



Program Studi Teknik Elektro ITB

Nama Kuliah (Kode) : Pemecahan Masalah dengan C (EL2008)
Tahun / Semester : 2023-2024 / Genap
Nama Tugas : PR Spesial Lebaran
Tema : *Linked List*
Nama / NIM : Pradigta Hisyam Ramadhan / 18322008

Tugas Spesial Lebaran

A. Source Code

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX_STRING 100

typedef struct Siswa{
    char *nama;
    int NIM;
    int matematika;
    int fisika;
    int biologi;
    int kimia;
    int akuntansi;
    int sosiologi;
    struct Siswa *next; // pointer untuk menyimpan alamat node selanjutnya
}Siswa;

void addStudent(Siswa **head, int id, char *nama, int nilai_mat, int nilai_fis,
int nilai_bio, int nilai_kim, int nilai_akuntansi, int nilai_sosiologi){
    Siswa *mahasiswa_baru = (Siswa*)malloc(sizeof(Siswa));
    if (mahasiswa_baru == NULL){
        printf("Alokasi memori gagal\n");
        exit(1);
    }

    // Alokasi memori untuk nama
    mahasiswa_baru->nama = strdup(nama);
    if (mahasiswa_baru->nama == NULL) {
        printf("Alokasi memori gagal\n");
        free(mahasiswa_baru);
        exit(1);
    }

    mahasiswa_baru->NIM = id;
    mahasiswa_baru->matematika = nilai_mat;
    mahasiswa_baru->fisika = nilai_fis;
    mahasiswa_baru->biologi = nilai_bio;
    mahasiswa_baru->kimia = nilai_kim;
    mahasiswa_baru->akuntansi = nilai_akuntansi;
    mahasiswa_baru->sosiologi = nilai_sosiologi;
```

```

mahasiswa_baru->next = NULL;

// Inisiasi data pada awal linked list, jika head belum terisi
if (*head == NULL){
    *head = mahasiswa_baru;
}
// apabila head sudah terisi
else{
    Siswa *temp = *head;
    while (temp->next != NULL){
        temp = temp->next;
    }
    temp->next = mahasiswa_baru;
}
}

void insertStudent(Siswa **head, int id, char *nama, int nilai_mat, int
nilai_fis, int nilai_bio, int nilai_kim, int nilai_akuntansi, int
nilai_sosiologi, int posisi){
    Siswa *mahasiswa_baru = (Siswa*)malloc(sizeof(Siswa));
    if (mahasiswa_baru == NULL){
        printf("Alokasi memori gagal\n");
        exit(1);
    }

    // Alokasi memori untuk nama
    mahasiswa_baru->nama = strdup(nama);
    if (mahasiswa_baru->nama == NULL) {
        printf("Alokasi memori gagal\n");
        free(mahasiswa_baru);
        exit(1);
    }

    mahasiswa_baru->NIM = id;
    mahasiswa_baru->matematika = nilai_mat;
    mahasiswa_baru->fisika = nilai_fis;
    mahasiswa_baru->biologi = nilai_bio;
    mahasiswa_baru->kimia = nilai_kim;
    mahasiswa_baru->akuntansi = nilai_akuntansi;
    mahasiswa_baru->sosiologi = nilai_sosiologi;

    // Kasus belum ada data sama sekali pada linked list
    if (posisi == 1 || *head == NULL){
        mahasiswa_baru->next = *head;
        *head = mahasiswa_baru;
    }
    else{
        Siswa *current = *head;
        int count = 1;
        while (count < posisi-1 && current->next != NULL){
            current = current->next;

```

```

        count++;
    }
    mahasiswa_baru->next = current->next;
    current->next = mahasiswa_baru;
}
}

void searchStudent(Siswa *head, int search_nim){
    Siswa *current = head;
    int found = 0;
    while (current != NULL){
        if (current->NIM == search_nim){
            printf("\nMahasiswa ditemukan!\n");
            printf("NIM: %d\nNama: %s\nMatematika: %d\nFisika: %d\nBiologi: %d\nKimia: %d\nAkuntansi: %d\nSosiologi: %d\n\n", current->NIM,
                current->nama, current->matematika, current->fisika, current->biologi, current->kimia, current->akuntansi, current->sosiologi);
            found = 1;
            break;
        }
        current = current->next;
    }
    if(!found){
        printf("Mahasiswa dengan NIM %d tidak ada :(\n", search_nim);
    }
}

void updateMarks(Siswa *head, int nim, int nilai_mat, int nilai_fis, int nilai_bio, int nilai_kim, int nilai_akuntansi, int nilai_sosiologi){
    Siswa *current = head;
    int found = 0;
    while (current != NULL){
        if (current->NIM == nim){
            current->matematika = nilai_mat;
            current->fisika = nilai_fis;
            current->biologi = nilai_bio;
            current->kimia = nilai_kim;
            current->akuntansi = nilai_akuntansi;
            current->sosiologi = nilai_sosiologi;
            printf("Nilai mahasiswa dengan NIM %d sudah diperbaharui!\n", nim);
            found = 1;
            break;
        }
        current = current->next;
    }
    if(!found){
        printf("Mahasiswa dengan NIM %d tidak ada :(", nim);
    }
}
}

```

```

void deleteStudent(Siswa **head, int nim){
    Siswa *current = *head;
    Siswa *prev = NULL;

    // Mencari mahasiswa pada node linked list dengan NIM yang diberikan
    while (current != NULL && current->NIM != nim){
        prev = current;
        current = current->next;
    }

    // Apabila node ditemukan
    if (current != NULL){
        // Node yang dihapus adalah head
        if (prev == NULL){
            *head = current->next;
        }
        // Apabila berada di tengah
        else{
            prev->next = current->next;
        }
        // Bebaskan memori
        free(current->nama);
        free(current);
        printf("Mahasiswa dengan NIM %d berhasil dihapus!\n", nim);
    }
    else{
        printf("Mahasiswa dengan NIM %d tidak ada :(", nim);
    }
}

void calculateAverage(Siswa *head){
    int total_mahasiswa = 0;
    int total_mat = 0, total_fis = 0, total_kim = 0, total_bio = 0, total_sosio =
0, total_akuntansi = 0;
    Siswa *current = head;
    while (current != NULL){
        total_mahasiswa++;
        total_mat += current->matematika;
        total_fis += current->fisika;
        total_kim += current->kimia;
        total_bio += current->biologi;
        total_sosio += current->sosiologi;
        total_akuntansi += current->akuntansi;
        current = current->next;
    }

    printf("\nRata-Rata nilai mahasiswa\n");
    printf("Matematika: %.2f\n", (float)total_mat/total_mahasiswa);
    printf("Fisika: %.2f\n", (float)total_fis/total_mahasiswa);
    printf("Kimia: %.2f\n", (float)total_kim/total_mahasiswa);
    printf("Biologi: %.2f\n", (float)total_bio/total_mahasiswa);

```

