

**LAPORAN AKHIR  
PRAKTIKUM IMPLEMENTASI STRUKTUR DATA**



Oleh:

Imam Khusain / NIM 123230018

**PROGRAM STUDI INFORMATIKA  
JURUSAN INFORMATIKA  
FAKULTAS TEKNIK INDUSTRI  
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”  
YOGYAKARTA  
2024**

## HALAMAN PENGESAHAN

### LAPORAN AKHIR

Disusun oleh :

Imam Khusain

123230018

Telah Diperiksa dan Disetujui oleh Asisten Praktikum Implementasi Struktur Data  
Pada Tanggal : 18 November 2024

**Menyetujui,**

**Asisten Praktikum**



Arvidion Havas Oktavian

NIM. 123220067

**Asisten Praktikum**



Laras Ayodya Sari

NIM. 123220081

## KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa yang senantiasa mencurahkan rahmat dan hidayah-Nya sehingga kami dapat menyelesaikan praktikum Implementasi Struktur Data serta laporan akhir praktikum Implementasi Struktur Data. Adapun laporan ini berisi tentang kumpulan tugas dan evaluasi dari hasil pembelajaran selama praktikum berlangsung.

Tidak lupa ucapan terima kasih kepada asisten dosen yang selalu membimbing dan mengajari kami dalam melaksanakan praktikum dan dalam menyusun laporan ini. Laporan ini masih sangat jauh dari kesempurnaan, oleh karena itu kritik serta saran yang membangun kami harapkan untuk menyempurnakan laporan akhir ini.

Atas perhatian dari semua pihak yang membantu penulisan ini, kami ucapkan terima kasih. Semoga laporan ini dapat dipergunakan seperlunya.

Yogyakarta, 18 November 2024

Penyusun

A handwritten signature in black ink, appearing to read 'Imam Khusain', with a horizontal line extending from the end of the signature.

Imam Khusain

## DAFTAR ISI

<b>LAPORAN AKHIR PRAKTIKUM IMPLEMENTASI STRUKTUR DATA .....</b>	<b>i</b>
<b>HALAMAN PENGESAHAN.....</b>	<b>ii</b>
<b>KATA PENGANTAR.....</b>	<b>iii</b>
<b>DAFTAR ISI .....</b>	<b>iv</b>
<b>TUGAS 1 QUICK SORT ATAU BUBBLE SORT .....</b>	<b>1</b>
1.1 Source Code Program .....	1
1.2 Catatan Revisi .....	6
1.3 Revisi Program.....	6
1.4 Screenshoot Program .....	11-12
<b>TUGAS 2 LINKED LIST KEPALA EKOR    MELINGKAR .....</b>	<b>13</b>
2.1 Source Code Program .....	13-16
2.2 Catatan Revisi .....	16
2.3 Revisi Program.....	17-22
2.4 Screenshot Program .....	22-23
<b>TUGAS 3 STACK.....</b>	<b>24</b>
3.1 Source Code Program .....	24-27
3.2 Catatan Revisi .....	27
3.3 Revisi Program.....	27
3.4 Screenshot Program .....	28-31
<b>TUGAS 4 QUEUE .....</b>	<b>32</b>
4.1 Source Code Program .....	32-34
4.2 Catatan Revisi .....	35
4.3 Revisi Program.....	35
4.4 Screenshot Program .....	35-36

## TUGAS 1

### QUICK SORT ATAU BUBBLE SORT

Buatlah Linked List Pointer tentang data gaji pegawai yang berisi nama pegawai (String) dan gaji pegawai (int) yang memiliki menu.

#### 1.1 Source Code Program

```
#include <iostream>
#include <string>

using namespace std;

struct Pegawai {
    string nPegawai;
    int gaji;
    Pegawai* next;
};

Pegawai* buatNode(string nPegawai, int gaji) {
    Pegawai* newNode = new Pegawai;
    newNode->nPegawai = nPegawai;
    newNode->gaji = gaji;
    newNode->next = NULL;
    return newNode;
}

void sisipDepan(Pegawai*& head, string nPegawai, int gaji) {
    Pegawai* newNode = buatNode(nPegawai, gaji);
    newNode->next = head;
    head = newNode;
}

void sisipTengah(Pegawai*& head, string nPegawai, int gaji) {
    if (head == NULL) {
        sisipDepan(head, nPegawai, gaji);
        return;
    }

    cout << "Masukkan nama pegawai sebagai acuan: ";
    cin.ignore();
    string acuan;
    getline(cin, acuan);

    Pegawai* temp = head;
    while (temp != NULL) {
        if (temp->nPegawai == acuan) {
```

```

        Pegawai* newNode = buatNode(nPegawai, gaji);
        newNode->next = temp->next;
        temp->next = newNode;
        return;
    }
    temp = temp->next;
}

cout << "Nama acuan tidak ditemukan!" << endl;
}

void sisipBelakang(Pegawai*& head, string nPegawai, int gaji)
{
    Pegawai* newNode = buatNode(nPegawai, gaji);
    if (head == NULL) {
        head = newNode;
    } else {
        Pegawai* temp = head;
        while (temp->next != NULL) {
            temp = temp->next;
        }
        temp->next = newNode;
    }
}

void hapusData(Pegawai*& head, string nPegawai) {
    if (head == NULL) {
        cout << "Data kosong, tidak ada yang bisa dihapus."
<< endl;
        return;
    }

    Pegawai* temp = head;
    Pegawai* prev = NULL;
    bool found = false;

    while (temp != NULL) {
        if (temp->nPegawai == nPegawai) {
            found = true;
            if (temp == head) {
                head = head->next;
            } else {
                prev->next = temp->next;
            }
            delete temp;
            break;
        }
        prev = temp;
        temp = temp->next;
    }
}

