```
======MENU=======
1. Menambah N data baru
2. Menampilkan semua data
3. Menampilkan semua data secara terbalik
4. Menampilkan saldo terkecil ke-2
0. Exit

Pilihan Menu : 1
Jumlah data yang akan ditambahkan : 3
Masukan data yang akan ditambahkan :
Nomor Rekening : 372846
Nama Nasabah : choi
Total Saldo : 15000

Masukan data yang akan ditambahkan :
Nomor Rekening : 928146
Nama Nasabah : lee
Total Saldo : 5000

Masukan data yang akan ditambahkan :
Nomor Rekening : 93864
Nama Nasabah : kang
Total Saldo : 10000
```

```
======MENU=======
1. Menambah N data baru
2. Menampilkan semua data
3. Menampilkan semua data secara terbalik
4. Menampilkan saldo terkecil ke-2
0. Exit

Pilihan Menu : 2
Data :
No. Rek : 93864
Nama : kang
Saldo : 10000
No. Rek : 928146
Nama : lee
Saldo : 5000
No. Rek : 372846
Nama : choi
Saldo : 15000
```

```
======MENU=======
1. Menambah N data baru
2. Menampilkan semua data
3. Menampilkan semua data secara terbalik
4. Menampilkan saldo terkecil ke-2
0. Exit

Pilihan Menu : 3
Data :
No. Rek : 372846
Nama : choi
Saldo : 15000
No. Rek : 928146
Nama : lee
Saldo : 5000
No. Rek : 93864
Nama : kang
Saldo : 10000
```

```
======MENU=======
1. Menambah N data baru
2. Menampilkan semua data
3. Menampilkan semua data secara terbalik
4. Menampilkan saldo terkecil ke-2
0. Exit

Pilihan Menu : 4
Data terkecil kedua :10000
```

main.cpp | MOD7.H | MOD7.cpp

```cpp
#ifndef MOD7_H_INCLUDED
#define MOD7_H_INCLUDED
#include <iostream>

using namespace std;

#define info(P) (P)->info
#define next(P) (P)->next
#define prev(P) (P)->prev
#define first(D) ((D).first)
#define last(D) ((D).last)

typedef struct nasabah infotype;
typedef struct element *adr;

struct nasabah{
    int norek;
    float saldo;
    string nama;
};

struct element{
    infotype info;
    adr next;
    adr prev;
};
```

```cpp
struct element{
    infotype info;
    adr next;
    adr prev;
};

struct List{
    adr first;
    adr last;
};

void CreateList_1301213072(List &D);
adr newElement_1301213072(infotype atk);
void show_1301213072(List D);
void insertFirst_1301213072(List &D, adr P);
void reverseList_1301213072(List &D);
float scndSmallest_1301213072(List &D);
int selectMenu_1301213072();

#endif // MOD7_H_INCLUDED
```

main.cpp | MOD7.H | MOD7.cpp

```cpp
#include "MOD7.H"

int main()
{
    List D;
    int pilihan = 0, K, x;
    adr P;
    infotype data;

    CreateList_1301213072(D);
    pilihan = selectMenu_1301213072();


  while(pilihan != 0){
    switch(pilihan){
    case 1:
        cout << "Jumlah data yang akan ditambahkan : ";
        cin >> x;
        for (int i = 1; i <= x; i++){
            cout << "Masukan data yang akan ditambahkan : " << endl;
            cout << "Nomor Rekening : ";
            cin >> data.norek;
            cout << "Nama Nasabah : ";
            cin >> data.nama;
            cout << "Total Saldo : ";
            cin >> data.saldo;
            cout << endl;
            P = newElement_1301213072(data);
            insertFirst_1301213072(D, P);
        }
        break;
    case 2:
        cout << "Data :" << endl;
        show_1301213072(D);
        cout << endl;
        break;
    case 3:
        cout << "Data :" << endl;
        reverseList_1301213072(D);
        cout << endl;
        break;
    case 4:
        cout << "Data terkecil kedua :";
        float Q = scndSmallest_1301213072(D);
        cout << Q << endl;
        break;
    }
    pilihan = selectMenu_1301213072();
  }
  cout << "done " << endl;

    return 0;
}
```

main.cpp MOD7.H MOD7.cpp

```cpp
#include "MOD7.H"


void CreateList_1301213072(List &D){
    first(D) = NULL;
    last(D) = NULL;

}

adr newElement_1301213072(infotype x){
    adr P = new element;
    info(P) = x;
    next(P) = NULL;
    prev(P) = NULL;
    return P;
}

void show_1301213072(List D){
    if (first(D) != NULL && last(D) != NULL){
        adr P = first(D);
        while (P != NULL){
            cout << "No. Rek : " << (info(P).norek) << endl;
            cout << "Nama : "<< (info(P).nama) << endl;
            cout << "Saldo : " << (info(P).saldo) << endl;
            P = next(P);
        }
        cout << endl;
    }else {
        cout << "List Kosong" << endl;
    }
}


void insertFirst_1301213072(List &D, adr P){
    if (first(D) == NULL && last(D) == NULL){
        first(D) = P;
        last(D) = P;
    }else {
        next(P) = first(D);
        prev(first(D)) = P;
        first(D) = P;
    }
}

void reverseList_1301213072(List &D){
    if (first(D) != NULL && last(D) != NULL){
        adr P = last(D);
        while (P != NULL){
            cout << "No. Rek : " << (info(P).norek) << endl;
            cout << "Nama : "<< (info(P).nama) << endl;
            cout << "Saldo : " << (info(P).saldo) << endl;
            P = prev(P);
        }
        cout << endl;
    }else {
        cout << "List Kosong" << endl;
    }
}

float scndSmallest_1301213072(List &D){
    adr P = first(D);
    adr Q = first(D);
    adr x = first(D);

    while (P != NULL){
        if (info(x).saldo >= info(P).saldo){
            Q = x;
            x = P;
        }
        P = next(P);
    }

    /*int norek = info(Q).norek;
    string nama = info(Q).nama;
    int saldo = info(Q).saldo;*/

    return info(Q).saldo;
}

int selectMenu_1301213072(){
    cout << endl << "======MENU=======" << endl;
    cout << "1. Menambah N data baru" << endl;
    cout << "2. Menampilkan semua data" << endl;
    cout << "3. Menampilkan semua data secara terbalik" << endl;
    cout << "4. Menampilkan saldo terkecil ke-2" << endl;
    cout << "0. Exit" << endl << endl;
    cout << "Pilihan Menu : ";

    int input = 0;
    cin >> input;

    return input;
}
```