TUGAS PRAKTIKUM ALGORITMA & STRUKTUR DATA

Jilid 8

Oleh :

Nama : Rosi Arif Mulyadi

NRP : 3121522021

Prodi : D3 Teknik Informatika PENS PSDKU Sumenep

Kelas : 1 ITA D3 Sumenep

Dosen :

LUSIANA AGUSTIEN M.Kom

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

MODUL 4

LINKED LIST (Double LINKED LIST NON CIRCULAR)

B. Kegiatan Praktikum

1. Lakukan Pembenahan terhadap listing program yang di berikan pada implementasi linked list non circular diatas.

Jawab :

Listing Program :

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

struct node
{
   struct node *prev;
   struct node *next;
   int data;
};
struct node *head;
void insertion_beginning();
void insertion_last();
void insertion_specified();
void deletion_beginning();
void deletion_last();
void deletion_specified();
void display();
void search();

void insertion_beginning()
{
   struct node *ptr;
   int item;
   ptr = (struct node *)malloc(sizeof(struct node));
   if(ptr == item)
   {
      printf("\nOVERFLOW");
   }
   else
```

```
    {
        printf("\n Input Data : ");
        scanf("%d", &item);
        if(head==NULL)
        {
            ptr->next = NULL;
            ptr->data = item;
            ptr->prev = NULL;
            head=ptr;
        }
        else
        {
            ptr->data = ptr;
            ptr->next = NULL;
            ptr->prev = head;
            head->prev = NULL;
            head=ptr;
        }
        printf("\nNode telah di inputkan\n");
    }
}
void insertion_last()
{
    struct node *ptr, *temp;
    int item;
    ptr = (struct node *)malloc(sizeof(struct node));
    if(ptr == NULL)
    {
        printf("\nOVERFLOW");
    }
    else
    {
        printf("\nInput nilai data : ");
        scanf("%d", &item);
        ptr->data=item;
        if(head == NULL)
        {
            ptr->next = NULL;
```

```
            ptr->prev = NULL;
            head = ptr;
        }
        else
        {
            temp = head;
            while(temp->next!=NULL)
            {
                temp = temp->next;
            }
            temp->next = ptr;
            ptr->prev = temp;
            ptr->next = NULL;
        }
    }
    printf("\nNode sudah di inputkan\n");
}
void insertion_specified()
{
    struct node *ptr, *temp;
    int item, loc, i;
    ptr = (struct node *)malloc(sizeof(struct node));
    if(ptr == NULL)
    {
        printf("\nOVERFLOW");
    }
    else
    {
        temp=head;
        printf("Input lokasi yang akan di sisipkan node baru : ");
        scanf("%d", &loc);
        for(i=0; i<loc; i++)
        {
            temp = temp->next;
            if(temp == NULL)
            {
                printf("\n Linked list hanya memiliki %d elements : ", loc);
                return;
```