```

```

        if (found) {
            cout << "+=====+" <<
endl;
            cout << "|          DATA TERBARU          |" <<
endl;
            cout << "+=====+" <<
endl;
            if (head == nullptr) {
                cout << "Tidak ada data pegawai." << endl;
            } else {
                Pegawai* temp = head;
                while (temp != NULL) {
                    cout << "Nama Pegawai : " << temp->nPegawai
<< endl;
                    cout << "Gaji Pegawai : Rp. " << temp->gaji
<< endl;
                    cout << "-----"
<< endl;
                    temp = temp->next;
                }
                cout << "Nama pegawai berhasil dihapus." << endl;
            } else {
                cout << "Nama tidak ditemukan." << endl;
            }
        }
    }

void tampilkanData(Pegawai* head) {
    if (head == NULL) {
        cout << "Tidak ada data pegawai." << endl;
        return;
    }
    cout << "+=====+" << endl;
    cout << "|          DATA PEGAWAI          |" << endl;
    cout << "+=====+" << endl;
    Pegawai* temp = head;
    while (temp != NULL) {
        cout << "Nama Pegawai : " << temp->nPegawai << endl;
        cout << "Gaji Pegawai : Rp. " << temp->gaji << endl;
        cout << "-----" << endl;
        temp = temp->next;
    }
}

int main() {
    Pegawai* head = NULL;
    char back;

    do {

```

```

endl;      cout << "+=====+" <<
endl;      cout << "|          MENU UTAMA          |" << endl;
           cout << "+=====+" << endl;
           cout << "|1. TAMBAH DATA          |" << endl;
           cout << "|2. HAPUS DATA          |" << endl;
           cout << "|3. TAMPILKAN DATA      |" << endl;
           cout << "|4. EXIT                |" << endl;
           cout << "+=====+" << endl;
           cout << "Masukkan pilihan anda (1-4): ";
           int pilihan;
           cin >> pilihan;

           switch (pilihan) {
               case 1: {
                   cout << "Masukkan nama pegawai: ";
                   cin.ignore();
                   string nama;
                   getline(cin, nama);

                   cout << "Masukkan gaji pegawai: Rp. ";
                   int gaji;
                   cin >> gaji;

                   cout << "+=====+" <<
endl;      cout << "|          TAMBAH DATA          |" <<
endl;      cout << "+=====+" <<
endl;      cout << "|1. SISIP DEPAN          |" <<
endl;      cout << "|2. SISIP TENGAH        |" <<
endl;      cout << "|3. SISIP BELAKANG      |" <<
endl;      cout << "+=====+" <<
endl;      cout << "Masukkan pilihan anda (1-3): ";
           int sisipPilihan;
           cin >> sisipPilihan;

           switch (sisipPilihan) {
               case 1:
                   sisipDepan(head, nama, gaji);
                   break;
               case 2:
                   sisipTengah(head, nama, gaji);
                   break;
               case 3:

```



```

        sisipBelakang(head, nama, gaji);
        break;
    default:
        cout << "Pilihan sisip tidak valid."
<< endl;
    }
    break;
}
case 2: {
    cout << "Masukkan nama pegawai yang akan
dihapus: ";
    cin.ignore();
    string nama;
    getline(cin, nama);
    hapusData(head, nama);
    break;
}
case 3:
    tampilkanData(head);
    break;
case 4:
    cout <<
"+=====+\n";
    cout << "|Senang sekali Anda mau mencoba
program ini! |\n";
    cout << "|Kami berharap program ini memenuhi
harapan Anda, |\n";
    cout << "|dan masukan Anda akan sangat
berharga bagi kami. |\n";
    cout <<
"+=====+\n";
    return 0;
    default:
        cout << "Pilihan tidak valid." << endl;
}

    cout << "Kembali ke menu ? (y/n) : ";
    cin >> back;
    cout << endl;

} while (back == 'y' || back == 'Y');

return 0;
}

```

## 1.2 Catatan Revisi

hallow, tolong diperbaiki sisip tengahnya agar bisa menyisip sesuai urutan gaji ya  
soalnya sisip tengahmu nyisip ke belakang semua, sama setelah hapus data  
tampilkan tampil data terbaru ya

## 1.3 Revisi Program

```
##include <iostream>
#include <string>

using namespace std;

struct Pegawai {
    string nPegawai;
    int gaji;
    Pegawai* next;
};

Pegawai* buatNode(string nPegawai, int gaji) {
    Pegawai* newNode = new Pegawai;
    newNode->nPegawai = nPegawai;
    newNode->gaji = gaji;
    newNode->next = NULL;
    return newNode;
}

void sisipDepan(Pegawai*& head, string nPegawai, int gaji) {
    Pegawai* newNode = buatNode(nPegawai, gaji);
    newNode->next = head;
    head = newNode;
}

void sisipTengah(Pegawai*& head, string nPegawai, int gaji) {
    if (head == NULL) {
        sisipDepan(head, nPegawai, gaji);
        return;
    }

    cout << "Masukkan nama pegawai sebagai acuan: ";
    cin.ignore();
    string acuan;
    getline(cin, acuan);

    Pegawai* temp = head;
    while (temp != NULL) {
        if (temp->nPegawai == acuan) {
            Pegawai* newNode = buatNode(nPegawai, gaji);
            newNode->next = temp->next;
            temp->next = newNode;
            return;
        }
        temp = temp->next;
    }
}
```

```

        }
        temp = temp->next;
    }

    cout << "Nama acuan tidak ditemukan!" << endl;
}

void sisipBelakang(Pegawai*& head, string nPegawai, int gaji)
{
    Pegawai* newNode = buatNode(nPegawai, gaji);
    if (head == NULL) {
        head = newNode;
    } else {
        Pegawai* temp = head;
        while (temp->next != NULL) {
            temp = temp->next;
        }
        temp->next = newNode;
    }
}

void hapusData(Pegawai*& head, string nPegawai) {
    if (head == NULL) {
        cout << "Data kosong, tidak ada yang bisa dihapus." <<
endl;
        return;
    }

    Pegawai* temp = head;
    Pegawai* prev = NULL;
    bool found = false;

    while (temp != NULL) {
        if (temp->nPegawai == nPegawai) {
            found = true;
            if (temp == head) {
                head = head->next;
            } else {
                prev->next = temp->next;
            }
            delete temp;
            break;
        }
        prev = temp;
        temp = temp->next;
    }

    if (found) {
        cout << "+=====+" << endl;
        cout << "|                DATA TERBARU                |" << endl;
    }
}