```

printf("Sosiologi: %.2f\n", (float)total_sosio/total_mahasiswa);
printf("Akuntansi: %.2f\n", (float)total_akuntansi/total_mahasiswa);
}

void displayStudents(Siswa *head){
    Siswa *current = head;
    printf("\nData Mahasiswa Institut Tidak Bobok 2044/2045\n");
    while (current != NULL){
        printf("NIM: %d, Nama: %s, Mat: %d, Fis: %d, Kim: %d, Bio: %d, Sosio: %d,
Akuntansi: %d\n",
            current->NIM, current->nama, current->matematika, current->fisika,
current->kimia, current->biologi, current->sosiologi, current->akuntansi);
        current = current->next;
    }
    printf("\n");
}

// Fungsi utama
int main(void){
    Siswa *head = NULL;
    int nim, mat, fis, kim, bio, sosio, akuntansi;
    int opsi, idx_nim;
    char nama[MAX_STRING];

    do{
        printf("\nSelamat Datang di Program Manajemen Mahasiswa\n");
        printf("1. Tambahkan Mahasiswa (Sisipkan di akhir)\n");
        printf("2. Tambahkan Mahasiswa (Sisipkan di tengah)\n");
        printf("3. Cari Mahasiswa berdasarkan NIM\n");
        printf("4. Update nilai mahasiswa berdasarkan NIM\n");
        printf("5. Hapus data mahasiswa berdasarkan NIM\n");
        printf("6. Hitung rerata tiap mata kuliah\n");
        printf("7. Tampilkan semua data mahasiswa\n");
        printf("8. Exit\n");

        // Mendapatkan user input
        printf("Enter your choice: ");
        scanf("%d", &opsi); // Read user's choice

        switch (opsi) {
            case 1:
                // Mendapatkan NIM mahasiswa
                printf("Masukkan NIM mahasiswa: ");
                scanf("%d", &nim);

                // Mendapatkan nama mahasiswa
                printf("Masukkan nama mahasiswa: ");
                getchar(); // mengambil karakter newline dari input sebelumnya
                fgets(nama, MAX_STRING, stdin);

```

```

        nama[strcspn(nama, "\n")] = '\0';

        // Mendapatkan nilai masing-masing mata kuliah
        printf("Masukkan nilai matematika: ");
        scanf("%d", &mat);

        printf("Masukkan nilai fisika: ");
        scanf("%d", &fis);

        printf("Masukkan nilai biologi: ");
        scanf("%d", &bio);

        printf("Masukkan nilai kimia: ");
        scanf("%d", &kim);

        printf("Masukkan nilai akuntansi: ");
        scanf("%d", &akuntansi);

        printf("Masukkan nilai sosiologi: ");
        scanf("%d", &sosio);

        addStudent(&head, nim, nama, mat, fis, bio, kim, akuntansi,
sosio);

        break;

    case 2:
        int posisi;

        printf("Pilih posisi untuk menyisipkan data mahasiswa: ");
        scanf("%d", &posisi);

        // Mendapatkan NIM mahasiswa
        printf("Masukkan NIM mahasiswa: ");
        scanf("%d", &nim);

        // Mendapatkan nama mahasiswa
        printf("Masukkan nama mahasiswa: ");
        getchar(); // mengambil karakter newline dari input sebelumnya
        fgets(nama, MAX_STRING, stdin);
        nama[strcspn(nama, "\n")] = '\0';

        // Mendapatkan nilai masing-masing mata kuliah
        printf("Masukkan nilai matematika: ");
        scanf("%d", &mat);

        printf("Masukkan nilai fisika: ");
        scanf("%d", &fis);

        printf("Masukkan nilai biologi: ");
        scanf("%d", &bio);

```

```

        printf("Masukkan nilai kimia: ");
        scanf("%d", &kim);

        printf("Masukkan nilai akuntansi: ");
        scanf("%d", &akuntansi);

        printf("Masukkan nilai sosiologi: ");
        scanf("%d", &sosio);

        insertStudent(&head, nim, nama, mat, fis, bio, kim, akuntansi,
sosio, posisi);
        break;

    case 3:
        printf("Masukkan NIM mahasiswa yang akan dicari: ");
        scanf("%d", &idx_nim);
        searchStudent(head, idx_nim);
        break;
    case 4:
        printf("Masukkan NIM mahasiswa yang akan diperbaharui datanya:
");

        scanf("%d", &idx_nim);

        // Mendapatkan nilai baru dari mahasiswa
        printf("Masukkan nilai matematika: ");
        scanf("%d", &mat);

        printf("Masukkan nilai fisika: ");
        scanf("%d", &fis);

        printf("Masukkan nilai biologi: ");
        scanf("%d", &bio);

        printf("Masukkan nilai kimia: ");
        scanf("%d", &kim);

        printf("Masukkan nilai akuntansi: ");
        scanf("%d", &akuntansi);

        printf("Masukkan nilai sosiologi: ");
        scanf("%d", &sosio);

        // memanggil fungsi
        updateMarks(head, idx_nim, mat, fis, bio, kim, akuntansi, sosio);
        break;
    case 5:
        printf("Masukkan NIM mahasiswa yang akan dihapus datanya: ");
        scanf("%d", &idx_nim);

        // memanggil fungsi untuk menghapus node
        deleteStudent(&head, idx_nim);

```

```

        break;
    case 6:
        calculateAverage(head);
        break;
    case 7:
        displayStudents(head);
        break;
    case 8:
        printf("Keluar dari program, have a nice day!\n");
        break;
    default:
        printf("Opsi yang anda masukkan tidak valid, silakan ketik angka
antara 1-8!\n");
    }
} while (opsi != 8);