```
            }
        }
        printf("Inputkan data : ");
        scanf("%d", &item);
        ptr->data = item;
        ptr->next = temp->next;
        ptr->prev = temp;
        temp->next = ptr;
        temp->next->prev=ptr;
        printf("\nNode sudah di inputkan\n");
    }
}
void deletion_beginning()
{
    struct node *ptr;
    if(head == NULL)
    {
        printf("\nUNDERFLOW");
    }
    else if(head->next == NULL)
    {
        head = NULL;
        free(head);
        printf("\nNode berhasil di hapus\n");
    }
    else
    {
        ptr = head;
        head = head->next;
        head->prev = NULL;
        free(ptr);
        printf("\nNode berhasil di hapus\n");
    }
}
void deletion_last()
{
    struct node *ptr;
    if(head == NULL)
```

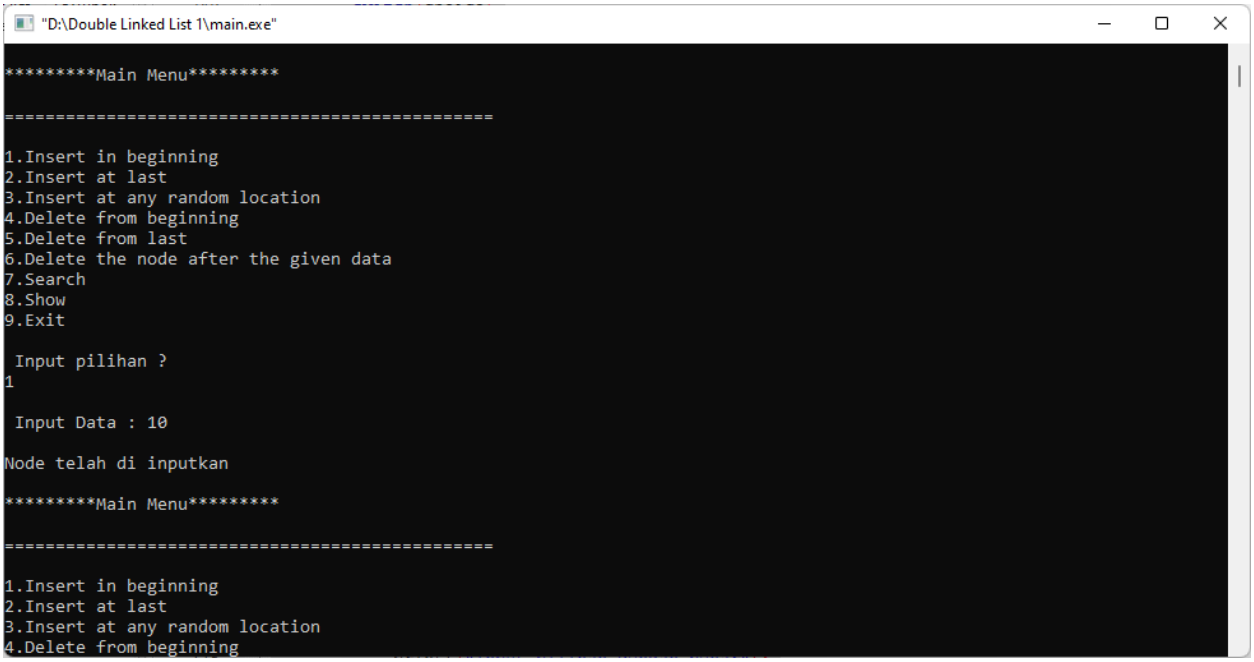
```
    {
        printf("\nUNDERFLOW");
    }
    else if(head->next == NULL)
    {
        head = NULL;
        free(head);
        printf("\nNode berhasil di hapus\n");
    }
    else
    {
        ptr = head;
        if(ptr->next != NULL)
        {
            ptr = ptr->next;
        }
        ptr->prev->next = NULL;
        free(head);
        printf("\nNode berhasil di hapus\n");
    }
}
void deletion_specified()
{
    struct node *ptr, *temp;
    int val;
    printf("\n Inputkan data yang akan di hapus : ");
    scanf("%d", &val);
    ptr = head;
    while(temp->next != NULL)
    {
        ptr = ptr->next;
        if(ptr->next == NULL)
        {
            printf("\n Tidak dapat di deleted\n");
        }
        else if(ptr->next->next == NULL)
        {
            ptr->next = NULL;
```

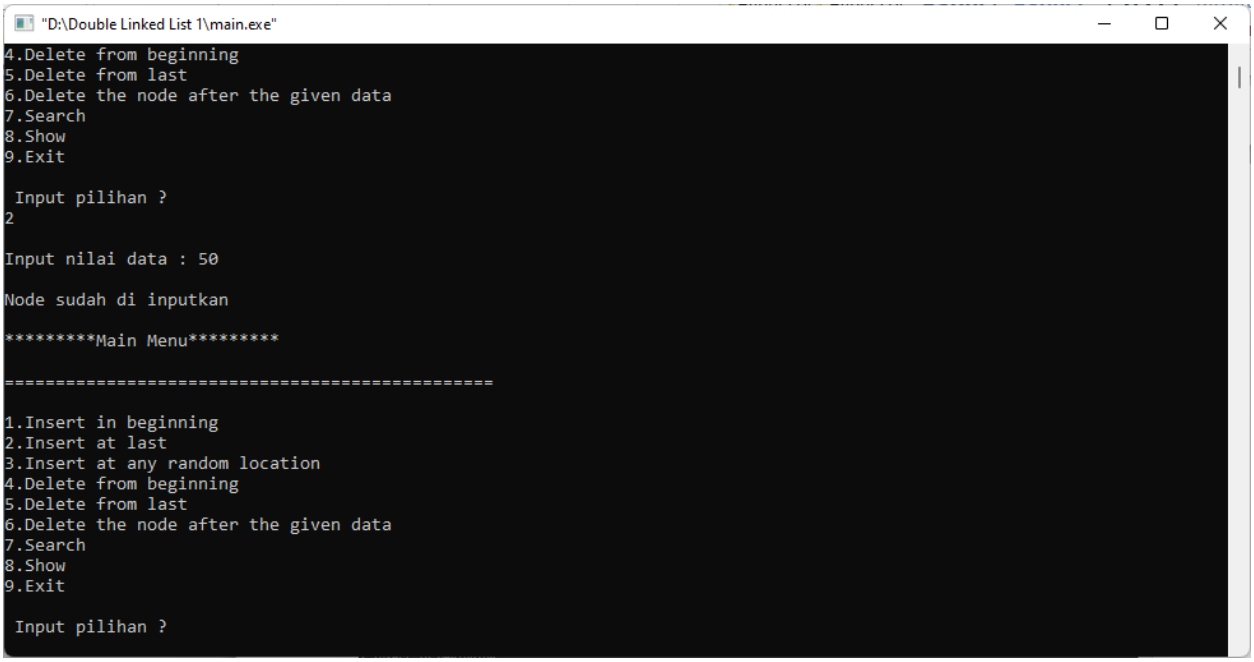
```
        }
        else
        {
            temp = ptr->next;
            ptr->next = temp->next;
            temp->next->prev = ptr;
            free(temp);
            printf("\nNode deleted\n");
        }
    }
}
void display()
{
    struct node *ptr;
    printf("\n Tampilkan list : \n");
    ptr = head;
    while(ptr != NULL)
    {
        printf("%d\n", ptr->data);
        ptr = ptr->next;
    }
}
void search()
{
    struct node *ptr;
    int item, i=0, flag;
    ptr = head;
    if(ptr == NULL)
    {
        printf("\nList kosong\n");
    }
    else
    {
        printf("\nInputkan data yang akan dicari : \n");
        scanf("%d", &item);
        while(ptr!=NULL)
        {
            if(ptr->data == item)
```

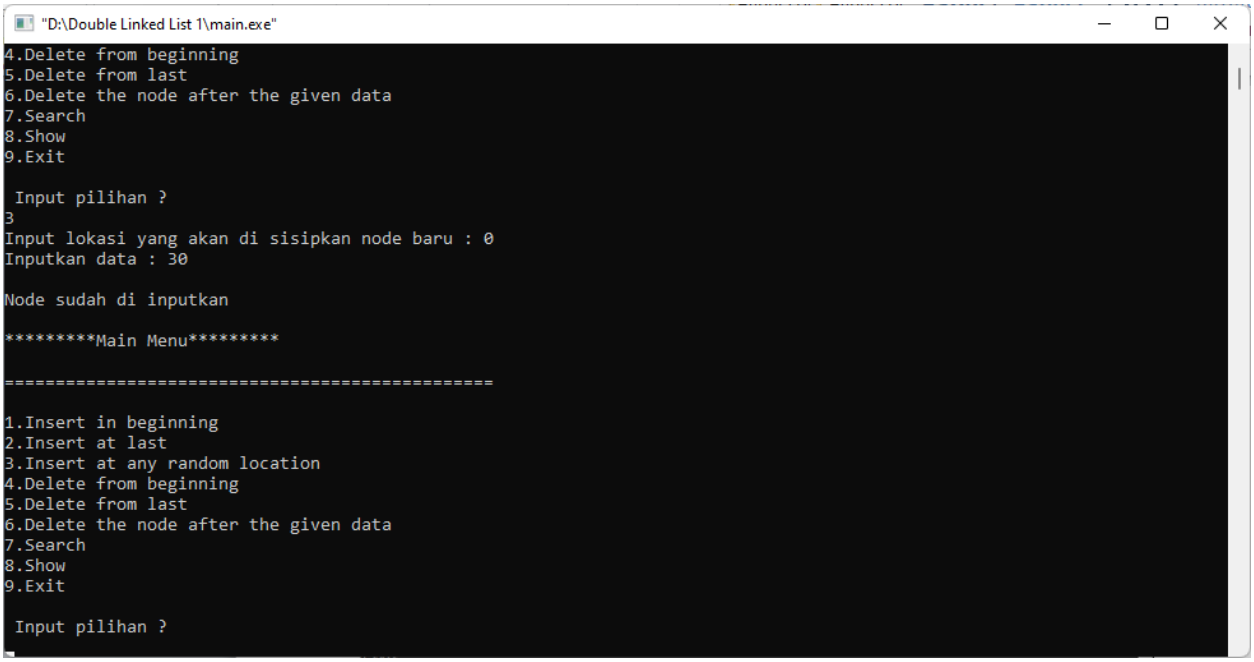
```
        {
          printf("\n Data yang dicari ada di dalam node %d ", i);
          flag=0;
          break;
        }
        else
        {
          flag=1;
        }
        i++;
        ptr = ptr->next;
      }
      if(flag==1)
      {
        printf("\nItem tidak ditemukan\n");
      }
    }
}
void main()
{
   int choice = 0;
   while(choice != 9)
   {
      printf("\n*********Main Menu*********\n");
      printf("\n================================================\n");
      printf("\n1.Insert in beginning\n2.Insert at last\n3.Insert at any random location\n4.Delete
from beginning\n5.Delete from last\n6.Delete the node after the given
data\n7.Search\n8.Show\n9.Exit\n");
      printf("\n Input pilihan ?\n");
      scanf("\n%d", &choice);
      switch(choice)
      {
      case 1:
        insertion_beginning();
        break;
      case 2:
        insertion_last();
        break;
```

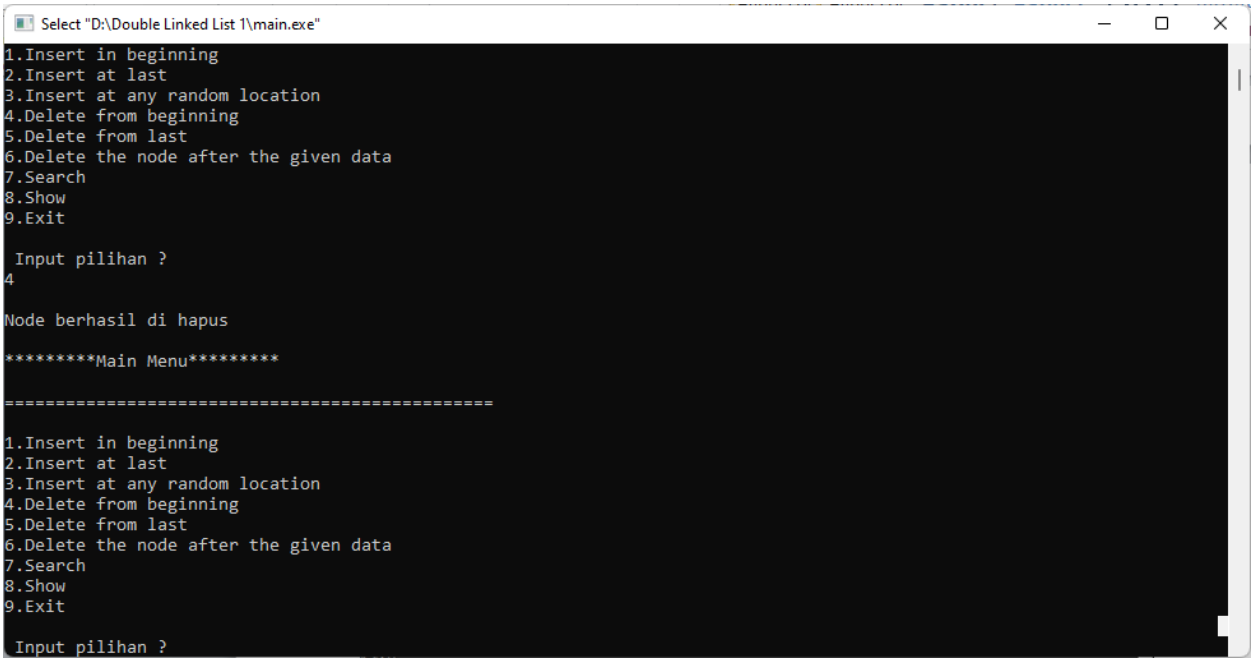
```
        case 3:
           insertion_specified();
           break;
        case 4 :
           deletion_beginning();
           break;
        case 5:
           deletion_last();
           break;
        case 6:
           deletion_specified();
           break;
        case 7:
           search();
           break;
        case 8:
           display();
           break;
        case 9:
           exit(0);
           break;
        default:
           printf("Input pilihan dengan benar");
        }
     }
}
```

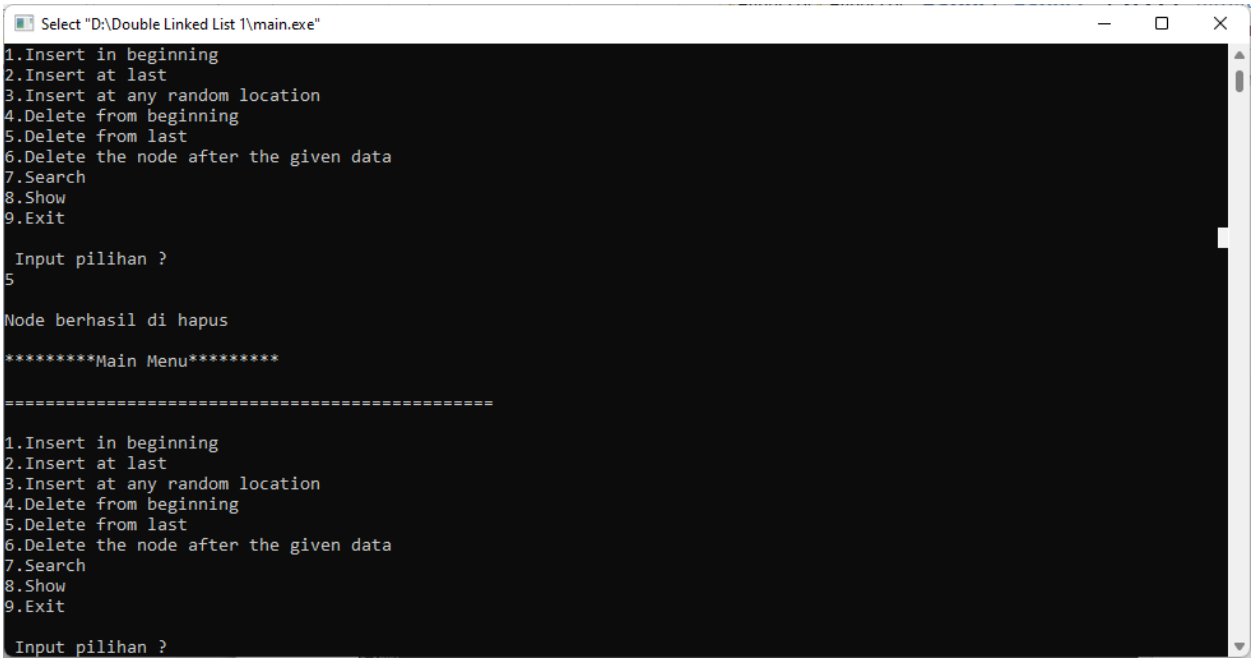
Output :

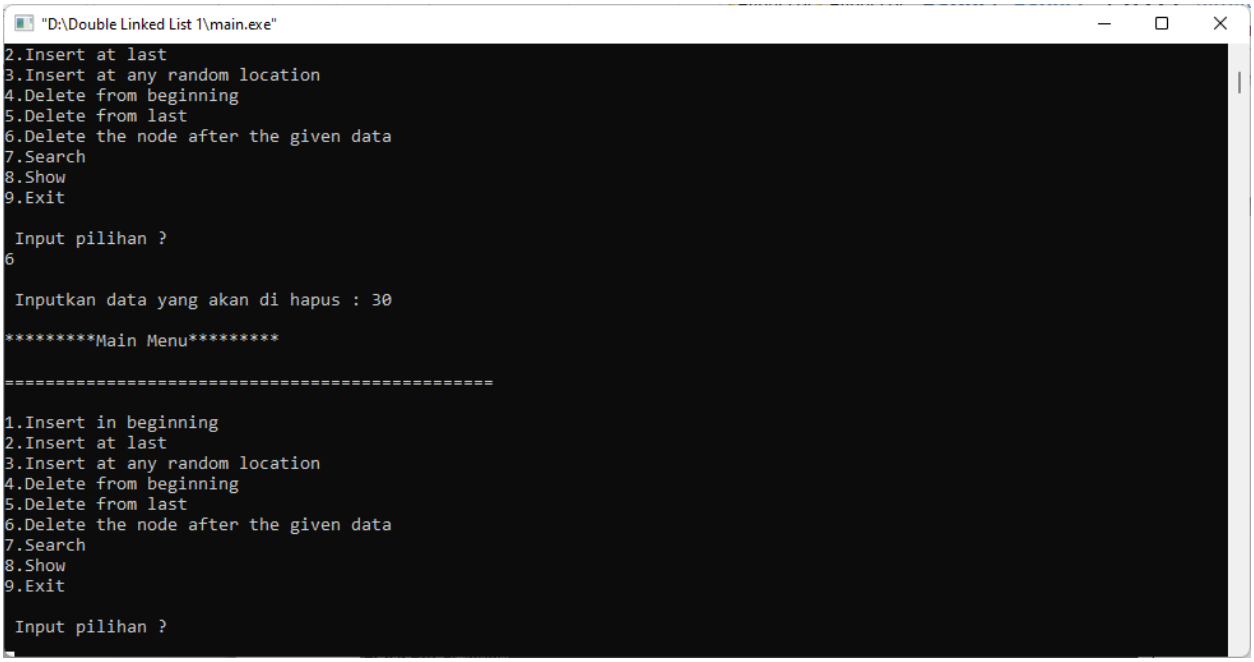

"D:\Double Linked List 1\main.exe"
*********Main Menu*********
==========
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Delete from beginning
5.Delete from last
6.Delete the node after the given data
7.Search
8.Show
9.Exit
Input pilihan ?
1
Input Data : 10
Node telah di inputkan
*********Main Menu*********
==========
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Delete from beginning

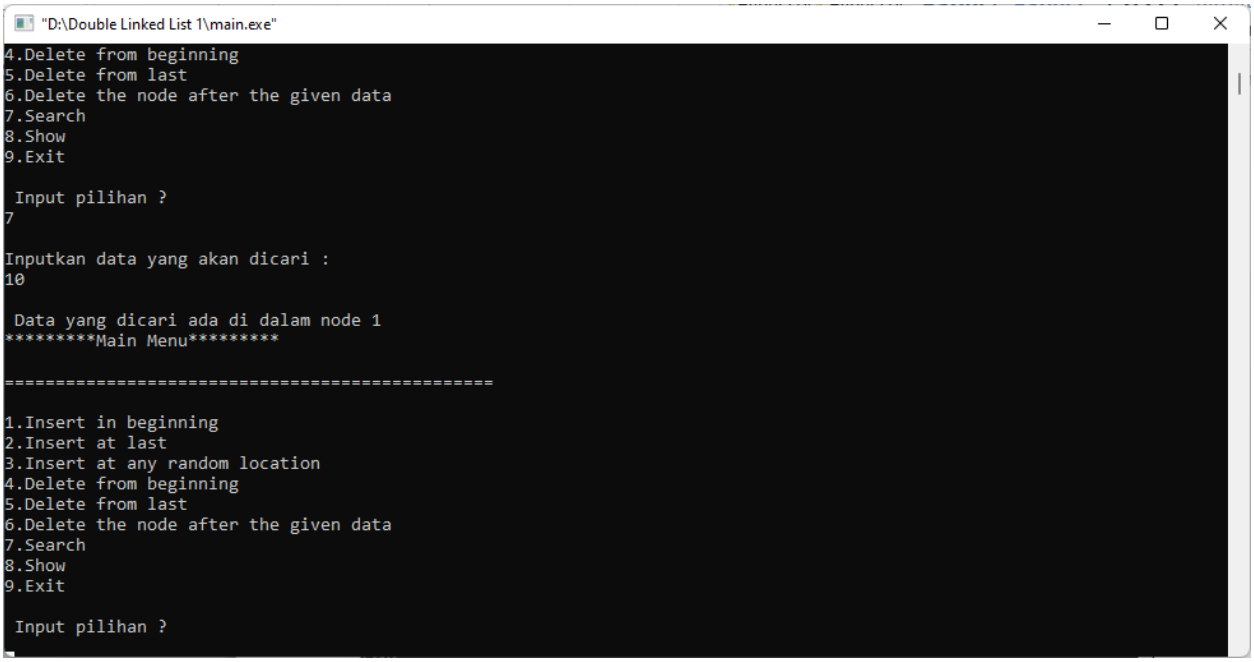

"D:\Double Linked List 1\main.exe"
4.Delete from beginning
5.Delete from last
6.Delete the node after the given data
7.Search
8.Show
9.Exit
Input pilihan ?
2
Input nilai data : 50
Node sudah di inputkan
*********Main Menu*********
==========
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Delete from beginning
5.Delete from last
6.Delete the node after the given data
7.Search
8.Show
9.Exit
Input pilihan ?

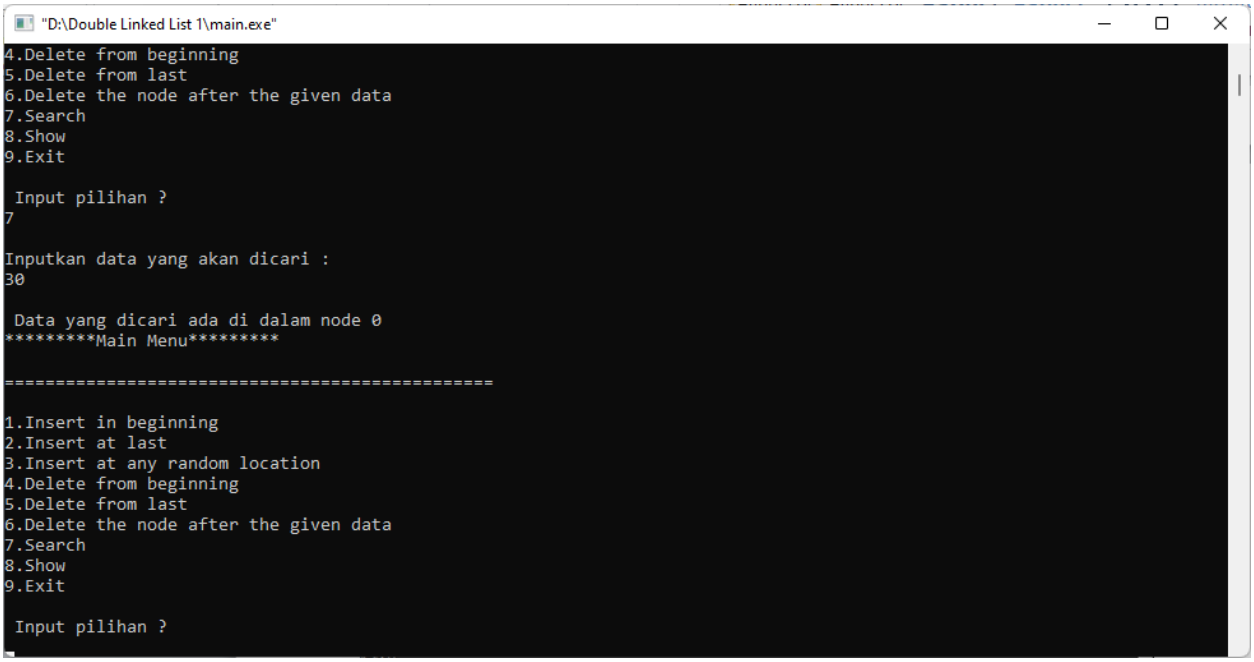

"D:\Double Linked List 1\main.exe"
4.Delete from beginning
5.Delete from last
6.Delete the node after the given data
7.Search
8.Show
9.Exit
Input pilihan ?
3
Input lokasi yang akan di sisipkan node baru : 0
Inputkan data : 30
Node sudah di inputkan
*********Main Menu*********
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Delete from beginning
5.Delete from last
6.Delete the node after the given data
7.Search
8.Show
9.Exit
Input pilihan ?

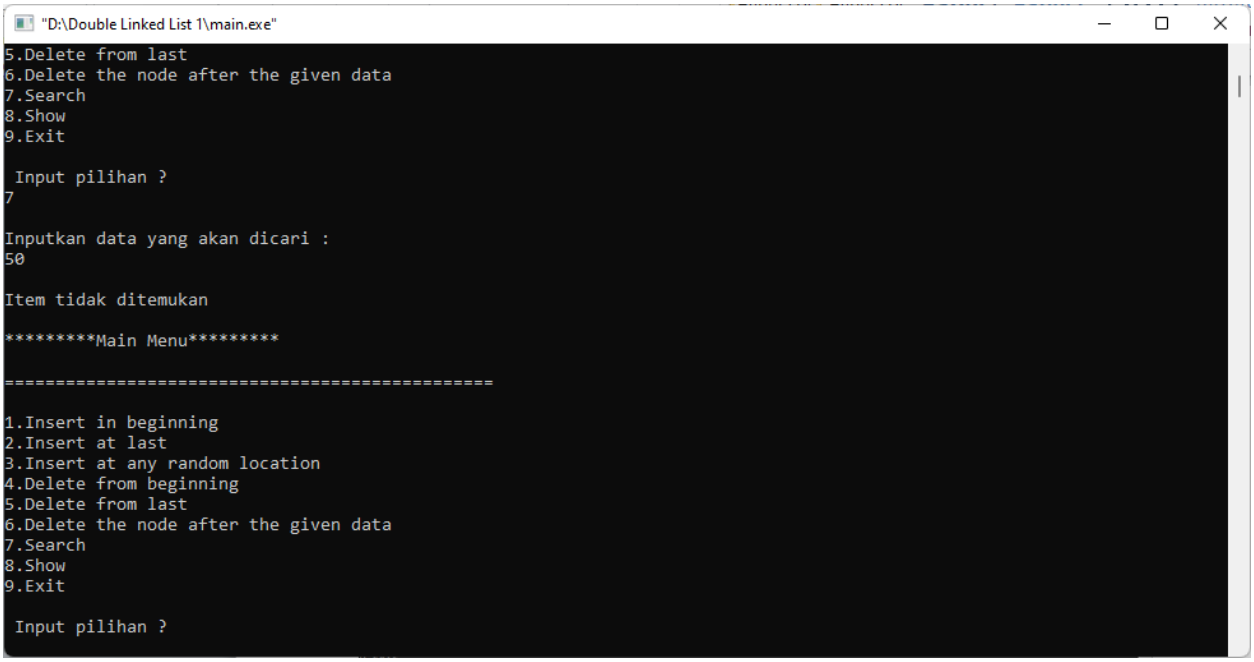

Select "D:\Double Linked List 1\main.exe"
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Delete from beginning
5.Delete from last
6.Delete the node after the given data
7.Search
8.Show
9.Exit
Input pilihan ?
4
Node berhasil di hapus
*********Main Menu*********
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Delete from beginning
5.Delete from last
6.Delete the node after the given data
7.Search
8.Show
9.Exit
Input pilihan ?

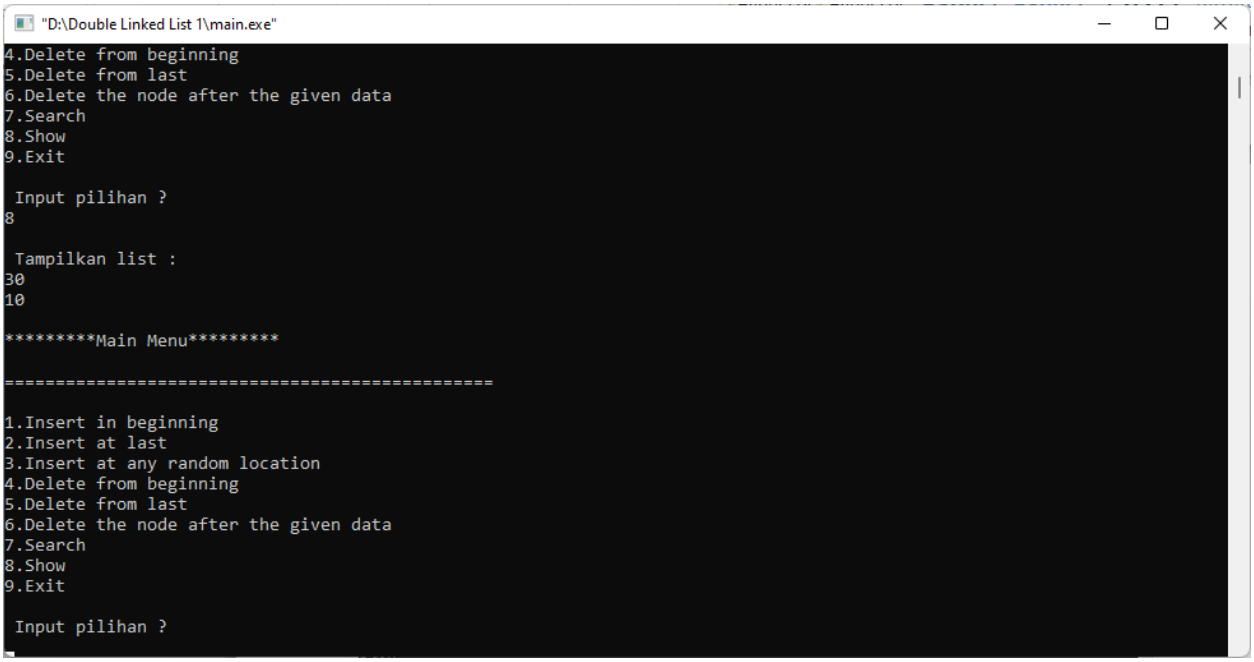

Select "D:\Double Linked List 1\main.exe"
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Delete from beginning
5.Delete from last
6.Delete the node after the given data
7.Search
8.Show
9.Exit
Input pilihan ?
5
Node berhasil di hapus
*********Main Menu*********
==========
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Delete from beginning
5.Delete from last
6.Delete the node after the given data
7.Search
8.Show
9.Exit
Input pilihan ?

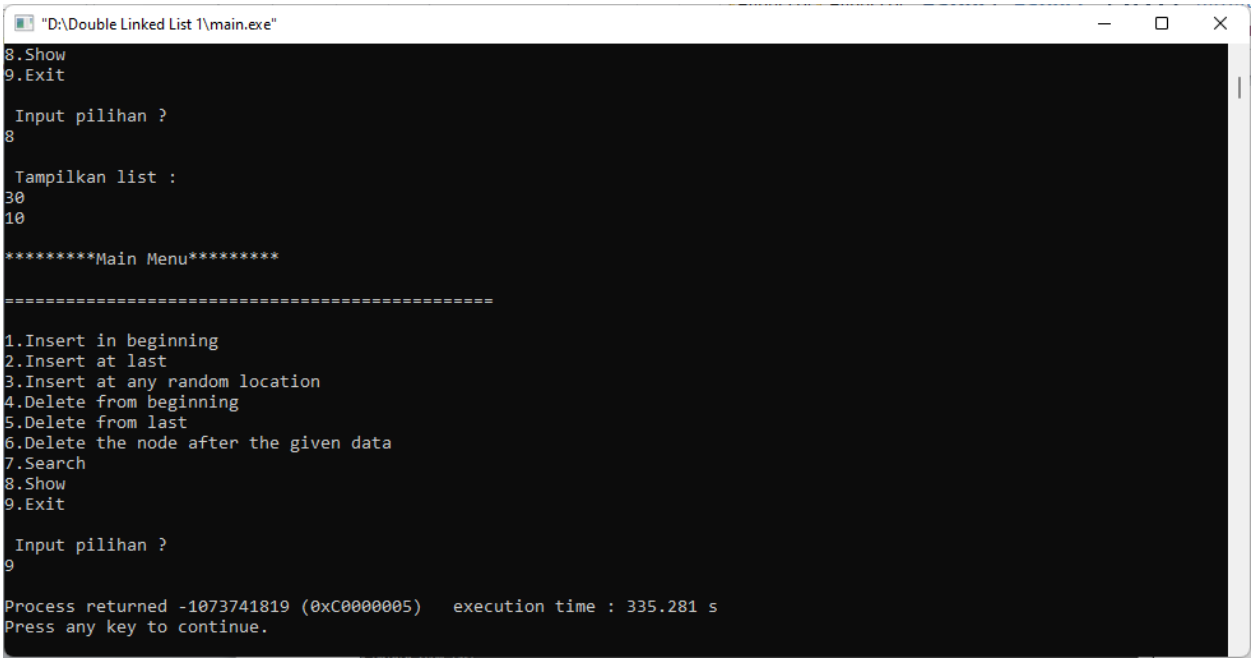

"D:\Double Linked List 1\main.exe"
2.Insert at last
3.Insert at any random location
4.Delete from beginning
5.Delete from last
6.Delete the node after the given data
7.Search
8.Show
9.Exit
Input pilihan ?
6
Inputkan data yang akan di hapus : 30
*********Main Menu*********
==========
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Delete from beginning
5.Delete from last
6.Delete the node after the given data
7.Search
8.Show
9.Exit
Input pilihan ?


"D:\Double Linked List 1\main.exe"
4.Delete from beginning
5.Delete from last
6.Delete the node after the given data
7.Search
8.Show
9.Exit
Input pilihan ?
7
Inputkan data yang akan dicari :
10
Data yang dicari ada di dalam node 1
*********Main Menu*********
==========
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Delete from beginning
5.Delete from last
6.Delete the node after the given data
7.Search
8.Show
9.Exit
Input pilihan ?


"D:\Double Linked List 1\main.exe"
4.Delete from beginning
5.Delete from last
6.Delete the node after the given data
7.Search
8.Show
9.Exit
Input pilihan ?
7
Inputkan data yang akan dicari :
30
Data yang dicari ada di dalam node 0
*********Main Menu*********
==========
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Delete from beginning
5.Delete from last
6.Delete the node after the given data
7.Search
8.Show
9.Exit
Input pilihan ?


"D:\Double Linked List 1\main.exe"
5.Delete from last
6.Delete the node after the given data
7.Search
8.Show
9.Exit
Input pilihan ?
7
Inputkan data yang akan dicari :
50
Item tidak ditemukan
*********Main Menu*********
==========
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Delete from beginning
5.Delete from last
6.Delete the node after the given data
7.Search
8.Show
9.Exit
Input pilihan ?


"D:\Double Linked List 1\main.exe"
4.Delete from beginning
5.Delete from last
6.Delete the node after the given data
7.Search
8.Show
9.Exit
Input pilihan ?
8
Tampilkan list :
30
10
*********Main Menu*********
==========
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Delete from beginning
5.Delete from last
6.Delete the node after the given data
7.Search
8.Show
9.Exit
Input pilihan ?



2. Buatlah dan tuliskan langkah – langkah algoritma yang disajikan pada tiap function yang dibuat pada implementasi linked list non circular diatas.

Jawab :

- **ALgoritma Menginpukan Data :**

Step 1: IF ptr = NULL Write OVERFLOW Go to Step 9 [END OF IF]
Step 2: SET NEW_NODE = ptr
Step 3: SET ptr = ptr -> NEXT
Step 4: SET NEW_NODE -> DATA = VAL
Step 5: SET NEW_NODE -> PREV = NULL
Step 6: SET NEW_NODE -> NEXT = START
Step 7: SET head -> PREV = NEW_NODE
Step 8: SET head = NEW_NODE
Step 9: EXIT

- **Algoritma Menghapus Data :**

Step 1: IF HEAD = NULL Write UNDERFLOW Go to Step 7 [END OF IF]
Step 2: SET TEMP = HEAD
Step 3: REPEAT STEP 4 WHILE TEMP->NEXT != NULL
Step 4: SET TEMP = TEMP->NEXT [END OF LOOP]
Step 5: SET TEMP ->PREV-> NEXT = NULL
Step 6: FREE TEMP
Step 7: EXIT

C. Tugas Praktikum

1. Buatlah program untuk menginputkan data pada double linked list (di awal, akhir, dan di tempat tertetu) namu pada saat kita menginputkan data 0, maka otomatis proses penginputan data akan berhenti.

Jawab :

Listing Program :

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

struct node
{
   struct node *prev;
   struct node *next;
   int data;
};
struct node *head;
void insertion_beginning();
void insertion_last();
void insertion_specified();

void insertion_beginning()
{
   struct node *ptr;
   int angka;
   ptr = (struct node *)malloc(sizeof(struct node));
   if(ptr == angka)
   {
      printf("\nOVERFLOW");
   }
   else
   {
      printf("\n Input Data : ");
      scanf("%d", &angka);
      if(angka==0){
         exit(0);
      }
      if(head==NULL)
      {
```

```
            ptr->next = NULL;
            ptr->data = angka;
            ptr->prev = NULL;
            head=ptr;
        }
        else
        {
            ptr->data = ptr;
            ptr->next = NULL;
            ptr->prev = head;
            head->prev = NULL;
            head=ptr;
        }
        printf("\nNode telah di inputkan\n");
    }
}
void insertion_last()
{
    struct node *ptr, *temp;
    int angka;
    ptr = (struct node *)malloc(sizeof(struct node));
    if(ptr == NULL)
    {
        printf("\nOVERFLOW");
    }
    else
    {
        printf("\nInput nilai data : ");
        scanf("%d", &angka);
        if(angka==0){
        exit(0);
        }
        ptr->data=angka;
        if(head == NULL)
        {
            ptr->next = NULL;
            ptr->prev = NULL;
            head = ptr;
```

```
        }
        else
        {
            temp = head;
            while(temp->next!=NULL)
            {
                temp = temp->next;
            }
            temp->next = ptr;
            ptr->prev = temp;
            ptr->next = NULL;
        }
    }
    printf("\nNode sudah di inputkan\n");
}
void insertion_specified()
{
    struct node *ptr, *temp;
    int angka, loc, i;
    ptr = (struct node *)malloc(sizeof(struct node));
    if(ptr == NULL)
    {
        printf("\nOVERFLOW");
    }
    else
    {
        temp=head;
        printf("Input lokasi yang akan di sisipkan node baru : ");
        scanf("%d", &loc);
        for(i=0; i<loc; i++)
        {
            temp = temp->next;
            if(temp == NULL)
            {
                printf("\n Linked list hanya memiliki %d elements : ", loc);
                return;
            }
        }
```