```

```

        cout << "+=====+" << endl;
        if (head == nullptr) {
            cout << "Tidak ada data pegawai." << endl;
        } else {
            Pegawai* temp = head;
            while (temp != NULL) {
                cout << "Nama Pegawai : " << temp->nPegawai <<
endl;
                cout << "Gaji Pegawai : Rp. " << temp->gaji <<
endl;
                cout << "-----" <<
endl;
                temp = temp->next;
            }
            cout << "Nama pegawai berhasil dihapus." << endl;
        } else {
            cout << "Nama tidak ditemukan." << endl;
        }
    }

void tampilkanData(Pegawai* head) {
    if (head == NULL) {
        cout << "Tidak ada data pegawai." << endl;
        return;
    }
    cout << "+=====+" << endl;
    cout << "|          DATA PEGAWAI          |" << endl;
    cout << "+=====+" << endl;
    Pegawai* temp = head;
    while (temp != NULL) {
        cout << "Nama Pegawai : " << temp->nPegawai << endl;
        cout << "Gaji Pegawai : Rp. " << temp->gaji << endl;
        cout << "-----" << endl;
        temp = temp->next;
    }
}

int main() {
    Pegawai* head = NULL;
    char back;

    do {
        cout << "+=====+" <<
endl;
        cout << "|          MENU UTAMA          |" << endl;
        cout << "+=====+" << endl;
        cout << "|1. TAMBAH DATA          |" << endl;
        cout << "|2. HAPUS DATA          |" << endl;
        cout << "|3. TAMPILKAN DATA       |" << endl;

```

```

        cout << "|4. EXIT                                |" << endl;
        cout << "+=====+" << endl;
        cout << "Masukkan pilihan anda (1-4): ";
        int pilihan;
        cin >> pilihan;

        switch (pilihan) {
            case 1: {
                cout << "Masukkan nama pegawai: ";
                cin.ignore();
                string nama;
                getline(cin, nama);

                cout << "Masukkan gaji pegawai: Rp. ";
                int gaji;
                cin >> gaji;

                cout << "+=====+" <<
endl;
                cout << "|          TAMBAH DATA          |" <<
endl;
                cout << "+=====+" <<
endl;
                cout << "|1. SISIP DEPAN                                |" <<
endl;
                cout << "|2. SISIP TENGAH                                |" <<
endl;
                cout << "|3. SISIP BELAKANG                                |" <<
endl;
                cout << "+=====+" <<
endl;

                cout << "Masukkan pilihan anda (1-3): ";
                int sisipPilihan;
                cin >> sisipPilihan;

                switch (sisipPilihan) {
                    case 1:
                        sisipDepan(head, nama, gaji);
                        break;
                    case 2:
                        sisipTengah(head, nama, gaji);
                        break;
                    case 3:
                        sisipBelakang(head, nama, gaji);
                        break;
                    default:
                        cout << "Pilihan sisip tidak valid."
<< endl;
                }
                break;

```

```

        }
        case 2: {
            cout << "Masukkan nama pegawai yang akan
dihapus: ";
            cin.ignore();
            string nama;
            getline(cin, nama);
            hapusData(head, nama);
            break;
        }
        case 3:
            tampilkanData(head);
            break;
        case 4:
            cout <<
"+=====+\n";
            cout << "|Senang sekali Anda mau mencoba
program ini!   |\n";
            cout << "|Kami berharap program ini memenuhi
harapan Anda, |\n";
            cout << "|dan masukan Anda akan sangat
berharga bagi kami. |\n";
            cout <<
"+=====+\n";
            return 0;
        default:
            cout << "Pilihan tidak valid." << endl;
    }

    cout << "Kembali ke menu ? (y/n) : ";
    cin >> back;
    cout << endl;

    } while (back == 'y' || back == 'Y');

    return 0;
}

```

## 1.4 Screenshoot Program

### 1. Menu Utama dan Tambah Data

```
+=====+
|          MENU UTAMA          |
+=====+
|1. TAMBAH DATA                |
|2. HAPUS DATA                |
|3. TAMPILKAN DATA            |
|4. EXIT                      |
+=====+
Masukkan pilihan anda (1-4): 1
Masukkan nama pegawai: Adi
Masukkan gaji pegawai: Rp. 250000
+=====+
|          TAMBAH DATA          |
+=====+
|1. SISIP DEPAN                |
|2. SISIP TENGAH              |
|3. SISIP BELAKANG            |
+=====+
Masukkan pilihan anda (1-3): 2
Masukkan nama pegawai sebagai acuan: Ikhsan
Kembali ke menu ? (y/n) : y
```

### 2. Tampilkan Data

```
+=====+
|          MENU UTAMA          |
+=====+
|1. TAMBAH DATA                |
|2. HAPUS DATA                |
|3. TAMPILKAN DATA            |
|4. EXIT                      |
+=====+
Masukkan pilihan anda (1-4): 3
+=====+
|          DATA PEGAWAI          |
+=====+
Nama Pegawai : Ikhsan
Gaji Pegawai : Rp. 100000
-----
Nama Pegawai : Adi
Gaji Pegawai : Rp. 250000
-----
Nama Pegawai : Affan
Gaji Pegawai : Rp. 500000
-----
Kembali ke menu ? (y/n) : y
```

### 3. Hapus Data

```
+=====+
|           MENU UTAMA           |
+=====+
|1. TAMBAH DATA                 |
|2. HAPUS DATA                  |
|3. TAMPILKAN DATA              |
|4. EXIT                         |
+=====+
Masukkan pilihan anda (1-4): 2
Masukkan nama pegawai yang akan dihapus: Adi
+=====+
|           DATA TERBARU        |
+=====+
Nama Pegawai : Ikhsan
Gaji Pegawai : Rp. 100000
-----
Nama Pegawai : Affan
Gaji Pegawai : Rp. 500000
-----
Nama pegawai berhasil dihapus.
Kembali ke menu ? (y/n) : y
```

### 4. Exit Program

```
+=====+
|           MENU UTAMA           |
+=====+
|1. TAMBAH DATA                 |
|2. HAPUS DATA                  |
|3. TAMPILKAN DATA              |
|4. EXIT                         |
+=====+
Masukkan pilihan anda (1-4): 4
+=====+
|Senang sekali Anda mau mencoba program ini! |
|Kami berharap program ini memenuhi harapan Anda, |
|dan masukan Anda akan sangat berharga bagi kami. |
+=====+
```



## TUGAS 2

### LINKED LIST KEPALA EKOR || MELINGKAR

Dengan menggunakan program (tugas) sebelumnya :

- a. Ubah program linked list pointer tersebut menjadi linked list kepala ekor atau melingkar
- b. NIM Ganjil = Linked List Kepala Ekor
- c. NIM Genap = Linked List Melingkar

Saya NIM Genap.

