

LAPORAN PROYEK UAS PRAKTIKUM STRUKTUR DATA



Dosen Pengampu Mata Kuliah :

Asep Jamaludin, S.Si., M.Kom.

Asisten Laboratorium :

Ihsan Fauzi

Disusun oleh :

Afridho Ikhsan	2210631170002
Ananta Ziaurohman Az Zaki	2210631170007
Reiza Alithian Syach	2210631170098
Ridhaka Gina Amalia	2210631170099
Risma Auliya Salsabilla	2210631170100
Yesaya Adhelyasa Vareen Tetuko	2210631170107

Kelas 2A

PROGRAM STUDI INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS SINGAPERBANGSA KARAWANG

2023

Laporan Proyek UAS : Program Pengolahan Data Sekolah

I. PENDAHULUAN

A. Identifikasi Permasalahan

Pengolahan data saat ini menjadi sangat penting dalam menjalankan berbagai proses di banyak sektor, termasuk di bidang pendidikan. Sekolah sebagai lembaga pendidikan memiliki kebutuhan yang mendesak dalam pengelolaan data siswa yang efisien dan akurat. Pengolahan data siswa yang baik akan memberikan manfaat dalam penyimpanan arsip data, pengambilan keputusan, pemantauan perkembangan siswa, serta pengelolaan administrasi dan keuangan sekolah.

Pengolahan data siswa yang dilakukan secara manual atau menggunakan *spreadsheet* saat ini dinilai sudah tidak lagi optimal karena mengakibatkan pemborosan waktu serta peningkatan resiko kesalahan dalam penginputan dan pemrosesan data.

Metode penyimpanan data siswa yang masih mengandalkan kertas juga memiliki resiko tinggi terjadinya kerugian data. Kertas rentan hilang, rusak ataupun basah, baik karena kesalahan manusia maupun faktor alam. Hal ini menyebabkan kesulitan lagi dalam pemulihan data yang rusak atau bahkan kehilangan data siswa yang penting. Ditambah pula akan ada peningkatan biaya untuk mencetak dan menyimpan kertas. Selain itu, penggunaan kertas dalam jumlah besar juga menghabiskan sumber daya alam dan meningkatkan dampak negatif terhadap lingkungan.

Perekapan data siswa secara manual memiliki keterbatasan karena tidak adanya fitur yang cukup memadai untuk mengelola dan memanipulasi data dengan lengkap sewaktu-waktu. Dilain hal, dalam mengelola data siswa, sekolah juga perlu mempertimbangkan aspek keuangan, terutama dalam hal pembayaran SPP (Sumbangan Pembinaan Pendidikan). Proses pembayaran yang dilakukan dengan cara biasa atau menggunakan sistem terpisah memberikan tantangan tersendiri baik dalam mencatat, menyimpan, memantau, ataupun mengelola data pembayaran siswa. Sehingga diperlukan kehati-hatian yang lebih karena resiko kesalahan dan kehilangan data yang tinggi.

B. Manfaat/Urgensi

1. Efisiensi dan Produktivitas

Program pengolahan data siswa memungkinkan staf sekolah untuk mengelola data secara lebih efisien dan juga menghemat waktu. Dengan otomatisasi proses pengolahan data, seperti pencatatan kehadiran, penyimpanan arsip, atau pembayaran SPP, staf sekolah dapat mengalokasikan waktu dan sumber daya mereka untuk kegiatan yang lebih produktif, seperti mengajar, memberikan pembinaan kepada siswa, atau mengembangkan kurikulum.

2. Akurasi dan Ketepatan Data

Program pengolahan data siswa memberikan jaminan akurasi dan ketepatan data yang lebih tinggi dibandingkan dengan metode manual atau menggunakan *spreadsheet*. Risiko kesalahan dalam penginputan data atau pemrosesan manual dapat diminimalisir sehingga informasi yang diperoleh menjadi lebih dapat dipastikan. Hal ini penting dalam pengambilan keputusan yang berdampak pada pendidikan dan perkembangan siswa.

3. Integrasi dan Kolaborasi

Sistem program pengolahan data siswa dapat diintegrasikan dengan sistem atau aplikasi lain yang digunakan di sekolah, seperti sistem manajemen pembelajaran, sistem keuangan, atau platform pembelajaran *online*. Hal ini memfasilitasi kolaborasi dan pertukaran data antara berbagai departemen atau komponen dalam lingkungan sekolah, memperkuat kerja tim, dan meningkatkan efektivitas proses kerja.

4. Manajemen Sumbangan Pembinaan Pendidikan (SPP)

Sistem program pengolahan data siswa dapat membantu dalam pengelolaan SPP atau sumbangan lainnya dengan lebih efektif. Fitur yang terintegrasi memungkinkan pencatatan pembayaran, pengingat tunggakan, dan laporan pembayaran secara otomatis. Hal ini membantu sekolah dalam melacak dan mengelola arus kas, mempercepat proses administrasi keuangan, dan meningkatkan transparansi dalam penggunaan dana sekolah.

5. Persyaratan Regulasi dan Pelaporan

Sekolah sering kali diharuskan memenuhi persyaratan regulasi dan melaporkan data siswa kepada otoritas pendidikan atau pihak berwenang. Dengan sistem program pengolahan data siswa, sekolah dapat dengan mudah menghasilkan laporan yang dibutuhkan secara cepat dan akurat.

6. Persiapan untuk Masa Depan Digital

Dalam era digital yang terus berkembang, sekolah perlu beradaptasi dengan teknologi dan mempersiapkan siswa untuk menghadapi tuntutan dunia digital. Dengan mengimplementasikan sistem program pengolahan data siswa, sekolah memberikan contoh positif dan memperlihatkan pentingnya penggunaan teknologi dalam kehidupan sehari-hari. Siswa juga dapat belajar dan terlibat dalam penggunaan teknologi ini, mempersiapkan mereka untuk kebutuhan masa depan yang semakin tergantung pada teknologi.

C. Justifikasi Permasalahan

Berdasarkan identifikasi masalah di atas, terdapat beberapa masalah yang berhubungan dengan penyusunan data yang masih dilakukan secara manual. Manusia memang sejatinya tidak pernah luput dari kesalahan. Oleh karena itu, dengan adanya program yang telah dibuat, diharapkan dapat membantu untuk meminimalisir kesalahan-kesalahan *input* data yang dilakukan secara manual.

Umumnya, program ini dirancang untuk mempermudah para staf sekolah untuk membantu mereka dalam melakukan penyusunan dan pemeriksaan data siswa. Program ini memaksimalkan fungsi-fungsi yang dapat memberikan *output* sesuai dengan yang diinginkan oleh para staf sekolah. Tidak hanya untuk stafnya saja, tetapi program ini juga dapat digunakan oleh para siswa dan siswi sekolah tersebut untuk mengakses fitur yang memang dirancang untuk memenuhi kebutuhan mereka.

Pengolahan data siswa yang baik melalui metode digital akan memberikan sejumlah manfaat yang signifikan dalam berbagai aspek pengelolaan sekolah. Dengan menggunakan metode digital, data siswa dapat disimpan secara elektronik dalam sistem yang terpusat. Ini menghilangkan kebutuhan akan penyimpanan fisik berkas dan kertas yang memakan ruang. Selain itu, data dapat diatur dan dicari dengan mudah melalui fitur pencarian atau kategori yang relevan. Hal ini menghemat waktu dalam mencari informasi tertentu dan mengurangi resiko kehilangan data tersebut.

Dengan menggunakan metode digital dalam pengolahan data siswa, sekolah juga dapat meningkatkan efisiensi waktu dan tenaga dalam penyimpanan arsip data, pemantauan perkembangan siswa, dan pengelolaan administrasi, serta keuangan sekolah. Ini memberikan manfaat yang signifikan dalam menghemat waktu,

Metode penyimpanan data siswa yang masih mengandalkan kertas memiliki risiko tinggi terjadinya kerugian data. Kertas rentan hilang, rusak, atau basah, baik karena kesalahan manusia maupun faktor alam. Dalam hal ini, pemulihan data yang rusak atau hilang akan sulit dilakukan. Dengan adanya pengolahan data secara digital, risiko tersebut dapat dikurangi dan data dapat di-*backup* secara teratur untuk meminimalisir risiko kehilangan data.

Penggunaan kertas dalam jumlah besar akan meningkatkan biaya untuk mencetak dan menyimpan kertas. Selain itu, penggunaan kertas yang berlebihan juga berkontribusi terhadap penggunaan sumber daya alam dan meningkatkan dampak negatif terhadap lingkungan. Dengan melakukan pengolahan data secara digital, sekolah dapat mengurangi penggunaan kertas, menghemat biaya, dan berkontribusi pada pelestarian lingkungan.

Pengolahan data siswa secara manual memiliki keterbatasan dalam mengelola data dengan lengkap sewaktu-waktu. Dengan menggunakan metode digital, sekolah dapat menggunakan fitur yang memadai untuk mengelola data dengan lebih efektif dan efisien. Hal ini akan mempermudah pengambilan keputusan dan pemantauan perkembangan siswa.

Dalam mengelola data siswa, sekolah juga perlu mempertimbangkan aspek keuangan, terutama dalam hal pembayaran SPP. Proses pembayaran yang dilakukan dengan cara biasa atau menggunakan sistem terpisah dapat menyebabkan tantangan dalam mencatat, menyimpan, memantau, dan mengelola data pembayaran siswa. Dengan pengolahan data secara digital, sekolah dapat memiliki sistem yang terintegrasi untuk memudahkan pengelolaan keuangan, termasuk pencatatan pembayaran dan pemantauan status pembayaran siswa.

Dengan demikian, pengolahan data siswa yang efisien dan akurat melalui metode digital menjadi penting bagi sekolah dalam menyediakan manfaat dalam berbagai aspek. Hal tersebut mencakup penghematan waktu dan biaya, keamanan data, pengelolaan keuangan, perlindungan lingkungan, dan masih banyak lagi. Oleh karena itu, program ini benar-benar dirancang untuk memenuhi semua hal yang telah disebutkan sebelumnya.

Melalui pembuatan sistem program pengolahan data siswa, sekolah dapat menghadapi tantangan modern dengan lebih baik. Pembuatan sistem tersebut juga dapat meningkatkan efisiensi operasional, meningkatkan kualitas pendidikan, dan mengikuti tren teknologi yang terus berkembang. Hal ini sudah pasti akan

memberikan sekolah tersebut sebuah keunggulan berkompetitif dalam menyediakan pendidikan berkualitas dan relevan untuk siswa.

II. PENYELESAIAN

A. Ide Penyelesaian

Dalam proyek ini, fokusnya adalah mengembangkan program pengolahan data sekolah yang efisien dan terstruktur. Berikut adalah ide penyelesaian yang dapat diimplementasikan:

a. Implementasi Sistem Manajemen Data Terintegrasi

Membangun atau mengadopsi sistem manajemen data sekolah yang terintegrasi dan berbasis database. Sistem ini harus mampu mengumpulkan, menyimpan, dan mengelola data yang berhubungan dengan siswa, data siswa, tambah data siswa, hapus data siswa, ubah data siswa, pembayaran SPP, dan bimbingan konseling.