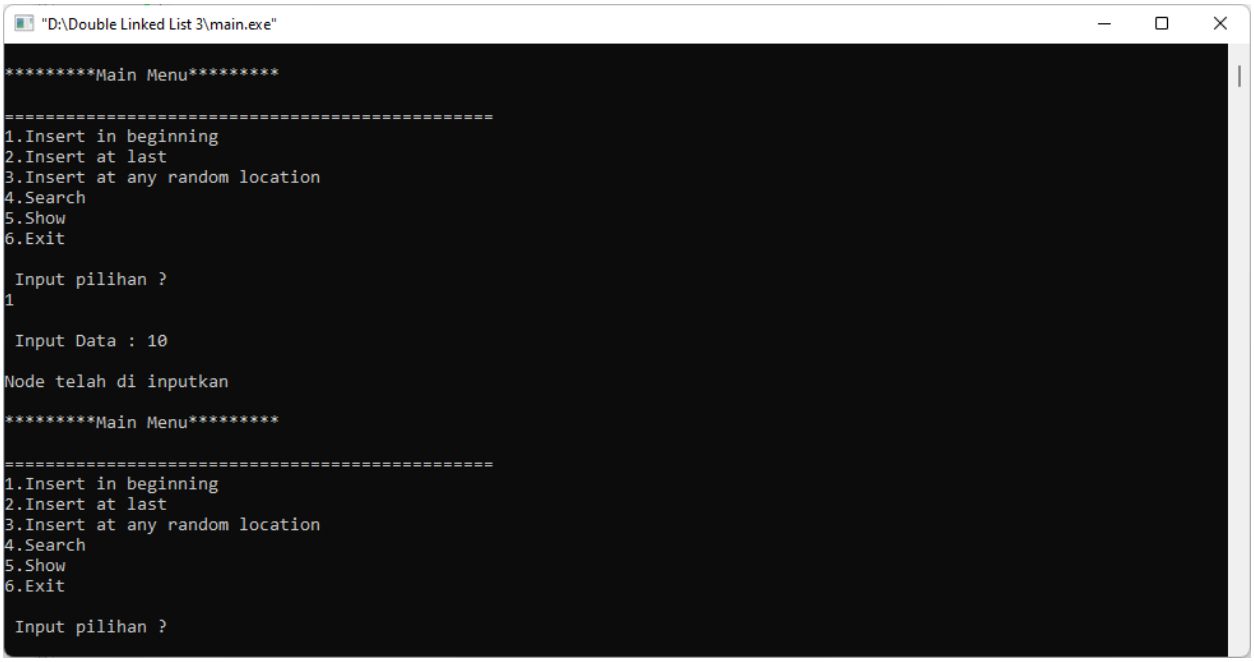
```
        printf("Inputkan data : ");
        scanf("%d", &angka);
        if(angka==0){
           exit(0);
        }
        ptr->data = angka;
        ptr->next = temp->next;
        ptr->prev = temp;
        temp->next = ptr;
        temp->next->prev=ptr;
        printf("\nNode sudah di inputkan\n");
    }
}
void display()
{
    struct node *ptr;
    printf("\n Tampilkan list : \n");
    ptr = head;
    while(ptr != NULL)
    {
        printf("%d\n", ptr->data);
        ptr = ptr->next;
    }
}
void search()
{
    struct node *ptr;
    int item, i=0, flag;
    ptr = head;
    if(ptr == NULL)
    {
        printf("\nList kosong\n");
    }
    else
    {
        printf("\nInputkan data yang akan dicari : \n");
        scanf("%d", &item);
        while(ptr!=NULL)
```