// Bebaskan memori untug masing-masing nama mahasiswa dan node
Siswa *temp;
while (head != NULL){
    temp = head;
    head = head->next;
    free(temp->nama);
    free(temp);
}

return 0;
}

```


B. Hasil Eksekusi Program

Kalimat yang diberi garis bawah adalah input dari pengguna

Selamat Datang di Program Manajemen Mahasiswa

1. Tambahkan Mahasiswa (Sisipkan di akhir)
2. Tambahkan Mahasiswa (Sisipkan di tengah)
3. Cari Mahasiswa berdasarkan NIM
4. Update nilai mahasiswa berdasarkan NIM
5. Hapus data mahasiswa berdasarkan NIM
6. Hitung rerata tiap mata kuliah
7. Tampilkan semua data mahasiswa
8. Exit

Enter your choice: 1

Masukkan NIM mahasiswa: 18322008

Masukkan nama mahasiswa: Digta

Masukkan nilai matematika: 87

Masukkan nilai fisika: 67

Masukkan nilai biologi: 48

Masukkan nilai kimia: 59

Masukkan nilai akuntansi: 90

Masukkan nilai sosiologi: 90

Selamat Datang di Program Manajemen Mahasiswa

1. Tambahkan Mahasiswa (Sisipkan di akhir)
2. Tambahkan Mahasiswa (Sisipkan di tengah)
3. Cari Mahasiswa berdasarkan NIM
4. Update nilai mahasiswa berdasarkan NIM