Mengatur struktur data yang jelas dan terstruktur untuk masing-masing entitas, seperti data siswa, tambah data siswa, hapus data siswa, dan sebagainya. Mengimplementasikan fungsi pencarian, pembaruan, dan penghapusan data yang efisien untuk memudahkan akses dan pengelolaan data sekolah.

b. Pengembangan Sistem Pelaporan:

Mengembangkan sistem pelaporan yang komprehensif dan mudah digunakan untuk menghasilkan laporan data sekolah yang relevan dan terbaru. Menyediakan laporan yang meliputi data siswa, data siswa baru, ubah data, data pembayaran SPP, dan data bimbingan konseling. Menyusun dashboard visual yang intuitif untuk memudahkan pemantauan dan analisis data.

B. Struktur Data yang Diterapkan

- a. Pointer : Pada program yang kami buat pointer digunakan untuk menyimpan data siswa. Dengan menggunakan pointer, dapat dilakukan pengelolaan data yang efisien. Misalnya, penggunaan pointer dalam linked list dapat mempermudah penambahan, penghapusan, dan pencarian data siswa
- b. Array : Di dalam program kami terdapat linked list untuk menampung siswa-siswa yang ada di setiap kelas, untuk membedakan penempatan data siswa tersebut perlu dibuatkan linked list yang berbeda, linked list" tersebut ditampung di dalam array

- c. Struct : Struct juga terdapat di dalamnya untuk menunjang dalam pembuatan linked list, stack, queue, dan mengubah data siswa untuk menjadikan variable siswa, Stack: Dalam implementasi stack, Juga dapat menggunakan struct untuk menyimpan data siswa dalam setiap elemen stack. Setiap elemen stack dapat berisi struct yang berisi nama-nama siswa. Ketika elemen baru ditambahkan (push) atau dihapus (pop) dari stack, data siswa dalam struct tersebut akan dimanipulasi.
- d. Linked List : Siswa direpresentasikan oleh node, node tersebut terhubung satu sama lain sehingga keterhubungan antar node tersebut dapat membentuk linked list. Yang dimana seperti yang sudah dijelaskan tadi terdapat linked list yang berbeda" untuk menampung siswa di setiap kelas
- e. Stack : Di dalam program terdapat menu dimana kita bisa menampilkan data siswa berdasarkan yang paling baru mendaftarkan sebagai siswa di sekolah, dengan penerapan stack dapat merealisasikan operasi yang diinginkan, seperti Menambahkan data siswa jadi setiap kali ada siswa baru yang mendaftar, dapat menggunakan operasi push pada stack untuk menambahkan data siswa tersebut ke stack. Data siswa yang baru akan menjadi elemen teratas dalam stack. Apabila pengguna memilih menu untuk menampilkan data siswa terbaru, dapat menggunakan fungsi bawaan yang ada pada kontruktor Stack untuk melihat data siswa teratas dalam stack. Dengan demikian, dapat menampilkan data siswa yang paling baru mendaftar di sekolah. Jika diperlukan, Jika ingin menyediakan opsi untuk menghapus data siswa dari stack menggunakan operasi pop. Misalnya, jika siswa tersebut telah keluar dari sekolah atau tidak aktif lagi, Anda dapat menghapus data siswanya dari stack.
- f. Queue : Pengelolaan data antrian BK pun juga ada di program yang kami buat, seperti menambah, melihat, dan menghapus, struktur data queue dapat menunjang kebutuhan fitur tersebut.

Menambah data: Ketika ada siswa baru yang mendaftar ke BK, dan dapat menggunakan operasi enqueue pada queue untuk menambahkan data siswa tersebut ke antrian. Siswa baru akan dimasukkan ke dalam bagian belakang antrian (rear).

Melihat data: Untuk melihat data antrian, yang dapat menggunakan operasi peek pada queue yang nanti akan dibuatkan fungsi nya sendiri untuk melihat elemen pertama dalam antrian, yaitu siswa yang berada di bagian depan antrian (front).

Dengan demikian, dapat menampilkan data siswa yang sedang berada di dalam antrian BK.

Menghapus data: Ketika BK selesai menangani siswa yang berada di bagian depan antrian, Kita bisa menggunakan operasi dequeue pada queue untuk menghapus siswa tersebut dari antrian. Setelah penghapusan, elemen berikutnya di depan antrian akan menjadi elemen baru yang akan dilayani selanjutnya.

III. IMPLEMENTASI STRUKTUR DATA

A. Source Code

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

using namespace std;
#define maks 20

struct siswa // Struct
{
    string nama, tanggalLahir;
    int kelas, IdSiswa, sisaSPP;
    bool lunasSPP;
};

struct dataSiswa // Node Link list
{
    siswa identitasSiswa;
    struct dataSiswa *prev, *next;
};

struct dataSiswa *bantu = NULL, *hapus = NULL;

struct dataSiswa *awalKelas[12] = {NULL, NULL, NULL, NULL, NULL, NULL,
NULL, NULL, NULL, NULL, NULL, NULL};
struct dataSiswa *akhirKelas[12] = {NULL, NULL, NULL, NULL, NULL, NULL,
NULL, NULL, NULL, NULL, NULL, NULL};

int nisnbawaan = 1000000001, tanyakelas, pil;

string namaSiswaBawaan[12][10] = {"Ahmad Fauzi", "Ahmad Sutrisno", "Andi
Nugroho", "Ani Puspitasari", "Arianto Wijaya", "Arif Sutrisno", "Budi
Santoso", "Budi Wijaya", "Dewi Indah", "Kartika Sari"}, {"Dedi Pratama",
"Dian Fitri", "Dian Purnama", "Dodi Santoso", "Evi Susanti", "Harianto
```



```
Prabowo", "Heriawan Putra", "Indah Fitriani", "Indah Sari", "Joko
Sutrisno"}, {"Maya Dewi", "Maya Sari", "Maya Siti", "Maman Santoso",
"Maman Sutrisno", "Rina Amelia", "Rina Astuti", "Rina Cahyani", "Rina
Dewi", "Rina Fitriani"}, {"Rina Indriani", "Rina Wahyuni", "Rina
Wijayanti", "Rina Wijayanto", "Ricky Wijaya", "Sari Purnama", "Sinta
Indriani", "Siti Rahmawati", "Suhardi Prabowo", "Surya Nugroho"}, {"Surya
Pratama", "Surya Wijaya", "Tuti Indriani", "Triana Dewi", "Yulia
Setiawati", "Yuli Susanti", "Yuni Susanti", "Yuni Susanto", "Yuni
Susilawati", "Yuny Susilawati"}, {"Maya Kusuma", "Aditya Wijaya", "Rani
Sari", "Devi Putri", "Anita Rahayu", "Indra Permana", "Ratna Asmila",
"Yoga Putra", "Farhan Santoso", "Putri Cahaya"}, {"Bagus Prayogo ", "Joko
Surya", "Yusuf Wijaya", "Fira Lestari", "Rina Aulia", "Eko Nugroho",
"Dina Aulia", "Hadi Prasetya", "Bambang Wijaya", "Sinta Dewi"}, {"Deny
Nugroho", "Aditya Oktavian", "Rapi Pratama", "Bulan putri", "Maya
Wijaya", "Riko Simanjuntak", "Dwi Septian", "Adnan Wijaya", "Novri
Herdianto", "Tiara Putri"}, {"Raka Setiawan", "Ebi Santoso", "Reza
Alfalah", "Devani Putri", "Saykila Maulida", "Aldo Pramata", "Medis
Putri", "Fadil Alfares", "Daffa Firmansyah", "sakia Wati"}, {"Ajis
Guswan", "Pratama Indah", "Didik Hermawan", "aulia syaputri", "Lola
Amalia", "Andika Alfarezy", "Wulan Dari", "Faldo Purnomo", "Muhamad
Rehan", "Rika Olin"}, {"Ragil Santoso", "Yuda Harianto", "Widi Pertiwi",
"Reisa elsa", "Najwa Indah", "Mail Muslikin", "Rara Kasturi", "Radit
Wijaya", "Altaf Safar", "Aprilia Putri"}, {"Agus Nugroho", "Bambang
Santoso", "Dedi Pratama", "Desi Permata", "Indah Fitriani", "Joko
Pratama", "Rina Astuti", "Rudi Wijaya", "Suhardi Prabowo", "Triana
Dewi"}}};
```

```
string tanggalLahir[12][10] = {"03 Januari 2005", "12 Februari 2005",
"27 Maret 2005", "09 April 2005", "21 Mei 2005", "06 Juni 2005", "17 Juli
2005", "29 Agustus 2005", "10 September 2005", "25 Oktober 2005"}, {"08
November 2005", "16 Desember 2005", "01 Januari 2005", "14 Februari
2005", "28 Maret 2005", "05 April 2005", "19 Mei 2005", "07 Juni 2005",
"20 Juli 2005", "02 Agustus 2005 {"13 September 2005", "30 Oktober 2005",
"04 November 2005", "23 Desember 2005", "11 Januari 2005", "02 Februari
2005", "19 Maret 2005", "13 April 2005", "26 Mei 2005", "08 Juni 2005"},
{"24 Juli 2005", "16 Agustus 2005", "09 September 2005", "28 Oktober
2005", "05 November 2005", "22 Desember 2005", "07 Januari 2005", "18
Februari 2005", "04 Maret 2005", "14 April 2005"}, {"30 Mei 2004", "06
Oktober 2004", "19 Juli 2004", "31 Agustus 2004", "12 September 2004",
"27 Oktober 2004", "03 November 2004", "21 Desember 2004", "09 Januari
2004", "22 Februari 2004"}, {"06 Maret 2004", "18 April 2004", "23 Mei
2004", "07 Juni 2004", "19 Juli 2004", "01 Agustus 2004", "15 September
2004", "29 Oktober 2004", "12 November 2004", "24 Desember 2004"}, {"21
Januari 2004", "18 Desember 2004", "21 Februari 2004", "19 Agustus 2004",
"09 Juni 2004", "27 Juli 2004", "26 Januari 2004", "24 September 2004",
```

```

"15 November 2004", "29 Februari 2004"}, {"27 Oktober 2004", "25 Desember
2004", "07 September 2004", "12 Maret 2004", "27 Januari 2004", "10
September 2004", "15 Juni 2004", "07 November 2004", "05 Januari 2004",
"20 November 2004"}, {"31 Juli 2003", "27 Mei 2003", "04 Oktober 2003",
"19 Desember 2003", "23 Agustus 2003", "11 Mei 2003", "07 Februari 2003",
"24 Juni 2003", "26 Juni 2003", "12 Desember 2003"}, {"27 Februari 2003",
"15 Agustus 2003", "19 Oktober 2003", "07 Maret 2003", "25 Agustus 2003",
"05 Februari 2003", "29 November 2003", "26 Februari 2003", "14 November
2003", "18 Juni 2003"}, {"11 Juni 2003", "18 Juli 2003", "29 April 2003",
"10 Agustus 2003", "08 Agustus 2003", "29 Desember 2003", "27 Maret
2003", "24 November 2003", "22 September 2003", "24 Januari 2003"}, {"06
Agustus 2003", "11 Februari 2003", "07 Desember 2003", "11 Juli 2003",
"21 Juli 2003", "17 Agustus 2003", "01 September 2003", "17 November
2003", "29 Maret 2003", "23 April 2003"}];

string kelasBawaan[12] = {"10 MIPA 1", "10 MIPA 2", "10 IPS 1", "10 IPS
2",
                        "11 MIPA 1", "11 MIPA 2", "11 IPS 1", "11 IPS
2",
                        "12 MIPA 1", "12 MIPA 2", "12 IPS 1", "12 IPS
2"};

struct Queue
{
    int rear, front;
    struct dataSiswa *siswa[20];
} antrianBimbinganKonseling;

struct Stack
{
    struct dataSiswa *top;
    Stack()
    {
        top = NULL;
    }

