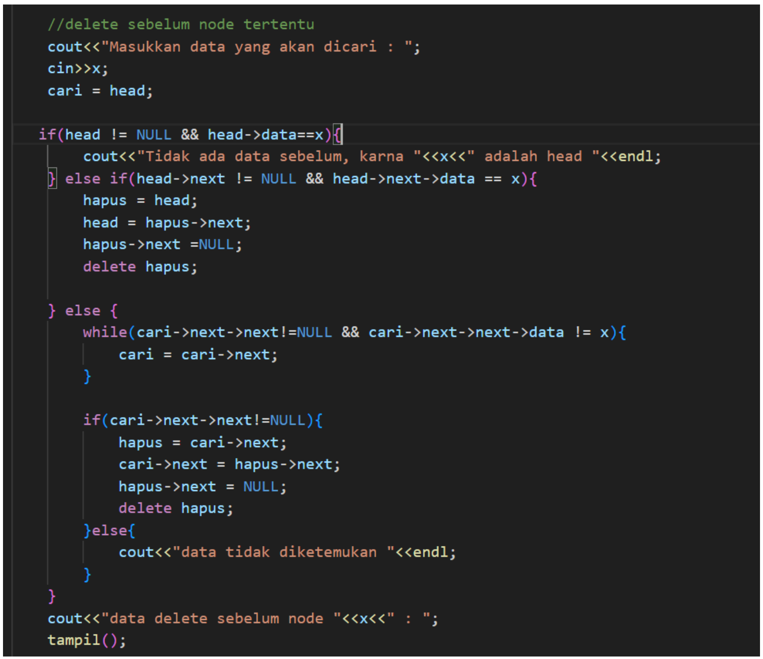
```
// -------------------OPERASI DELETE-------------------
//delete di awal node
hapus = head;
head = head->next;
hapus->next =NULL;
delete hapus;
cout<<"Data setelah delete node awal : ";
tampil();

//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){
        cout<<"Tidak ada elemen selanjutnya"<<endl;
    } else { //jika elemen selanjutnya bukan NULL
        hapus = cari->next;
        cari->next = hapus->next;
        hapus->next = NULL;
        delete hapus;
    }
} else {
    cout<<"data tidak ditemukan"<<endl;
}
cout<<"data setelah delete node "<<x<<" : ";
tampil();
```



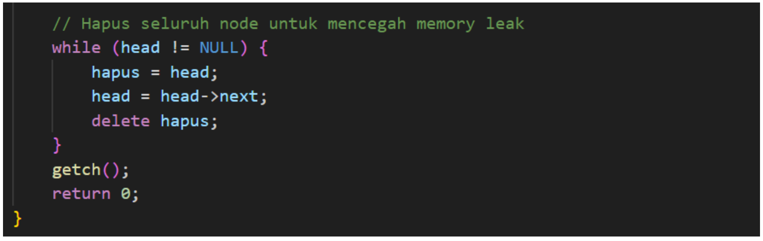
```
//delete node terakhir
cari = head;
if(cari->next == NULL){
    // hapus = head;
    // delete hapus; //jangan hapus head karna kalo di delete berarti head akan hilang
    head = NULL;
}else{
    while(cari->next->next != NULL){
        cari = cari->next;
    }
}
hapus = cari->next;
cari->next = NULL;
delete hapus;

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



```
 //delete node tertentu
 cout<<"Masukkan data yang akan dicari : ";
 cin>>x;
 cari = head;

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

     if(cari->next!=NULL){
         hapus = cari->next;
         cari->next = hapus->next;
         hapus->next = NULL;
         delete hapus;
     }else{
         cout<<"data tidak diketemukan "<<endl;
     }
 }
 cout<<"data delete node "<<x<<" : ";
 tampil();
```



