

# **PRAKTIKUM STRUKTUR DATA**

## **LAPORAN TUGAS AKHIR PROGRAM**



Disusun Oleh :

Alvin Anugerah Pratama  
(22343019)

Dosen Pengampu :  
Randi Proska Sandra, S.Pd., M.Sc.  
Seksi : 202223430029

**PRODI INFORMATIKA**  
**DEPARTEMEN TEKNIK ELEKTRONIKA**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS NEGERI PADANG**  
**2023**

**Nama : Alvin Anugerah Pratama**

**NIM : 22343019**

**Prodi : Informatika NK**

**Nama Program : Program Rekening Nasabah Bank**

**Source Code Program :**

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

typedef struct Bank{
    char nama[60];
    int nomor;
    struct Bank* lanjut;
}Bank;

Bank* kepala = NULL;

void inputAntrian(int nomor, char nama[]){
    Bank* antrian = (Bank*)malloc(sizeof(Bank));
    antrian -> nomor = nomor;
    strcpy(antrian->nama, nama);
    antrian -> lanjut = NULL;

    if(kepala == NULL){
        kepala = antrian;
    }
    else{
        Bank* sekarang = kepala;
        while (sekarang -> lanjut != NULL){
            sekarang = sekarang -> lanjut;
        }
        sekarang -> lanjut = antrian;
    }
}

void cetakAntrian(){
    Bank* sekarang = kepala;
    printf("\nDaftar Data Nasabah : \n");
    printf("=====\n");
    printf("|No. Rek. | Nasabah | \n");
    printf("=====\n");
    while (sekarang != NULL) {
        printf("| %-6d | %-10s | \n", sekarang -> nomor, sekarang -> nama);
        sekarang = sekarang -> lanjut;
    }
}

void swap(Bank* a, Bank* b){
    int ksgnomor = a -> nomor;
    char ksgnama [60];
    strcpy (ksgnama, a -> nama);

    a -> nomor = b -> nomor;
    strcpy (a -> nama, b -> nama);

    b -> nomor = ksgnomor;
    strcpy (b -> nama, ksgnama);
}
```

```

}

void sortirNasabah(){
    if (kepala == NULL || kepala -> lanjut == NULL)
        return;
    int tukar;
    Bank* urut1;
    Bank* urutn = NULL;

    do {
        tukar = 0;
        urut1 = kepala;

        while (urut1 -> lanjut != urutn){
            if (urut1 -> nomor > urut1 -> lanjut -> nomor){
                swap(urut1, urut1 -> lanjut);
                tukar = 1;
            }
            urut1 = urut1 -> lanjut;
        }
        urutn = urut1;
    }while (tukar);
}

void ubah(int nomor){
    Bank* sekarang = kepala;
    while (sekarang != NULL){
        if (sekarang -> nomor == nomor){
            printf("Masukkan nama baru : ");
            scanf("%s", sekarang -> nama);
            printf("Data nasabah dengan nomor %d berhasil diubah \n",
nomor);
            return;
        }
        sekarang = sekarang -> lanjut;
    }
    printf("Data nasabah dengan nomor %d tidak ditemukan \n ", nomor);
}

void hapusData(int nomor){
    if (kepala == NULL){
        printf("Rekening Nasabah Kosong\n");
        return;
    }

    Bank* hapus = NULL;
    if (kepala -> nomor == nomor){
        hapus = kepala;
        kepala = kepala -> lanjut;
    }
    else {
        Bank* sekarang = kepala;
        while (sekarang -> lanjut != NULL){
            if (sekarang -> lanjut -> nomor == nomor){
                hapus = sekarang -> lanjut;
                sekarang -> lanjut = sekarang -> lanjut -> lanjut;
                break;
            }
            sekarang = sekarang -> lanjut;
        }
    }
}

```

```

    if (hapus != NULL){
        free(hapus);
        printf("Data nasabah %d berhasil dihapus\n", nomor);
    }
    else{
        printf("Data nasabah %d tidak ditemukan\n", nomor);
    }
}