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

    void push(struct dataSiswa *siswaYangDipush)
    {
        struct dataSiswa *siswa = new dataSiswa;
        siswa->identitasSiswa.nama = siswaYangDipush->identitasSiswa.nama;
    }
}

```

```

        siswa->identitasSiswa.kelas = siswaYangDipush->identitasSiswa.kelas;
        siswa->identitasSiswa.IdSiswa = siswaYangDipush->identitasSiswa.IdSiswa;
        siswa->identitasSiswa.tanggalLahir = siswaYangDipush->identitasSiswa.tanggalLahir;
        siswa->identitasSiswa.lunasSPP = siswaYangDipush->identitasSiswa.lunasSPP;
        siswa->identitasSiswa.sisaSPP = siswaYangDipush->identitasSiswa.sisaSPP;
        siswa->next = top;
        top = siswa;
    }

    void pop()
    {
        struct dataSiswa *temp = top;
        top = top->next;

        delete temp;
    }

    void tampilSiswaTerbaru()
    {
        if (top == NULL)
        {
            cout << "List siswa kosong. \n";
            return;
        }
        struct dataSiswa *currentStudent = top;
        cout << "List Siswa (Berdasarkan Data Terbaru Siswa SMAN 01 Karawang) : \n";
        int listNumber = 1;
        while (currentStudent != NULL)
        {
            cout << listNumber++ << ". " << currentStudent->identitasSiswa.nama << endl;
            currentStudent = currentStudent->next;
        }
    }
};

struct Stack daftarSiswaTerbaru,
    temp;

```

```

bool isFull(Queue *selectedQueue)
{
    return selectedQueue->rear == maks;
}

bool isEmpty(Queue *selectedQueue)
{
    return selectedQueue->rear == 0;
}

void enqueue(Queue *selectedQueue, struct dataSiswa *enqueuedPointer)
{
    if (isFull(selectedQueue))
    {
        std::cout << "Antrian sudah penuh!" << endl;
    }
    else
    {
        selectedQueue->siswa[selectedQueue->rear] = *enqueuedPointer;
        selectedQueue->rear++;
    }
}

void dequeue(Queue *selectedQueue)
{
    for (int i = selectedQueue->front; i < selectedQueue->rear; i++)
    {
        selectedQueue->siswa[i] = selectedQueue->siswa[i + 1];
    }
    selectedQueue->rear--;
}

void tampilMenu()
{
    cout << "\n\n\t\t===== SELAMAT DATANG DI PROGRAM PENGELOLAAN DATA  

    SISWA SMAN 01 KARAWANG =====\n\n"
        << "\n1. Data Siswa\n"
        << "\n2. Tambah Data Siswa\n"
        << "\n3. Hapus Data Siswa\n"
        << "\n4. Ubah Data Siswa\n"
        << "\n5. Pembayaran SPP\n"
        << "\n6. Bimbingan Konseling\n"
        << "\n7. Keluar\n"
        << endl;
}

```

```

}

void dataKelas()
{
    cout << "=====" << endl;
    for (int i = 0; i < 12; i++)
    {
        cout << "\n"
            << i + 1 << " ) " << kelasBawaan[i] << endl;
    }
    cout << endl;
}

void pencarianSiswa(int idSiswaYangDicari)
{
    for (int i = 0; i < 12; i++)
    {
        bantu = awalKelas[i];
        while (bantu != NULL)
        {
            if (bantu->identitasSiswa.IdSiswa == idSiswaYangDicari)
                break;
            bantu = bantu->next;
        }

        if (bantu != NULL && bantu->identitasSiswa.IdSiswa ==
idSiswaYangDicari)
            break;
    }
}

void tampilDataSiswa()
{
    cout << "Nama                : " << bantu->identitasSiswa.nama <<
"\n"
        << "Kelas                : " << kelasBawaan[bantu-
>identitasSiswa.kelas - 1] << "\n"
        << "ID Siswa                : " << bantu->identitasSiswa.IdSiswa <<
"\n"
        << "Tanggal Lahir            : " << bantu-
>identitasSiswa.tanggalLahir << "\n"
        << "Status Pembayaran SPP : " << ((bantu-
>identitasSiswa.lunasSPP) ? "Sudah Lunas" : "Belum Lunas") << "\n"
        << "Sisa Tunggalan            : " << bantu->identitasSiswa.sisaSPP <<
"\n"

```

```

        << "\n===== \n"
        << endl;
    }

void dataSiswaDipilih()
{
    cout << "Data Siswa yang Dipilih" << endl;
    tampilDataSiswa();
}

void tampilkanSiswaTertentu(int id)
{
    pencarianSiswa(id);
    if (bantu == NULL)
    {
        cout << "Siswa yang Anda cari tidak ditemukan." << endl;
    }
    else
    {
        cout << "Data Siswa yang Dicari : " << endl;
        tampilDataSiswa();
    }
}

void listDataSiswa(struct dataSiswa *dataSiswaKelas)
{
    int numberList = 1;
    if (dataSiswaKelas == NULL)
    {
        cout << "Data siswa tidak tersedia." << endl;
    }
    else
    {
        bantu = dataSiswaKelas;
        while (bantu != NULL)
        {
            cout << "Absen : " << numberList++ << endl;
            tampilDataSiswa();
            bantu = bantu->next;
        }
    }
}

void subMenu1()
{

```

```

system("cls");
dataKelas();
cout << "Pilih Kelas : ";
cin >> tanyakelas;

system("cls");
cout << "Data Siswa Kelas " << kelasBawaan[tanyakelas - 1] << "\n"
    << "-----\n"
    << endl;
switch (tanyakelas)
{
case 1:
    listDataSiswa(awalKelas[0]);
    break;
case 2:
    listDataSiswa(awalKelas[1]);
    break;
case 3:
    listDataSiswa(awalKelas[2]);
    break;
case 4:
    listDataSiswa(awalKelas[3]);
    break;
case 5:
    listDataSiswa(awalKelas[4]);
    break;
case 6:
    listDataSiswa(awalKelas[5]);
    break;
case 7:
    listDataSiswa(awalKelas[6]);
    break;
case 8:
    listDataSiswa(awalKelas[7]);
    break;
case 9:
    listDataSiswa(awalKelas[8]);
    break;
case 10:
    listDataSiswa(awalKelas[9]);
    break;
case 11:
    listDataSiswa(awalKelas[10]);
    break;
case 12:

```

```

        listDataSiswa(awalKelas[11]);
        break;
    default:
        cout << "Input yang Anda masukkan salah." << endl;
    }
}

void menu1()
{
    system("cls");
    cout << "Tampilkan Data Siswa\n"
        << "=====\n"
        << "\n1. Berdasarkan Kelas\n"
        << "\n2. Berdasarkan Pendaftaran Siswa yang Terbaru\n"
        << "\n3. Cari ID Siswa\n"
        << "\nPilih : ";

    cin >> pil;
    switch (pil)
    {
    case 1:
        subMenu1();
        break;
    case 2:
        system("cls");
        daftarSiswaTerbaru.tampilSiswaTerbaru();
        break;
    case 3:
        int idCari;
        cout << "\nInputkan ID Siswa yang Ingin Anda Lihat : ";
        cin >> idCari;

        if (cin.fail())
        {
            cout << "Input yang Anda masukkan salah." << endl;
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
        }
        else
        {
            tampilkanSiswaTertentu(idCari);
        }
    }
}

```



```

void penempatanPosisiNodeBaru(struct dataSiswa *nodeBaru, int
kelasSiswaIndex)
{
    bantu = awalKelas[kelasSiswaIndex - 1];
    if (bantu == NULL)
    {
        awalKelas[kelasSiswaIndex - 1] = nodeBaru;
        nodeBaru->next = NULL;
        nodeBaru->prev = NULL;
    }
    else if (bantu->next == NULL)
    {
        if (nodeBaru->identitasSiswa.nama < bantu->identitasSiswa.nama)
        {
            nodeBaru->next = bantu;
            bantu->prev = nodeBaru;
            awalKelas[kelasSiswaIndex - 1] = nodeBaru;
        }
        else
        {
            bantu->next = nodeBaru;
            nodeBaru->prev = bantu;
            nodeBaru->next = NULL;
            akhirKelas[kelasSiswaIndex - 1] = nodeBaru;
        }
    }
    else
    {
        while (bantu != NULL)
        {
            if (nodeBaru->identitasSiswa.nama < bantu-
>identitasSiswa.nama)
            {
                if (bantu == awalKelas[kelasSiswaIndex - 1])
                {
                    nodeBaru->next = bantu;
                    bantu->prev = nodeBaru;
                    awalKelas[kelasSiswaIndex - 1] = nodeBaru;
                }
                else
                {
                    nodeBaru->prev = bantu->prev;
                    nodeBaru->next = bantu;
                    bantu->prev->next = nodeBaru;
                    bantu->prev = nodeBaru;
                }
            }
        }
    }
}

```

```

        }
        break;
    }
    else if (bantu->next == NULL || nodeBaru->identitasSiswa.nama
< bantu->next->identitasSiswa.nama)
    {
        nodeBaru->prev = bantu;
        nodeBaru->next = bantu->next;
        bantu->next = nodeBaru;
        if (nodeBaru->next != NULL)
        {
            nodeBaru->next->prev = nodeBaru;
        }
        else
        {
            akhirKelas[kelasSiswaIndex - 1] = nodeBaru;
        }
        break;
    }
    bantu = bantu->next;
}
}
}

```

```

void tambahSiswaFunction(bool insertPengguna, string namaSiswa = "", int
kelasSiswa = 0, int IdSiswa = 0, string tanggalLahir = "", bool lunasSPP
= true, int sisaSPP = 0)
{

```

```

    system("cls");
    dataSiswa *siswaBaru = new dataSiswa();

    if (insertPengguna)
    {
        cout << "Inputkan Data Siswa yang Ingin Ditambah\n"
        << "=====\n"
        << endl
        << "Nama          : ";

        cin.ignore();
        getline(cin, siswaBaru->identitasSiswa.nama);
        cout << "ID Siswa      : ";
        cin >> siswaBaru->identitasSiswa.IdSiswa;
        cout << "Tanggal Lahir : ";
        cin.ignore();
        getline(cin, siswaBaru->identitasSiswa.tanggalLahir);
        dataKelas();
    }
}

```