```
    //delete sebelum node tertentu
    cout<<"Masukkan data yang akan dicari : ";
    cin>>x;
    cari = head;

   if(head != NULL && head->data==x){
        cout<<"Tidak ada data sebelum, karna "<<x<<" adalah head "<<endl;
    } else if(head->next != NULL && head->next->data == x){
        hapus = head;
        head = hapus->next;
        hapus->next =NULL;
        delete hapus;

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

        if(cari->next->next!=NULL){
            hapus = cari->next;
            cari->next = hapus->next;
            hapus->next = NULL;
            delete hapus;
        }else{
            cout<<"data tidak diketemukan "<<endl;
        }
    }
    cout<<"data delete sebelum node "<<x<<" : ";
    tampil();
```

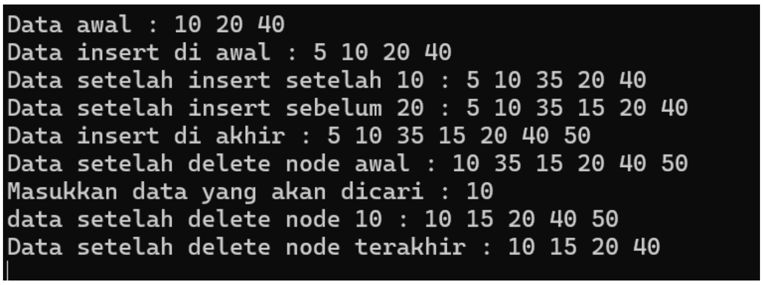


```
    // Hapus seluruh node untuk mencegah memory leak
    while (head != NULL) {
        hapus = head;
        head = head->next;
        delete hapus;
    }
    getch();
    return 0;
}
```

BAB IV

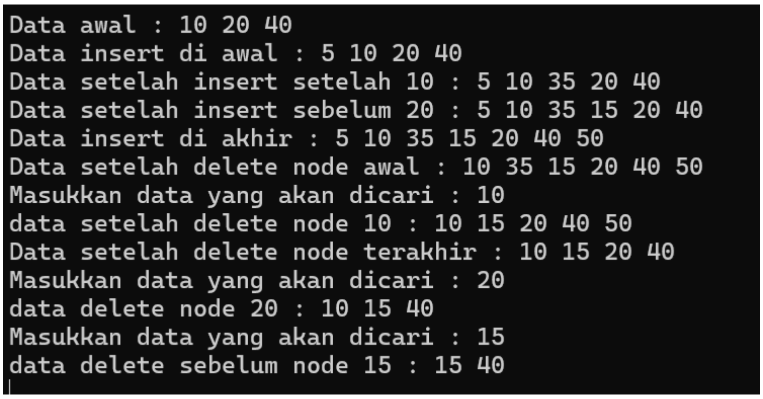
TAMPILAN PROGRAM

Percobaan 1



```
Data awal : 10 20 40
Data insert di awal : 5 10 20 40
Data setelah insert setelah 10 : 5 10 35 20 40
Data setelah insert sebelum 20 : 5 10 35 15 20 40
Data insert di akhir : 5 10 35 15 20 40 50
Data setelah delete node awal : 10 35 15 20 40 50
Masukkan data yang akan dicari : 10
data setelah delete node 10 : 10 15 20 40 50
Data setelah delete node terakhir : 10 15 20 40
```

Latihan 1



```
Data awal : 10 20 40
Data insert di awal : 5 10 20 40
Data setelah insert setelah 10 : 5 10 35 20 40
Data setelah insert sebelum 20 : 5 10 35 15 20 40
Data insert di akhir : 5 10 35 15 20 40 50
Data setelah delete node awal : 10 35 15 20 40 50
Masukkan data yang akan dicari : 10
data setelah delete node 10 : 10 15 20 40 50
Data setelah delete node terakhir : 10 15 20 40
Masukkan data yang akan dicari : 20
data delete node 20 : 10 15 40
Masukkan data yang akan dicari : 15
data delete sebelum node 15 : 15 40
```