Praktikum

Linked List

# NIM/Nama : 3311401066/Firchan Risanto, 3311401047/Dwi Teguh Prasetyo

|  |
| --- |
| Setelah mengikuti praktikum ini, mahasiswa diharapkan dapat:   * Menggambarkan konsep linked list pada memory * Membuat struktur data linked list dalam bahasa C * Melakukan operasi pada linked list * Membedakan macam-macam struktur linked list dan operasiny * Mengaplikasikan linked list pada program |

1. Buatlah fungsi/prosedur untuk menampilkan semua data di suatu linked list.

void tampil\_isi\_linked\_list(){

Simpul \*Q;

Q=HEAD;

if (HEAD != NULL)

{

do{

printf("Q = %d \n\n", Q->data);

Q=Q->next;

}while(Q != NULL);

}

else

printf("Linked list belum ada \n\n");

}

Berdasarkan deklarasi struktur dan fungsi-fungsi yang telah diberikan di slide teori Linked List, buatlah program yang dapat menampilkan menu aneka macam operasi insert dan delete pada linked list, dan menjalankan operasi sesuai menu yang dipilih. Gunakan fungsi tampil isi linked list yang telah dibuat sebelumnya untuk mengecek kebenaran operasi yang dilakukan. Menu akan terus ditampilkan dan diproses selama pengguna belum memilih menu untuk keluar program.

Salin kode C Anda di sini.

#include <stdio.h>

#include <stdlib.h>

struct Node{

int data;

struct Node \*next; };

typedef struct Node Simpul;

Simpul \*P, \*HEAD, \*TAIL;

void tampil\_isi\_linked\_list(){

Simpul \*Q;

Q=HEAD;

if (HEAD != NULL)

{

do{

printf("Q = %d \n\n", Q->data);

Q=Q->next;

}while(Q != NULL);

}

else

printf("Linked list belum ada \n\n");

}

void buatSimpul(int X)

{

P = (Simpul \*) malloc(sizeof(Simpul));

if (P != NULL)

{

P->data = X;

} else

printf("Pembuatan simpul gagal. \n");

}

void awal()

{

if (HEAD == NULL)

{

HEAD = P;

TAIL = P;

P->next = NULL;

}

else

printf("Linked List sudah ada");

}

void insertBelakang()

{

TAIL->next = P;

TAIL = P;

P->next = NULL;

}

void insertDepan()

{

if (HEAD != NULL)

{

P->next = HEAD;

HEAD = P;

}

else

printf("Linked list belum ada");

}

void insertSetelah(int X)

{

Simpul \*Q;

Q = HEAD;

while (Q->data != X){

Q = Q->next;

}

P->next = Q->next;

Q->next = P;

}

void insertSebelum(int X)

{

Simpul \*Q, \*sebelumQ;

if (HEAD->data == X)

insertDepan();

else

{

Q = HEAD;

do{

sebelumQ = Q;

Q = Q->next;}

while (Q->data != X);

P->next = Q;

sebelumQ->next = P;

}

}

void deleteDepan()

{

Simpul \*Q;

Q = HEAD;

HEAD = Q->next;

free(Q);

}

void deleteBelakang()

{

Simpul \*Q;

Q = HEAD;

while(Q->next != TAIL)

Q = Q->next;

free(TAIL);

TAIL = Q;

Q->next = NULL;

}

void deleteSimpul(int X)

{

Simpul \*Q, \*sebelumQ;

if (HEAD == NULL)

printf("Linked list kosong\n");

else if (HEAD->data == X){

deleteDepan();

} else

{

Q = HEAD;

do {

sebelumQ = Q;

Q = Q->next;

} while (Q->data != X);

sebelumQ->next = Q->next;

free(Q);

}

}

main(){

int m,X;

do {

printf ("Insert Linked List \n");

printf ("1. Buat Simpul \n");

printf ("2. Buat Linked List \n");

printf ("3. Insert Depan \n");

printf ("4. Insert Belakang \n");

printf ("5. Insert Sebelum \n");

printf ("6. Insert Setelah \n");

printf ("Delete Linked List \n");

printf ("7. Delete Depan \n");

printf ("8. Delete Belakang \n");

printf ("9. Delete Simpul \n");

printf ("10. Tampilkan Linked List \n");

printf ("0. Keluar \n");

printf ("\n Masukkan Kode Menu : ");

scanf ("%d",&m);

switch (m)

{

case 1 : printf ("\n Masukkan Data Simpul : "); scanf ("%d",&X);

buatSimpul(X);

system("pause");

break ;

case 2 : awal();

system("pause");

break ;

case 3 : insertDepan();

system("pause");

break ;

case 4 : insertBelakang();

system("pause");

break;

case 5 : printf ("\n Masukkan Data Simpul : "); scanf ("%d",&X);

insertSebelum(X);

system("pause");

break ;

case 6 : printf ("\n Masukkan Data Simpul : "); scanf ("%d",&X);

insertSetelah(X);

system("pause");

break ;

case 7 : deleteDepan();

break ;

case 8 : deleteBelakang();

system("pause");

break ;

case 9 : printf ("\n Masukkan Data Simpul : "); scanf ("%d",&X);

deleteSimpul(X);

system("pause");

break ;

case 10 : tampil\_isi\_linked\_list();

system("pause");

break ;

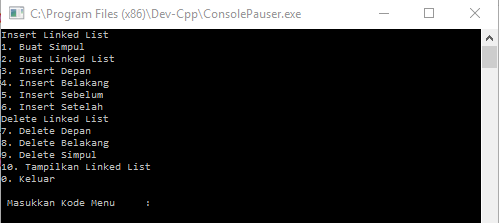
}

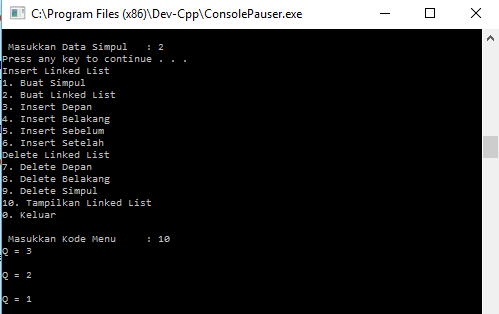
}while(m!=0);

return(0);

}

Tampilkan dan jelaskan hasil running program di sini.