```

        cout << "Masukkan di Kelas (1-12) : ";
        cin >> siswaBaru->identitasSiswa.kelas;
        cout << endl;

        penempatanPosisiNodeBaru(siswaBaru, siswaBaru-
>identitasSiswa.kelas);
        cout << "Siswa telah ditambahkan." << endl;
    }
    else
    {
        siswaBaru->identitasSiswa.nama = namaSiswa;
        siswaBaru->identitasSiswa.kelas = kelasSiswa;
        siswaBaru->identitasSiswa.IdSiswa = IdSiswa;
        siswaBaru->identitasSiswa.tanggalLahir = tanggalLahir;
        siswaBaru->identitasSiswa.lunasSPP = lunasSPP;
        siswaBaru->identitasSiswa.sisaSPP = sisaSPP;
        penempatanPosisiNodeBaru(siswaBaru, kelasSiswa);
    }
    daftarSiswaTerbaru.push(siswaBaru);
}

void penentuanPosisiHapusNode(int indexKelas)
{
    if (bantu->next == NULL && bantu->prev == NULL)
    {
        awalKelas[indexKelas - 1] = NULL;
        delete hapus;
    }
    else if (bantu == awalKelas[indexKelas - 1])
    {
        awalKelas[indexKelas - 1] = awalKelas[indexKelas - 1]->next;
        awalKelas[indexKelas - 1]->prev = NULL;
        delete hapus;
    }
    else if (bantu->next == NULL)
    {
        bantu->prev->next = NULL;
        delete hapus;
    }
    else
    {
        bantu->prev->next = bantu->next;
        bantu->next->prev = bantu->prev;
        delete hapus;
    }
}

```

```

    if (bantu->identitasSiswa.IdSiswa == daftarSiswaTerbaru.top->identitasSiswa.IdSiswa)
    {
        daftarSiswaTerbaru.pop();
    }
    else
    {
        while (daftarSiswaTerbaru.top->identitasSiswa.IdSiswa != bantu->identitasSiswa.IdSiswa)
        {
            temp.push(daftarSiswaTerbaru.top);
            daftarSiswaTerbaru.pop();
        }
        daftarSiswaTerbaru.pop();

        while (!temp.isEmpty())
        {
            daftarSiswaTerbaru.push(temp.top);
            temp.pop();
        }
    }
}

void hapusSiswa(bool insertPengguna, int kelasSiswaIndex = 0)
{
    if (insertPengguna)
    {
        system("cls");
        cout << "PILIH \n"
              << "1. ID Siswa\n"
              << "2. Siswa Terbaru\n"
              << "Hapus Siswa Berdasarkan : ";
        cin >> pil;

        switch (pil)
        {
            case 1:
                system("cls");
                dataKelas();
                cout << "Pilih Kelas : ";
                cin >> tanyakelas;
                cout << endl;

                if (awalKelas[tanyakelas - 1] == NULL)
                {

```

```

        cout << "Data siswa tidak tersedia." << endl;
    }
    else
    {
        int IdHapusSiswa;
        listDataSiswa(awalKelas[tanyakelas - 1]);
        cout << "Masukkan ID Siswa yang Ingin Dihapus : ";
        cin >> IdHapusSiswa;
        cout << endl;

        pencarianSiswa(IdHapusSiswa);

        if (bantu == NULL)
        {
            cout << "Siswa dengan ID yang dimasukkan tidak
ditemukan." << endl;
        }
        else
        {
            dataSiswaDipilih();
            char konfir;
            cout << "Apakah Anda ingin menghapus data siswa
tersebut? (Y/N) : ";
            cin >> konfir;
            if (konfir == 'Y' || konfir == 'y')
            {
                penentuanPosisiHapusNode(tanyakelas);
                cout << "\nData siswa telah dihapus." << endl;
            }
            else if (konfir == 'N' || konfir == 'n')
            {
                cout << "\nData siswa batal dihapus." << endl;
            }
        }
        cout << endl;
    }
    break;
case 2:
    daftarSiswaTerbaru.tampilSiswaTerbaru();
    char konfir;
    cout << "Apakah Anda yakin ingin menghapus data siswa terbaru?
(Y/N) : ";
    cin >> konfir;
    if (konfir == 'Y' || konfir == 'y')
    {

```

```

        bantu = awalKelas[daftarSiswaTerbaru.top-
>identitasSiswa.kelas - 1];
        while (bantu->identitasSiswa.IdSiswa !=
daftarSiswaTerbaru.top->identitasSiswa.IdSiswa)
        {
            if (bantu->next == NULL)
            {
                bantu = NULL;
                break;
            }
            else
                bantu = bantu->next;
        }

        penentuanPosisiHapusNode(daftarSiswaTerbaru.top-
>identitasSiswa.kelas);
        cout << "\nData siswa telah dihapus." << endl;
    }
    else if (konfir == 'N' || konfir == 'n')
    {
        cout << "\nData siswa batal dihapus." << endl;
    }
    break;
}
}
else
{
    penentuanPosisiHapusNode(kelasSiswaIndex);
}
}

void namaBaru()
{
    cout << "Masukkan Nama Baru          : ";
    cin.ignore();
    getline(cin, bantu->identitasSiswa.nama);

    // Meng-update nama pada stack
    while (daftarSiswaTerbaru.top->identitasSiswa.IdSiswa != bantu-
>identitasSiswa.IdSiswa)
    {
        temp.push(daftarSiswaTerbaru.top);
        daftarSiswaTerbaru.pop();
    }
}

```

```

    daftarSiswaTerbaru.top->identitasSiswa.nama = bantu-
>identitasSiswa.nama;

    while (!temp.isEmpty())
    {
        daftarSiswaTerbaru.push(temp.top);
        temp.pop();
    }
}

void kelasBaru()
{
    if (bantu->identitasSiswa.kelas != tanyakelas)
    {
        // Hapus siswa dari kelas asal
        hapusSiswa(false, bantu->identitasSiswa.kelas);

        // Tambahkan siswa ke kelas baru
        tambahSiswaFunction(false, bantu->identitasSiswa.nama, bantu->
identitasSiswa.kelas, bantu->identitasSiswa.IdSiswa, bantu-
>identitasSiswa.tanggalLahir, bantu->identitasSiswa.lunasSPP, bantu-
>identitasSiswa.sisaSPP);
    }
}

void bayarSPP()
{
    cout << "Pembayaran sudah lunas?\n"
        << "1. Iya\n"
        << "2. Tidak\n"
        << "Pilih : ";
    cin >> pil;

    switch (pil)
    {
    case 1:
        bantu->identitasSiswa.lunasSPP = true;
        bantu->identitasSiswa.sisaSPP = 0;
        break;
    case 2:
        bantu->identitasSiswa.lunasSPP = false;
        cout << "\nSisa Tunggakan yang Belum Dibayar : ";
        cin >> bantu->identitasSiswa.sisaSPP;
        break;
    }
}

```

```

}

void ubahDataSiswa()
{
    system("cls");
    dataKelas();
    cout << "Pilih Kelas : ";
    cin >> tanyakelas;
    cout << endl;

    if (awalKelas[tanyakelas - 1] == NULL)
    {
        cout << "Data siswa tidak tersedia." << endl;
    }
    else
    {
        int idUbahSiswa;
        listDataSiswa(awalKelas[tanyakelas - 1]);
        cout << "Masukkan ID Siswa yang Ingin Diubah: ";
        cin >> idUbahSiswa;
        cout << endl;

        pencarianSiswa(idUbahSiswa);

        if (bantu == NULL)
        {
            cout << "Siswa dengan ID yang dimasukkan tidak ditemukan." <<
endl;
        }
        else
        {
            system("cls");
            dataSiswaDipilih();
            cout << "Pilih Data yang Ingin Diubah\n"
                << "~~~~~\n"
                << "\n1. Nama\n"
                << "\n2. Kelas\n"
                << "\n3. Tanggal Lahir\n"
                << "\n4. Status Pembayaran\n"
                << "\n5. Semua Data\n"
                << endl;

            // int pil;
            cout << "Pilih Menu : ";
            cin >> pil;

```



```

switch (pil)
{
case 1:
    system("cls");
    dataSiswaDipilih();
    namaBaru();
    cout << "\n=====\\n";
    cout << "Nama berhasil diubah." << endl;
    break;
case 2:
    system("cls");
    dataSiswaDipilih();
    dataKelas();
    cout << "Masukkan Kelas Baru      : ";
    cin >> bantu->identitasSiswa.kelas;
    cout << "\n=====\\n";
    cout << "Kelas berhasil diubah." << endl;
    kelasBaru();
    break;
case 3:
    system("cls");
    dataSiswaDipilih();
    cout << "Masukkan Tanggal Lahir Baru : ";
    cin.ignore();
    getline(cin, bantu->identitasSiswa.tanggalLahir);
    cout << "\n=====\\n";
    cout << "Tanggal lahir berhasil diubah." << endl;
    break;
case 4:
    system("cls");
    dataSiswaDipilih();
    bayarSPP();
    break;
case 5:
    system("cls");
    dataSiswaDipilih();
    namaBaru();
    dataKelas();
    cout << "Masukkan Kelas Baru      : ";
    cin >> bantu->identitasSiswa.kelas;
    cout << "Masukkan Tanggal Lahir Baru : ";
    cin.ignore();
    getline(cin, bantu->identitasSiswa.tanggalLahir);
    bayarSPP();
    cout << "\n=====\\n";

```

```

        cout << "\nSemua data siswa berhasil diubah." << endl;
        kelasBaru();
        break;
    default:
        cout << "\n===== \n";
        cout << "Input yang Anda masukkan salah." << endl;
        break;
    }
}
}

void pembayaranSPP()
{
    system("cls");
    dataKelas();
    cout << "Pilih Kelas : ";
    cin >> tanyakelas;
    cout << endl;

    // Menampilkan daftar siswa dalam kelas yang dipilih
    listDataSiswa(awalKelas[tanyakelas - 1]);

    // Meminta input ID Siswa untuk melakukan pembayaran SPP
    int idSiswa;
    cout << "Masukkan ID Siswa : ";
    cin >> idSiswa;

    // Mencari siswa dengan ID Siswa yang sesuai
    bantu = awalKelas[tanyakelas - 1];
    while (bantu != NULL)
    {
        if (bantu->identitasSiswa.IdSiswa == idSiswa)
        {
            // Memeriksa apakah siswa telah membayar SPP
            if (bantu->identitasSiswa.lunasSPP)
            {
                cout << "Siswa dengan ID Siswa " << idSiswa << " sudah
melakukan pembayaran SPP." << endl;
            }
            else
            {
                // Meminta input jumlah pembayaran SPP
                int jumlahBayar;
                cout << "Masukkan Jumlah Pembayaran SPP : ";
            }
        }
    }
}

```

```

        cin >> jumlahBayar;

        // Mengupdate status pembayaran dan sisa SPP
        bantu->identitasSiswa.sisaSPP -= jumlahBayar;
        if (bantu->identitasSiswa.sisaSPP == 0)
            bantu->identitasSiswa.lunasSPP = true;

        cout << "Pembayaran SPP sebesar " << jumlahBayar << "
berhasil dilakukan untuk siswa dengan ID Siswa "
            << idSiswa << "." << endl;
    }
    break;
}
bantu = bantu->next;
}