5. Hapus data mahasiswa berdasarkan NIM
6. Hitung rerata tiap mata kuliah
7. Tampilkan semua data mahasiswa
8. Exit

Enter your choice: 1

Masukkan NIM mahasiswa: 18322001

Masukkan nama mahasiswa: Ronaldo siu

Masukkan nilai matematika: 87

Masukkan nilai fisika: 67

Masukkan nilai biologi: 88

Masukkan nilai kimia: 90

Masukkan nilai akuntansi: 89

Masukkan nilai sosiologi: 90

Selamat Datang di Program Manajemen Mahasiswa

1. Tambahkan Mahasiswa (Sisipkan di akhir)
2. Tambahkan Mahasiswa (Sisipkan di tengah)
3. Cari Mahasiswa berdasarkan NIM
4. Update nilai mahasiswa berdasarkan NIM
5. Hapus data mahasiswa berdasarkan NIM
6. Hitung rerata tiap mata kuliah
7. Tampilkan semua data mahasiswa
8. Exit

Enter your choice: 2

Pilih posisi untuk menyisipkan data mahasiswa: 2

Masukkan NIM mahasiswa: 18322016

Masukkan nama mahasiswa: zaki

Masukkan nilai matematika: 89
Masukkan nilai fisika: 76
Masukkan nilai biologi: 99
Masukkan nilai kimia: 80
Masukkan nilai akuntansi: 100
Masukkan nilai sosiologi: 100

Selamat Datang di Program Manajemen Mahasiswa

