

LAPORAN PRAKTIKUM
Tugas Pendahuluan Modul 5
“Single Linked List – Bagian 2”



Disusun Oleh:
Muhammad Atsal Rizandri - 21104057
Struktur Data SE07-01 / SE05-02

Dosen :
Yudha Islami Sulistya

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM PURWOKERTO
2024

1. Mencari nilai tertentu dalam SLL. Membuat program yang mengizinkan pengguna memasukkan 6 elemen integer ke dalam list. Selanjutnya, mengimplementasikan function searchElement untuk mencari apakah sebuah nilai tertentu ada dalam list. Dalam program, pengguna memasukkan nilai ke 6 elemen, lalu memasukkan nilai yang ingin dicari, kemudian program akan memproses dengan pencarian nilai pada elemen menggunakan linier search.

Kode:

```
05_Single_Linked_List_Bagian_2 > TP > SOAL_01.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  struct Node
5  {
6      int data;
7      Node *next;
8  };
9
10 typedef Node *List;
11
12 void insertLast_21104057(List &L, int value)
13 {
14     Node *newNode = new Node;
15     newNode->data = value;
16     newNode->next = nullptr;
17
18     if (L == nullptr)
19     {
20         L = newNode;
21     }
22     else
23     {
24         Node *temp = L;
25         while (temp->next != nullptr)
26         {
27             temp = temp->next;
28         }
29         temp->next = newNode;
30     }
31 }
```

```

void searchElement_21104057(List L, int x)
{
    Node *current = L;
    int position = 1;

    while (current != nullptr)
    {
        if (current->data == x)
        {
            cout << "Element ditemukan pada alamat: " << current << " dan pada posisi: " << position << endl;
            return;
        }
        current = current->next;
        position++;
    }

    cout << "Element " << x << " tidak ada di dalam list." << endl;
}

int main()
{
    List L = nullptr;

    for (int y = 1; y <= 6; y++)
    {
        int value;
        cout << "Masukkan nilai pada elemen " << y << ": ";
        cin >> value;
        insertLast_21104057(L, value);
    }

    int searchValue;
    cout << "Masukkan nilai yang dicari: ";
    cin >> searchValue;

    searchElement_21104057(L, searchValue);

    return 0;
}

```

- Jika nilai ditemukan

```

Masukkan nilai pada elemen 1: 24
Masukkan nilai pada elemen 2: 72
Masukkan nilai pada elemen 3: 59
Masukkan nilai pada elemen 4: 12
Masukkan nilai pada elemen 5: 32
Masukkan nilai pada elemen 6: 66
Masukkan nilai yang dicari: 24
Element ditemukan pada alamat: 0xdb760 dan pada posisi: 1
PS D:\Data atsal\= Kuliah-ITTP-SE24' SMT 7\Perkuliahan\SD\Praktikum\STD_Muhammad_Atsal_Rizandri_21104057>

```

- Jika nilai tidak ditemukan

```

Masukkan nilai pada elemen 1: 24
Masukkan nilai pada elemen 2: 72
Masukkan nilai pada elemen 3: 59
Masukkan nilai pada elemen 4: 12
Masukkan nilai pada elemen 5: 32
Masukkan nilai pada elemen 6: 66
Masukkan nilai yang dicari: 9
Element 9 tidak ada di dalam list.
PS D:\Data atsal\= Kuliah-ITTP-SE24' SMT 7\Perkuliahan\SD\Praktikum\STD_Muhammad_Atsal_Rizandri_21104057>

```

2. Mengurutkan List menggunakan bubble sort. Membuat program dimana pengguna dapat memasukkan 5 elemen integer ke dalam list. Program mengimplementasikan procedure bubbleSortList untuk mengurutkan elemen-elemen dalam list dari nilai terkecil ke terbesar.

- Kode

```
SOAL_02.cpp U X
05_Single_Linked_List_Bagian_2 > TP > SOAL_02.cpp > bubbleSortList_21104057(List &L)
1  #include <iostream>
2  using namespace std;
3
4  struct Node
5  {
6      int data;
7      Node *next;
8  };
9
10 typedef Node *List;
11
12 // Fungsi memasukkan elemen ke dalam list di akhir
13 void insertLast_21104057(List &L, int value)
14 {
15     Node *newNode = new Node;
16     newNode->data = value;
17     newNode->next = nullptr;
18
19     if (L == nullptr)
20     {
21         L = newNode;
22     }
23     else
24     {
25         Node *temp = L;
26         while (temp->next != nullptr)
27         {
28             temp = temp->next;
29         }
30         temp->next = newNode;
31     }
32 }
```

```

// Prosedur pengurutan list menggunakan bubble sort
void bubbleSortList_21104057(List &L)
{
    if (L == nullptr)
        return;

    bool swapped;
    do
    {
        swapped = false;
        Node *current = L;

        while (current->next != nullptr)
        {
            if (current->data > current->next->data)
            {
                // Pertukaran data
                int temp = current->data;
                current->data = current->next->data;
                current->next->data = temp;
                swapped = true;
            }
            current = current->next;
        }
    } while (swapped);
}

void printList_21104057(List L)
{
    Node *temp = L;
    while (temp != nullptr)
    {
        cout << temp->data << " ";
        temp = temp->next;
    }
    cout << endl;
}

int main()
{
    List L = nullptr;

    for (int x = 1; x <= 5; x++)
    {
        int value;
        cout << "Masukkan nilai pada elemen " << x << ": ";
        cin >> value;
        insertLast_21104057(L, value);
    }

    cout << "List sebelum di-urut: ";
    printList_21104057(L);

    bubbleSortList_21104057(L);

    cout << "List setelah di-urut: ";
    printList_21104057(L);

    return 0;
}

