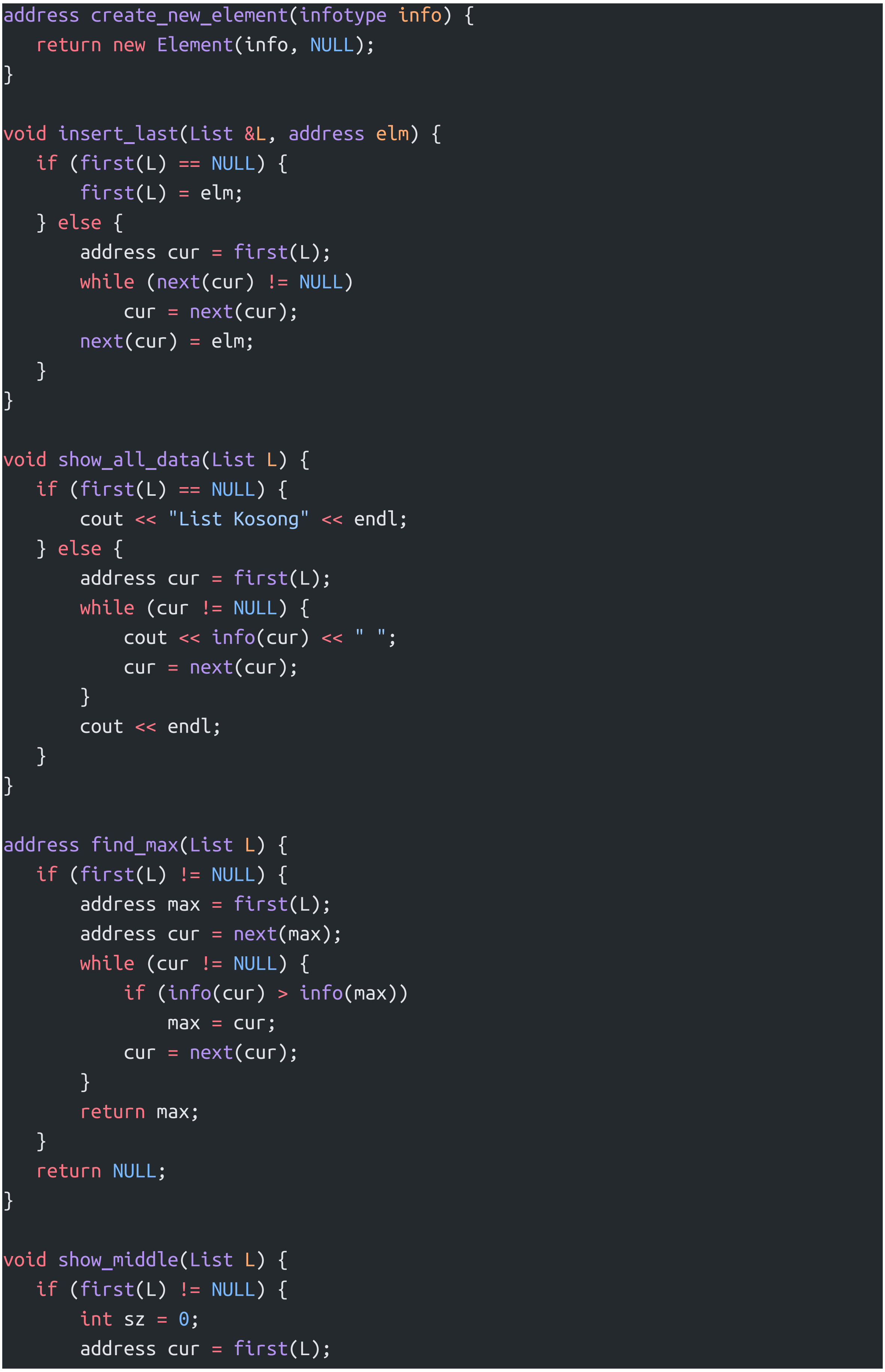
Nizam Abdullah
1301213232
IF-45-10

sll.h

```cpp
#include <iostream>

using namespace std;

#define first(L) (L).first
#define next(P) (P)->next
#define info(P) (P)->info

typedef int infotype;
typedef struct Element *address;

struct Element {
   infotype info;
   address next;
   Element(infotype info, Element *next){
       this->info = info;
       this->next = NULL;
   };
};
struct List {
   struct Element *first;
};

void create_list(List &L);
address create_new_element(infotype info);
void insert_last(List &L, address elm);
void show_all_data(List L);
address find_max(List L);
void show_middle(List L);
```