1. Tambahkan Mahasiswa (Sisipkan di akhir)
2. Tambahkan Mahasiswa (Sisipkan di tengah)
3. Cari Mahasiswa berdasarkan NIM
4. Update nilai mahasiswa berdasarkan NIM
5. Hapus data mahasiswa berdasarkan NIM
6. Hitung rerata tiap mata kuliah
7. Tampilkan semua data mahasiswa
8. Exit

Enter your choice: 3

Masukkan NIM mahasiswa yang akan dicari: 18322009

Mahasiswa dengan NIM 18322009 tidak ada :(

Selamat Datang di Program Manajemen Mahasiswa

1. Tambahkan Mahasiswa (Sisipkan di akhir)
2. Tambahkan Mahasiswa (Sisipkan di tengah)
3. Cari Mahasiswa berdasarkan NIM
4. Update nilai mahasiswa berdasarkan NIM
5. Hapus data mahasiswa berdasarkan NIM
6. Hitung rerata tiap mata kuliah
7. Tampilkan semua data mahasiswa
8. Exit

Enter your choice: 3

Masukkan NIM mahasiswa yang akan dicari: 18322008

Mahasiswa ditemukan!

NIM: 18322008

Nama: Digta

Matematika: 87

Fisika: 67

Biologi: 48

Kimia: 59

Akuntansi: 90

Sosiologi: 90

Selamat Datang di Program Manajemen Mahasiswa

1. Tambahkan Mahasiswa (Sisipkan di akhir)
2. Tambahkan Mahasiswa (Sisipkan di tengah)