#### 2.1 Source Code Program

```
#include <iostream>
#include <string>

using namespace std;

struct Pegawai {
    string nPegawai;
    int gaji;
};

struct node {
    Pegawai pgw;
    node* next;
};

node* awal = NULL;
node* akhir = NULL;

bool listKosong() {
    return (awal == NULL);
}

void sisipnode(Pegawai pgw) {
    node* nb = new node();
    nb->pgw = pgw;
    nb->next = NULL;

    if (listKosong()) {
        awal = nb;
        akhir = nb;
    } else if (pgw.gaji < awal->pgw.gaji) {
        nb->next = awal;
        awal = nb;
    } else {
        node* bantu = awal;
```

```

        while (bantu->next != NULL && pgw.gaji >= bantu-
>next->pgw.gaji) {
            bantu = bantu->next;
        }
        nb->next = bantu->next;
        bantu->next = nb;

        if (nb->next == NULL) {
            akhir = nb;
        }
    }
}

void hapusnode(string nama) {
    if (listKosong()) {
        cout << "Tidak ada data." << endl;
        return;
    }

    node* bantu = awal;
    node* hapus = NULL;

    if (awal->pgw.nPegawai == nama) {
        hapus = awal;
        awal = awal->next;
        delete hapus;
        return;
    }

    while (bantu->next != NULL && bantu->next->pgw.nPegawai
!= nama) {
        bantu = bantu->next;
    }

    if (bantu->next == NULL) {
        cout << "Node dengan nama " << nama << " tidak
ditemukan." << endl;
        return;
    }

    hapus = bantu->next;
    bantu->next = hapus->next;

    if (hapus == akhir) {
        akhir = bantu;
    }

    delete hapus;
}

```

```

void tampilkanData() {
    if (listKosong()) {
        cout << "Tidak ada data pegawai." << endl;
        return;
    }

    node* temp = awal;
    while (temp != NULL) {
        cout << "Nama pegawai: " << temp->pgw.nPegawai <<
endl;
        cout << "Gaji pegawai: Rp. " << temp->pgw.gaji <<
endl;
        cout << "=====" << endl;
        temp = temp->next;
    }
}

int main() {
    char back;

    do {
        cout << "+=====+" <<
endl;
        cout << "|          MENU UTAMA          |" << endl;
        cout << "+=====+" << endl;
        cout << "|1. TAMBAH DATA          |" << endl;
        cout << "|2. HAPUS DATA          |" << endl;
        cout << "|3. TAMPILAN DATA       |" << endl;
        cout << "|4. EXIT                 |" << endl;
        cout << "+=====+" << endl;
        cout << "Masukkan pilihan anda (1-4): ";
        int pilihan;
        cin >> pilihan;

        switch (pilihan) {
            case 1: {
                cout << "Masukkan nama pegawai: ";
                string nama;
                cin.ignore();
                getline(cin, nama);

                cout << "Masukkan gaji pegawai: Rp. ";
                int gaji;
                cin >> gaji;

                Pegawai pgw = {nama, gaji};
                sisipnode(pgw);
                cout << "Data berhasil disimpan." << endl;
                break;
            }

```

```

        case 2: {
            cout << "Masukkan nama pegawai yang akan
dihapus: ";
            string nama;
            cin.ignore();
            getline(cin, nama);

            hapusnode(nama);
            break;
        }
        case 3:
            tampilkanData();
            break;
        case 4:
            cout << "=====" <<
endl;
            cout << "|          Terima kasih!          |" <<
endl;
            cout << "|Untuk Anda Yang Selalu Ada!|" <<
endl;
            cout << "=====" <<
endl;
            return 0;
        default:
            cout << "Pilihan tidak valid." << endl;
    }

    cout << endl;
    cout << "Kembali ke menu? (y/n): ";
    cin >> back;
    cout << endl;

    } while (back == 'y' || back == 'Y');

    return 0;
}

```

## 2.2 Catatan Revisi

hallow tolong di ubah jadi melingkar sesuai ketentuan ya karena kamu kebagiannya itu bukan melingkar, sama tampilkan data terbaru setelah hapus data

## 2.3 Revisi Program

```
#include <iostream>
#include <string>

using namespace std;

struct Pegawai {
    string nPegawai;
    int gaji;
    Pegawai* next;
};

Pegawai* buatNode(string nPegawai, int gaji) {
    Pegawai* newNode = new Pegawai;
    newNode->nPegawai = nPegawai;
    newNode->gaji = gaji;
    newNode->next = newNode;
    return newNode;
}

void sisipDepan(Pegawai*& head, string nPegawai, int gaji) {
    Pegawai* newNode = buatNode(nPegawai, gaji);
    if (head == nullptr) {
        head = newNode;
    } else {
        Pegawai* temp = head;
        while (temp->next != head) {
            temp = temp->next;
        }
        temp->next = newNode;
        newNode->next = head;
        head = newNode;
    }
}

void sisipTengah(Pegawai*& head, string nPegawai, int gaji) {
    if (head == nullptr) {
        sisipDepan(head, nPegawai, gaji);
        return;
    }
    cout << "Masukkan nama pegawai sebagai acuan: ";
    string acuan;
    cin.ignore();
    getline(cin, acuan);

    Pegawai* temp = head;
    do {
        if (temp->nPegawai == acuan) {
            Pegawai* newNode = buatNode(nPegawai, gaji);
```

```

        newNode->next = temp->next;
        temp->next = newNode;
        return;
    }
    temp = temp->next;
} while (temp != head);

cout << "Nama acuan tidak ditemukan!" << endl;
}

void sisipBelakang(Pegawai*& head, string nPegawai, int gaji)
{
    Pegawai* newNode = buatNode(nPegawai, gaji);
    if (head == nullptr) {
        head = newNode;
    } else {
        Pegawai* temp = head;
        while (temp->next != head) {
            temp = temp->next;
        }
        temp->next = newNode;
        newNode->next = head;
    }
}

void hapusData(Pegawai*& head, string nPegawai) {
    if (head == nullptr) {
        cout << "Data kosong, tidak ada yang bisa dihapus."
<< endl;
        return;
    }

    Pegawai* temp = head;
    Pegawai* prev = nullptr;
    bool found = false;

    do {
        if (temp->nPegawai == nPegawai) {
            found = true;
            if (temp == head && temp->next == head) {
                head = nullptr;
            } else {
                if (temp == head) {
                    Pegawai* last = head;
                    while (last->next != head) {
                        last = last->next;
                    }
                    head = head->next;
                    last->next = head;
                } else {