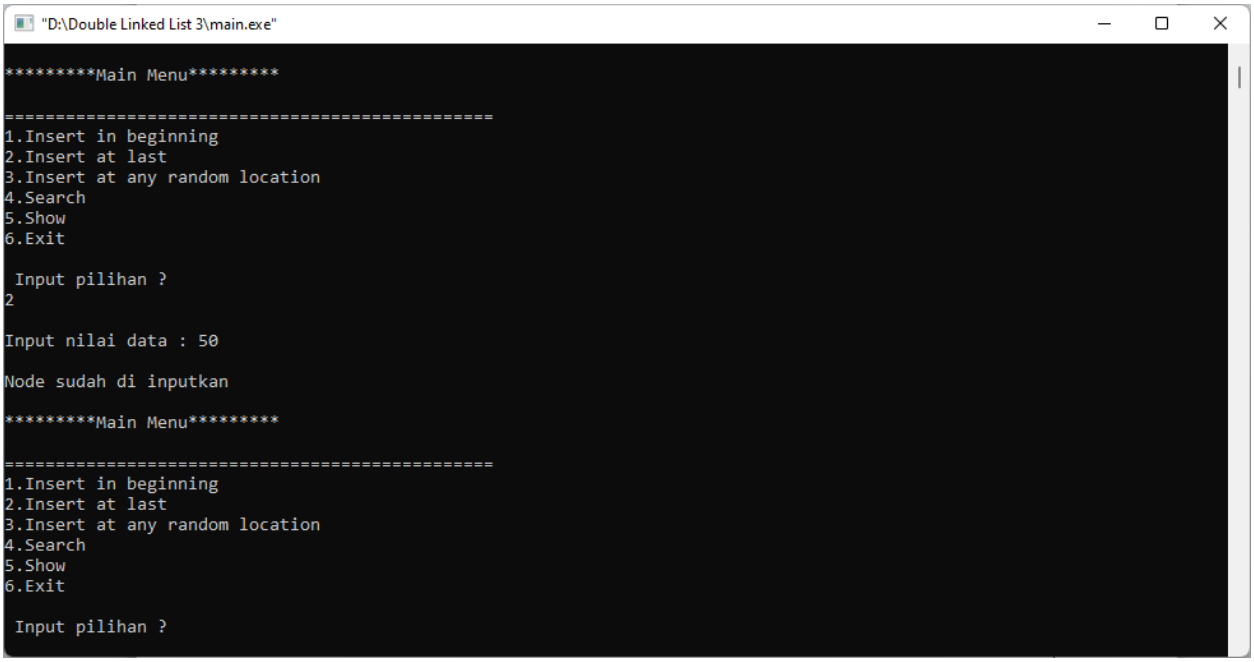
```
        {
            if(ptr->data == item)
            {
                printf("\n Data yang dicari ada di dalam node %d ", i);
                flag=0;
                break;
            }
            else
            {
                flag=1;
            }
            i++;
            ptr = ptr->next;
        }
        if(flag==1)
        {
            printf("\nItem tidak ditemukan\n");
        }
    }
}
void main()
{
    int choice = 0;
    while(choice != 6)
    {
        printf("\n*********Main Menu*********\n");
        printf("\n================================================\n");
        printf("1.Insert in beginning\n");
        printf("2.Insert at last\n");
        printf("3.Insert at any random location\n");
        printf("4.Search\n");
        printf("5.Show\n");
        printf("6.Exit\n");
        printf("\n Input pilihan ?\n");
        scanf("\n%d", &choice);
        switch(choice)
        {
        case 1:
```

