

**LAPORAN AKHIR
PRAKTIKUM ALGORITMA DAN PEMROGRAMAN LANJUT**



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 Algoritma dan Pemrograman Lajut

Pada Tanggal : 10 Juni 2024

Menyetujui,

Asisten Praktikum

Asisten Praktikum

Daniel Ridho Abadi

NIM 123220064

Habib Maulana Akbar

NIM 124220022

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa yang senantiasa mencurahkan rahmat dan hidayah-Nya sehingga kami dapat menyelesaikan praktikum Algoritma dan Pemrograman Lanjut serta laporan akhir praktikum Algoritma dan Pemrograman Lanjut. 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, 10 Juni 2024

Penyusun

Imam Khusain

DAFTAR ISI

LAPORAN AKHIR PRAKTIKUM ALGORITMA DAN PEMROGRAMAN LANJUT	i
HALAMAN PENGESAHAN.....	ii
KATA PENGANTAR.....	iii
DAFTAR ISI	iv
TUGAS 1 QUICK SORT ATAU BUBBLE SORT	1
1.1 Source Code Program	1-5
1.2 Catatan Revisi	5
1.3 Revisi Program.....	5
1.4 Screenshot Program	6
TUGAS 2 FILE	7
2.1 Source Code Program	7-10
2.2 Catatan Revisi	11
2.3 Revisi Program.....	11
2.4 Screenshot Program	11
TUGAS 3 FILE LATIHAN KUIS	12
3.1 Source Code Program	12-14
3.2 Catatan Revisi	14
3.3 Revisi Program.....	14
3.4 Screenshot Program	115
TUGAS 4 LATHAN RESPONSI.....	16
4.1 Source Code Program	17-20
4.2 Catatan Revisi	20
4.3 Revisi Program.....	20
4.4 Screenshot Program	21

TUGAS 1

QUICK SORT ATAU BUBBLE SORT

Buatlah sebuah program untuk menyimpan buku telepon..

1.1 Source Code Program

```
#include <iostream>
#include <string>
using namespace std;

struct Kontak{
    string nama;
    string nomorTelepon;
};

// Fungsi untuk membagi array dan mengembalikan indeks pivot
int partisi(Kontak contacts[], int low, int high, int nomorTelepon) {
    string pivot;
    if (nomorTelepon)
        pivot = contacts[high].nomorTelepon;
    else
        pivot = contacts[high].nama;
    int i = low - 1;

    for (int j = low; j < high; j++) {
        if ((nomorTelepon && contacts[j].nomorTelepon <= pivot) ||
            (!nomorTelepon && contacts[j].nama <= pivot)) {
            i++;
            swap(contacts[i], contacts[j]);
        }
    }
    swap(contacts[i + 1], contacts[high]);
    return i + 1;
}

// Fungsi rekursif untuk mengurutkan array dengan algoritma Quick Sort
void quickSort(Kontak contacts[], int low, int high, int nomorTelepon)
{
    if (low < high) {
        int pi = partisi(contacts, low, high, nomorTelepon);

        quickSort(contacts, low, pi - 1, nomorTelepon);
        quickSort(contacts, pi + 1, high, nomorTelepon);
    }
}
```

```

// Fungsi untuk mencetak semua kontak
void printContacts(Kontak contacts[], int size) {
    for (int i = 0; i < size; i++) {
        cout << contacts[i].nama << " - " << contacts[i].nomorTelepon
<< endl;
    }
}

// Fungsi template untuk Bubble Sort nama
void bubbleSortName(Kontak contacts[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (contacts[j].nama > contacts[j + 1].nama) {
                swap(contacts[j], contacts[j + 1]);
            }
        }
    }
}

// Bubble sort function for sorting by phone number
void bubbleSortPhone(Kontak contacts[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (contacts[j].nomorTelepon > contacts[j +
1].nomorTelepon) {
                swap(contacts[j], contacts[j + 1]);
            }
        }
    }
}

#include <iostream>
#include <string>
using namespace std;

struct kontak{
    string nama;
    string nomorTelepon;
};

// Fungsi untuk membagi array dan mengembalikan indeks pivot
int partisi(kontak person[], int low, int high, int nomorTelepon) {
    string pivot;
    if (nomorTelepon == 1)
        pivot = person[high].nomorTelepon;
    else
        pivot = person[high].nama;
    int i = low - 1;