```

- Output

```

Masukkan nilai pada elemen 1: 87
Masukkan nilai pada elemen 2: 12
Masukkan nilai pada elemen 3: 21
Masukkan nilai pada elemen 4: 78
Masukkan nilai pada elemen 5: 61
List sebelum di-urut: 87 12 21 78 61
List setelah di-urut: 12 21 61 78 87
PS D:\Data atsal\= Kuliah-ITTP-SE24' SMT 7\Perkuliah\SD\Praktikum\STD_Muhammad_Atsal_Rizandri_21104057>

```

3. Menambahkan elemen secara terurut. Membuat program dengan pengguna dapat memasukkan 4 elemen integer ke dalam list secara manual, lalu pengguna memasukkan elemen tambahan yang harus ditempatkan di posisi yang sesuai sehingga list tetap terurut.

- Kode

```
SOAL_03.cpp U X
05_Single_Linked_List_Bagian_2 > TP > SOAL_03.cpp > insertSorted_21104057(List &, int)
1  #include <iostream>
2  using namespace std;
3
4  struct Node
5  {
6      int data;
7      Node *next;
8  };
9
10 typedef Node *List;
11
12 // Fungsi memasukkan elemen ke dalam list secara terurut
13 void insertSorted_21104057(List &L, int value)
14 {
15     Node *newNode = new Node;
16     newNode->data = value;
17     newNode->next = nullptr;
18
19     // Menambahkan elemen di awal jika list kosong atau elemen baru lebih kecil dari elemen pertama
20     if (L == nullptr || L->data >= value)
21     {
22         newNode->next = L;
23         L = newNode;
24         cout << "Elemen " << value << " ditambahkan di awal list." << endl;
25     }
26     else
27     {
28         Node *current = L;
29         Node *prev = nullptr;
30
31         // Mencari posisi yang sesuai untuk elemen baru
32         while (current != nullptr && current->data < value)
33         {
34             prev = current;
35             current = current->next;
36         }
37
38         // Menambahkan elemen baru di antara prev dan current
39         newNode->next = current;
40         prev->next = newNode;
41         cout << "Elemen " << value << " ditambahkan ke dalam list." << endl;
42     }
43 }
```

```

void printList(List L)
{
    Node *temp = L;
    while (temp != nullptr)
    {
        cout << temp->data << " ";
        temp = temp->next;
    }
    cout << endl;
}

int main()
{
    List L = nullptr;

    cout << "Masukkan 4 elemen pertama ke dalam list" << endl;
    for (int x = 1; x <= 4; x++)
    {
        int value;
        cout << "Masukkan nilai elemen " << x << ": ";
        cin >> value;
        insertSorted_21104057(L, value);
    }

    cout << "List sebelum penambahan elemen baru: ";
    printList(L);

    int newValue;
    cout << "Masukkan elemen tambahan yang ingin ditambahkan ke dalam list: ";
    cin >> newValue;
    insertSorted_21104057(L, newValue);

    cout << "List setelah penambahan elemen baru: ";
    printList(L);

    return 0;
}

```

- Output

```

Masukkan 4 elemen pertama ke dalam list
Masukkan nilai elemen 1: 12
Elemen 12 ditambahkan di awal list.
Masukkan nilai elemen 1: 12
Elemen 12 ditambahkan di awal list.
Masukkan nilai elemen 2: 18
Elemen 12 ditambahkan di awal list.
Masukkan nilai elemen 2: 18
Masukkan nilai elemen 2: 18
Elemen 18 ditambahkan ke dalam list.
Masukkan nilai elemen 3: 22
Elemen 22 ditambahkan ke dalam list.
Masukkan nilai elemen 4: 30
Masukkan nilai elemen 3: 22
Elemen 22 ditambahkan ke dalam list.
Masukkan nilai elemen 4: 30
Elemen 22 ditambahkan ke dalam list.
Masukkan nilai elemen 4: 30
Masukkan nilai elemen 4: 30
Elemen 30 ditambahkan ke dalam list.
List sebelum penambahan elemen baru: 12 18 22 30
Masukkan elemen tambahan yang ingin ditambahkan ke dalam list: 13
Elemen 13 ditambahkan ke dalam list.
List setelah penambahan elemen baru: 12 13 18 22 30
PS D:\Data atsal\= Kuliah-ITTP-SE24\ SMT 7\Perkuliahan\SD\Praktikum\STD_Muhammad_Atsal_Rizandri_21104057>

```