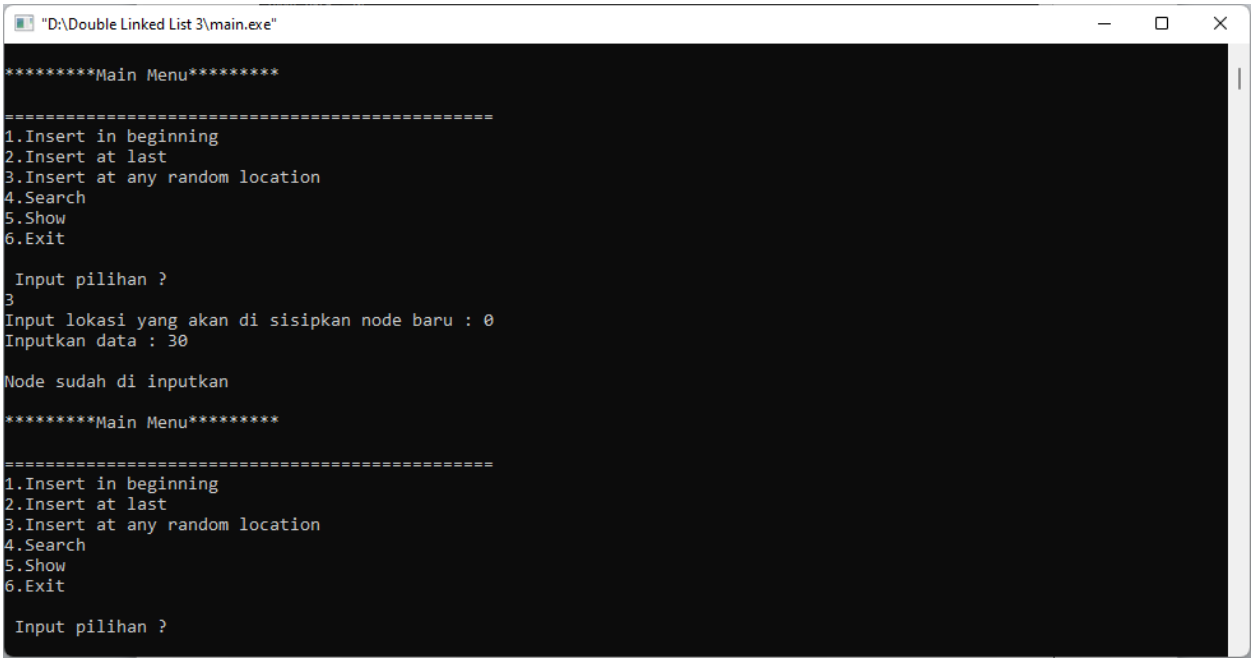
```
            insertion_beginning();
            break;
        case 2:
            insertion_last();
            break;
        case 3:
            insertion_specified();
            break;
        case 4:
            search();
            break;
        case 5:
            display();
            break;
        case 6:
            exit(0);
            break;
        default:
            printf("Input pilihan dengan benar");
        }
    }
}
```