```

```

        for (int j = low; j < high; j++) {
            if ((nomorTelepon == 1 && person[j].nomorTelepon <= pivot) ||
                (nomorTelepon == 0 && person[j].nama <= pivot)) {
                i++;
                swap(person[i], person[j]);
            }
        }
        swap(person[i + 1], person[high]);
        return i + 1;
    }

// Fungsi rekursif untuk mengurutkan array dengan algoritma Quick Sort
void quickSort(kontak person[], int low, int high, int nomorTelepon) {
    if (low < high) {
        int pi = partisi(person, low, high, nomorTelepon);

        quickSort(person, low, pi - 1, nomorTelepon);
        quickSort(person, pi + 1, high, nomorTelepon);
    }
}

// Fungsi untuk mencetak semua kontak
void printContact(kontak person[], int size) {
    for (int i = 0; i < size; i++) {
        cout << person[i].nama << " - " << person[i].nomorTelepon <<
endl;
    }
}

// Fungsi template untuk Bubble Sort nama
void bubbleSortName(kontak person[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (person[j].nama > person[j + 1].nama) {
                swap(person[j], person[j + 1]);
            }
        }
    }
}

// Bubble sort function for sorting by phone number
void bubbleSortPhone(kontak person[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (person[j].nomorTelepon > person[j + 1].nomorTelepon) {
                swap(person[j], person[j + 1]);
            }
        }
    }
}

```

```

    }
}

int main(){
    int size = 10;
    kontak person[10] = { {"Joko", "0813"}, {"Rendi", "1890"},
{"Farel", "0978"}, {"Ida", "1765"},
{"April", "7890"}, {"Koko", "3456"},
{"Hari", "6723"}, {"Yusuf", "3498"},
{"Mita", "5623"} };

    int pilih_1;
    string ulang;

    do{
        cout << "MENU : " << endl;
        cout <<
"+=====+" << endl;
        cout << "|                      PROGRAM
MEMILIH                      |" << endl;
        cout <<
"|=====|" << endl;
        cout << "| 1. Quick
Sort                      |" << endl;
        cout << "| 2. Bubble
Sort                      |" << endl;
        cout <<
"+=====+" << endl;
        cout << "Pilih no : ";
        cin >> pilih_1;

        if (pilih_1 == 1){
            int pilihan;
            cout << "+=====+" << endl;
            cout << "|Pilih cara mengurutkan buku telepon : |" << endl;
            cout << "|1. Berdasarkan nama                      |" << endl;
            cout << "|2. Berdasarkan nomor telepon              |" << endl;
            cout << "+=====+" << endl;
            cout << "Pilih no : ";
            cin >> pilihan;

            if (pilihan == 1) {
                quickSort(person, 0, size - 1, false);
                cout << "Kontak setelah diurutkan berdasarkan nama: " << endl;
                printContact(person, size);
            } else if (pilihan == 2) {
                quickSort(person, 0, size - 1, true);
            }
        }
    } while (ulang != "q");
}

```



```

        cout << "Kontak setelah diurutkan berdasarkan nomor telepon: "
<< endl;
        printContact(person, size);
        cout <<
"===== " << endl;
    }
}
if (pilih_1 == 2){
    int pilihan;
    cout << "+=====+" << endl;
    cout << "|Pilih cara mengurutkan buku telepon : |" << endl;
    cout << "|1. Berdasarkan nama |" << endl;
    cout << "|2. Berdasarkan nomor telepon |" << endl;
    cout << "+=====+" << endl;
    cout << "Pilih no : ";
    cin >> pilihan;

    if (pilihan == 1) {
        bubbleSortName(person, size);
        cout << "Kontak setelah diurutkan berdasarkan nama:" << endl;
        printContact(person, size);
    } else if (pilihan == 2) {
        bubbleSortPhone (person, size);
        cout << "Kontak setelah diurutkan berdasarkan nomor telepon: "
<< endl;
        printContact(person, size);
        cout <<
"===== " << endl;
    }
}
cout<<"Mau Ulangi Lagi? (Y/N) ";
cin>>ulang;
}
while(ulang == "y" || ulang == "Y");
return 0;
}