```

```

        prev->next = temp->next;
    }
    }
    delete temp;
    break;
}
prev = temp;
temp = temp->next;
} while (temp != head);

if (found) {
    cout << "+=====+" << endl;
    cout << "|          DATA TERBARU          |" << endl;
    cout << "+=====+" << endl;
    if (head == nullptr) {
        cout << "Tidak ada data pegawai." << endl;
    } else {
        Pegawai* temp = head;
        do {
            cout << "Nama Pegawai : " << temp->nPegawai
<< endl;
            cout << "Gaji Pegawai : Rp. " << temp->gaji
<< endl;
            cout << "-----" <<
endl;
            temp = temp->next;
        } while (temp != head);
    }
    cout << "Nama pegawai berhasil dihapus." << endl;
} else {
    cout << "Nama tidak ditemukan." << endl;
}
}

void tampilkanData(Pegawai* head) {
    if (head == nullptr) {
        cout << "+=====+" << endl;
        cout << "|          DATA PEGAWAI          |" << endl;
        cout << "+=====+" << endl;
        cout << "Tidak ada data pegawai." << endl;
        return;
    }

    cout << "+=====+" << endl;
    cout << "|          DATA PEGAWAI          |" << endl;
    cout << "+=====+" << endl;

    Pegawai* temp = head;
    do {
        cout << "Nama Pegawai : " << temp->nPegawai << endl;

```

```

        cout << "Gaji Pegawai : Rp. " << temp->gaji << endl;
        cout << "-----" << endl;
        temp = temp->next;
    } while (temp != head);
}

int main() {
    Pegawai* head = nullptr;
    char back;

    do {
        cout << "+=====+" <<
endl;
        cout << "|          MENU UTAMA          |" << endl;
        cout << "+=====+" << endl;
        cout << "|1. TAMBAH DATA                    |" << endl;
        cout << "|2. HAPUS DATA                    |" << endl;
        cout << "|3. TAMPILKAN DATA                |" << endl;
        cout << "|4. EXIT                          |" << endl;
        cout << "+=====+" << endl;
        cout << "Masukkan pilihan anda (1-4): ";
        int pilihan;
        cin >> pilihan;

        switch (pilihan) {
            case 1: {
                cout << "Masukkan nama pegawai: ";
                string nama;
                cin.ignore();
                getline(cin, nama);

                cout << "Masukkan gaji pegawai: Rp. ";
                int gaji;
                cin >> gaji;
                cin.ignore();

                cout << "+=====+" <<
endl;
                cout << "|          TAMBAH DATA          |" <<
endl;
                cout << "+=====+" <<
endl;
                cout << "|1. SISIP DEPAN                    |" <<
endl;
                cout << "|2. SISIP TENGAH                  |" <<
endl;
                cout << "|3. SISIP BELAKANG                |" <<
endl;
                cout << "+=====+" <<
endl;

```



```

        cout << "Masukkan pilihan anda (1-3): ";
        int sisipPilihan;
        cin >> sisipPilihan;

        switch (sisipPilihan) {
            case 1:
                sisipDepan(head, nama, gaji);
                break;
            case 2:
                sisipTengah(head, nama, gaji);
                break;
            case 3:
                sisipBelakang(head, nama, gaji);
                break;
            default:
                cout << "Pilihan sisip tidak valid."
<< endl;
        }
        break;
    }
    case 2: {
        cout << "Masukkan nama pegawai yang akan
dihapus: ";

        string nama;
        cin.ignore();
        getline(cin, nama);
        hapusData(head, nama);
        break;
    }
    case 3:
        tampilkanData(head);
        break;
    case 4:
        cout <<
"+=====+" << endl;
        cout << "|      Terima kasih telah
mencoba!  |" << endl;
        cout <<
"+=====+" << endl;
        return 0;
    default:
        cout << "Pilihan tidak valid." << endl;
    }

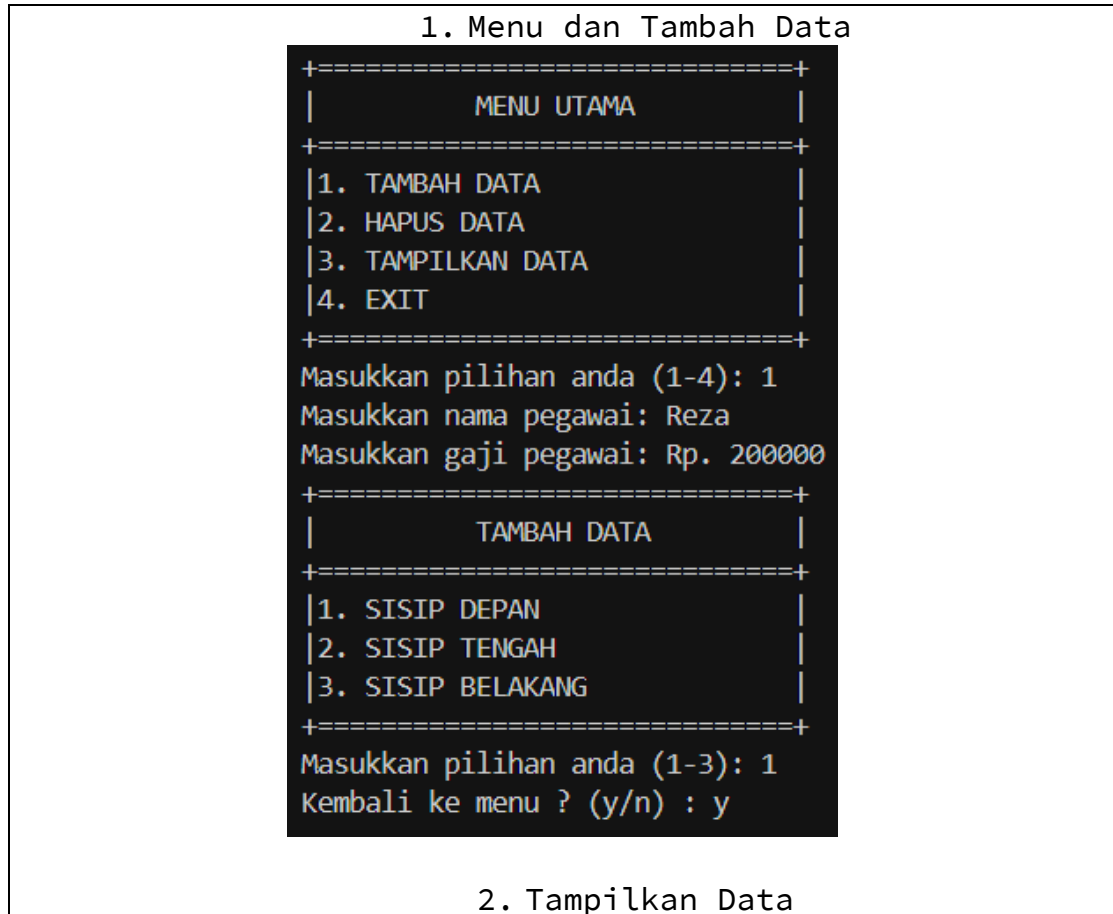
    cout << "Kembali ke menu ? (y/n) : ";
    cin >> back;
    cout << endl;

} while (back == 'y' || back == 'Y');