// Jika ID Siswa tidak ditemukan
if (bantu == NULL)
{
    cout << "Siswa dengan ID Siswa " << idSiswa << " tidak ditemukan."
<< endl;
}
}

void tampilAntrianBK()
{
    cout << "\nAntrian Bimbingan Konseling" << endl;
    int urutanAntrian = antrianBimbinganKonseling.front;
    while (urutanAntrian != antrianBimbinganKonseling.rear)
    {
        cout << urutanAntrian + 1 << ". " <<
antrianBimbinganKonseling.siswa[urutanAntrian++]->identitasSiswa.nama <<
endl
            << endl;
    }
}

void bimbinganKonseling()
{
    system("cls");
    int idSiswa;
    cout << "Bimbingan Konseling\n"
        << "=====\n"
        << "\n1. Tambah Antrian Bimbingan Konseling\n"
        << "\n2. Lihat Antrian Bimbingan Konseling\n"

```

```

        << "\n3. Hapus Antrian\n"
        << "\nPilih Menu : ";
    cin >> pil;

    switch (pil)
    {
    case 1:
        bantu = NULL;
        cout << "Masukkan ID Siswa yang ingin melakukan Bimbingan
Konseling : ";
        cin >> idSiswa;

        pencarianSiswa(idSiswa);

        if (bantu != NULL)
        {
            dataSiswaDipilih();
            char konfir;
            cout << "Apakah ID siswa yang Anda inputkan sudah benar? (Y/N)
: ";

            cin >> konfir;
            if (konfir == 'Y' || konfir == 'y')
            {
                cout << "Siswa telah ditambahkan ke antrian." << endl;
                enqueue(&antrianBimbinganKonseling, bantu);
            }
            else if (konfir == 'N' || konfir == 'n')
            {
                cout << "\nPenambahan antrian siswa tersebut dibatalkan."
<< endl;
            }
        }
        else
            cout << "Data siswa tidak ditemukan." << endl;

        break;
    case 2:
        if (isEmpty(&antrianBimbinganKonseling))
            cout << "Antrian masih kosong." << endl;
        else
        {
            tampilAntrianBK();
        }

        break;

```

```

case 3:
    char konfir;
    if (isEmpty(&antrianBimbinganKonseling))
    {
        cout << "Antrian masih kosong." << endl;
    }
    else
    {
        tampilAntrianBK();

        cout << "Apakah Anda yakin ingin memanggil siswa selanjutnya?
(Y/N) : ";
        cin >> konfir;
        if (konfir == 'Y' || konfir == 'y')
        {
            dequeue(&antrianBimbinganKonseling);
            cout << "Antrian telah diperbarui." << endl;
        }
        else if (konfir == 'N' || konfir == 'n')
        {
            cout << "\nPembaruan antrian dibatalkan." << endl;
        }
    }
}

void menuUtama()
{
    char balikMenu;

    do
    {
        system("cls");
        tampilMenu();
        cout << "Pilih Menu : ";
        cin >> pil;
        switch (pil)
        {
            case 1:
                menu1();
                break;
            case 2:
                tambahSiswaFunction(true);
                break;
            case 3:

```

```

        hapusSiswa(true);
        break;
    case 4:
        ubahDataSiswa();
        break;
    case 5:
        pembayaranSPP();
        break;
    case 6:
        bimbinganKonseling();
        break;
    case 7:
        system("cls");
        cout << "\n\n\n\t\t\tTerima kasih telah menggunakan program
ini. Sampai jumpa!\n\n\n";
        break;
    default:
        cout << "Input yang Anda masukkan salah." << endl;
        break;
}

while (balikMenu != 'N' || balikMenu != 'n')
{
    cout << "Apakah Anda ingin kembali ke menu utama (Y/N) ? ";
    cin >> balikMenu;
    cin.clear();
    cin.ignore();
    if (balikMenu == 'N' || balikMenu == 'n')
    {
        pil = 7;
        system("cls");
        cout << "\n\n\n\t\t\tTerima kasih telah menggunakan
program ini. Sampai jumpa!\n\n\n";
        break;
    }
    else if (balikMenu == 'Y' || balikMenu == 'y')
    {
        menuUtama();
        break;
    }
    else
    {
        cout << "Input yang Anda masukkan salah." << endl;
    }
    cin.clear();
}

```

```

        cin.ignore(numeric_limits<streamsize>::max(), '\n');
    }

    } while (pil != 7);
}

int main()
{
    for (int i = 0; i < 12; i++)
    {
        for (int j = 0; j < 10; j++)
        {
            tambahSiswaFunction(false, namaSiswaBawaan[i][j], i + 1,
nisnbawaan++, tanggalLahir[i][j]);
        }
    }

    menuUtama();
    system("pause");

    return 0;
}

```

B. Penjelasan Code

```

#include <iostream>
#include <string>
#include <limits>

```

Baris (1-4) : Untuk mengimpor fungsi-fungsi yang sudah didefinisikan pada header file.

```

using namespace std;
#define maks 20

```

Baris (5 - 6) : Mengimpor seluruh simbol dan definisi dari namespace std ke dalam ruang lingkup saat ini. std adalah namespace standar dalam C++ yang berisi banyak fungsi, kelas, dan objek yang diperlukan untuk pemrograman dasar seperti input/output, manipulasi string, penggunaan vektor, dll.

Selain itu, pada baris ke-6 juga mendefinisikan variabel maks yang nanti akan digunakan untuk nilai maksimum queue.

```

struct siswa // Struct
{
    string nama, tanggalLahir;
    int kelas, IdSiswa, sisaSPP;
    bool lunasSPP;
};

```

Baris (8 - 13) : Mendeklarasikan struct baru yang dinamakan dengan siswa yang akan digunakan untuk menggambarkan entitas siswa dengan berbagai properti terkait, yaitu nama, tanggal lahir, kelas, ID Siswa, sisa tunggakan SPP, dan juga status pembayaran SPP. Semua properti tersebut juga di deklarasikan dengan berbagai tipe data yang berbeda.

```

struct dataSiswa // Node Link list
{
    siswa identitasSiswa;
    struct dataSiswa *prev, *next;
};

```

Baris (15 - 19) : Mendefinisikan sebuah struct dengan nama dataSiswa yang digunakan untuk membangun linked list dari objek siswa.

```

struct dataSiswa *bantu = NULL, *hapus = NULL;

struct dataSiswa *awalKelas[12] = {NULL, NULL, NULL, NULL, NULL, NULL,
NULL, NULL, NULL, NULL, NULL, NULL};
struct dataSiswa *akhirKelas[12] = {NULL, NULL, NULL, NULL, NULL, NULL,
NULL, NULL, NULL, NULL, NULL, NULL};

int nisnbawaan = 1000000001, tanyakelas, pil;

```

Baris 21 : Mendefinisikan dua pointer ke struct dataSiswa, yaitu bantu dan hapus, yang awalnya diinisialisasi sebagai NULL. Pointer bantu dan hapus akan digunakan dalam operasi traversal dan penghapusan node dalam linked list.

Baris 23 : Mendefinisikan array awalKelas yang berisi 12 elemen pointer ke struct dataSiswa. Setiap elemen diinisialisasi sebagai NULL. Array ini akan digunakan untuk menyimpan pointer ke node pertama dalam linked list siswa untuk setiap kelas

Baris 24 : Sama seperti awalKelas yang berisi 12 elemen pointer ke struct dataSiswa. Bedanya array akhirKelas digunakan untuk menyimpan pointer ke node akhir dalam linked list siswa untuk di setiap kelas.

Baris 26 : Melakukan inisialisasi variabel nisnbawaan yang merupakan nilai awal untuk Nomor Induk Siswa Nasional (NISN) yang diberikan secara otomatis kepada siswa yang baru ditambahkan ke linked list & juga ada deklarasi 2 variabel lainnya yaitu tanyakelas & pil yang nantinya akan digunakan untuk kebutuhan program.

Baris (28 - 32) : Melakukan inisialisasi pada variabel namaSiswaBawaan, tanggalLahir, & kelasBawaan yang akan digunakan untuk menyimpan data awal (dummy data) siswa dalam linked list berdasarkan kelas. Data ini dapat digunakan sebagai contoh untuk mengisi linked list atau melakukan operasi lainnya yang melibatkan data siswa.

```
struct Queue
{
    int rear, front;
    struct dataSiswa *siswa[20];
} antrianBimbinganKonseling;
```

Baris (34 - 38) : Mendefinisikan sebuah struct bernama Queue yang digunakan untuk merepresentasikan antrian (queue) untuk bimbingan konseling siswa.

Baris (40 - 90) : Mendefinisikan sebuah struct bernama Stack yang digunakan untuk merepresentasikan tumpukan (stack) data siswa. Struct Stack pada program kami terdapat beberapa anggota dan fungsi-fungsi berikut:

- struct dataSiswa *top: Pointer top bertipe struct dataSiswa digunakan sebagai penanda elemen teratas (top) dalam tumpukan. Pointer ini akan menunjuk ke elemen teratas tumpukan.
- Stack(): Konstruktor Stack digunakan untuk menginisialisasi nilai awal top dengan NULL. Konstruktor ini akan dipanggil saat objek Stack dibuat.
- bool isEmpty(): Fungsi isEmpty() digunakan untuk memeriksa apakah tumpukan kosong atau tidak. Jika top bernilai NULL, berarti tumpukan kosong, dan fungsi ini akan mengembalikan true. Jika top tidak NULL, berarti tumpukan tidak kosong, dan fungsi ini akan mengembalikan false.

- d. `void push(struct dataSiswa *siswaYangDipush):` Fungsi `push()` digunakan untuk menambahkan elemen baru ke dalam tumpukan. Fungsi ini menerima parameter `siswaYangDipush` yang merupakan pointer ke `struct dataSiswa`. Di dalam fungsi ini, sebuah objek baru siswa dari `struct dataSiswa` akan dibuat dan nilainya disalin dari `siswaYangDipush`. Elemen siswa kemudian ditambahkan ke tumpukan dengan mengatur `next` untuk menunjuk ke elemen `top` sebelumnya, dan `top` diperbarui untuk menunjuk ke elemen siswa yang baru ditambahkan.
- e. `void pop():` Fungsi `pop()` digunakan untuk menghapus elemen teratas (`top`) dari tumpukan. Di dalam fungsi ini, pointer `temp` digunakan untuk menyimpan alamat elemen `top` saat ini. `top` diperbarui untuk menunjuk ke elemen di bawahnya (elemen sebelumnya dalam tumpukan), dan elemen yang dihapus (elemen awalnya berada di `top`) dilepaskan dari memori menggunakan operator `delete`.
- f. `void tampilSiswaTerbaru():` Fungsi `tampilSiswaTerbaru()` digunakan untuk menampilkan nama-nama siswa dalam tumpukan secara terurut, dimulai dari elemen teratas (`top`) sebagai data terbaru. Jika tumpukan kosong, akan ditampilkan pesan bahwa daftar siswa kosong. Jika tumpukan tidak kosong, fungsi ini akan menampilkan nama-nama siswa secara berurutan dengan nomor urutannya.