```

1.2 Catatan Revisi

Tidak ada

1.3 Revisi Program

Tidak ada

1.4 Screenshot Program

```
MENU :
+=====+
|                                     |
|                PROGRAM MEMILIH    |
|=====+
| 1. Quick Sort                     |
| 2. Bubble Sort                    |
|=====+
Pilih no : 1
+=====+
|Pilih cara mengurutkan buku telepon : |
|1. Berdasarkan nama                  |
|2. Berdasarkan nomor telepon         |
|=====+
Pilih no : 1
Kontak setelah diurutkan berdasarkan nama:
-
April - 7890
Farel - 0978
Hari - 6723
Ida - 1765
Joko - 0813
Koko - 3456
Mita - 5623
Rendi - 1890
Yusuf - 3498
```

```
MENU :
+=====+
|                                     |
|                PROGRAM MEMILIH    |
|=====+
| 1. Quick Sort                     |
| 2. Bubble Sort                    |
|=====+
Pilih no : 2
+=====+
|Pilih cara mengurutkan buku telepon : |
|1. Berdasarkan nama                  |
|2. Berdasarkan nomor telepon         |
|=====+
Pilih no : 1
Kontak setelah diurutkan berdasarkan nama:
-
April - 7890
Farel - 0978
Hari - 6723
Ida - 1765
Joko - 0813
Koko - 3456
Mita - 5623
Rendi - 1890
Yusuf - 3498
```

TUGAS 2

FILE

Kerjasama antar divisi di perusahaan ABC akan mengadakan pertemuan dengan beberapa direksi lain di perusahaan. Pertemuan itu dilakukan untuk membahas kerjasama untuk kolaborasi produk baru yang akan di launching bulan depan. Rapat harus dilaksanakan dalam satu hari dengan ketentuan bahwa dalam ruang rapat hanya ada satu divisi dan tepat satu direksi lain. Perwakilan sekretaris bingung bagaimana harus mengatur jadwal agar tidak ada jadwal tabrakan antar pertemuan karena setiap divisi punya waktu untuk rapat yang mereka ajukan sendiri. Tidak hanya itu setiap direksi juga memiliki lamanya durasi rapat. Sebagai seorang programmer bantulah perusahaan ABC untuk menyelesaikan masalah ini. Berikan jadwal terbaik untuk perusahaan agar dapat bekerjasama dengan banyak divisi.

2.1 Source Code Program

```
#include <iostream>
#include <fstream>
#include <iomanip>
#include <string>

using namespace std;

struct inputt {
    string jurusan;
    string jam;
    string durasi;
};

inputt data[100];
char namafile[20];
int n;

void input() {
    int i;
    cout << "INPUT DATA\n";

    cout << "Nama File = ";
    cin >> namafile;
    ofstream Arsip(namafile, ios::out);
    if (Arsip.is_open()) {
        cout << "Jumlah data = ";
        cin >> n;
        cin.ignore(); // To clear the newline character from the
input buffer
        for (i = 0; i < n; i++) {
            cout << "Data ke-" << i + 1 << " = " << endl;
            cout << "Jurusan = ";
            getline(cin, data[i].jurusan);
```

```

        cout << "Jam = ";
        getline(cin, data[i].jam);
        cout << "Durasi = ";
        getline(cin, data[i].durasi);
        Arsip << data[i].jurusan << endl << data[i].jam << endl <<
data[i].durasi << endl;
    }
    Arsip.close();
}

void output() {
    int i = 0;
    cout << "OUTPUT DATA\n";

    cout << "Nama File = ";
    cin >> namafile;
    ifstream Arsip(namafile);
    cout << "+" << setfill('=') << setw(50) << "+" << "\n";
    cout << "|" << setw(12) << setfill(' ') << "JURUSAN" << setw(5) <<
"|";
    cout << setw(10) << setfill(' ') << "JAM" << setw(7) << "|";
    cout << setw(10) << setfill(' ') << "DURASI" << setw(6) << "|" <<
"\n";
    cout << "+" << setfill('=') << setw(50) << "+" << "\n";
    if (Arsip.is_open()) {
        while (getline(Arsip, data[i].jurusan) && getline(Arsip,
data[i].jam) && getline(Arsip, data[i].durasi)) {
            i++;
        }
    } else {
        cout << "File Not Found!\n";
    }
    for (int a = 0; a < i; a++) {
        cout << "|" << left << setfill(' ') << setw(14) <<
data[a].jurusan << " |";
        cout << left << setfill(' ') << setw(14) << data[a].jam << " |";
        cout << left << setfill(' ') << setw(14) << data[a].durasi <<
"| " << endl;
        cout << "+";
        cout << right << setfill('=') << setw(50) << "+" << "\n";
    }
    Arsip.close();
}

void hasil() {
    int i = 0;