```

```
    return 0;  
}
```

## 2.4 Screenshot Program



```

+=====+
|          MENU UTAMA          |
+=====+
|1. TAMBAH DATA               |
|2. HAPUS DATA                |
|3. TAMPILKAN DATA            |
|4. EXIT                       |
+=====+
Masukkan pilihan anda (1-4): 3
Nama pegawai: Reza
Gaji pegawai: Rp. 200000
=====
Nama pegawai: Andika
Gaji pegawai: Rp. 400000
=====
Nama pegawai: Aal
Gaji pegawai: Rp. 600000
=====
Kembali ke menu ? (y/n) : y

```

### 3. Hapus Data

```

+=====+
|          MENU UTAMA          |
+=====+
|1. TAMBAH DATA               |
|2. HAPUS DATA                |
|3. TAMPILKAN DATA            |
|4. EXIT                       |
+=====+
Masukkan pilihan anda (1-4): 2
Masukkan nama pegawai yang akan dihapus: Reza
+=====+
|          HAPUS DATA          |
+=====+
Nama pegawai yang akan dihapus : Reza
+=====+
|          DATA TERBARU        |
+=====+
Andika
Gaji Pegawai Rp. 400000 (Silahkan minta gaji anda!)
Aal
Gaji Pegawai Rp. 600000 (Silahkan minta gaji anda!)
Kembali ke menu ? (y/n) : y

```

## TUGAS 3 STACK

Membuat Program Perpustakaan Rajawali sedang merapihkan buku-buku yang dikembalikan dari peminjam. Buatlah program stack / tumpukan untuk buku-buku tersebut.

### 3.1 Source Code Program

```
#include <bits/stdc++.h>
#define MAX 30
using namespace std;

struct Buku {
    string namaBuku;
    string penulis;
    Buku* next;
};

Buku* top = NULL;
Buku* awal = NULL;
Buku* akhir = NULL;

int countBuku();

void buatStack() {
    top = NULL;
    awal = NULL;
    akhir = NULL;
}

bool StackKosong() {
    return top == NULL;
}

void pushBuku(string namaBuku, string penulis) {
    Buku* newBuku = new Buku();
    newBuku->namaBuku = namaBuku;
    newBuku->penulis = penulis;
    newBuku->next = NULL;

    if (StackKosong()) {
        awal = akhir = top = newBuku;
    } else {
        newBuku->next = awal;
        awal = newBuku;
        top = awal;
    }
    cout << "Buku berhasil ditambahkan.\n";
}
```

```

void popBuku() {
    if (StackKosong()) {
        cout << "Tidak Ada Buku Kawan!\n";
        return;
    }
    Buku* hapus = awal;
    awal = awal->next;
    top = awal;
    cout << "Data Buku yang Berhasil Diambil:\n Nama Buku: "
    << hapus->namaBuku
        << "\n Nama Penulis: " << hapus->penulis << "\n";
    delete hapus;
}

void daftarBuku() {
    if (StackKosong()) {
        cout << "Stack Kosong\n";
    } else {
        cout << "+=====+\n";
        cout << "|          Daftar Buku          |\n";
        cout << "+=====+\n";
        Buku* temp = top;
        while (temp != NULL) {
            cout << "[" << temp->namaBuku << ", " << temp->
            >penulis << "]\n";
            temp = temp->next;
        }
        cout << "Jumlah Buku: " << countBuku() << "\n";
    }
}

void hapusStack() {
    while (!StackKosong()) {
        popBuku();
    }
    cout << "[SEMUA BUKU TELAH DIHAPUS]\n";
}

void editBuku(string bukuLama) {
    Buku* temp = top;
    while (temp != NULL) {
        if (temp->namaBuku == bukuLama) {
            cout << "Edit Data Buku:\n";
            cout << "Judul Buku Lama: " << temp->namaBuku <<
            "\n";
            cout << "Penulis Lama: " << temp->penulis <<
            "\n";
            cout << "Judul Buku Baru: ";
            getline(cin, temp->namaBuku);
            cout << "Penulis Baru: ";