```
struct Stack daftarSiswaTerbaru, temp;
```

Baris 92 : Mendeklarasikan dua objek dari `struct Stack`, yaitu `daftarSiswaTerbaru` dan `temp`.

1. `daftarSiswaTerbaru` adalah objek `Stack` yang digunakan untuk menyimpan daftar siswa dengan data terbaru. Objek ini akan digunakan untuk menambahkan siswa baru dan menampilkan daftar siswa terbaru.
2. `temp` adalah objek `Stack` yang digunakan sebagai variabel sementara dalam beberapa operasi yang melibatkan manipulasi tumpukan siswa. Objek ini dapat digunakan untuk melakukan penyimpanan sementara atau pemrosesan data sebelum data tersebut ditambahkan ke `daftarSiswaTerbaru` atau digunakan untuk tujuan lain yang diperlukan dalam konteks program.

```

bool isFull(Queue *selectedQueue)
{
    return selectedQueue->rear == maks;
}

bool isEmpty(Queue *selectedQueue)
{
    return selectedQueue->rear == 0;
}

void enqueue(Queue *selectedQueue, struct dataSiswa *enqueuedPointer)
{
    if (isFull(selectedQueue))
    {
        std::cout << "Antrian sudah penuh!" << endl;
    }
    else
    {
        selectedQueue->siswa[selectedQueue->rear] = enqueuedPointer;
        selectedQueue->rear++;
    }
}

void dequeue(Queue *selectedQueue)
{
    for (int i = selectedQueue->front; i < selectedQueue->rear; i++)
    {
        selectedQueue->siswa[i] = selectedQueue->siswa[i + 1];
    }
    selectedQueue->rear--;
}

```

Baris (94-125) : Seperti yang sudah kita ketahui bahwa di program ini terdapat pendeklarasian queue yang nantinya akan digunakan untuk mengelola antrian bimbingan konseling. Pada baris 94-125 ini adalah pembuatan fungsi yang akan digunakan untuk operasi yang dilakukan untuk perubahan pada queue nanti, fungsi-fungsi tersebut antara lain :

1. `bool isEmpty(Queue *selectedQueue)`: Fungsi ini digunakan untuk memeriksa apakah antrian yang dipilih kosong atau tidak. Mengembalikan nilai true jika antrian kosong (`rear == 0`), dan false jika tidak kosong.
2. `void enqueue(Queue *selectedQueue, struct dataSiswa *enqueuedPointer)`: Fungsi ini digunakan untuk menambahkan elemen baru ke antrian. Jika antrian

sudah penuh (mencapai batas maksimum), pesan "Antrian sudah penuh!" akan dicetak. Jika antrian belum penuh, pointer `enqueuePointer` akan ditambahkan ke elemen antrian pada indeks `rear`, dan `rear` akan ditingkatkan.

3. `void dequeue(Queue *selectedQueue)`: Fungsi ini digunakan untuk menghapus elemen pertama (elemen yang berada di depan) dari antrian. Elemen-elemen yang tersisa akan maju ke posisi sebelumnya. `rear` akan dikurangi 1 untuk menandakan bahwa elemen telah dihapus.

```
void tampilMenu()
{
    cout << "\n\n\t\t===== SELAMAT DATANG DI PROGRAM PENGELOLAAN DATA
SISWA SMAN 01 KARAWANG =====\n\n"
        << "\n1. Data Siswa\n"
        << "\n2. Tambah Data Siswa\n"
        << "\n3. Hapus Data Siswa\n"
        << "\n4. Ubah Data Siswa\n"
        << "\n5. Pembayaran SPP\n"
        << "\n6. Bimbingan Konseling\n"
        << "\n7. Keluar\n"
        << endl;
}
```

Baris (127 - 138) : Untuk menampilkan menu utama yang ada pada program

```
void dataKelas()
{
    cout << "===== " << endl;
    for (int i = 0; i < 12; i++)
    {
        cout << "\n"
            << i + 1 << " ) " << kelasBawaan[i] << endl;
    }
    cout << endl;
}
```

Baris (140-149) : Untuk menampilkan list kelas yang ada pada program (Terdapat 12 kelas)

```
void pencarianSiswa(int idSiswaYangDicari)
{
    for (int i = 0; i < 12; i++)
    {
        bantu = awalKelas[i];
```

```

while (bantu != NULL)
{
    if (bantu->identitasSiswa.IdSiswa == idSiswaYangDicari)
        break;
    bantu = bantu->next;
}

if (bantu != NULL && bantu->identitasSiswa.IdSiswa ==
idSiswaYangDicari)
    break;
}
}

```

Baris (151-166) : Untuk melakukan pencarian data siswa berdasarkan ID yang diberikan.

Baris (168 - 217) : Fungsi-fungsi yang dibuat untuk menampilkan data-data siswa, yang nantinya akan digunakan berulang kali dan menghindari penulisan code yang sama secara berulang.

Baris (219 - 271) : Fungsi subMenu1 merupakan sebuah submenu dari menu 1 yang memberikan pengguna arahan dan pilihan mengenai data siswa dari kelas mana yang ingin dilihat dan menampilkannya setelah pengguna memilih.

Baris (273 - 308) : Fungsi untuk menu 1 yang mengarahkan pengguna untuk memilih mengenai data siswa berdasarkan apa yang ingin dilihat.

Baris (310 - 374) : Fungsi penempatanPosisiNodeBaru bertujuan untuk menempatkan node baru (data siswa baru) dalam urutan yang sesuai berdasarkan abjad nama siswa

Baris (376 - 413) : Fungsi tambahSiswaFunction digunakan untuk menambahkan data siswa baru ke dalam sistem, baik dengan input dari pengguna maupun dengan nilai yang diberikan melalui parameter.

Baris (415 - 459) : Fungsi penentuanPosisiHapusNode digunakan untuk menghapus sebuah node siswa dari linked list kelas dan juga menghapus node tersebut dari stack daftarSiswaTerbaru.

Baris (461 – 551) : Fungsi hapusSiswa digunakan untuk menghapus data siswa dari struktur data yang ada.

Baris (553 - 573) : Fungsi namaBaru digunakan untuk mengganti nama siswa dengan nama baru. Pada fungsi tersebut, selain mengganti nama siswa yang ada di struktur data linked list, juga melakukan update perubahan nama pada struktur data Stack untuk list siswa berdasarkan yang pendaftaran terbaru.

Baris (575 - 587) : Fungsi kelasBaru digunakan untuk mengganti identitas dari siswa mengenai kelasnya, yang dimana konsepnya menambahkan node dari linked list yang lama ke linked list yang baru terlebih dahulu, setelah itu baru dihapus node dari linked list yang lama.

Baris (589 - 609) : Fungsi bayarSPP digunakan untuk mengubah status pembayaran SPP beserta tunggakannya jika pengguna hendak mengubah data mengenai pembayaran SPP nya dari siswa yang dipilih (pada menu 4)

Baris (611 - 708) : Fungsi ubahDataSiswa digunakan untuk mengubah data-data dari siswa. Di dalam fungsi tersebut program dapat memberikan opsi kepada pengguna mengenai data apa yang ingin dirubah, sehingga pengguna bisa hanya mengubah data tertentu saja dan tidak diharuskan mengubah semua data sekaligus.

Baris (710 - 762) : fungsi pembayaranSPP ditujukan untuk menu 5 yang digunakan untuk mengupdate status pembayaran SPP beserta sisa tunggakannya dari siswa tertentu yang sudah terdaftar.

Baris (764 - 773) : fungsi tampilAntrianBK digunakan untuk sub menu dari menu 5 yang digunakan untuk menampilkan antrian bimbingan konseling dari siswa yang sudah melakukan booking. Dan list yang ditampilkan, diurutkan berdasarkan siswa yang lebih dulu melakukan booking pada TU (yang menjalankan program).

Baris (775 - 848) : fungsi bimbinganKonseling digunakan untuk menampilkan menu 5 yang di dalamnya terdapat pilihan untuk pengguna mengenai aktivitas apa yang

ingin dilakukan terhadap kegiatan bimbingan konseling nya, pilihan-pilihan tersebut antara lain menambah antrian, melihat antrian, & menghapus antrian.

Baris (850 - 916) : fungsi menuUtama digunakan untuk menampung menu utama yang ada pada program. Seperti yang sudah dijelaskan di atas bahwa terdapat 6 menu untuk pengelolaan data-data di sekolah, antara lain melihat, menambahkan, menghapus, mengubah data siswa, pengelolaan dan pencatatan status pembayaran SPP dari siswa yang terdaftar, & pengelolaan antrian bimbingan konseling.

Baris (918 - 933) : Fungsi int main() adalah titik awal atau pintu masuk utama dari program. Yang mengembalikan nilai bertipe data int yang menunjukkan status keluaran program. Biasanya, nilai 0 menunjukkan bahwa program selesai berjalan dengan sukses, sedangkan nilai selain 0 menunjukkan ada masalah atau kesalahan dalam eksekusi program.

Di awal int main() terdapat perulangan yang terjadi selama 120 kali (merekpresentasikan jumlah kelas dan jumlah siswa di setiap kelas) yang di dalamnya terdapat syntax yang bertujuan untuk melakukan inisialisasi data-data dari node linked list yang terdapat di array awalKelas. Yang artinya kami mengisi setiap kelas dengan 10 siswa sebelum program tersebut benar-benar dijalankan dari sudut pandang pengguna.

C. Hasil Testing Code

1. Menu Utama

```
===== SELAMAT DATANG DI PROGRAM PENGELOLAAN DATA SISWA SMAN 01 KARAWANG =====  
  
1. Data Siswa  
2. Tambah Data Siswa  
3. Hapus Data Siswa  
4. Ubah Data Siswa  
5. Pembayaran SPP  
6. Bimbingan Konseling  
7. Keluar  
Pilih Menu :
```

Ini adalah tampilan menu utama ketika user menjalankan program pertama kali. Di menu utama akan ada 7 submenu yang dapat dipilih. User dapat menginput angka dari 1-7 untuk mengakses menu yang terdapat pada program sesuai angka yang user inputkan.

2. Submenu Data Siswa

```
Tampilkan Data Siswa  
=====  
  
1. Berdasarkan Kelas  
2. Berdasarkan Pendaftaran Siswa yang Terbaru  
3. Cari ID Siswa  
Pilih :
```

Jika user mengetik angka "1" pada menu utama maka tampilan submenu data siswa akan muncul. Di dalam sub menu ini user dapat melihat dan mencari data siswa yang terdapat pada program. Sub menu ini memiliki 3 pilihan yang diberikan untuk menampilkan atau mencari data siswa. User dapat menginput angka 1-3 sesuai urutan untuk mengakses program pencarian siswa.


```

=====
1) 10 MIPA 1
2) 10 MIPA 2
3) 10 IPS 1
4) 10 IPS 2
5) 11 MIPA 1
6) 11 MIPA 2
7) 11 IPS 1
8) 11 IPS 2
9) 12 MIPA 1
10) 12 MIPA 2
11) 12 IPS 1
12) 12 IPS 2
Pilih Kelas :

```

Jika user mengetik angka “1”, maka akan muncul tampilan berikut. Dalam tampilan menu ini, user dapat memilih kelas siswa yang ingin user cari. Terdapat 12 kelas yang dapat user akses untuk mencari siswa. User dapat menginput angka 1-12 untuk mengakses kelas tersebut dan mencari siswa yang dicari.

