Nama: Sheren Aulia Azahra

NIM: 103032400036

Kelas: IT-48-05

Jawaban Tugas Pendahuluan Modul 5

1. Pembuatan ADT

sll2.h sll2.cpp

```h
#ifndef SLL2_H_INCLUDED
#define SLL2_H_INCLUDED

typedef int infotype;
typedef struct elmlist *adr;

struct elmlist {
    infotype info;
    adr next;
};

struct list {
    adr first;
};

void searchElement_103032400036(list &L, infotype x);
void selectionSort_103032400036(list &L);
void insertSorted_103032400036(list &L, adr p);

#endif // SLL2_H_INCLUDED
```

2. Mencari Elemen Tertentu dalam List

sll2.h sll2.cpp

```cpp
#include <iostream>
#include "sll2.h"

using namespace std;

void searchElement_103032400036(list &L, infotype x){
    adr current;
    int position;

    current = L.first;
    position = 1;
    while (current != nullptr && current->info != x){
        position += 1;
        current = current->next;
    }
    if (current != nullptr){
        cout << "Elemen terdapat pada posisi ke- " << position
             << ", dengan nilai: " << current->info <<  endl;
    } else {
        cout << "Elemen tersebut tidak ada dalam list" << endl;
    }
}
```

3. Mengurutkan List menggunakan Selection Sort

```
void selectionSort_103032400036(list &L){
    adr p, min, temp;
    infotype x;

    p = L.first;
    while (p != nullptr){
        min = p;
        temp = p;
        while (temp != nullptr){
            if (temp->info < min->info){
                min = temp;
            }
            temp = temp->next;
        }
        x = p->info;
        p->info = min->info;
        min->info = x;
        p = p->next;
    }
}
```

4. Menambahkan Elemen secara Terurut

```
44
45  void insertSorted_103032400036(list &L, adr p){
46      adr q, prev;
47      bool found;
48
49      q = L.first;
50      found = false;
51      prev = nullptr;
52      while (q != nullptr && found == false) {
53          if (q->info < p->info) {
54              prev = q;
55              q = q->next;
56          } else {
57              found = true;
58          }
59      }
60      if (prev == nullptr){
61          p->next = L.first;
62          L.first = p;
63      } else {
64          p->next = q;
65          prev->next = p;
66      }
67  }
68
```