Praktikum Algoritma & Struktur Data

Prak 3.1. Single Linked List: SLL – Delete

Dosen Pengampu

Dr. Tita Karlita S.Kom, M.Kom

Disusun Oleh :

Nama : M. Faza Nur Husain

Nrp : 3121550004

D3 PJJ AK TEKNIK INFORMATIKA

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

TAHUN AKADEMIK 2021/2022

Mengerjakan soal:

1. Delete Awal
2. Delete Akhir
3. Delete Node Tertentu
4. Menu Insert Delete

1. Delete Awal

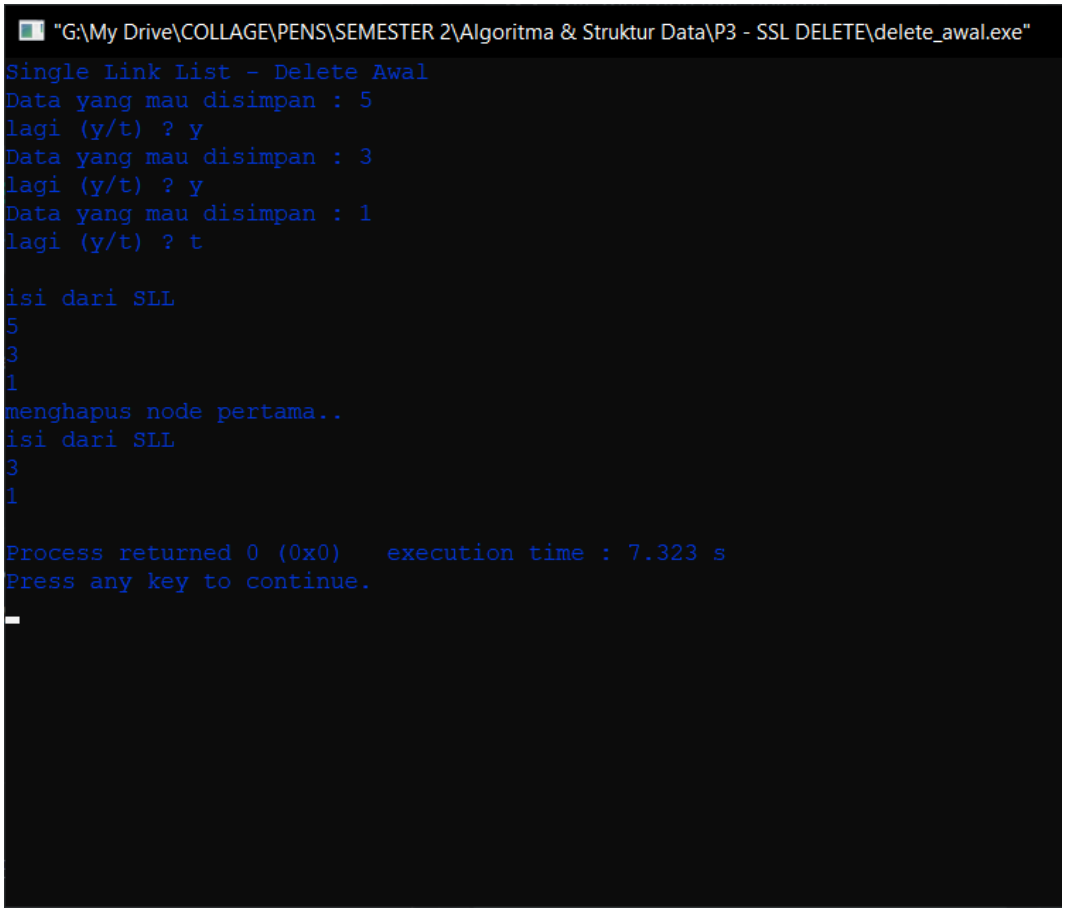
Source Code :

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

typedef struct simpul Node;
struct simpul{
     int data;
     Node *next;
};