Bank* cariData(int nomor){
    Bank* sekarang = kepala;
    while (sekarang != NULL){
        if (sekarang -> nomor == nomor){
            return sekarang;
        }
        sekarang = sekarang -> lanjut;
    }
    return NULL;
}

int main(){
    int pilih, nomor;
    char nama[60];

    printf("=====\n");
    printf("= Nama      : ALVIN  ANUGERAH  PRATAMA    =\n");
    printf("= NIM       : 22343019                    =\n");
    printf("= Mata Kuliah : Prak. Struktur Data         =\n");
    printf("= Sesi      : 202223430029                 =\n");
    printf("=====\n\n");

    printf(" Program Rekening Nasabah Bank\n");
    printf("=====\n\n");

    do{
        printf("Data Nasabah Bank : \n");
        printf("1. Tambah No. Rekening\n");
        printf("2. Lihat Rekening Nasabah\n");
        printf("3. Urutkan Data Nasabah\n");
        printf("4. Cari Data Nasabah \n");
        printf("5. Ubah Data Nasabah\n");
        printf("6. Hapus Data Nasabah\n");
        printf("0. Keluar\n");
        printf("Masukkan Pilihan Anda : ");
        scanf("%d", &pilih);
        switch(pilih){
            case 1:
                printf("\nMasukkan Nomor Rekening Anda : ");
                scanf("%d", &nomor);
                printf("Masukkan nama                : ");
                scanf("%s", &nama);
                inputAntrian(nomor, nama);
                printf("No. Rekening berhasil di tambahkan! \n\n");
                break;

            case 2:
                cetakAntrian();
                printf("\nDaftar Rekening Data Nasabah \n");
                break;

            case 3:

```

```

        sortirNasabah();
        printf("\nDaftar Nasabah Telah Diurutkan : \n\n");
        break;

    case 4:
        printf("Masukkan Nomor Rekening Yang Ingin Dicari : ");
        scanf("%d", &nomor);
        {
            Bank* hasil = cariData(nomor);
            if (hasil != NULL){
                printf("Rekening dengan nomor %d ditemukan: %s \n",
hasil -> nomor, hasil -> nama);
            }
            else {
                printf("Rekening dengan nomor %d Tidak di temukan
\n", nomor);
            }
        }

    case 5:
        printf("\nMasukkan Data Nasabahh yang ingin diubah : \n");
        scanf("%d", &nomor);
        ubah(nomor);
        break;

    case 6:
        printf("Masukkan Data Nasabah yang ingin dihapus : \n");
        scanf("%d", &nomor);
        hapusData(nomor);
        break;

    case 0:
        printf("\nThank You!!!\n");
        printf("Senang Berbisnis Dengan Anda...\n");
        break;

    default:
        printf("Nomor pilihan tidak valid\n\n");

    }
} while (pilih !=0);