```

```

    inputt data_filtered[100]; // array untuk menyimpan data yang
    sudah difilter
    int filtered_count = 0; // jumlah data yang sudah difilter

    cout << "HASIL\n";

    cout << "Nama File = ";
    cin >> namafile;
    ifstream Arsip(namafile);
    if (Arsip.is_open()) {
        while (getline(Arsip, data[i].jurusan) && getline(Arsip,
data[i].jam) && getline(Arsip, data[i].durasi)) {
            bool jam_exists = false;
            for (int j = 0; j < filtered_count; j++) {
                if (data_filtered[j].jam == data[i].jam) {
                    jam_exists = true;
                    break;
                }
            }

            if (!jam_exists) {
                data_filtered[filtered_count] = data[i];
                filtered_count++;
            }

            i++;
        }
        Arsip.close();

        cout << "+" << setfill('=') << setw(50) << "+" << "\n";
        cout << "|" << setw(12) << setfill(' ') << "JURUSAN" <<
setw(5) << "|";
        cout << setw(10) << setfill(' ') << "JAM" << setw(7) << "|";
        cout << setw(10) << setfill(' ') << "DURASI" << setw(6) << "| "
<< "\n";
        cout << "+" << setfill('=') << setw(50) << "+" << "\n";

        for (int j = 0; j < filtered_count; j++) {
            cout << "| " << setw(14) << setfill(' ') << left <<
data_filtered[j].jurusan << " | ";
            cout << setw(14) << setfill(' ') << left <<
data_filtered[j].jam << " | ";
            cout << setw(14) << setfill(' ') << left <<
data_filtered[j].durasi << "| " << endl;
            cout << "+";
            cout << right << setfill('=') << setw(50) << "+" << "\n";
        }
    } else {

```

```

        cout << "File Not Found!\n";
    }
}

int main() {
    int pilih_1;
    string ulang;

    do {
        cout << "MENU : " << endl;
        cout <<
        "+=====+" << endl;
        cout << "|          PROGRAM
MEMILIH          |" << endl;
        cout <<
        "|=====|" << endl;
        cout << "| 1.
Input          |" << endl;
        cout << "| 2.
Output         |" << endl;
        cout << "| 3.
Hasil          |" << endl;
        cout << "| 4.
Exit           |" << endl;
        cout <<
        "+=====+" << endl;
        cout << "Pilih no : ";
        cin >> pilih_1;

        if (pilih_1 == 1) {
            input();
        } else if (pilih_1 == 2) {
            output();
        } else if (pilih_1 == 3) {
            hasil();
        } else if (pilih_1 == 4) {
            cout << "Terimakasih :)" << endl;
        }
        cout<<"Mau Ulangi Lagi? (Y/N) ";
        cin>>ulang;
    }
    while(ulang == "y" || ulang == "Y");
    return 0;
}

```

2.2 Catatan Revisi

Tidak ada

2.3 Revisi Program

Tidak ada

2.4 Screenshot Program

```
MENU :
+=====+
|                                     |
|                PROGRAM MEMILIH    |
|=====+
| 1. Input                          |
| 2. Output                         |
| 3. Hasil                         |
| 4. Exit                          |
|=====+
Pilih no : 2
OUTPUT DATA
Nama File = file1
+=====+
|  JURUSAN  |  JAM  |  DURASI  |
+=====+
| Divisi Pemasaran | 6      | 2      |
+=====+
| Divisi Produksi  | 8      | 3      |
+=====+
| Divisi HRD       | 12     | 3      |
+=====+
| Divisi Umum      | 12     | 6      |
+=====+
| Divisi Pembelanjaan | 16    | 5      |
+=====+
```

```
Pilih no : 3
HASIL
Nama File = file1
+=====+
|  JURUSAN  |  JAM  |  DURASI  |
+=====+
| Divisi Pemasaran | 6      | 2      |
+=====+
| Divisi Produksi  | 8      | 3      |
+=====+
| Divisi HRD       | 12     | 3      |
+=====+
| Divisi Pembelanjaan | 16    | 5      |
+=====+
```

TUGAS 3

LATIHAN KUIS

Membuat Program yang RS Harapan Bersama inginkan untuk program ini, yaitu :
1. Program memiliki 4 menu (Menambah Data Pasien, Melihat Data Pasien, Mencari Data Pasien, Exit).