3. Cari Mahasiswa berdasarkan NIM
4. Update nilai mahasiswa berdasarkan NIM
5. Hapus data mahasiswa berdasarkan NIM
6. Hitung rerata tiap mata kuliah
7. Tampilkan semua data mahasiswa
8. Exit

Enter your choice: 4

Masukkan NIM mahasiswa yang akan diperbaharui datanya: 18322001

Masukkan nilai matematika: 100

Masukkan nilai fisika: 100
Masukkan nilai biologi: 100
Masukkan nilai kimia: 100
Masukkan nilai akuntansi: 100
Masukkan nilai sosiologi: 100
Nilai mahasiswa dengan NIM 18322001 sudah diperbaharui!

Selamat Datang di Program Manajemen Mahasiswa

1. Tambahkan Mahasiswa (Sisipkan di akhir)
2. Tambahkan Mahasiswa (Sisipkan di tengah)
3. Cari Mahasiswa berdasarkan NIM
4. Update nilai mahasiswa berdasarkan NIM
5. Hapus data mahasiswa berdasarkan NIM
6. Hitung rerata tiap mata kuliah
7. Tampilkan semua data mahasiswa
8. Exit

Enter your choice: 5

Masukkan NIM mahasiswa yang akan dihapus datanya: 18322001

Mahasiswa dengan NIM 18322001 berhasil dihapus!

Selamat Datang di Program Manajemen Mahasiswa

1. Tambahkan Mahasiswa (Sisipkan di akhir)