return 0;
}

```

## Latar Belakang Program

Alasan saya memilih program di atas dikarenakan program sederhana untuk mengelola data rekening nasabah bank. Program ini menggunakan struktur data linked list untuk menyimpan data nasabah. Setiap data nasabah memiliki nomor rekening dan nama.

Latar belakang program ini adalah untuk membantu bank dalam mengelola data nasabah dengan cara memungkinkan bank untuk melakukan operasi dasar seperti menambahkan data nasabah, melihat daftar data nasabah, mengurutkan data nasabah berdasarkan nomor rekening, mencari data nasabah berdasarkan nomor rekening, mengubah data nasabah, dan menghapus data nasabah.

## Penjelasan Umum Program

Program di atas adalah sebuah program sederhana untuk mengelola **Data Rekening Nasabah Bank**. Program ini menggunakan bahasa pemrograman C dan menggunakan struktur data linked list untuk menyimpan informasi pelanggan. Program ini memiliki beberapa fitur utama, antara lain:

1. **Tambah No. Rekening:** Fitur ini memungkinkan pengguna untuk menambahkan data nasabah baru ke dalam sistem. Pengguna diminta untuk memasukkan nomor rekening dan nama nasabah, kemudian data tersebut akan ditambahkan ke dalam linked list.
2. **Lihat Rekening Nasabah:** Fitur ini akan mencetak seluruh data nasabah yang tersimpan dalam linked list. Data nasabah akan ditampilkan dalam bentuk tabel yang terdiri dari nomor rekening dan nama nasabah.
3. **Urutkan Data Nasabah:** Fitur ini akan mengurutkan data nasabah berdasarkan nomor rekening secara ascending menggunakan algoritma bubble sort. Dengan menggunakan fitur ini, data nasabah akan tersusun dengan rapi berdasarkan nomor rekeningnya.
4. **Cari Data Nasabah:** Fitur ini memungkinkan pengguna untuk mencari data nasabah berdasarkan nomor rekening. Jika nomor rekening yang dicari ditemukan, program akan menampilkan nomor rekening dan nama nasabah yang sesuai. Jika tidak ditemukan, program akan memberikan pesan bahwa nomor rekening tidak ditemukan.
5. **Ubah Data Nasabah:** Fitur ini memungkinkan pengguna untuk mengubah data nasabah berdasarkan nomor rekening. Pengguna diminta untuk memasukkan nomor rekening yang ingin diubah, kemudian program akan meminta pengguna memasukkan nama baru untuk nasabah tersebut. Jika nomor rekening ditemukan, data nasabah akan diubah sesuai dengan input pengguna.
6. **Hapus Data Nasabah:** Fitur ini memungkinkan pengguna untuk menghapus data nasabah berdasarkan nomor rekening. Pengguna diminta untuk memasukkan nomor rekening yang ingin dihapus. Jika nomor rekening ditemukan, data nasabah akan dihapus dari linked list. Jika tidak ditemukan, program akan memberikan pesan bahwa nomor rekening tidak ditemukan.

## Penjelasan Baris Program

### 1. Baris 1-3

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

Penjelasan :

- `#include <stdio.h>`: Digunakan untuk menyertakan file header `stdio.h`, yang berisi fungsi-fungsi standar untuk input dan output seperti `printf` dan `scanf`.

- `#include <stdlib.h>`: Digunakan untuk menyertakan file header `stdlib.h`, yang berisi fungsi-fungsi standar untuk alokasi memori dinamis, konversi tipe data, dan fungsi-fungsi utilitas lainnya seperti `malloc` dan `free`.
- `#include <string.h>`: Digunakan untuk menyertakan file header `string.h`, yang berisi fungsi-fungsi yang digunakan untuk manipulasi string, seperti `strcpy`, `strcat`, dan `strlen`.

## 2. Baris 5-11

```
typedef struct Bank{
    char nama[60];
    int nomor;
    struct Bank* lanjut;
}Bank;

Bank* kepala = NULL;
```

Penjelasan :

Baris berikut adalah membuat struct "Bank".

- Mendefinisikan sebuah tipe data baru yang disebut "Bank". Tipe data ini menggunakan kata kunci "typedef" agar dapat digunakan secara langsung tanpa menuliskan kata kunci "struct" ketika mendeklarasikan variabel dengan tipe data "Bank". Struktur data ini memiliki tiga anggota, yaitu:
  - `char nama[60]`: Menyimpan nama bank dengan panjang maksimum 60 karakter.
  - `int nomor`: Menyimpan nomor bank dalam bentuk bilangan bulat.
  - `struct Bank* lanjut`: Menyimpan alamat dari elemen berikutnya dalam linked list.
- Mendeklarasikan sebuah variabel pointer bernama "kepala" dengan tipe data "Bank". Variabel ini bertindak sebagai kepala atau awal dari linked list.
- Pada saat awal, variabel "kepala" diatur ke nilai `NULL`, menunjukkan bahwa linked list kosong.

## 3. Baris 13-29

```
void inputAntrian(int nomor, char nama[]){
    Bank* antrian = (Bank*)malloc(sizeof(Bank));
    antrian -> nomor = nomor;
    strcpy(antrian->nama, nama);
    antrian -> lanjut = NULL;

    if(kepala == NULL){
        kepala = antrian;
    }
    else{
        Bank* sekarang = kepala;
        while (sekarang -> lanjut != NULL){
            sekarang = sekarang -> lanjut;
        }
        sekarang -> lanjut = antrian;
    }
}
```

Penjelasan :

Baris tersebut mendefinisikan sebuah fungsi untuk melakukan input antrian, dengan nama inputAntrian. Fungsi ini bertujuan untuk menampilkan fungsi memasukkan data no rekening dan nama dari nasabah

#### 4. Baris 31-42