3.1 Source Code Program

```
#include <iostream>
using namespace std;

struct Pasien{
    int id;
    string nama;
    int umur;
    string penyakit;
};

int jumlahPasien;
Pasien pasienbaru[10];

void inputdataPasien(){

    cout << "Masukkan jumlah pasien yang akan ditambahkan: ";
    cin >> jumlahPasien;

    for (int i = 0; i < jumlahPasien; i++){

        cout << "Input Data Pasien" << endl;
        pasienbaru[i].id = i;
        cout << "Id Pasien : ";
        cin >> pasienbaru[i].id;
        cout << "Nama Pasien : ";
        cin >> pasienbaru[i].nama;
        cout << "Penyakit Pasien : ";
        cin >> pasienbaru[i].penyakit;
        cout << "Umur Pasien : ";
        cin >> pasienbaru[i].umur;
    }
}

void lihatdataPasien(){
    cout << "Data Pasien:" << endl;
    cout << "ID\tNama" << endl;
    for (int i = 0; i < jumlahPasien; ++i) {
        cout << pasienbaru[i].id << "\t" << pasienbaru[i].nama << endl;
    }
}

void caridataPasien(){
    int idCari;
```



```

    cout << "Masukkan ID pasien yang ingin dicari: ";
    cin >> idCari;

    bool ditemukan = false;
    for (int i = 0; i < jumlahPasien; ++i) {
        if (pasienbaru[i].id == idCari) {
            cout << "Data Pasien:" << endl;
            cout << "ID: " << pasienbaru[i].id<< endl;
            cout << "Nama: " << pasienbaru[i].nama<< endl;
            cout << "Umur: " << pasienbaru[i].umur<< endl;
            cout << "Penyakit: " << pasienbaru[i].penyakit<< endl;
            ditemukan = true;
            break;
        }
    }

    if (!ditemukan) {
        cout << "Data pasien dengan ID " << idCari << " tidak
ditemukan." << endl;
    }
}

int main(){

    int pilih_1;
    string ulang;

    do{
        cout << "MENU : " << endl;
        cout <<
        "+=====+" << endl;
        cout << "|                MENU RUMAH
SAKIT                |" << endl;
        cout <<
        "|=====|" << endl;
        cout << "| 1. Input Data
Pasien                |" << endl;
        cout << "| 2. Lhat Data
Pasien                |" << endl;
        cout << "| 3. Cari Data
Pasien                |" << endl;
        cout << "| 4.
Exit                |" << endl;
        cout <<
        "+=====+" << endl;
        cout << "Pilih Menu : ";
        cin >> pilih_1;
        if (pilih_1 == 1){

```

```

        inputdataPasien ();
        cout <<
"===== " << endl;
    }
    if (pilih_1 == 2){
        lihatdataPasien ();
        cout <<
"===== " << endl;
    }
    if (pilih_1 == 3){
        caridataPasien ();
        cout <<
"===== " << endl;
    }
    if (pilih_1 == 4){

        cout<<"SEMANGAT BERJUANG KAWAN!!!" << endl;
        cout <<
"===== " << endl;
    }
    cout<<"Mau Ulangi Lagi? (Y/N) ";
    cin>>ulang;}
    while(ulang == "y" || ulang == "Y");
}