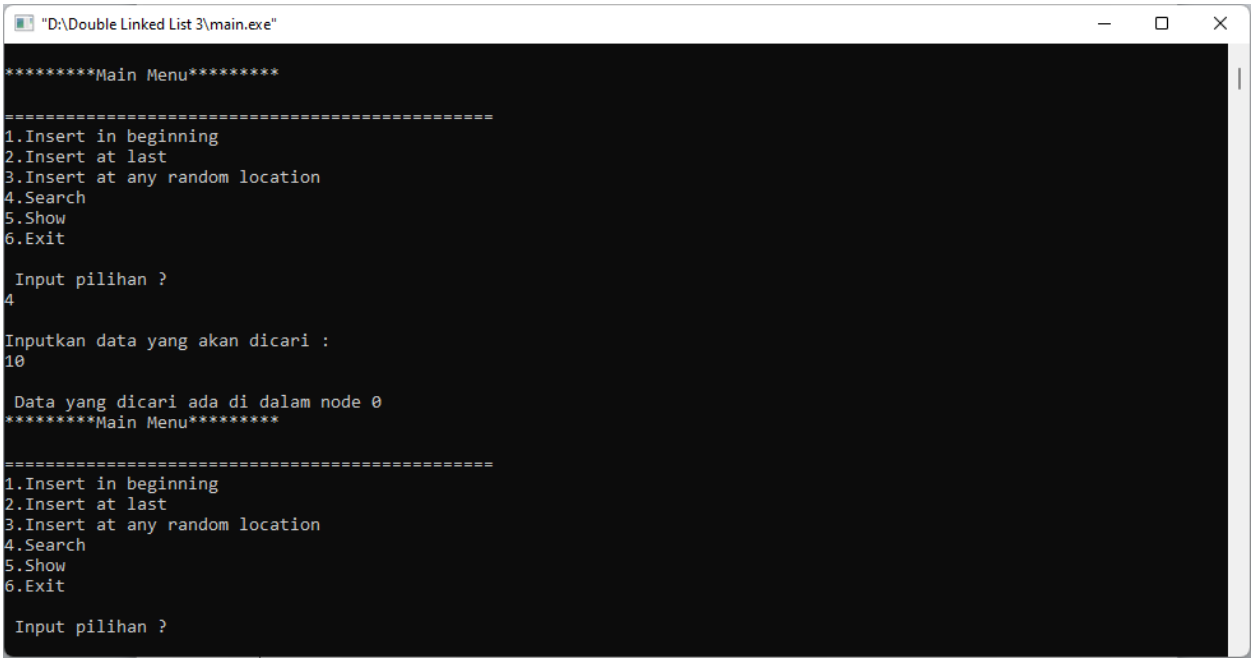
Output :