1. Buatlah deklarasi struktur simpul linked list yang dapat menyimpan data NIM dan Nama mahasiswa di setiap simpulnya.

struct Mhs{

int nim;

char nama[100];

struct Mhs \*next; };

typedef struct Mhs Simpul;

Modifikasi kode dari hasil soal nomor 1 sehingga dapat memproses data mahasiswa berupa NIM dan Nama.

Salin kode C Anda di sini. Jelaskan perubahan-perubahan yang dilakukan.

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct Mhs{

int nim;

char nama[100];

struct Mhs \*next; };

typedef struct Mhs Simpul;

Simpul \*P, \*HEAD, \*TAIL;

void tampil\_isi\_linked\_list(){

Simpul \*Q;

Q=HEAD;

if (HEAD != NULL)

{

do{

printf("Nim = %d \n", Q->nim);

printf("Nama = %s \n\n", Q->nama);

Q=Q->next;

}while(Q != NULL);

}

else

printf("Linked list belum ada \n\n");

}

void buatSimpul(int X, char Y[100])

{

P = (Simpul \*) malloc(sizeof(Simpul));

if (P != NULL)

{

P->nim = X;

strcpy(P->nama, Y);

} else

printf("Pembuatan simpul gagal. \n");

}

void awal()

{

if (HEAD == NULL)

{

HEAD = P;

TAIL = P;

P->next = NULL;

}

else

printf("Linked List sudah ada");

}

void insertBelakang()

{

TAIL->next = P;

TAIL = P;

P->next = NULL;

}

void insertDepan()

{

if (HEAD != NULL)

{

P->next = HEAD;

HEAD = P;

}

else

printf("Linked list belum ada");

}

void insertSetelah(int X)

{

Simpul \*Q;

Q = HEAD;

while (Q->nim != X){

Q = Q->next;

}

P->next = Q->next;

Q->next = P;

}

void insertSebelum(int X)

{

Simpul \*Q, \*sebelumQ;

if (HEAD->nim == X)

insertDepan();

else

{

Q = HEAD;

do{

sebelumQ = Q;

Q = Q->next;}

while (Q->nim != X);

P->next = Q;

sebelumQ->next = P;

}

}

void deleteDepan()

{

Simpul \*Q;

Q = HEAD;

HEAD = Q->next;

free(Q);

}

void deleteBelakang()

{

Simpul \*Q;

Q = HEAD;

while(Q->next != TAIL)

Q = Q->next;

free(TAIL);

TAIL = Q;

Q->next = NULL;

}

void deleteSimpul(int X)

{

Simpul \*Q, \*sebelumQ;

if (HEAD == NULL)

printf("Linked list kosong\n");

else if (HEAD->nim == X){

deleteDepan();

} else

{

Q = HEAD;

do {

sebelumQ = Q;

Q = Q->next;

} while (Q->nim != X);

sebelumQ->next = Q->next;

free(Q);

}

}

main(){

int m,X;

char Y[100];

do {

printf ("Insert Linked List \n");

printf ("1. Buat Simpul \n");

printf ("2. Buat Linked List \n");

printf ("3. Insert Depan \n");

printf ("4. Insert Belakang \n");

printf ("5. Insert Sebelum \n");

printf ("6. Insert Setelah \n");

printf ("Delete Linked List \n");

printf ("7. Delete Depan \n");

printf ("8. Delete Belakang \n");

printf ("9. Delete Simpul \n");

printf ("10. Tampilkan Linked List \n");

printf ("0. Keluar \n");

printf ("\n Masukkan Kode Menu : ");

scanf ("%d",&m);

switch (m)

{

case 1 : printf ("\n Masukkan NIM : "); scanf ("%d",&X);

printf ("\n Masukkan Nama : "); scanf ("%s",&Y);

buatSimpul(X,Y);

system("pause");

break ;

case 2 : awal();

system("pause");

break ;

case 3 : insertDepan();

system("pause");

break ;

case 4 : insertBelakang();

system("pause");

break;

case 5 : printf ("\n Masukkan Data Simpul : "); scanf ("%d",&X);

insertSebelum(X);

system("pause");

break ;

case 6 : printf ("\n Masukkan Data Simpul : "); scanf ("%d",&X);

insertSetelah(X);

system("pause");

break ;

case 7 : deleteDepan();

break ;

case 8 : deleteBelakang();

system("pause");

break ;

case 9 : printf ("\n Masukkan Data Simpul : "); scanf ("%d",&X);

deleteSimpul(X);

system("pause");

break ;

case 10 : tampil\_isi\_linked\_list();

system("pause");

break ;

}

}while(m!=0);

return(0);

}

Tampilkan dan jelaskan hasil running program di sini.