Node *head=NULL, *p;

void alokasi();
void awal();
void tampil();
void hapus_awal();
void bebaskan(Node *);

int main()
{
     char jwb;
     puts("Single Link List - Delete Awal");
     do {
          fflush(stdin);
          alokasi();
          akhir();
          fflush(stdin);
          printf("lagi (y/t) ? ");
          jwb = getchar();
     }while((jwb == 'y')||(jwb == 'Y'));
    puts("");
     tampil();
```

```
    puts("menghapus node pertama..");
    hapus_awal();
    tampil();
     return 0;
}

void bebaskan(Node *x){
    free(x);
    x = NULL;
}

void hapus_awal(){
    Node *hapus;

    hapus = head;
    if (hapus -> next == NULL)
        head = NULL;
    else{
        head = hapus->next;
        bebaskan(hapus);
    }

}


void akhir(){
    Node *tail;

    if(head == NULL)
        head = p;
    else{
        tail = head;
        while(tail->next != NULL)
            tail=tail->next;
        tail->next = p;
        tail=tail->next;
    }
}

void tampil(){
     Node *baca;

     puts("isi dari SLL");
     baca = head;
     while(baca !=NULL){
          printf("%d\n", baca->data);
          baca = baca->next;
     }
}
```

```
void alokasi(){
     int x;
     printf("Data yang mau disimpan : ");
     scanf("%d", &x);

     p = (Node *) malloc(sizeof(Node));
     if(p==NULL){
          puts("alokasi gagal");
          exit(0);
     }else{
          p->data = x;
          p->next = NULL;
     }
}

void awal(){
     if(head != NULL)
          p->next = head;
     head = p;
}
```

Output :



2. Delete Akhir

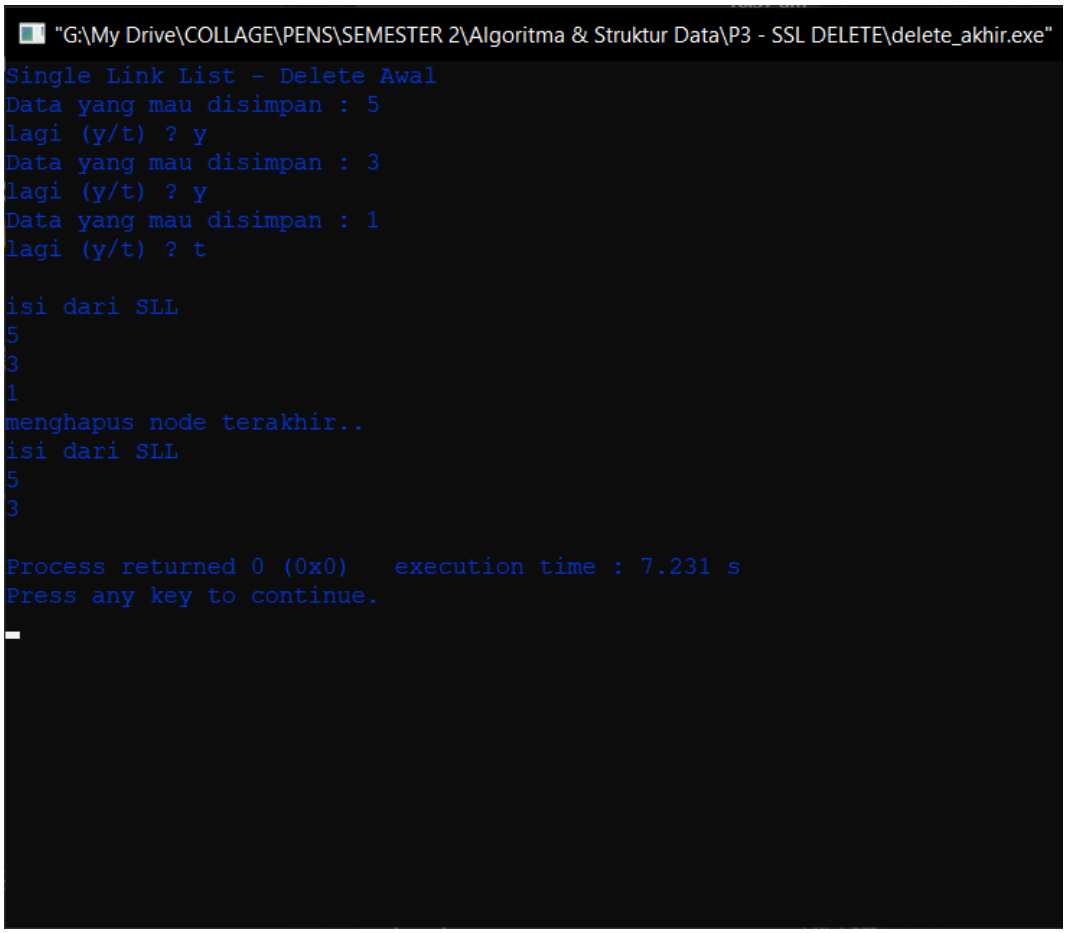
Source Code :

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

typedef struct simpul Node;
struct simpul{