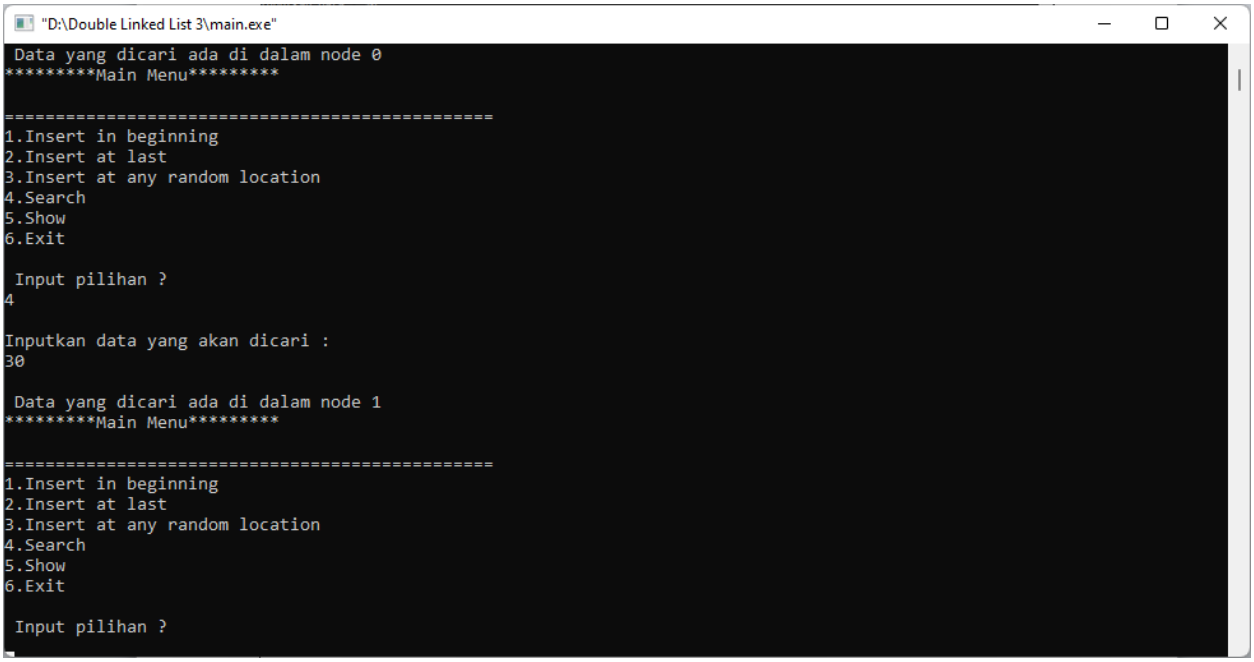
Tanpa Memasukkan Data 0

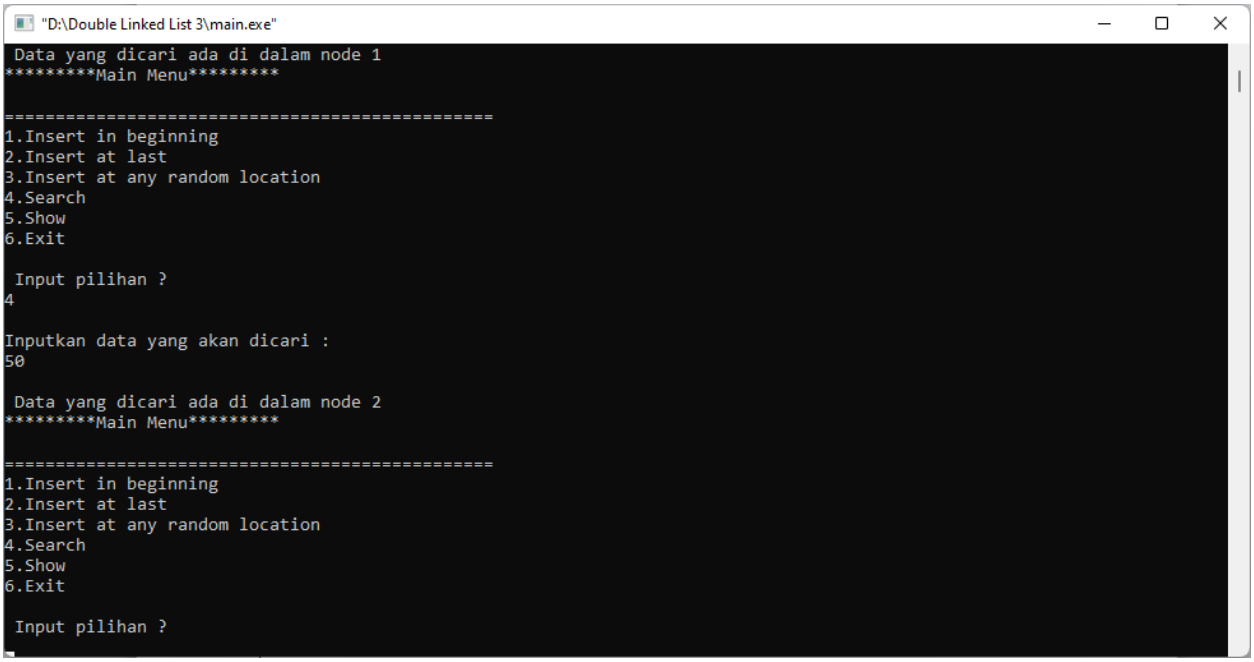

"D:\Double Linked List 3\main.exe"
*********Main Menu*********
==========
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Search
5.Show
6.Exit
Input pilihan ?
1
Input Data : 10
Node telah di inputkan
*********Main Menu*********
==========
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Search
5.Show
6.Exit
Input pilihan ?

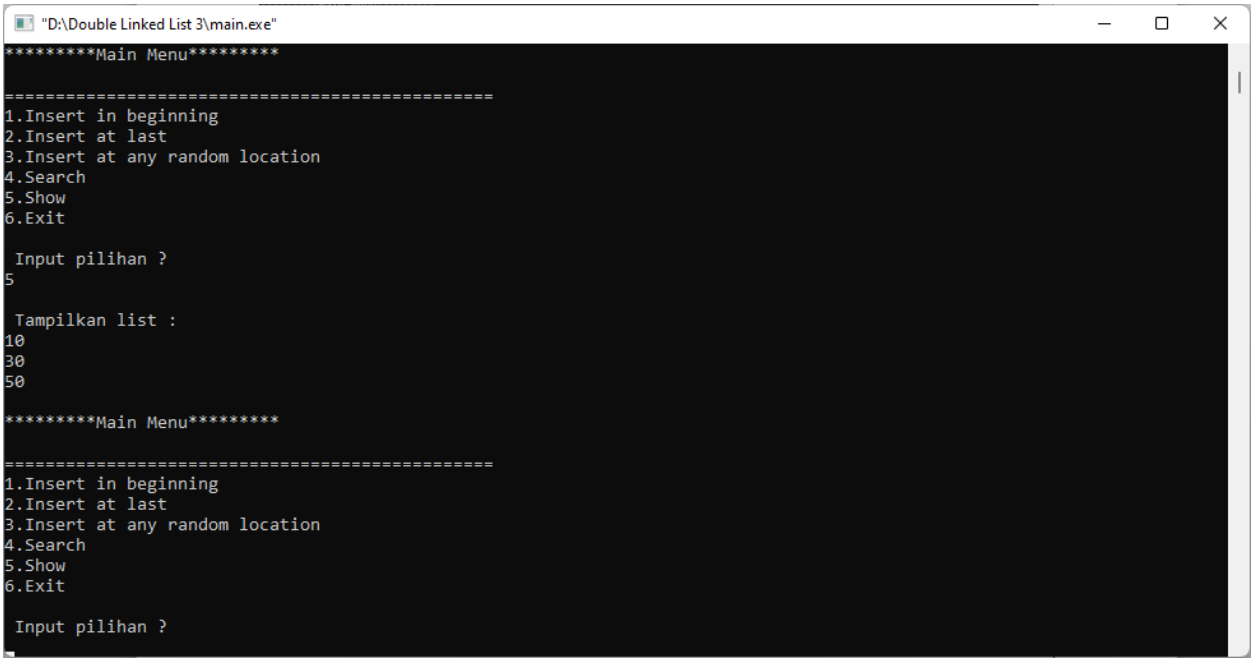

"D:\Double Linked List 3\main.exe"
*********Main Menu*********
==========
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Search
5.Show
6.Exit
Input pilihan ?
2
Input nilai data : 50
Node sudah di inputkan
*********Main Menu*********
==========
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Search
5.Show
6.Exit
Input pilihan ?

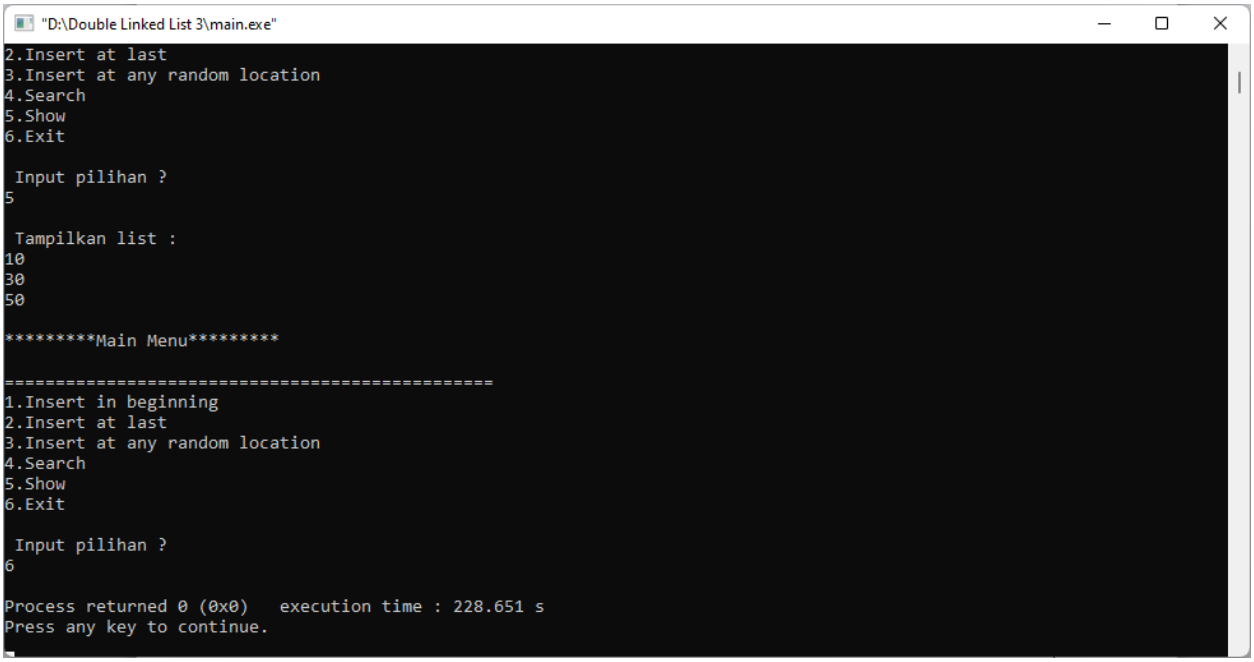

"D:\Double Linked List 3\main.exe"
*********Main Menu*********
==========
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Search
5.Show
6.Exit
Input pilihan ?
3
Input lokasi yang akan di sisipkan node baru : 0
Inputkan data : 30
Node sudah di inputkan
*********Main Menu*********
==========
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Search
5.Show
6.Exit
Input pilihan ?


"D:\Double Linked List 3\main.exe"
*********Main Menu*********
==========
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Search
5.Show
6.Exit
Input pilihan ?
4
Inputkan data yang akan dicari :
10
Data yang dicari ada di dalam node 0
*********Main Menu*********
==========
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Search
5.Show
6.Exit
Input pilihan ?