2. Tambahkan Mahasiswa (Sisipkan di tengah)
3. Cari Mahasiswa berdasarkan NIM
4. Update nilai mahasiswa berdasarkan NIM
5. Hapus data mahasiswa berdasarkan NIM
6. Hitung rerata tiap mata kuliah
7. Tampilkan semua data mahasiswa
8. Exit

Enter your choice: 6

Rata-Rata nilai mahasiwa

Matematika: 88.00

Fisika: 71.50

Kimia: 69.50

Biologi: 73.50

Sosiologi: 95.00

Akuntansi: 95.00

Selamat Datang di Program Manajemen Mahasiswa

1. Tambahkan Mahasiswa (Sisipkan di akhir)
2. Tambahkan Mahasiswa (Sisipkan di tengah)
3. Cari Mahasiswa berdasarkan NIM
4. Update nilai mahasiswa berdasarkan NIM
5. Hapus data mahasiswa berdasarkan NIM
6. Hitung rerata tiap mata kuliah
7. Tampilkan semua data mahasiswa
8. Exit

Enter your choice: 7

Data Mahasiswa Institut Tidak Bobok 2044/2045

NIM: 18322008, Nama: Digta, Mat: 87, Fis: 67, Kim: 59, Bio: 48, Sosio: 90, Akuntansi: 90

NIM: 18322016, Nama: zaki, Mat: 89, Fis: 76, Kim: 80, Bio: 99, Sosio: 100, Akuntansi: 100

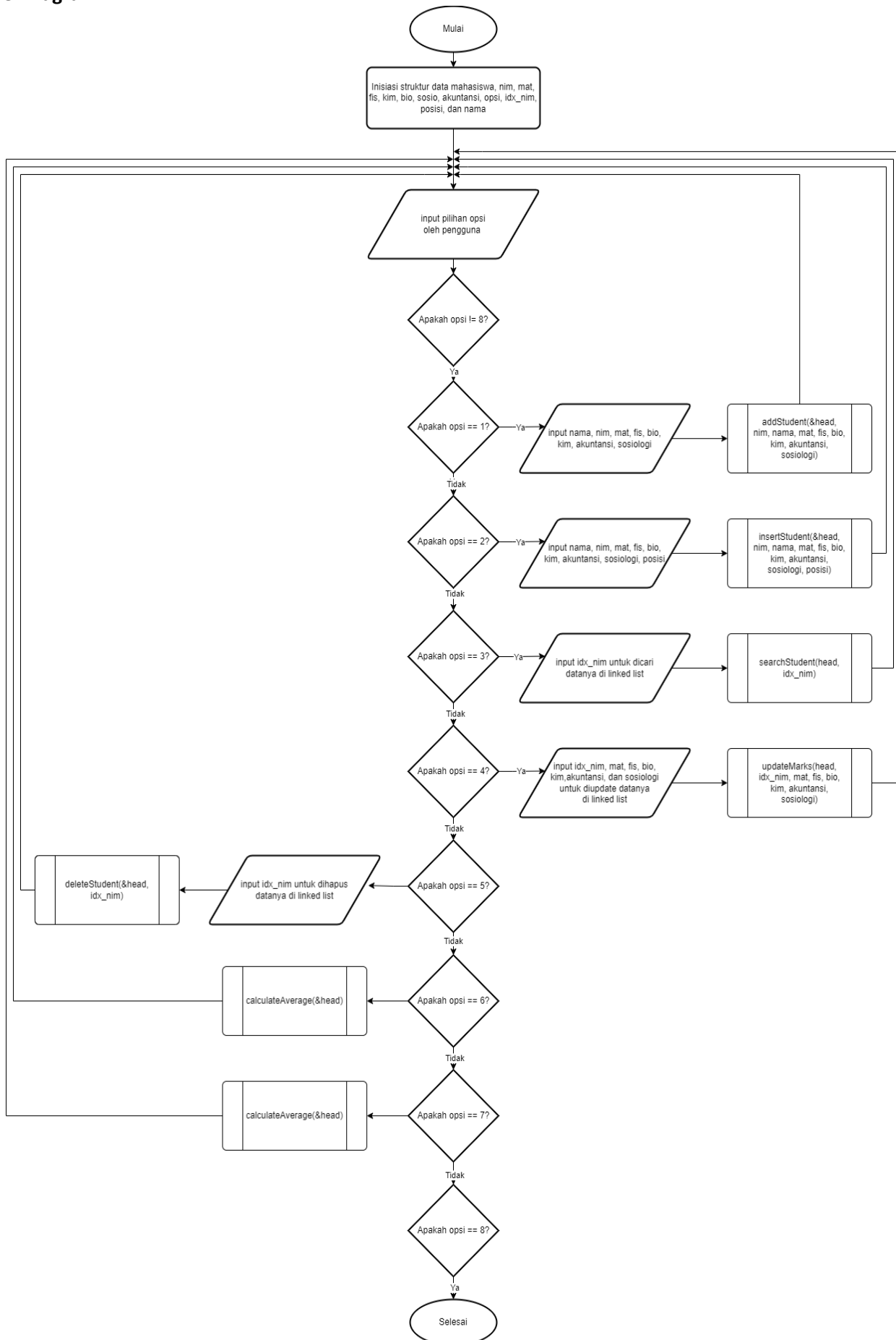
Selamat Datang di Program Manajemen Mahasiswa