     int data;
     Node *next;
};

Node *head=NULL, *p;

void alokasi();
void awal();
void tampil();
void bebaskan(Node *);
void hapus_akhir();

int main()
{
     char jwb;
     puts("Single Link List - Delete Akhir");
     do {
          fflush(stdin);
          alokasi();
          akhir();
          fflush(stdin);
          printf("lagi (y/t) ? ");
          jwb = getchar();
     }while((jwb == 'y')||(jwb == 'Y'));
    puts("");
     tampil();
    puts("menghapus node terakhir..");
    hapus_akhir();
    tampil();
     return 0;
}

void bebaskan(Node *x){
    free(x);
    x = NULL;
}

void hapus_akhir(){
     Node *hapus, *phapus;
```

```
     hapus = head;
     if(hapus->next==NULL)
          head = NULL;
     else{
          while (hapus->next != NULL)
          {
               phapus = hapus;
               hapus = hapus -> next;
          }
          phapus -> next = NULL;
     }
     bebaskan(hapus);
}



void akhir(){
    Node *tail;

    if(head == NULL)
        head = p;
    else{
        tail = head;
        while(tail->next != NULL)
            tail=tail->next;
        tail->next = p;
        tail=tail->next;
    }
}

void tampil(){
     Node *baca;

     puts("isi dari SLL");
     baca = head;
     while(baca !=NULL){
          printf("%d\n", baca->data);
          baca = baca->next;
     }
}


void alokasi(){
     int x;
     printf("Data yang mau disimpan : ");
     scanf("%d", &x);

     p = (Node *) malloc(sizeof(Node));
     if(p==NULL){
          puts("alokasi gagal");
          exit(0);
```

```
      }else{
            p->data = x;
            p->next = NULL;
      }
}

void awal(){
      if(head != NULL)
            p->next = head;
      head = p;
}
```