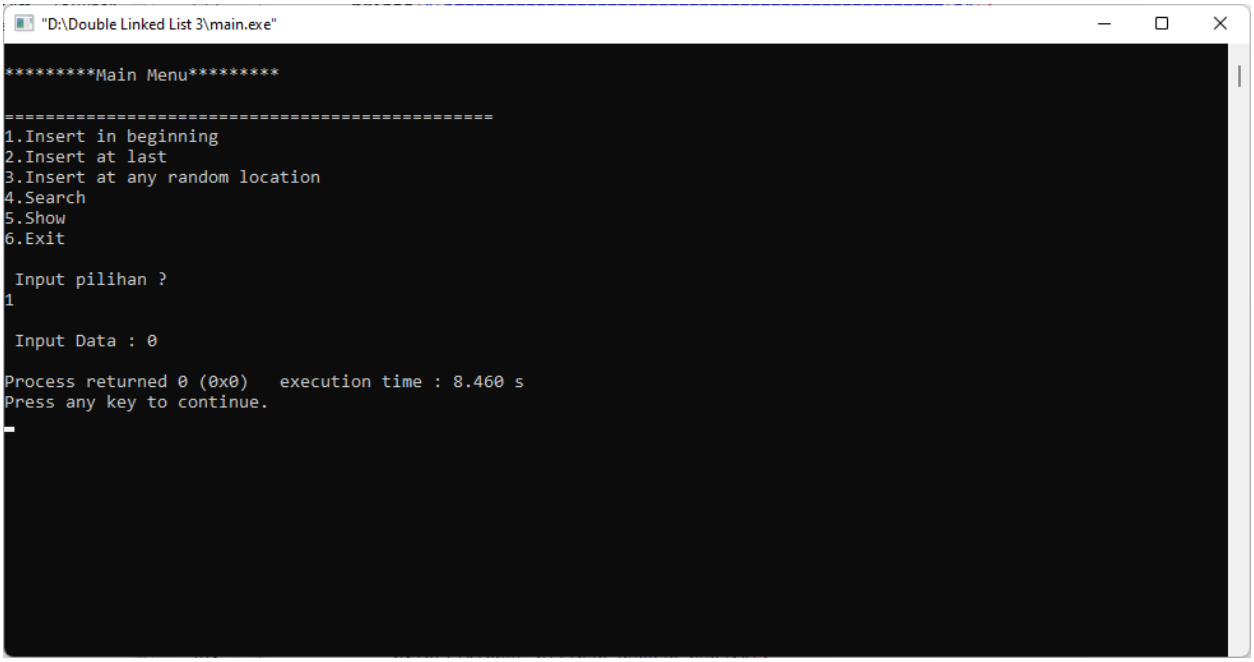

"D:\Double Linked List 3\main.exe"
Data yang dicari ada di dalam node 0
*********Main Menu*********
==
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Search
5.Show
6.Exit
Input pilihan ?
4
Inputkan data yang akan dicari :
30
Data yang dicari ada di dalam node 1
*********Main Menu*********
==
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Search
5.Show
6.Exit
Input pilihan ?

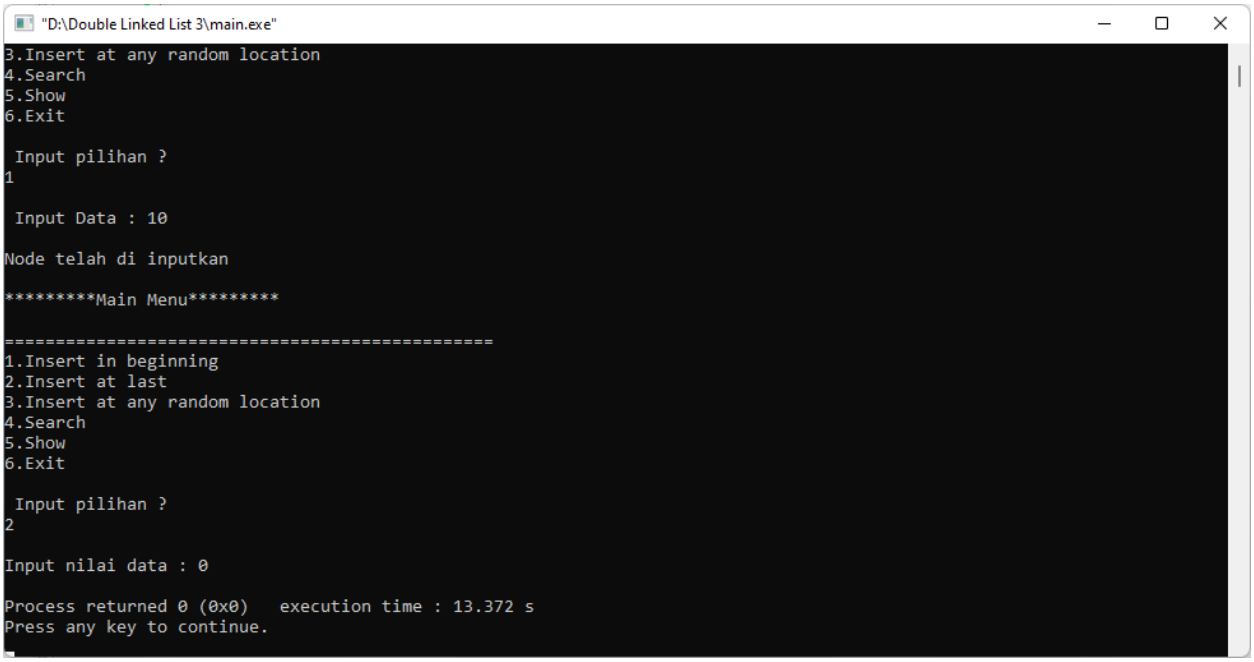

"D:\Double Linked List 3\main.exe"
Data yang dicari ada di dalam node 1
*********Main Menu*********
==
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Search
5.Show
6.Exit
Input pilihan ?
4
Inputkan data yang akan dicari :
50
Data yang dicari ada di dalam node 2
*********Main Menu*********
==
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Search
5.Show
6.Exit
Input pilihan ?

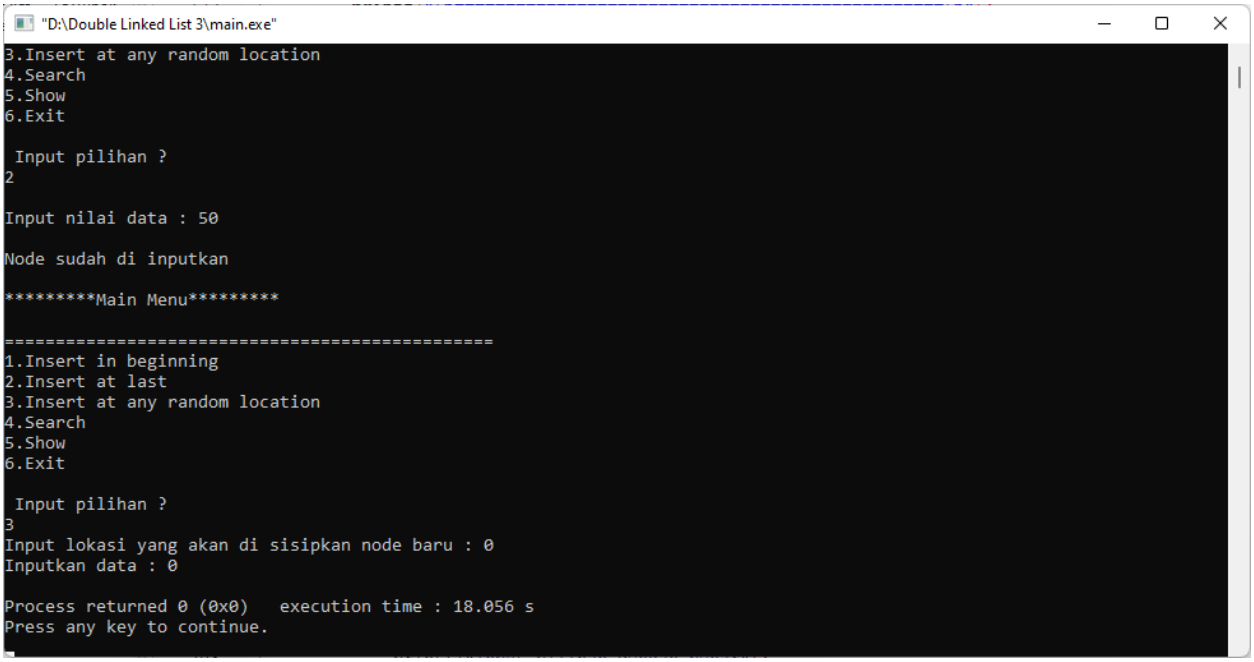




Dengan Memasukkan Data 0


"D:\Double Linked List 3\main.exe"
*********Main Menu*********
==========
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Search
5.Show
6.Exit
Input pilihan ?
1
Input Data : 0
Process returned 0 (0x0) execution time : 8.460 s
Press any key to continue.


"D:\Double Linked List 3\main.exe"
3.Insert at any random location
4.Search
5.Show
6.Exit
Input pilihan ?
1
Input Data : 10
Node telah di inputkan
*********Main Menu*********
==========
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Search
5.Show
6.Exit
Input pilihan ?
2
Input nilai data : 0
Process returned 0 (0x0) execution time : 13.372 s
Press any key to continue.


"D:\Double Linked List 3\main.exe"
3.Insert at any random location
4.Search
5.Show
6.Exit
Input pilihan ?
2
Input nilai data : 50
Node sudah di inputkan
*********Main Menu*********
==========
1.Insert in beginning
2.Insert at last
3.Insert at any random location
4.Search
5.Show
6.Exit
Input pilihan ?
3
Input lokasi yang akan di sisipkan node baru : 0
Inputkan data : 0
Process returned 0 (0x0) execution time : 18.056 s
Press any key to continue.