```

Data Siswa Kelas 10 MIPA 1
=====
Absen : 1
Nama           : Ahmad Fauzi
Kelas         : 10 MIPA 1
ID Siswa       : 1000000001
Tanggal Lahir  : 03 Januari 2005
Status Pembayaran SPP : Sudah Lunas
Sisa Tunggalan : 0

=====
Absen : 2
Nama           : Ahmad Sutrisno
Kelas         : 10 MIPA 1
ID Siswa       : 1000000002
Tanggal Lahir  : 12 Februari 2005
Status Pembayaran SPP : Sudah Lunas
Sisa Tunggalan : 0

=====
Absen : 3
Nama           : Andi Nugroho
Kelas         : 10 MIPA 1
ID Siswa       : 1000000003
Tanggal Lahir  : 27 Maret 2005
Status Pembayaran SPP : Sudah Lunas
Sisa Tunggalan : 0

=====
Absen : 8
Nama           : Budi Wijaya
Kelas         : 10 MIPA 1
ID Siswa       : 1000000008
Tanggal Lahir  : 29 Agustus 2005
Status Pembayaran SPP : Sudah Lunas
Sisa Tunggalan : 0

=====
Absen : 9
Nama           : Dewi Indah
Kelas         : 10 MIPA 1
ID Siswa       : 1000000009
Tanggal Lahir  : 10 September 2005
Status Pembayaran SPP : Sudah Lunas
Sisa Tunggalan : 0

=====
Absen : 10
Nama           : Kartika Sari
Kelas         : 10 MIPA 1
ID Siswa       : 1000000010
Tanggal Lahir  : 25 Oktober 2005
Status Pembayaran SPP : Sudah Lunas
Sisa Tunggalan : 0

=====
Apakah Anda ingin kembali ke menu utama (Y/N) ?

```

Setelah user mengetik salah satu dari angka 1-12, maka akan muncul tampilan berikut. Ini adalah tampilan daftar siswa yang diurutkan berdasarkan

kelas. Di tampilan ini user dapat melihat berbagai informasi siswa yang telah terinput pada program. Jika sudah menemukan siswa yang dicari, user dapat mengetik “Y” untuk kembali ke menu utama atau “N” untuk pergi ke menu keluar.

```
List Siswa (Berdasarkan Data Terbaru Siswa SMAN 01 Karawang) :
1. Triana Dewi
2. Suhardi Prabowo
3. Rudi Wijaya
4. Rina Astuti
5. Joko Pratama
6. Indah Fitriani
7. Desi Permata
8. Dedi Pratama
9. Bambang Santoso
10. Agus Nugroho
11. Aprilia Putri
12. Altaf Safar
13. Radit Wijaya
14. Rara Kasturi
15. Mail Muslikin
16. Najwa Indah
17. Reisa elsa
18. Widi Pertiwi
19. Yuda Harianto
20. Ragil Santoso
21. Rika Olin
22. Muhamad Rehan
23. Faldo Purnomo
24. Wulan Dari
25. Andika Alfarezy
26. Lola Amalia
27. aulia syaputri
28. Didik Hermawan
29. Pratama Indah
30. Ajis Guswan
31. Sakia Wati
32. Daffa Firmansyah
33. Fadil Alfares
```

Jika user mengetik angka “2” pada submenu data siswa, maka akan muncul tampilan berikut. Ini adalah tampilan daftar siswa yang diurutkan berdasarkan data terbaru yang diinput. User dapat melihat semua siswa yang terdapat pada program diurutkan berdasarkan data terbaru. Jika user menginputkan siswa baru pada program, maka siswa baru tersebut akan berada di urutan pertama. Setelah itu, user dapat mengetik “Y” untuk kembali ke menu utama atau “N” untuk pergi ke menu keluar.

```

Tampilkan Data Siswa
=====

1. Berdasarkan Kelas
2. Berdasarkan Pendaftaran Siswa yang Terbaru
3. Cari ID Siswa

Pilih : 3

Inputkan ID Siswa yang Ingin Anda Lihat : 1000000010

Data Siswa yang Dicari
-----

Nama           : Kartika Sari
Kelas         : 10 MIPA 1
ID Siswa       : 1000000010
Tanggal Lahir  : 25 Oktober 2005
Status Pembayaran SPP : Sudah Lunas
Sisa Tunggakan : 0

=====

Apakah Anda ingin kembali ke menu utama (Y/N) ?

```

Jika user mengetik angka “3” pada submenu data siswa, maka akan muncul tampilan seperti ini. Ini adalah tampilan yang muncul jika user mencari siswa berdasarkan ID siswa. User perlu memasukkan ID siswa yang ingin dicari terlebih dahulu. Program akan otomatis mencari siswa dengan ID yang user inputkan. Program akan menampilkan data siswa yang dicari jika ID siswa berhasil ditemukan. Jika berhasil ditemukan, user dapat mengetik “Y” untuk kembali ke menu utama atau “N” untuk pergi ke menu keluar.

3. Submenu Tambah Data Siswa

```

Inputkan Data Siswa yang Ingin Ditambah
=====

Nama       : Reiza Alithian
ID Siswa   : 1000000121
Tanggal Lahir : 28 September 2004

=====

1) 10 MIPA 1
2) 10 MIPA 2
3) 10 IPS 1
4) 10 IPS 2
5) 11 MIPA 1
6) 11 MIPA 2
7) 11 IPS 1
8) 11 IPS 2
9) 12 MIPA 1
10) 12 MIPA 2
11) 12 IPS 1
12) 12 IPS 2

Masukkan di Kelas (1-12) : |

```

```

Masukkan di Kelas (1-12) : 1

=====

Siswa telah ditambahkan.

Apakah Anda ingin kembali ke menu utama (Y/N) ?

```

Jika user mengetik angka “2” pada menu utama, maka akan muncul tampilan submenu tambah data siswa. Di dalam submenu ini user dapat

menambahkan data siswa. User perlu menginputkan nama, ID siswa, tanggal lahir, dan kelas sebelum menambahkan data siswa baru tersebut. Setelah berhasil menginput, maka data siswa tersebut otomatis masuk ke dalam kelas tersebut. Setelah selesai menginput, user dapat mengetik “Y” untuk kembali ke menu utama atau “N” untuk pergi ke menu keluar.

4. Submenu Hapus Data Siswa

```
Hapus Siswa Berdasarkan
1. ID Siswa
2. Siswa Terbaru
Pilih :
```

Jika user mengetik angka “3” pada menu utama, maka akan muncul tampilan submenu hapus data siswa. Dalam submenu ini user dapat menghapus data siswa berdasarkan ID siswa atau data siswa terbaru yang diinput. Jika ingin menghapus data siswa berdasarkan ID siswa maka ketik “1” dan jika ingin menghapus data siswa terbaru ketik “2”.

```
Masukkan ID Siswa yang Ingin Dihapus : 1000000121

Data Siswa yang Dipilih
-----
Nama           : Reiza Alithian
Kelas         : 10 MIPA 1
ID Siswa       : 1000000121
Tanggal Lahir  : 28 September 2004
Status Pembayaran SPP : Belum Lunas
Sisa Tunggakan : 0

=====

Apakah Anda ingin menghapus data siswa tersebut? (Y/N) : y
Data siswa telah dihapus.

Apakah Anda ingin kembali ke menu utama (Y/N) ?
```

Jika user memilih untuk menghapus data siswa berdasarkan ID siswa, user perlu menginputkan ID siswa yang datanya ingin dihapus. Setelah menginputkan, program akan memunculkan data siswa yang ID siswanya diinput sebelumnya. Setelah itu akan ada konfirmasi penghapusan, Jika iya maka ketik “Y” dan jika tidak maka ketik “N”.

```
93. Rina Cahyani
94. Rina Astuti
95. Rina Amelia
96. Maman Sutrisno
97. Maman Santoso
98. Maya Siti
99. Maya Sari
100. Maya Dewi
101. Joko Sutrisno
102. Indah Sari
103. Indah Fitriani
104. Heriawan Putra
105. Harianto Prabowo
106. Evi Susanti
107. Dodi Santoso
108. Dian Purnama
109. Dian Fitri
110. Dedi Pratama
111. Kartika Sari
112. Dewi Indah
113. Budi Wijaya
114. Budi Santoso
115. Arif Sutrisno
116. Arianto Wijaya
117. Ani Puspitasari
118. Andi Nugroho
119. Ahmad Sutrisno
120. Ahmad Fauzi

Apakah Anda yakin ingin menghapus data siswa terbaru? (Y/N) : y

Data siswa telah dihapus.

Apakah Anda ingin kembali ke menu utama (Y/N) ?
```

```
114. Arif Sutrisno
115. Arianto Wijaya
116. Ani Puspitasari
117. Andi Nugroho
118. Ahmad Sutrisno
119. Ahmad Fauzi

Apakah Anda ingin kembali ke menu utama (Y/N) ?
```

Jika user memilih untuk menghapus data siswa berdasarkan data terbaru, maka akan muncul tampilan berikut. User perlu mengkonfirmasi penghapusan, jika iya maka ketik “Y” dan jika tidak maka ketik “N”.

5. Submenu Ubah Data Siswa

```
=====
1) 10 MIPA 1
2) 10 MIPA 2
3) 10 IPS 1
4) 10 IPS 2
5) 11 MIPA 1
6) 11 MIPA 2
7) 11 IPS 1
8) 11 IPS 2
9) 12 MIPA 1
10) 12 MIPA 2
11) 12 IPS 1
12) 12 IPS 2
Pilih Kelas :
```

Jika user mengetik angka “4” pada menu utama, maka akan muncul submenu ubah data siswa. Di dalam tampilan ini akan muncul daftar kelas yang dapat dipilih. User dapat menginputkan angka 1-12 untuk mengakses kelas yang data siswanya ingin diubah.

```
=====
Absen : 8
Nama           : Budi Wijaya
Kelas          : 10 MIPA 1
ID Siswa       : 1000000008
Tanggal Lahir  : 29 Agustus 2005
Status Pembayaran SPP : Sudah Lunas
Sisa Tunggakan : 0
=====
Absen : 9
Nama           : Dewi Indah
Kelas          : 10 MIPA 1
ID Siswa       : 1000000009
Tanggal Lahir  : 10 September 2005
Status Pembayaran SPP : Sudah Lunas
Sisa Tunggakan : 0
=====
Absen : 10
Nama           : Kartika Sari
Kelas          : 10 MIPA 1
ID Siswa       : 1000000010
Tanggal Lahir  : 25 Oktober 2005
Status Pembayaran SPP : Sudah Lunas
Sisa Tunggakan : 0
=====
Masukkan ID Siswa yang Ingin Diubah :
```

Setelah user menginput kelas yang ingin diakses, maka akan muncul tampilan data siswa yang terdapat pada kelas yang dipilih sebelumnya. Untuk mencari data siswa yang ingin diubah, user perlu menginputkan ID siswa yang ingin diubah datanya.