Output :



3. Delete Node Tertentu

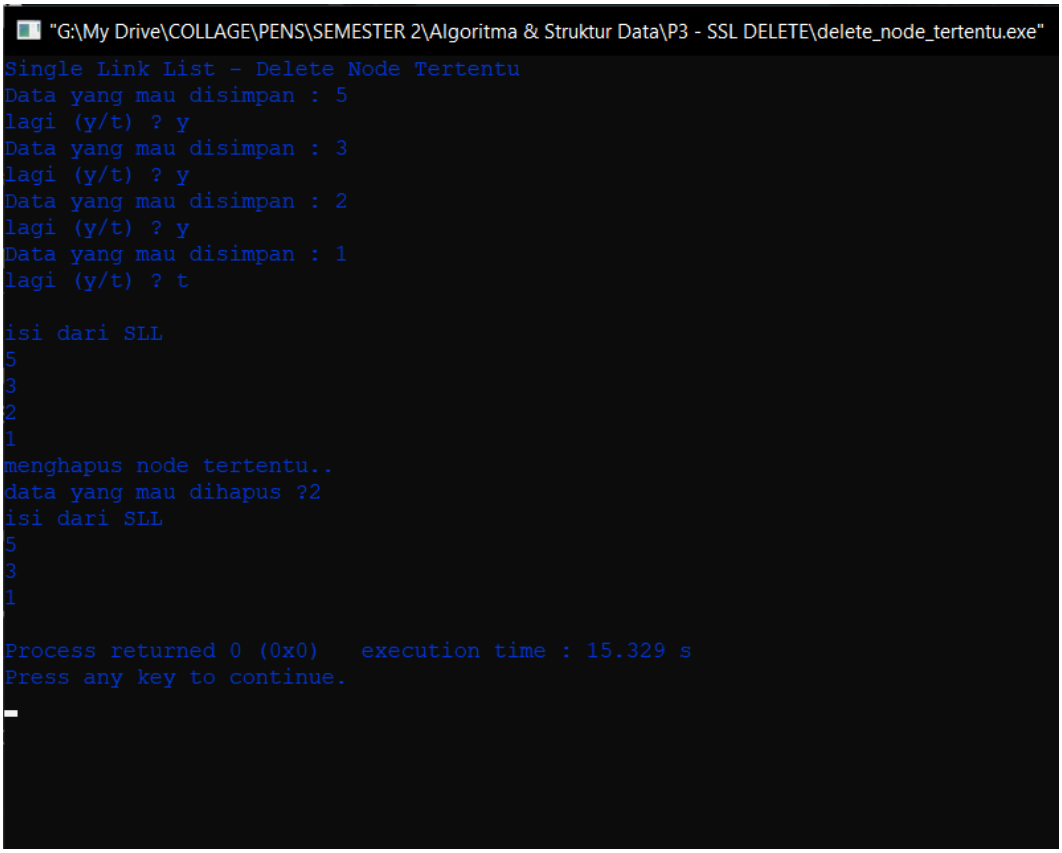
Souce Code :

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

typedef struct simpul Node;
struct simpul{
     int data;
     Node *next;
};

Node *head=NULL, *p;

void alokasi();
void awal();
void tampil();
void bebaskan(Node *);
void hapus_awal();
void hapus_node_tertentu();

int main()
{
     char jwb;
     puts("Single Link List - Delete Node Tertentu");
     do {
          fflush(stdin);
          alokasi();
          akhir();
          fflush(stdin);
          printf("lagi (y/t) ? ");
          jwb = getchar();
     }while((jwb == 'y')||(jwb == 'Y'));
    puts("");
     tampil();
    puts("menghapus node tertentu..");
    fflush(stdin);
    hapus_node_tertentu();
    tampil();
     return 0;
}

void bebaskan(Node *x){
    free(x);
    x = NULL;
}

void hapus_node_tertentu(){
```

```
     Node *hapus, *phapus;
     int key;

     printf("data yang mau dihapus ?");
     scanf("%d", &key);

     hapus = head;
     hapus = head;
     if(hapus->data==key)
          hapus_awal();
     else{
          while (hapus->data != key)
          {
               if (hapus->next==NULL)
               {
                    printf("%d tidak ada dalam SLL\n",
key);
                    exit(0);
               }else{
                    phapus = hapus;
                    hapus = hapus->next;
               }

          }
          phapus->next = hapus->next;
          bebaskan(hapus);
     }
}

void hapus_awal(){
    Node *hapus;

    hapus = head;
    if (hapus -> next == NULL)
        head = NULL;
    else{
        head = hapus->next;
        bebaskan(hapus);
    }

}

void akhir(){
    Node *tail;

    if(head == NULL)
        head = p;
    else{
        tail = head;
        while(tail->next != NULL)
            tail=tail->next;
```

```
         tail->next = p;
         tail=tail->next;
     }
}

void tampil(){
     Node *baca;

     puts("isi dari SLL");
     baca = head;
     while(baca !=NULL){
          printf("%d\n", baca->data);
          baca = baca->next;
     }
}


void alokasi(){
     int x;
     printf("Data yang mau disimpan : ");
     scanf("%d", &x);

     p = (Node *) malloc(sizeof(Node));
     if(p==NULL){
          puts("alokasi gagal");
          exit(0);
     }else{
          p->data = x;
          p->next = NULL;
     }
}

void awal(){
     if(head != NULL)
          p->next = head;
     head = p;
}
```

Output :



4. Menu Insert – Delete

Source Code :