```

```

        getline(cin, temp->penulis);
        cout << "[DATA BUKU DIPERBARUI]\n";
        return;
    }
    temp = temp->next;
}
cout << "Judul buku tidak ditemukan.\n";
}

int countBuku() {
    int count = 0;
    Buku* temp = top;
    while (temp != NULL) {
        count++;
        temp = temp->next;
    }
    return count;
}

int main() {
    int choice;
    string namaBuku, penulis;

    do {
        cout << "+=====+\n";
        cout << "|          DAFTAR MENU          |\n";
        cout << "+=====+\n";
        cout << "|1. Taruh Buku                  |\n";
        cout << "|2. Ambil Buku                  |\n";
        cout << "|3. Daftar Buku                 |\n";
        cout << "|4. Buang Semua Buku            |\n";
        cout << "|5. Edit Data Buku              |\n";
        cout << "|6. Keluar                      |\n";
        cout << "+=====+\n";
        cout << "Pilih Menu: ";
        cin >> choice;
        cin.ignore();

        switch (choice) {
            case 1: {
                int jumlah;
                cout << "Jumlah Buku yang akan dimasukkan: ";
                cin >> jumlah;
                cin.ignore();

                if (countBuku() + jumlah > MAX) {
                    cout << "Error: Jumlah buku yang
dimasukkan akan melebihi batas maksimal (" << MAX << ").\n";
                    break;
                }
            }

```

```

        for (int i = 0; i < jumlah; i++) {
            cout << "Judul Buku: ";
            getline(cin, namaBuku);
            cout << "Penulis: ";
            getline(cin, penulis);
            pushBuku(namaBuku, penulis);
        }
        break;
    }
    case 2: {
        popBuku();
        break;
    }
    case 3:
        daftarBuku();
        break;
    case 4:
        hapusStack();
        break;
    case 5: {
        cout << "Masukkan Judul Buku yang ingin
diedit: ";
        getline(cin, namaBuku);
        editBuku(namaBuku);
        break;
    }
    case 6:
        cout << "Terima Kasih! Sampai Jumpa Lagi
:)\n";
        break;
    default:
        cout << "Pilihan tidak valid.\n";
    }
} while (choice != 6);
return 0;
}

```

### 3.2 Catatan Revisi

Tidak ada catatan revisi

### 3.3 Revisi Program

Tidak ada revisi

### 3.4 Screenshot Program

#### 1. Menu Utama dan Input Data Buku

```
+=====+
|          DAFTAR MENU          |
+=====+
|1. Taruh Buku                  |
|2. Ambil Buku                  |
|3. Daftar Buku                 |
|4. Buang Semua Buku            |
|5. Edit Data Buku              |
|6. Keluar                      |
+=====+
Pilih Menu: 1
Jumlah Buku yang akan dimasukkan: 3
Judul Buku: Laskar Pelangi
Penulis: Andrea Hirata
Buku berhasil ditambahkan.
Judul Buku: Laut Bercerita
Penulis: Leila S. Chudori
Buku berhasil ditambahkan.
Judul Buku: Bumi
Penulis: Tere Liye
Buku berhasil ditambahkan.
```

#### 2. Tampil Data Buku

```
+=====+
|          DAFTAR MENU          |
+=====+
|1. Taruh Buku                  |
|2. Ambil Buku                  |
|3. Daftar Buku                 |
|4. Buang Semua Buku            |
|5. Edit Data Buku              |
|6. Keluar                      |
+=====+
Pilih Menu: 3
+=====+
|          Daftar Buku          |
+=====+
[Bumi, Tere Liye]
[Laut Bercerita, Leila S. Chudori]
[Laskar Pelangi, Andrea Hirata]
Jumlah Buku: 3
```



### 3. Edit Data Buku

```
+=====+
|          DAFTAR MENU          |
+=====+
|1. Taruh Buku                  |
|2. Ambil Buku                  |
|3. Daftar Buku                 |
|4. Buang Semua Buku            |
|5. Edit Data Buku              |
|6. Keluar                      |
+=====+
Pilih Menu: 5
Masukkan Judul Buku yang ingin diedit: Laut Bercerita
Edit Data Buku:
Judul Buku Lama: Laut Bercerita
Penulis Lama: Leila S. Chudori
Judul Buku Baru: Negeri 5 Menara
Penulis Baru: Ahmad Fuadi
[DATA BUKU DIPERBARUI]
```

### 4. Tampil Data Buku Terbaru

```
+=====+
|          DAFTAR MENU          |
+=====+
|1. Taruh Buku                  |
|2. Ambil Buku                  |
|3. Daftar Buku                 |
|4. Buang Semua Buku            |
|5. Edit Data Buku              |
|6. Keluar                      |
+=====+
Pilih Menu: 3
+=====+
|          Daftar Buku          |
+=====+
[Bumi, Tere Liye]
[Negeri 5 Menara, Ahmad Fuadi]
[Laskar Pelangi, Andrea Hirata]
Jumlah Buku: 3
```

### 5. Ambil Data Buku

```
+=====+
|          DAFTAR MENU          |
+=====+
|1. Taruh Buku                  |
|2. Ambil Buku                  |
|3. Daftar Buku                 |
|4. Buang Semua Buku           |
|5. Edit Data Buku             |
|6. Keluar                     |
+=====+
Pilih Menu: 2
Data Buku yang Berhasil Diambil:
Nama Buku: Bumi
Nama Penulis: Tere Liye
```

### 6. Hapus Data Buku

```
+=====+
|          DAFTAR MENU          |
+=====+
|1. Taruh Buku                  |
|2. Ambil Buku                  |
|3. Daftar Buku                 |
|4. Buang Semua Buku           |
|5. Edit Data Buku             |
|6. Keluar                     |
+=====+
Pilih Menu: 4
Data Buku yang Berhasil Diambil:
Nama Buku: Negeri 5 Menara
Nama Penulis: Ahmad Fuadi
Data Buku yang Berhasil Diambil:
Nama Buku: Laskar Pelangi
Nama Penulis: Andrea Hirata
[SEMUA BUKU TELAH DIHAPUS]
```

### 7. Tampil Data Buku Setelah Dihapus

```
+=====+
|          DAFTAR MENU          |
+=====+
|1. Taruh Buku                  |
|2. Ambil Buku                  |
|3. Daftar Buku                 |
|4. Buang Semua Buku           |
|5. Edit Data Buku             |
|6. Keluar                     |
+=====+
Pilih Menu: 3
Stack Kosong
```

## 8. Keluar Program Buku

```
+=====+
|          DAFTAR MENU          |
+=====+
|1. Taruh Buku                  |
|2. Ambil Buku                  |
|3. Daftar Buku                 |
|4. Buang Semua Buku           |
|5. Edit Data Buku             |
|6. Keluar                      |
+=====+
Pilih Menu: 6
Terima Kasih! Sampai Jumpa Lagi :)
```

## TUGAS 4

### QUEUE

Membuat Program antrian paket dengan menggunakan algoritma queue.

