Praktikum Algoritma & Struktur Data

**Prak 2.1. Single Linked List: Insert**

**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. Insert awal

2. Insert akhir

3. Insert after

4. Insert before

Buat fungsi insert dengan tipe yg bersesuaian untuk tiap soal.

Contoh no 1. fungsi yg dibuat adalah

void alokasi();

void awal();

void tampil();

Contoh no 2. fungsi yg dibuat adalah

void alokasi();

void akhir();

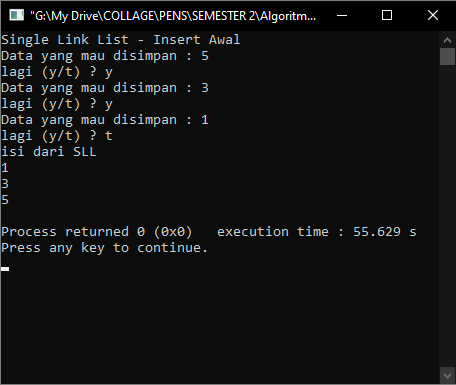
void tampil();

1. Insert awal

Sorce 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();  int main()  {  char jwb;  puts("Single Link List - Insert Awal");  do {  fflush(stdin);  alokasi();  awal();  fflush(stdin);  printf("lagi (y/t) ? ");  jwb = getchar();  }while((jwb == 'y')||(jwb == 'Y'));  tampil();  return 0;  }  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 :

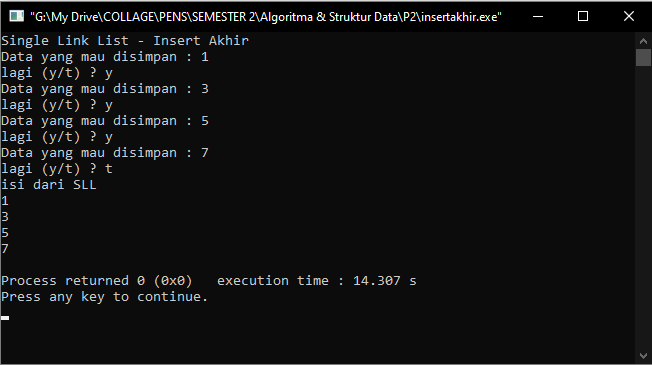


1. Insert Akhir

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

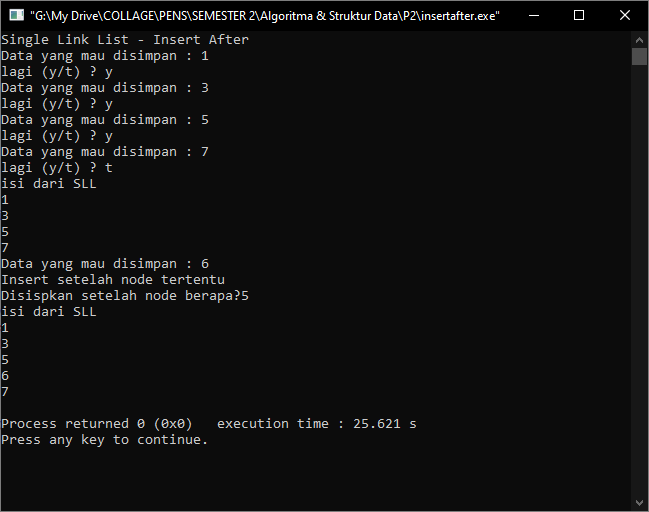


1. Insert After

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 akhir();  void setelah();  int main()  {  char jwb;  puts("Single Link List - Insert After");  do {  fflush(stdin);  alokasi();  akhir();  fflush(stdin);  printf("lagi (y/t) ? ");  jwb = getchar();  }while((jwb == 'y')||(jwb == 'Y'));  tampil();  fflush(stdin);  alokasi();  fflush(stdin);  puts("Insert setelah node tertentu");  setelah();  tampil();  return 0;  }  void setelah(){  Node \*after;  int key;  printf("Disispkan setelah node berapa?");  scanf("%d", &key);  after = head;  while(after->data != key){  if(after->next == NULL){  printf("&d tidak ada dalam SLL/n", key);  exit(0);  }  else  after = after ->next;  }  p->next = after->next;  after->next=p;  }  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 :



1. Insert Before

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 akhir();  void tampil();  void before();  int main(){  char jwb;  puts("Single Link List - Insert Before");  do{  fflush(stdin);  alokasi();  akhir();  fflush(stdin);  printf("Lagi? (y/n)");  jwb = getchar();  }while(jwb == 'Y'||jwb == 'y');  tampil();  fflush (stdin);  puts("Insert SEBELUM Node tertentu?");  alokasi();  fflush (stdin);  before();  tampil();  return 0;  }    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 before(){  Node \*bef, \*pbef;  int key;  printf("Disipkan sebelum node berapa ? ");  scanf("%d", &key);  bef = head;  while(bef -> data != key){  if(bef -> next == NULL){  printf("%d tdk ada dlm SLL\n", key);  exit(0);  }else{  pbef = bef;  bef = bef -> next;  }  }  p -> next = bef;  pbef ->next = p;  }    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;  }  } |

Output :