1. Tambahkan Mahasiswa (Sisipkan di akhir)
2. Tambahkan Mahasiswa (Sisipkan di tengah)
3. Cari Mahasiswa berdasarkan NIM
4. Update nilai mahasiswa berdasarkan NIM
5. Hapus data mahasiswa berdasarkan NIM
6. Hitung rerata tiap mata kuliah
7. Tampilkan semua data mahasiswa
8. Exit

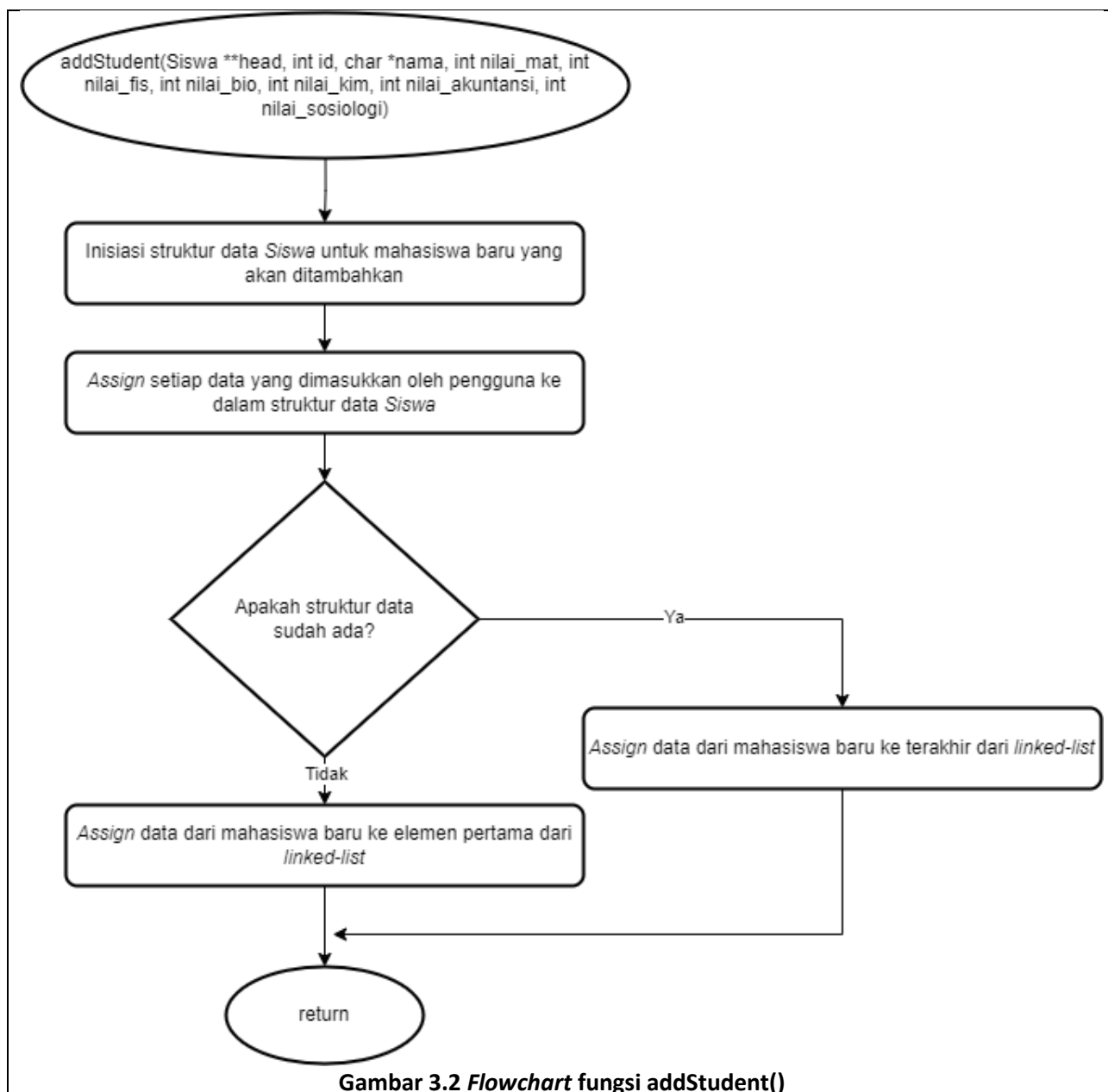
Enter your choice: 8

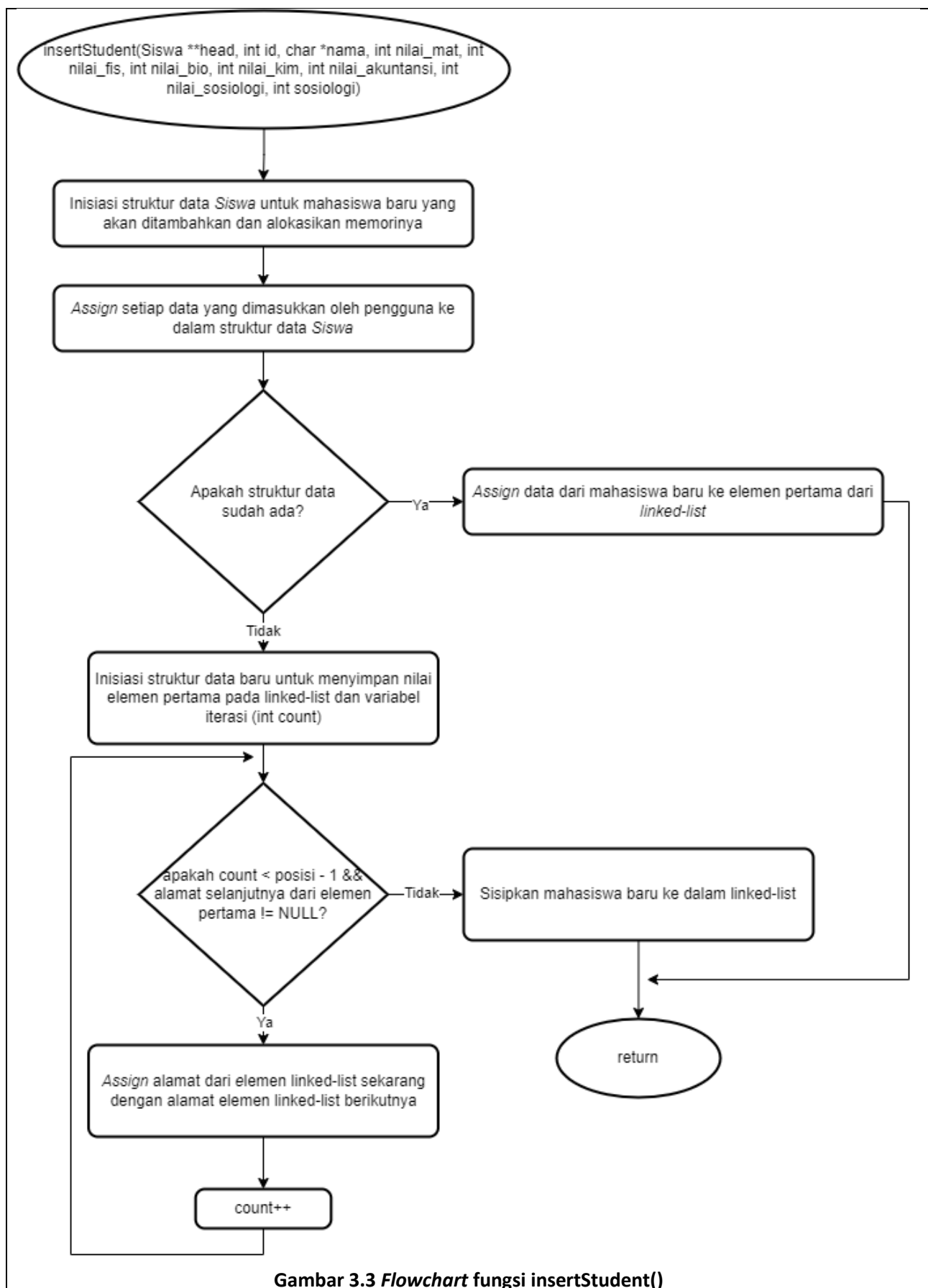
Keluar dari program, have a nice day!

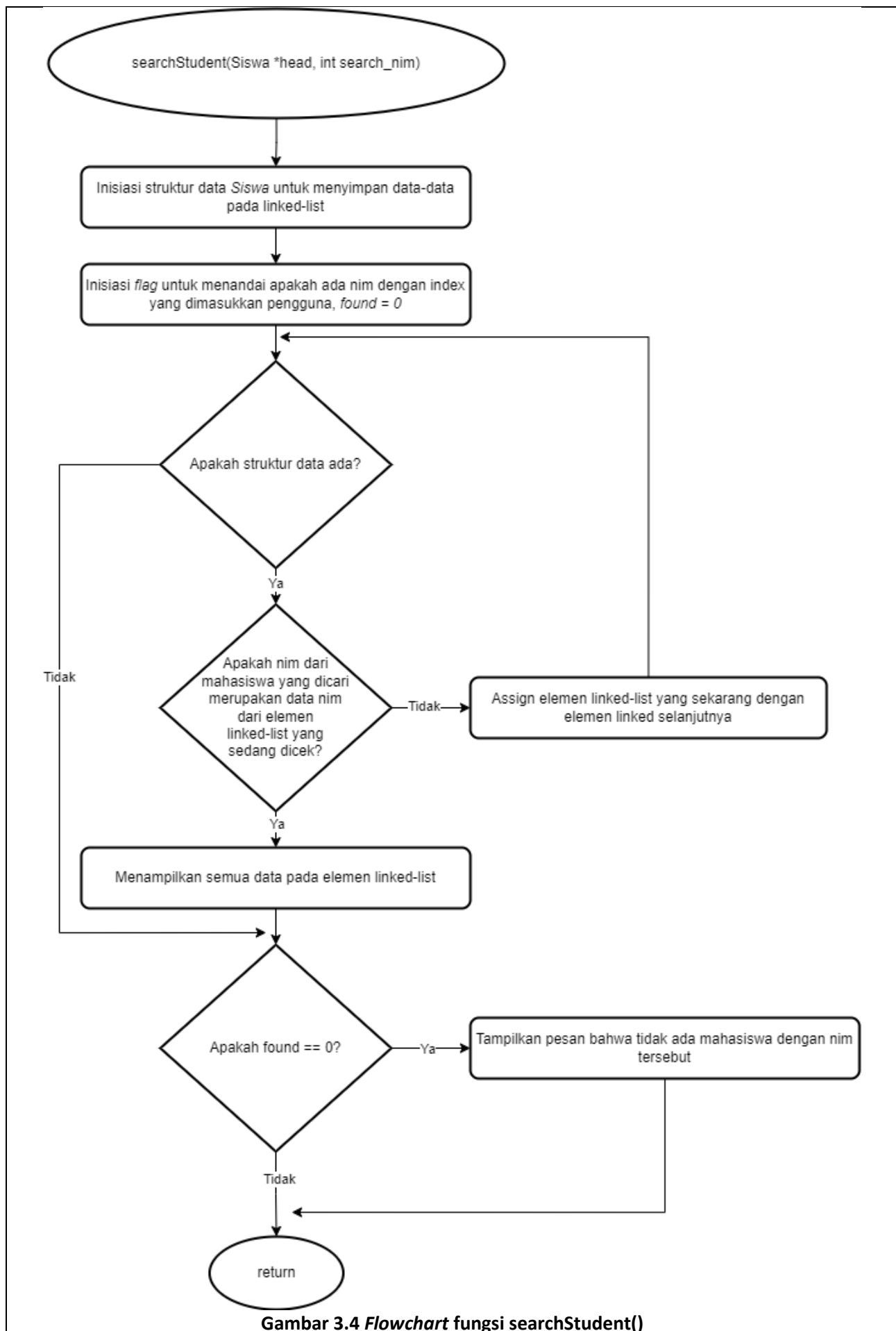
C. Diagram Alir



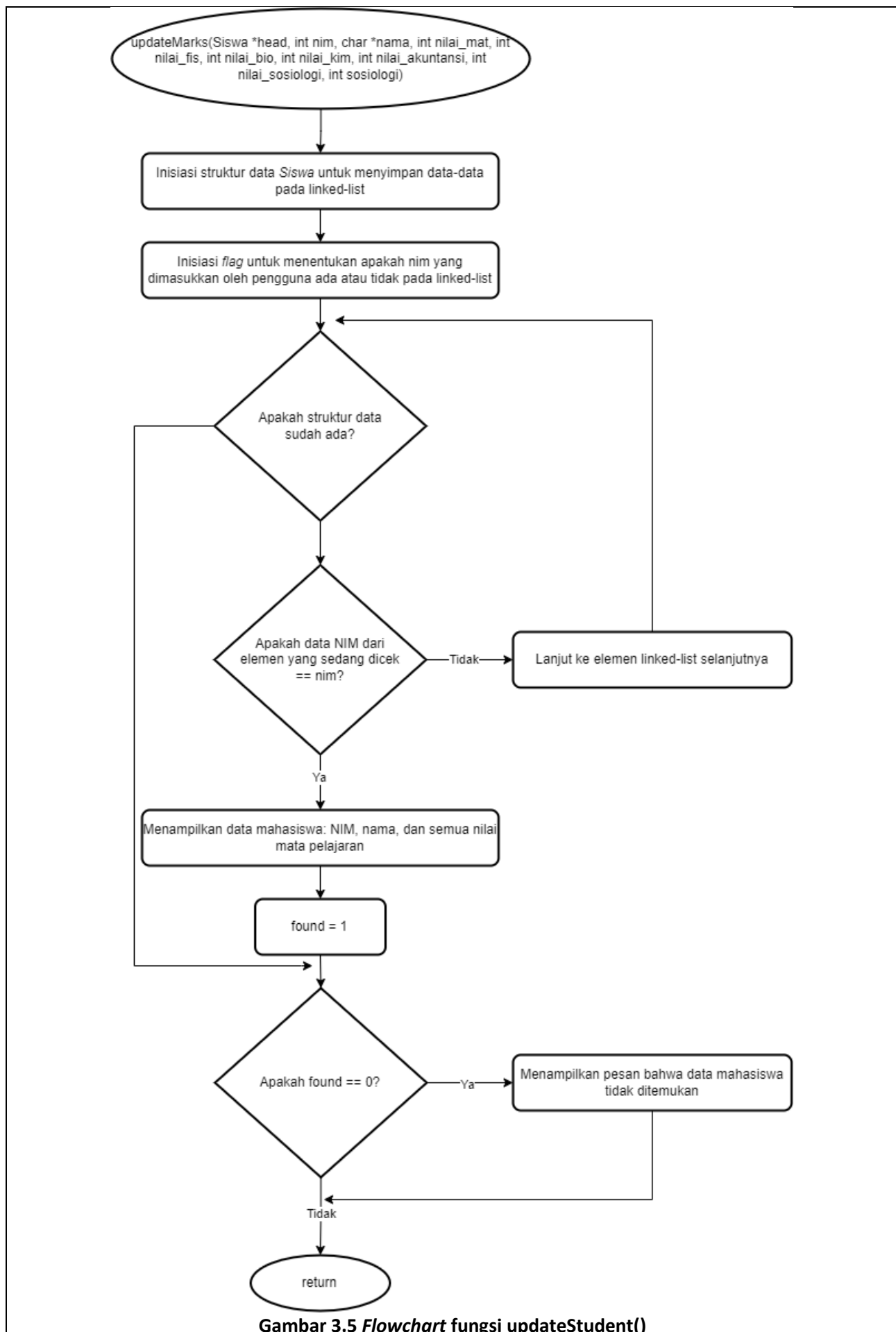
Gambar 3.1 Flowchart program utama

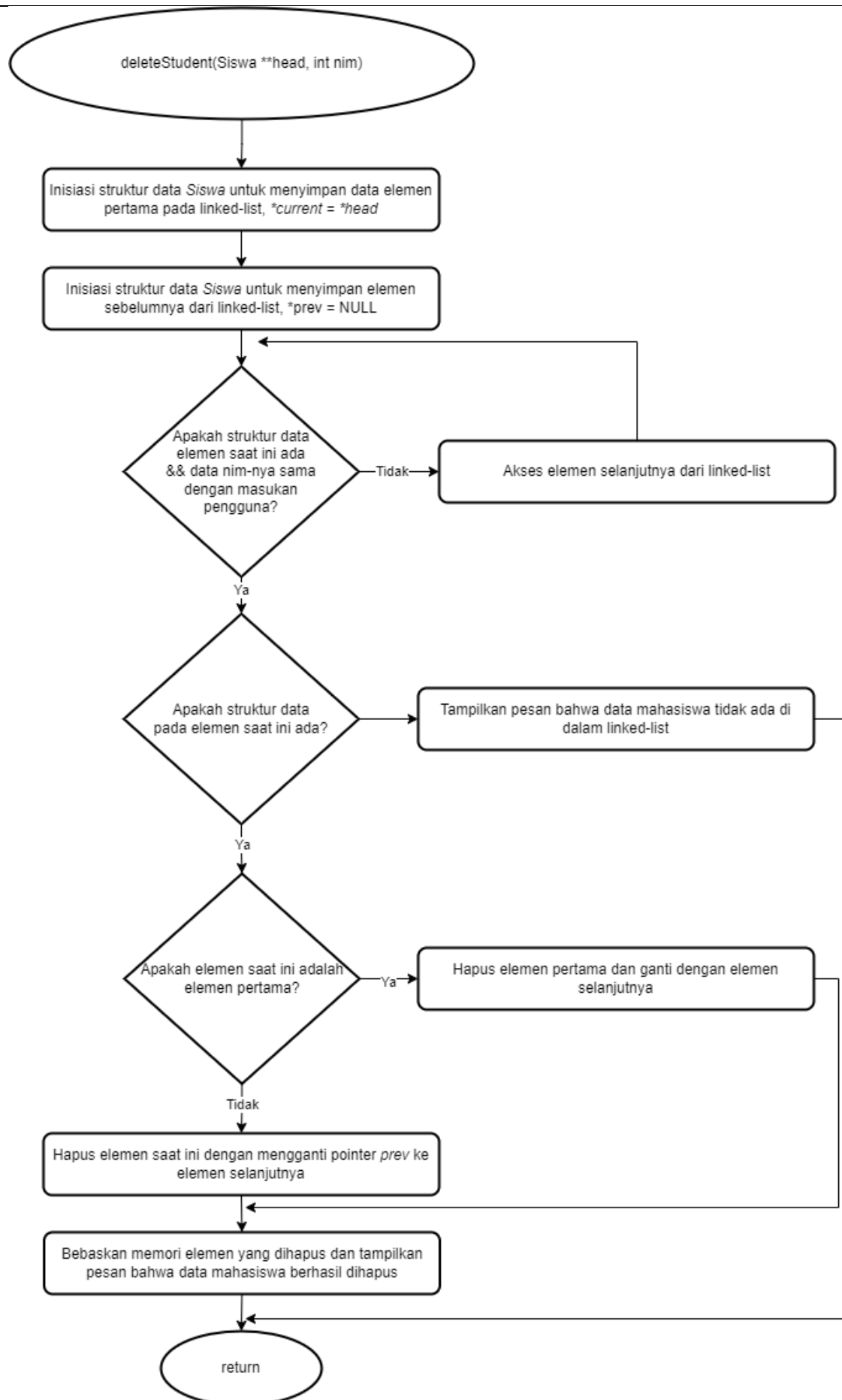




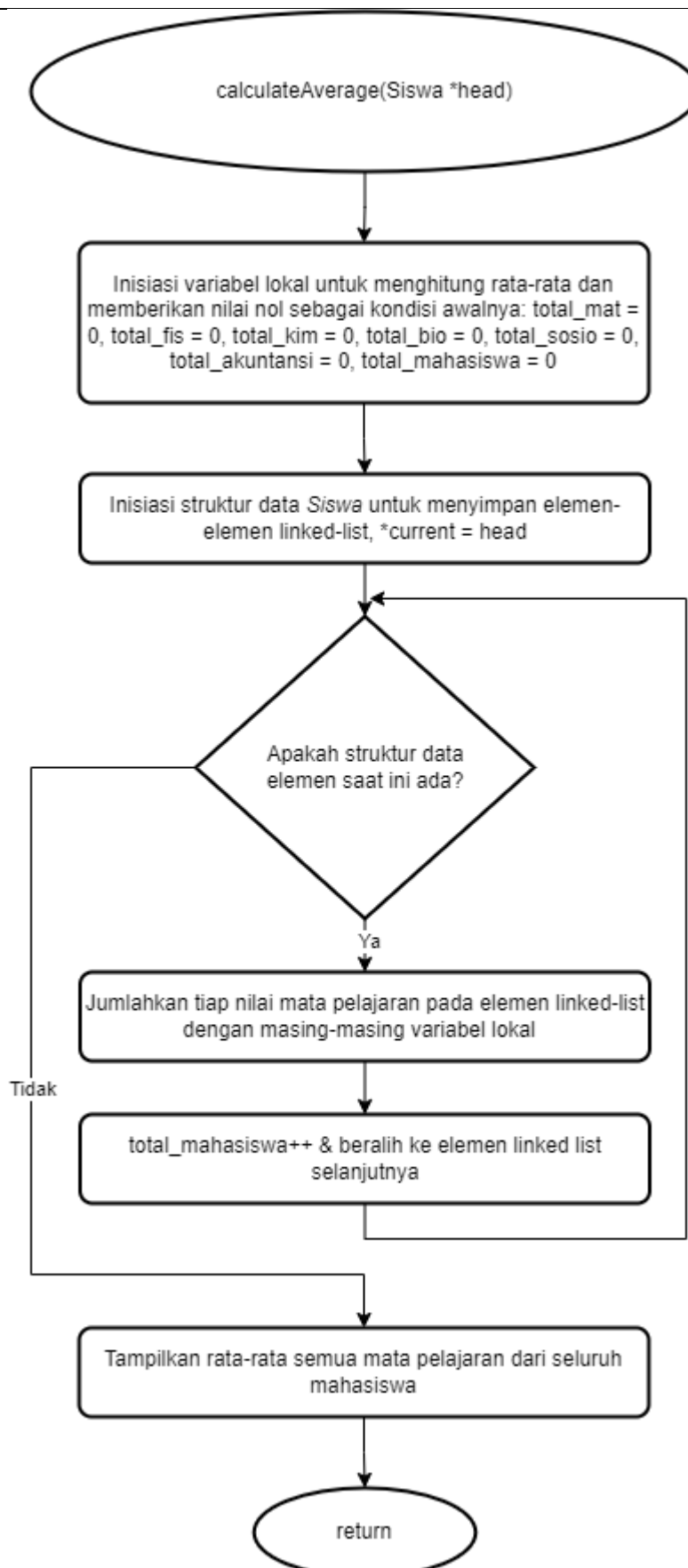


Gambar 3.4 Flowchart fungsi searchStudent()

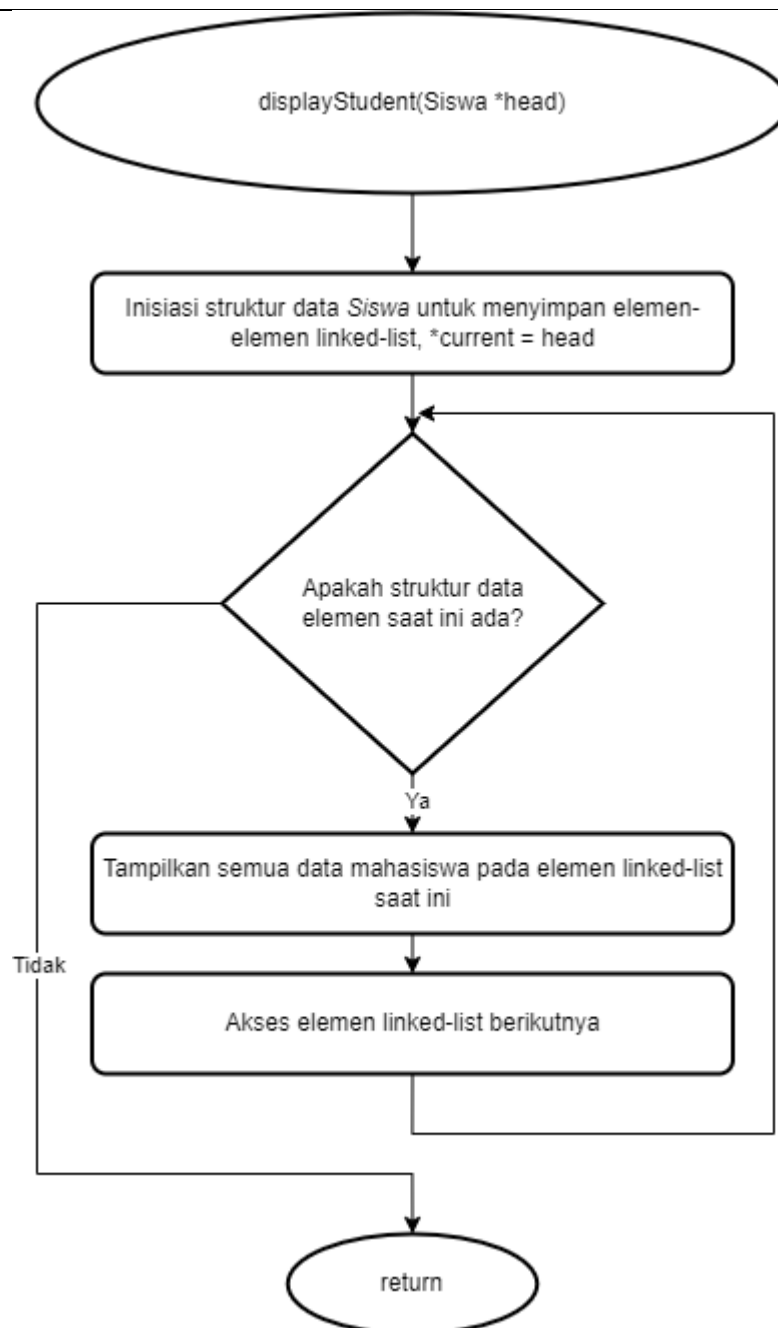




Gambar 3.6 Flowchart fungsi deleteStudent()

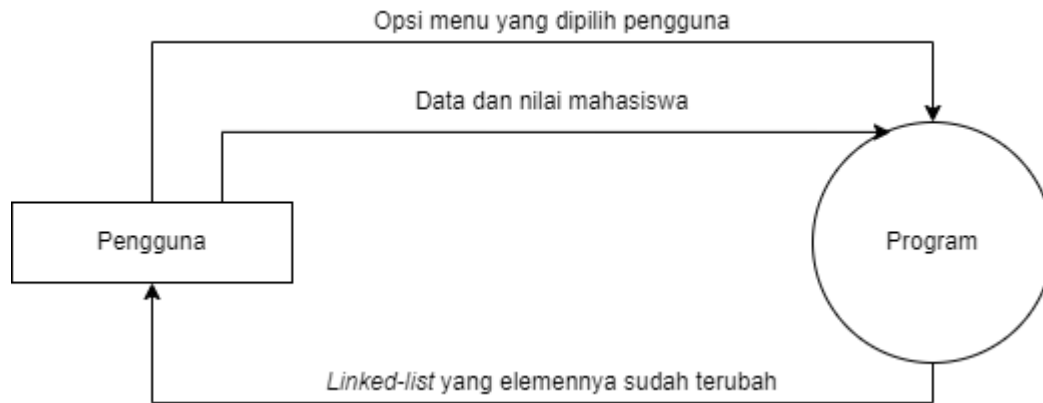


Gambar 3.7 Flowchart fungsi calculateAverage()

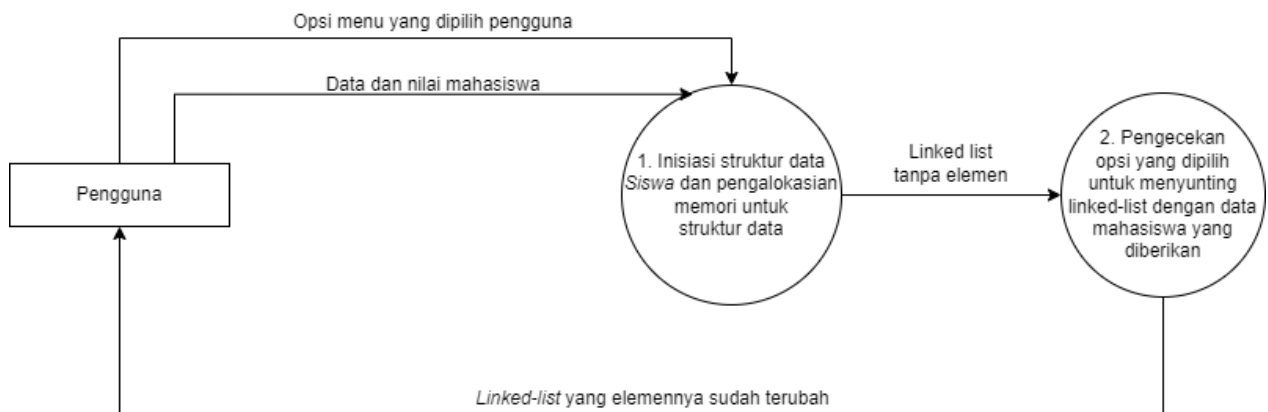


Gambar 3.8 Flowchart fungsi displayStudent()

D. Data Flow Diagram



Gambar 4.1 DFD level 0 dari program



Gambar 4.2 DFD level 1 dari program