#### 4.1 Source Code Program

```
#include <bits/stdc++.h>
using namespace std;

struct Paket {
    string nBarang; // Nama Barang
    string aTujuan; // Alamat Tujuan Pengiriman
    string nResi; // Nomor Resi
    Paket *next;
};

Paket *depan, *belakang, *NB, *hapus, *bantu;

void buatQueue(){
    depan = NULL;
    belakang = NULL;
}

bool queueKosong() {
    return depan == NULL;
}

void enqueue(string nBarang, string aTujuan, string nResi) {
    NB = new Paket();
    NB->nBarang = nBarang;
    NB->aTujuan = aTujuan;
    NB->nResi = nResi;
    NB->next = NULL;

    if(queueKosong()) {
        depan = NB;
        belakang = NB;
    } else {
        belakang->next = NB;
        belakang = NB;
    }
}

void dequeue() {
    if (queueKosong()) {
        cout << "Antrian Kosong, Paket Sudah Dikirim Semua Kawan!!!\n";
    } else {
        hapus = depan;
```

```

        cout << "+=====+\n";
        cout << "|          KIRIM PAKET          |\n";
        cout << "+=====+\n";
        cout << "|Nama Barang : " << setw(16) << left << hapus-
>nBarang << "|\n";
        cout << "|No Resi      : " << setw(16) << left << hapus-
>nResi << "|\n";
        cout << "+=====+\n";
        cout << "Paket Sedang Dikirimkan ke " << hapus->aTujuan <<
"\n";

        depan = hapus->next;
        delete hapus;
    }
}

void cetakQueue() {
    if(queueKosong()) {
        cout << "Antrian kosong.\n";
    } else {
        bantu = depan;
        int jumlah = 0;
        cout << "+=====+\n";
        cout << "|          PAKET MASUK          |\n";
        cout << "+=====+\n";

        while(bantu != NULL) {
            jumlah++;
            cout << "Nama Barang : " << bantu->nBarang << "\n";
            cout << "Tujuan      : " << bantu->aTujuan << "\n";
            cout << "No Resi      : " << bantu->nResi << "\n\n";
            bantu = bantu->next;
        }
        cout << "Jumlah Paket Masuk : " << jumlah << "\n";
    }
}

int main() {
    int pilihan;
    do {
        cout << "+=====+\n";
        cout << "|          MENU PAKET          |\n";
        cout << "+=====+\n";
        cout << "|1. Paket Masuk                |\n";
        cout << "|2. Kirim Paket                |\n";
        cout << "|3. Lihat Antrian              |\n";
        cout << "|4. Keluar                     |\n";
        cout << "+=====+\n";
        cout << "Pilih menu: ";
        cin >> pilihan;
    }
}

```

```

        cin.ignore();

        switch(pilihan) {
            case 1: {
                int jumlahPaket;
                cout << "Masukkan jumlah paket yang ingin diinput:
";

                cin >> jumlahPaket;
                cin.ignore();

                for(int i = 0; i < jumlahPaket; i++) {
                    string nBarang, aTujuan, nResi;
                    cout << "Nama Barang: ";
                    getline(cin, nBarang);
                    cout << "Alamat Tujuan: ";
                    getline(cin, aTujuan);
                    cout << "Nomor Resi: ";
                    getline(cin, nResi);
                    enqueue(nBarang, aTujuan, nResi);
                    cout << "\n";
                }
                cout << "Paket berhasil ditambahkan ke dalam
antrian.\n";
                break;
            }
            case 2:
                dequeue();
                break;
            case 3:
                cetakQueue();
                break;
            case 4:
                cout <<
"+=====+\n";
                cout << "|Senang sekali Anda mau mencoba program
ini!      |\n";
                cout << "|Kami berharap program ini memenuhi
harapan Anda, |\n";
                cout << "|dan masukan Anda akan sangat berharga
bagi kami. |\n";
                cout <<
"+=====+\n";
                break;
            default:
                cout << "Pilihan tidak valid. Silakan coba
lagi.\n";
                break;
        }
    } while(pilihan != 4);
    return 0;

```

```
}
```

## 4.2 Catatan Revisi

Tidak ada

## 4.3 Revisi Program

Tidak ada

## 4.4 Screenshot Program

### 1. Menu Utama dan Tambah Paket

```
+=====+
|           MENU PAKET           |
+=====+
| 1. Paket Masuk                 |
| 2. Kirim Paket                 |
| 3. Lihat Antrian                |
| 4. Keluar                      |
+=====+
Pilih menu: 1
Masukkan jumlah paket yang ingin diinput: 2
Nama Barang: Beras
Alamat Tujuan: Jakarta
Nomor Resi: 1234

Nama Barang: Jagung
Alamat Tujuan: Malang
Nomor Resi: 5678

Paket berhasil ditambahkan ke dalam antrian.
```

### 2. Tampilkan Paket

```
Pilih menu: 3
+=====+
|           PAKET MASUK           |
+=====+
Nama Barang : Beras
Tujuan      : Jakarta
No Resi     : 1234

Nama Barang : Jagung
Tujuan      : Malang
No Resi     : 5678

Jumlah Paket Masuk : 2
```

### 3. Kirim Paket

```
+=====+
|          MENU PAKET          |
+=====+
|1. Paket Masuk                |
|2. Kirim Paket                |
|3. Lihat Antrian              |
|4. Keluar                     |
+=====+
Pilih menu: 2
+=====+
|          KIRIM PAKET          |
+=====+
|Nama Barang : Beras           |
|No Resi      : 1234           |
+=====+
Paket Sedang Dikirimkan ke Jakarta
```