```
Data Siswa yang Dipilih
-----

Nama           : Kartika Sari
Kelas          : 10 MIPA 1
ID Siswa        : 1000000010
Tanggal Lahir   : 25 Oktober 2005
Status Pembayaran SPP : Sudah Lunas
Sisa Tunggalan  : 0

=====

Pilih Data yang Ingin Diubah
-----

1. Nama
2. Kelas
3. Tanggal Lahir
4. Status Pembayaran
5. Semua Data

Pilih Menu :
```

Setelah user menginputkan ID siswa yang ingin diubah, maka akan muncul tampilan berikut. Program akan secara otomatis mencari ID siswa yang diinputkan sebelumnya dan akan memunculkannya. User dapat memilih data apa saja yang ingin diubah dengan mengetik angka 1-5 sesuai urutannya.

```
=====

Masukkan Nama Baru      : Rei
=====

1) 10 MIPA 1
2) 10 MIPA 2
3) 10 IPS 1
4) 10 IPS 2
5) 11 MIPA 1
6) 11 MIPA 2
7) 11 IPS 1
8) 11 IPS 2
9) 12 MIPA 1
10) 12 MIPA 2
11) 12 IPS 1
12) 12 IPS 2

Masukkan Kelas Baru      : 1
Masukkan Tanggal Lahir Baru : 3 Februari 2004
Pembayaran sudah lunas?
1. Iya
2. Tidak
Pilih : 1

=====

Semua data siswa berhasil diubah.
Apakah Anda ingin kembali ke menu utama (Y/N) ?
```

Setelah user memilih data apa saja yang diubah, maka akan muncul tampilan seperti berikut. User akan diminta memasukkan nama baru, kelas baru, tanggal lahir baru, atau lunasnya pembayaran. Jika pembayaran belum lunas, maka user perlu memasukkan jumlah tunggaknya. Setelah selesai menginput, user dapat mengetik “Y” untuk kembali ke menu utama atau “N” untuk pergi ke menu keluar.

6. Submenu Pembayaran SPP

```
=====
1) 10 MIPA 1
2) 10 MIPA 2
3) 10 IPS 1
4) 10 IPS 2
5) 11 MIPA 1
6) 11 MIPA 2
7) 11 IPS 1
8) 11 IPS 2
9) 12 MIPA 1
10) 12 MIPA 2
11) 12 IPS 1
12) 12 IPS 2
Pilih Kelas :
```

Jika user mengetik angka “5” pada menu utama, maka user akan mengakses submenu membayar SPP. Di dalam submenunya akan muncul tampilan daftar kelas. User dapat mengakses ke salah satu kelas tersebut untuk mencari siswa yang ingin diubah data SPP-nya. User dapat mengetik salah satu angka dari 1-12 untuk mengakses salah satu kelasnya.


```

Absen : 9
Nama      : Dewi Indah
Kelas    : 10 MIPA 1
ID Siswa  : 100000009
Tanggal Lahir : 10 September 2005
Status Pembayaran SPP : Sudah Lunas
Sisa Tunggalan : 0

```

```

Absen : 10
Nama      : Kartika Sari
Kelas    : 10 MIPA 1
ID Siswa  : 100000010
Tanggal Lahir : 25 Oktober 2005
Status Pembayaran SPP : Sudah Lunas
Sisa Tunggalan : 0

```

```

Absen : 11
Nama      : Rei
Kelas    : 10 MIPA 1
ID Siswa  : 12345
Tanggal Lahir : 3 Februari 2004
Status Pembayaran SPP : Belum Lunas
Sisa Tunggalan : 100000

```

```

Masukkan ID Siswa :

```

Setelah user memilih salah satu kelas pada tampilan menu sebelumnya, maka akan masuk ke tampilan data siswa yang terdapat pada kelas tersebut. Di dalamnya akan terdapat data siswa juga status pembayaran SPP-nya. User perlu memasukkan ID siswa yang ingin diubah status pembayaran SPP-nya.

```

Absen : 11
Nama      : Rei
Kelas    : 10 MIPA 1
ID Siswa  : 12345
Tanggal Lahir : 3 Februari 2004
Status Pembayaran SPP : Belum Lunas
Sisa Tunggalan : 100000
=====
Masukkan ID Siswa : 12345
Masukkan Jumlah Pembayaran SPP : 500000
Pembayaran SPP sebesar 500000 berhasil dilakukan untuk siswa dengan ID Siswa 12345.
Apakah Anda ingin kembali ke menu utama (Y/N) ?

```

Setelah menginput ID siswa yang ingin diubah data status pembayaran SPP-nya, maka akan muncul tampilan seperti di atas. Program akan secara otomatis mencari ID siswa yang sebelumnya di input dan menampilkan data siswa tersebut. Setelah itu, user menginput jumlah pembayaran SPP yang ingin ditambahkan. Program akan secara otomatis mengubah status pembayaran jika pembayaran yang dilakukan sudah lunas atau mengurangi jumlah tunggakan jika pembayaran yang dilakukan belum lunas. Setelah selesai menginput, user dapat mengetik “Y” untuk kembali ke menu utama atau “N” untuk pergi ke menu keluar.

7. Submenu Bimbingan Konseling

```
Bimbingan Konseling
=====

1. Tambah Antrian Bimbingan Konseling
2. Lihat Antrian Bimbingan Konseling
3. Hapus Antrian

Pilih Menu : _
```

Jika user mengetik angka “6” pada menu utama, maka user akan mengakses submenu bimbingan konseling. Di dalam tampilan submenu ini terdapat 3 pilihan yang diberikan, yaitu tambah antrian, lihat antrian, dan hapus antrian. User dapat memilih 3 pilihan tersebut dengan mengetik salah satu angka 1-3 sesuai urutan.

```
Bimbingan Konseling
=====

1. Tambah Antrian Bimbingan Konseling
2. Lihat Antrian Bimbingan Konseling
3. Hapus Antrian

Pilih Menu : 1
Masukkan ID Siswa yang ingin melakukan Bimbingan Konseling : 12345
Data Siswa yang Dipilih
Nama          : Rei
Kelas        : 10 MIPA 1
ID Siswa      : 12345
Tanggal Lahir : 3 Februari 2004
Status Pembayaran SPP : Belum Lunas
Sisa Tunggakan : 500000

=====

Apakah ID siswa yang Anda inputkan sudah benar? (Y/N) :
```

Jika user mengetik angka “1”, maka akan muncul tampilan berikut. User akan diminta untuk memasukan ID siswa yang ingin melakukan bimbingan konseling. Program akan otomatis mencari data siswa yang sesuai dengan ID siswa dan menampilkannya. User akan diminta mengonfirmasi siswa yang ingin melakukan bimbingan konseling, jika iya ketik “Y” dan jika tidak ketik “N”.

```

Bimbingan Konseling
=====

1. Tambah Antrian Bimbingan Konseling
2. Lihat Antrian Bimbingan Konseling
3. Hapus Antrian

Pilih Menu : 2

Antrian Bimbingan Konseling
1. Rei

Apakah Anda ingin kembali ke menu utama (Y/N) ? _

```

Jika user mengetik angka “2”, maka akan menampilkan tampilan berikut. Program akan otomatis menampilkan antrian bimbingan konseling. Setelah selesai menampilkan, user dapat mengetik “Y” untuk kembali ke menu utama atau “N” untuk pergi ke menu keluar.

```

Bimbingan Konseling
=====

1. Tambah Antrian Bimbingan Konseling
2. Lihat Antrian Bimbingan Konseling
3. Hapus Antrian

Pilih Menu : 3

Antrian Bimbingan Konseling
1. Rei

Apakah Anda yakin ingin memanggil siswa selanjutnya? (Y/N) : y
Antrian telah diperbarui.
Apakah Anda ingin kembali ke menu utama (Y/N) ?

```

Jika user mengetik angka “3”, maka akan muncul tampilan berikut. Program akan menampilkan daftar antrian bimbingan konseling. User akan diminta untuk mengonfirmasi pemanggilan siswa, jika iya ketik “Y” dan jika tidak ketik “N”. Setelah selesai menginput, user dapat mengetik “Y” untuk kembali ke menu utama atau “N” untuk pergi ke menu keluar.

8. Submenu Keluar

```

Terima kasih telah menggunakan program ini. Sampai jumpa!

Press any key to continue . . .

Process returned 0 (0x0)   execution time : 1213.400 s
Press any key to continue.

```

Jika user mengetik angka ”7”, maka akan muncul tampilan menu keluar. Di dalam submenu ini, menandakan bahwa user telah selesai menggunakan program yang dijalankan. Ketik apa saja untuk keluar dari program.

IV. PENUTUP

A. Kesimpulan

Pengolahan data yang cukup efisien dan akurat yaitu dengan melalui metode digital. Metode digital menjadi sangat penting dalam bidang pendidikan. Penggunaan program pengolahan data siswa membawa sejumlah manfaat, antara lain efisiensi dan produktivitas, akurasi dan ketepatan data, pemantauan perkembangan siswa yang lebih baik, pengambilan keputusan yang lebih baik, peningkatan kualitas pelayanan dan komunikasi, keamanan dan privasi data, integrasi dan kolaborasi, manajemen SPP, persyaratan regulasi dan pelaporan, serta persiapan untuk masa depan digital.

Program pengolahan data siswa membantu meminimalisir kesalahan input data yang dilakukan secara manual dan meningkatkan efisiensi waktu dan tenaga dalam pengelolaan data siswa. Selain itu, penggunaan metode digital mengurangi risiko kehilangan data dan penggunaan kertas berlebihan, sehingga membantu menghemat biaya dan berkontribusi pada pelestarian lingkungan.

Dengan menggunakan program pengolahan data siswa, sekolah dapat memperoleh informasi yang akurat dan terkini untuk pengambilan keputusan yang lebih baik, serta memantau dan mengevaluasi perkembangan siswa secara keseluruhan. Program ini juga memungkinkan sekolah meningkatkan kualitas pelayanan dan komunikasi dengan siswa dan orang tua, serta memenuhi persyaratan regulasi dan melaporkan data siswa secara cepat dan akurat.

B. Saran

Berikut beberapa saran yang dapat diterapkan dalam pengembangan program pengolahan data siswa sebagai berikut:

- Menjaga keamanan dan privasi data

Memastikan sistem yang digunakan memiliki fitur keamanan yang memadai, seperti enkripsi data dan pengaturan izin akses. Selain itu, tetap patuhi aturan

privasi data siswa dan pastikan bahwa data tersebut hanya dapat diakses oleh pihak yang berwenang.

- Melakukan pelatihan dan pemahaman terhadap staf sekolah

Sebelum mengimplementasikan sistem baru, sebaiknya memberikan pelatihan kepada staf sekolah agar mereka dapat memahami dan menggunakan sistem tersebut dengan baik dan memastikan mereka mengerti cara menggunakan fitur-fitur yang disediakan dan memaksimalkan manfaat dari pengolahan data siswa yang efisien.

- Melakukan pemantauan dan evaluasi secara berkala

Melakukan pemantauan dan evaluasi secara berkala untuk memastikan sistem berjalan dengan baik dan memenuhi kebutuhan sekolah. Mengidentifikasi area yang perlu ditingkatkan atau perbaikan yang diperlukan untuk meningkatkan efisiensi dan efektivitas pengolahan data.