```
void cetakAntrian() {
    Bank* sekarang = kepala;
    printf("\nDaftar Data Nasabah      : \n");
    printf("===== \n");
    printf("|No. Rek. |    Nasabah | \n");
    printf("===== \n");
    while (sekarang != NULL) {
        printf("| %-6d | %-10s | \n", sekarang -> nomor, sekarang ->
nama);
        sekarang = sekarang -> lanjut;
    }
}
```

Penjelasan :

Baris program tersebut merupakan berfungsi untuk menampilkan data dari nasabah yang telah di masukkan sebelumnya.

Berikut adalah penjelasannya:

- Mendeklarasikan variabel pointer sekarang dengan tipe data Bank. Variabel ini digunakan sebagai penanda saat mengiterasi melalui linked list, dimulai dari kepala kepala
- Mencetak nomor rekening dan nama nasabah yang terkait dengan elemen saat ini dalam linked list menggunakan printf(). %6d dan %10s adalah format specifier yang digunakan untuk mengatur lebar output untuk nomor rekening dan nama nasabah
- Menggerakkan sekarang ke elemen berikutnya dalam linked list dengan mengakses lanjut dari sekarang. Ini memindahkan pointer sekarang ke elemen berikutnya untuk mempersiapkan iterasi berikutnya dalam loop while.

#### 5. Baris 44-54

```
void swap(Bank* a, Bank* b) {
    int ksgnomor = a -> nomor;
    char ksgnama [60];
    strcpy (ksgnama, a -> nama);

    a -> nomor = b -> nomor;
    strcpy (a -> nama, b -> nama);

    b -> nomor = ksgnomor;
    strcpy (b -> nama, ksgnama);
}
```



Penjelasan :

Source code di atas adalah sebuah fungsi bernama swap yang bertujuan untuk menukar nilai antara dua node (elemen) dalam linked list yang direpresentasikan oleh tipe data Bank. Fungsi ini menerima dua argumen, yaitu dua pointer ke dua node yang akan ditukar nilainya.

Berikut adalah penjelasan mengenai setiap bagian dari source code tersebut:

- Baris pertama mendeklarasikan variabel lokal ksgnomor dengan tipe data int. Variabel ini digunakan untuk menyimpan sementara nilai nomor dari node pertama.
- Baris kedua mendeklarasikan array lokal ksgnama dengan tipe data char dan panjang 60 karakter. Array ini digunakan untuk menyimpan sementara nilai nama dari node pertama.
- Baris ketiga mengkopikan nilai nomor dari node pertama (a) ke variabel lokal ksgnomor menggunakan operator -> untuk mengakses anggota nomor dari a.
- Baris keempat menggunakan fungsi strcpy untuk mengkopikan nilai nama dari node pertama (a) ke array lokal ksgnama. strcpy digunakan untuk menghindari masalah penugasan langsung antar array karakter.
- Baris keenam menukar nilai nomor antara node pertama (a) dan node kedua (b) dengan menggunakan operator -> untuk mengakses anggota nomor dari a dan b.
- Baris ketujuh menukar nilai nama antara node pertama (a) dan node kedua (b) dengan menggunakan fungsi strcpy. Hal ini dilakukan untuk menukar nilai string nama.
- Baris kedelapan menukar nilai nomor antara node kedua (b) dan nilai sementara ksgnomor yang awalnya berasal dari node pertama (a).
- Baris kesembilan menukar nilai nama antara node kedua (b) dan nilai sementara ksgnama yang awalnya berasal dari node pertama (a).

## 6. Baris 56-132

```
void sortirNasabah() {
    if (kepala == NULL || kepala -> lanjut == NULL)
        return;
    int tukar;
    Bank* urutl;
    Bank* urutn = NULL;

    do {
        tukar = 0;
        urutl = kepala;

        while (urutl -> lanjut != urutn) {
            if (urutl -> nomor > urutl -> lanjut -> nomor) {
                swap(urutl, urutl -> lanjut);
                tukar = 1;
            }
            urutl = urutl -> lanjut;
        }
        urutn = urutl;
    } while (tukar == 1);
}
```

```

        }
        urut1 = urut1 -> lanjut;
    }
    urutn = urut1;
}while (tukar);
}

void ubah(int nomor){
    Bank* sekarang = kepala;
    while (sekarang != NULL){
        if (sekarang -> nomor == nomor){
            printf("Masukkan nama baru : ");
            scanf("%s", sekarang -> nama);
            printf("Data nasabah dengan nomor %d berhasil diubah
\n", nomor);
            return;
        }
        sekarang = sekarang -> lanjut;
    }
    printf("Data nasabah dengan nomor %d tidak ditemukan \n ",
nomor);
}

void hapusData(int nomor){
    if (kepala == NULL){
        printf("Rekening Nasabah Kosong\n");
        return;
    }