```

3.2 Catatan Revisi

Tidak ada catatan revisi

3.1 Revisi Program

Tidak ada revisi

3.1 Screenshot Program

```
MENU :
+=====+
|           MENU RUMAH SAKIT           |
+=====+
| 1. Input Data Pasien                  |
| 2. Lhat Data Pasien                  |
| 3. Cari Data Pasien                  |
| 4. Exit                              |
+=====+
Pilih Menu : 1
Masukkan jumlah pasien yang akan ditambahkan: 2
Input Data Pasien
Id Pasien : 1
Nama Pasien : Affan
Penyakit Pasien : Pilek
Umur Pasien : 25
Input Data Pasien
Id Pasien : 2
Nama Pasien : Resa
Penyakit Pasien : TBC
Umur Pasien : 30
=====
MENU :
+=====+
|           MENU RUMAH SAKIT           |
+=====+
| 1. Input Data Pasien                  |
| 2. Lhat Data Pasien                  |
| 3. Cari Data Pasien                  |
| 4. Exit                              |
+=====+
Pilih Menu : 3
Masukkan ID pasien yang ingin dicari: 1
Data Pasien:
ID: 1
Nama: Affan
Umur: 25
Penyakit: Pilek
=====
```

TUGAS 4

LATIHAN RESPONSI

Membuat Program untuk menginput data kependudukan berupa nomor kependudukan, nama, jenis kelamin, dan tanggal lahir.

4.1 Source Code Program

```
#include <iostream>
#include <fstream>
#include <iomanip>
#include <string>
#include <algorithm>

using namespace std;

struct DataPenduduk {
    string NIK;
    string nama;
    string jenisKelamin;
    string tanggalLahir;
};

DataPenduduk data[100];
int jumlahData = 0;

void input() {
    cout <<
    "+=====+\n";
    cout << "|                INPUT DATA\n";
    "KEPENDUDUKAN                |\n";
    cout <<
    "+=====+\n";
    ofstream file("penduduk.txt", ios::app);
    if (file.is_open()) {
        int jumlahTambah;
        cout << "Masukkan jumlah data penduduk yang ingin ditambahkan: ";
        cin >> jumlahTambah;
        cin.ignore();

        for (int i = 0; i < jumlahTambah; i++) {
            DataPenduduk dapen;

            cout << "\nMasukkan data penjualan baru ke-" << i + 1 << ":" <<
endl;

            cout << "Masukkan nomor NIK\t\t\t: ";
            getline(cin, dapen.NIK);
            cout << "Masukkan Nama KTP\t\t\t: ";
            getline(cin, dapen.nama);
```

```

        cout << "Jenis Kelamin (L/P)\t\t: ";
        cin >> dapen.jenisKelamin;
        cout << "Tanggal Lahir (YYYY-MM-DD)\t: ";
        cin >> dapen.tanggalLahir;

        data[jumlahData++] = dapen;

        file << dapen.NIK << " " << dapen.nama << " " <<
dapen.jenisKelamin << " " << dapen.tanggalLahir << endl;

        cin.ignore();
    }

    file.close();

    cout << "\nData Penduduk berhasil ditambahkan." << endl;
} else {
    cout << "File penduduk.txt tidak dapat dibuka." << endl;
}
}

void openDataPenduduk() {
    ifstream file("penduduk.txt");
    if (file.is_open()) {
        int count = 0;
        while (file >> data[count].NIK >> data[count].nama >>
data[count].jenisKelamin >> data[count].tanggalLahir) {
            count++;
        }
        file.close();

        sort(data, data + count, [](const DataPenduduk &a, const
DataPenduduk &b) {
            return a.NIK < b.NIK;
        });

        cout << "Data Penduduk:" << endl;
        cout << "-----"
--" << endl;
        cout << setw(16) << left << "Nomor NIK" << " | " << setw(10) << left
<< "Nama" << " | " << setw(10) << left << "Jenis Kelamin" << " | " <<
setw(10) << left << "Tanggal Lahir" << endl;
        cout << "-----"
--" << endl;

        for (int i = 0; i < count; i++) {