sll.cpp

```cpp
#include "sll.h"

void create_list(List &L) {
   first(L) = NULL;
}
```



```cpp
address create_new_element(infotype info) {
    return new Element(info, NULL);
}

void insert_last(List &L, address elm) {
    if (first(L) == NULL) {
        first(L) = elm;
    } else {
        address cur = first(L);
        while (next(cur) != NULL)
            cur = next(cur);
        next(cur) = elm;
    }
}

void show_all_data(List L) {
    if (first(L) == NULL) {
        cout << "List Kosong" << endl;
    } else {
        address cur = first(L);
        while (cur != NULL) {
            cout << info(cur) << " ";
            cur = next(cur);
        }
        cout << endl;
    }
}

address find_max(List L) {
    if (first(L) != NULL) {
        address max = first(L);
        address cur = next(max);
        while (cur != NULL) {
            if (info(cur) > info(max))
                max = cur;
            cur = next(cur);
        }
        return max;
    }
    return NULL;
}

void show_middle(List L) {
    if (first(L) != NULL) {
        int sz = 0;
        address cur = first(L);
```

```
       while (cur != NULL) {
           sz++;
           cur = next(cur);
       }
       cur = first(L);
       sz = sz / 2;
       while (sz--)
           cur = next(cur);
       cout << info(cur) << endl;
   } else {
       cout << "List Kosong" << endl;
   }
}
```

main.cpp

```
#include "sll.h"

#define JUM_DATA_PROMPT "Jumlah data yang ditambahkan: "
#define DATA_BARU_PROMPT "Masukkan data baru: "

infotype read_data(string msg) {
   infotype data;
   cout << msg;
   cin  >> data;
   return data;
}

int main() {
   int choice, n;
   List L;
   address max;

   create_list(L);

   for (;;) {
       cout << "== MENU ==\n"
               "1. Menambahkan N data baru\n"
               "2. Menampilkan semua data\n"
               "3. Menampilkan nilai maksimum\n"
               "4. Menampilkan nilai tengah\n"
               "0. Exit\n"
               "Masukkan menu: ";
       cin >> choice;
```

```
        switch (choice) {
            case 1:
                first(L) = NULL;
                n = read_data(JUM_DATA_PROMPT);
                while (n--)
                    insert_last(L,
create_new_element(read_data(DATA_BARU_PROMPT)));
                break;
            case 2:
                show_all_data(L);
                break;
            case 3:
                max = find_max(L);
                if (max != NULL)
                    cout << info(max) << endl;
                else
                    cout << "List Kosong" << endl;
                break;
            case 4:
                show_middle(L);
                break;
            case 0:
                goto EXIT;
                break;
            default:
                cout << "Invalid choice" << endl;
                break;
        }
    }

    EXIT:
    return 0;
}
```

output

```
== MENU ==
1. Menambahkan N data baru
2. Menampilkan semua data
3. Menampilkan nilai maksimum
4. Menampilkan nilai tengah
0. Exit
Masukkan menu: 1
Jumlah data yang ditambahkan: 7
Masukkan data baru: 1
Masukkan data baru: 2
Masukkan data baru: 3
Masukkan data baru: 4
Masukkan data baru: 5
Masukkan data baru: 6
Masukkan data baru: 7
== MENU ==
1. Menambahkan N data baru
2. Menampilkan semua data
3. Menampilkan nilai maksimum
4. Menampilkan nilai tengah
0. Exit
Masukkan menu: 2
1 2 3 4 5 6 7
== MENU ==
1. Menambahkan N data baru
2. Menampilkan semua data
3. Menampilkan nilai maksimum
4. Menampilkan nilai tengah
0. Exit
Masukkan menu: 3
7
== MENU ==
1. Menambahkan N data baru
2. Menampilkan semua data
3. Menampilkan nilai maksimum
4. Menampilkan nilai tengah
0. Exit
Masukkan menu: 4
4
== MENU ==
1. Menambahkan N data baru
2. Menampilkan semua data
3. Menampilkan nilai maksimum
4. Menampilkan nilai tengah
0. Exit
Masukkan menu: 0
```