    Bank* hapus = NULL;
    if (kepala -> nomor == nomor){
        hapus = kepala;
        kepala = kepala -> lanjut;
    }
    else {
        Bank* sekarang = kepala;
        while (sekarang -> lanjut != NULL){
            if (sekarang -> lanjut -> nomor == nomor){
                hapus = sekarang -> lanjut;
                sekarang -> lanjut = sekarang -> lanjut -> lanjut;
                break;
            }
            sekarang = sekarang -> lanjut;
        }
    }
    if (hapus != NULL){
        free(hapus);
        printf("Data nasabah %d berhasil dihapus\n", nomor);
    }
    else{
        printf("Data nasabah %d tidak ditemukan\n", nomor);
    }
}

Bank* cariData(int nomor){
    Bank* sekarang = kepala;
    while (sekarang != NULL){
        if (sekarang -> nomor == nomor){
            return sekarang;
        }
        sekarang = sekarang -> lanjut;
    }
}

```

```

    }
    return NULL;
}

```

Penjelasan :

- `void sortirNasabah()`: Fungsi ini bertanggung jawab untuk mengurutkan daftar nasabah dalam linked list berdasarkan nomor rekening. Fungsi ini menggunakan algoritma pengurutan Bubble Sort. Pertama, fungsi memeriksa apakah linked list kosong atau hanya memiliki satu elemen. Jika ya, fungsi akan mengembalikan kontrol. Jika tidak, fungsi akan melakukan pengurutan dengan menukar posisi node jika nomor rekening pada node saat ini lebih besar dari nomor rekening pada node berikutnya. Iterasi dilakukan hingga tidak ada lagi pertukaran yang dilakukan selama satu iterasi penuh. Ini memastikan bahwa daftar nasabah diurutkan secara ascending (menaik) berdasarkan nomor rekening.
- `void ubah(int nomor)`: Fungsi ini bertanggung jawab untuk mengubah nama nasabah dengan nomor rekening tertentu dalam linked list. Fungsi ini menerima nomor rekening sebagai argumen. Fungsi akan mencari node dengan nomor rekening yang sesuai dalam linked list. Jika ditemukan, fungsi akan meminta input pengguna untuk memasukkan nama baru dan mengubah nama nasabah dalam node yang sesuai. Jika nomor rekening tidak ditemukan, fungsi akan mencetak pesan yang sesuai.
- `void hapusData(int nomor)`: Fungsi ini bertanggung jawab untuk menghapus data nasabah dengan nomor rekening tertentu dari linked list. Fungsi ini menerima nomor rekening sebagai argumen. Fungsi akan memeriksa apakah linked list kosong. Jika ya, fungsi akan mencetak pesan yang sesuai dan mengembalikan kontrol. Jika tidak, fungsi akan mencari node dengan nomor rekening yang sesuai dalam linked list. Jika ditemukan, fungsi akan menghapus node tersebut dengan memperbarui pointer sebelumnya. Jika nomor rekening tidak ditemukan, fungsi akan mencetak pesan yang sesuai.
- `Bank* cariData(int nomor)`: Fungsi ini bertanggung jawab untuk mencari dan mengembalikan node dengan nomor rekening tertentu dalam linked list. Fungsi ini menerima nomor rekening sebagai argumen. Fungsi akan melakukan iterasi melalui linked list untuk mencari node dengan nomor rekening yang sesuai. Jika ditemukan, fungsi akan mengembalikan pointer ke node tersebut. Jika nomor rekening tidak ditemukan, fungsi akan mengembalikan nilai NULL.

## 7. Baris 134-216

```

int main(){
    int pilih, nomor;
    char nama[60];

    printf("=====\n");
    printf("=  Nama      :  ALVIN  ANUGERAH  PRATAMA  =\n");
    printf("=  NIM        :  22343019  =\n");
}

```