```

```

        cout << setw(15) << left << data[i].NIK << " | " << setw(10) <<
left << data[i].nama << " | " << setw(13) << left << data[i].jenisKelamin <<
" | " << setw(10) << left << data[i].tanggalLahir << endl;
    }
    cout << "-----"
--" << endl;
    } else {
        cout << "File penduduk.txt tidak dapat dibuka." << endl;
    }
}

void searchDataPenduduk() {
    cout << "Masukkan Tanggal Lahir yang ingin dicari (YYYY-MM-DD): ";
    string tanggalLahirCari;
    cin >> tanggalLahirCari;

    ifstream file("penduduk.txt");
    if (file.is_open()) {
        bool found = false;
        DataPenduduk temp;
        while (file >> temp.NIK >> temp.nama >> temp.jenisKelamin >>
temp.tanggalLahir) {
            if (temp.tanggalLahir == tanggalLahirCari) {
                cout << "-----"
-----" << endl;
                cout << " | DATA DITEMUKAN
BRO! | " << endl;
                cout << "-----"
-----" << endl;
                cout << "Nomor NIK: " << temp.NIK << endl;
                cout << "Nama: " << temp.nama << endl;
                cout << "Jenis Kelamin: " << temp.jenisKelamin << endl;
                cout << "Tanggal Lahir: " << temp.tanggalLahir << endl;
                found = true;
            }
        }
        file.close();

        if (!found) {
            cout << "Data dengan Tanggal Lahir " << tanggalLahirCari << "
tidak ditemukan." << endl;
        }
    } else {
        cout << "File penduduk.txt tidak dapat dibuka." << endl;
    }
}

void menuOutput() {

```

```

    int pilihan;
    do {
        cout <<
        "+=====+\n";
        cout << " |                      MENU
OUTPUT | \n";
        cout <<
        "+=====+\n";
        cout << " | 1. Tampilkan Seluruh
Data | \n";
        cout << " | 2. Cari
Data | \n";
        cout << " | 3. Kembali ke Menu
Utama | \n";
        cout <<
        "+=====+\n";
        cout << "Pilih menu: ";
        cin >> pilihan;

        switch (pilihan) {
            case 1:
                openDataPenduduk();
                break;
            case 2:
                searchDataPenduduk();
                break;
            case 3:
                return;
            default:
                cout << "\nANDA SALAH DALAM MEMILIH KAWAN :(. \n";
        }
    } while (true);
}

int main() {
    int pilih_1;
    string ulang;

    do {
        cout << "MENU : " << endl;
        cout << "+=====+"
<< endl;
        cout << " |                      PROGRAM MEMILIH | "
<< endl;
        cout << " |===== | "
<< endl;
        cout << " | 1. Input | "
<< endl;

```

```

        cout << "| 2. Output" << endl;
    cout << "| 3. Exit" << endl;
    cout << "+=====+" << endl;
    cout << "Pilih no : ";
    cin >> pilih_1;

    if (pilih_1 == 1) {
        input();
    } else if (pilih_1 == 2) {
        menuOutput();
    } else if (pilih_1 == 3) {
        cout << "Terimakasih :)" << endl;
        break;
    }
    cout << "Mau Ulangi Lagi? (Y/N) ";
    cin >> ulang;
} while (ulang == "y" || ulang == "Y");

return 0;
}

```

4.2 Catatan Revisi

Tidak ada

4.3 Revisi Program

Tidak ada

4.4 Screenshot Program

```
MENU :
+=====+
|                                     |
|               PROGRAM MEMILIH     |
|=====+
| 1. Input                               |
| 2. Output                             |
| 3. Exit                               |
|=====+
Pilih no : 2
+=====+
|                                     |
|               MENU OUTPUT          |
|=====+
| 1. Tampilkan Seluruh Data           |
| 2. Cari Data                       |
| 3. Kembali ke Menu Utama           |
|=====+
Pilih menu: 1
Data Penduduk:
-----
Nomor NIK      | Nama      | Jenis Kelamin | Tanggal Lahir
-----
3577020503690001 | NONO      | L              | 1969-03-05
3577022903040001 | IMAM      | L              | 2004-03-29
3577023903740001 | WATI      | P              | 1974-03-39
3577024607990001 | IJUNG     | P              | 1999-07-07
-----
```

```
Pilih menu: 2
Masukkan Tanggal Lahir yang ingin dicari (YYYY-MM-DD): 2004-03-29
-----
|                                     |
|               DATA DITEMUKAN BRO! |
|=====+
Nomor NIK: 3577022903040001
Nama: IMAM
Jenis Kelamin: L
Tanggal Lahir: 2004-03-29
+=====+
|                                     |
|               MENU OUTPUT          |
|=====+
| 1. Tampilkan Seluruh Data           |
| 2. Cari Data                       |
| 3. Kembali ke Menu Utama           |
|=====+
```