```

printf("= Mata Kuliah : Prak. Struktur Data      =\n");
printf("= Sesi          : 202223430029          =\n");
printf("=====\\n\\n");

printf(" Program Rekening Nasabah Bank\\n");
printf("=====\\n\\n");

do{
    printf("Data Nasabah Bank : \\n");
    printf("1. Tambah No. Rekening\\n");
    printf("2. Lihat Rekening Nasabah\\n");
    printf("3. Urutkan Data Nasabah\\n");
    printf("4. Cari Data Nasabah \\n");
    printf("5. Ubah Data Nasabah\\n");
    printf("6. Hapus Data Nasabah\\n");
    printf("0. Keluar\\n");
    printf("Masukkan Pilihan Anda : ");
    scanf("%d", &pilih);
    switch(pilih){
        case 1:
            printf("\\nMasukkan Nomor Rekening Anda : ");
            scanf("%d", &nomor);
            printf("Masukkan nama                : ");
            scanf("%s", &nama);
            inputAntrian(nomor, nama);
            printf("No. Rekening berhasil di tambahkan!
\\n\\n");
            break;
        case 2:
            cetakAntrian();
            printf("\\nDaftar Rekening Data Nasabah \\n");
            break;

        case 3:
            sortirNasabah();
            printf("\\nDaftar Nasabah Telah Diurutkan : \\n\\n");
            break;

        case 4:
            printf("Masukkan Nomor Rekening Yang Ingin
Dicari : ");
            scanf("%d", &nomor);
            {
                Bank* hasil = cariData(nomor);
                if (hasil != NULL){
                    printf("Rekening dengan nomor %d
ditemukan: %s \\n", hasil -> nomor, hasil -> nama);
                }
                else {
                    printf("Rekening dengan nomor %d Tidak di
temukan \\n", nomor);
                }
            }

        case 5:
            printf("\\nMasukkan Data Nasabah yang ingin
diubah : \\n");
            scanf("%d", &nomor);
            ubah(nomor);
            break;
    }
}

```

```

        case 6:
            printf("Masukkan Data Nasabah yang ingin dihapus :
\n");

            scanf("%d", &nomor);
            hapusData(nomor);
            break;

        case 0:
            printf("\nThank You!!!\n");
            printf("Senang Berbisnis Dengan Anda...\n");
            break;

        default:
            printf("Nomor pilihan tidak valid\n\n");

    }
} while (pilih !=0);

return 0;
}

```

Penjelasan :

Berikut adalah penjelasan dari menu int(main):

Source code di atas adalah implementasi utama dari program rekening nasabah bank. Ini adalah fungsi main() yang berisi logika program untuk menginteraksi dengan pengguna dan melakukan operasi pada linked list yang menyimpan data nasabah.

- Mencetak informasi mengenai pembuat program, seperti nama, NIM, mata kuliah, dan sesi.
- Loop do-while yang berjalan selama pengguna tidak memilih opsi "Keluar" (pilih != 0). Di dalam loop ini, pengguna diminta untuk memilih opsi melalui scanf("%d", &pilih).
- Ketika pengguna memilih opsi "Tambah No. Rekening" (pilih == 1). Pengguna diminta untuk memasukkan nomor rekening dan nama nasabah, kemudian fungsi inputAntrian() dipanggil untuk menambahkan data nasabah ke linked list.
- Ketika pengguna memilih opsi "Lihat Rekening Nasabah" (pilih == 2). Fungsi cetakAntrian() dipanggil untuk mencetak daftar rekening nasabah yang ada dalam linked list.
- Ketika pengguna memilih opsi "Urutkan Data Nasabah" (pilih == 3). Fungsi sortirNasabah() dipanggil untuk mengurutkan daftar nasabah dalam linked list berdasarkan nomor rekening.
- Ketika pengguna memilih opsi "Cari Data Nasabah" (pilih == 4). Pengguna diminta untuk memasukkan nomor rekening yang ingin dicari. Fungsi cariData() dipanggil untuk mencari data nasabah dengan nomor rekening yang sesuai dalam linked list. Hasil pencarian dicetak ke layar.

- Ketika pengguna memilih opsi "Ubah Data Nasabah" (pilih == 5). Pengguna diminta untuk memasukkan nomor rekening data nasabah yang ingin diubah. Fungsi ubah() dipanggil untuk mengubah data nasabah dengan nomor rekening yang sesuai dalam linked list.
- Ketika pengguna memilih opsi "Hapus Data Nasabah" (pilih == 6). Pengguna diminta untuk memasukkan nomor rekening data nasabah yang ingin dihapus. Fungsi hapusData() dipanggil untuk menghapus data nasabah dengan nomor rekening yang sesuai dalam linked list.
- Ketika pengguna memilih opsi "Keluar" (pilih == 0). Pesan penutup dicetak ke layar.