

**PENERAPAN KONSEP ALGORITMA DAN
STRUKTUR DATA UNTUK SISTEM PENDAFTARAN DAN PENERIMAAN
SISWA SMP WACANA**

Diajukan guna memenuhi Ujian Akhir Semester mata kuliah

Algoritma dan Struktur Data 2

Dosen Pengampu : Indah Nofikasari, M.Kom



Disusun oleh :

Bachrul Rizal Fauzi (210103009)

Muhammad Lintang Pamungkas (210103028)

**FAKULTAS ILMU KOMPUTER
UNIVERSITAS DUTA BANGSA SURAKARTA
2022**

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa karena berkah dan rahmat-Nya kami dapat menyelesaikan tugas laporan “Sistem Pendaftaran Dan Penerimaan Siswa SMP Wacana” mata kuliah Algoritma dan Struktur data 2. Laporan ini dilakukan untuk memenuhi nilai mata kuliah Algoritma dan Struktur Data 2.

Dengan diselesaikannya tugas laporan ini, kami harapkan dapat memenuhi syarat penilaian tugas Algoritma dan Struktur Data 2 dan berguna untuk para pembacanya. Kritik dan saran yang membangun sangat kami harapkan sebagai bahan penyempurnaan pada laporan ini. Semoga bermanfaat.

Surakarta, Juli 2022

Penulis

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI.....	ii
BAB I PENDAHULUAN	1
A. LATAR BELAKANG.....	1
B. RUMUSAN MASALAH	2
C. BATASAN ALGORITMA DAN KONSEP	2
D. Tujuan.....	3
BAB II PEMBAHASAN	4
A. Permasalahn Studi Kasus	4
B. Konsep Algoritma dan Struktur Data.....	5
C. Flowchart.....	5
D. Implementasi Bahasa Pemrograman	6
BAB III PENUTUP	23
A. KESIMPULAN	23
B. SARAN	23
DAFTAR PUSTAKA	24

BAB I PENDAHULUAN

A. LATAR BELAKANG

Pendaftaran siswa baru merupakan siklus tahunan yang dialami oleh semua instansi pendidikan baik itu instansi negeri maupun swasta. Proses penerimaan siswa baru merupakan salah satu kegiatan yang menunjang kegiatan belajar mengajar nantinya. Dalam proses inilah interaksi antara calon siswa baru dengan instansi pendidikan terjalin. (Iswanto, 2006)

Biasanya setelah tiba waktu pendaftaran, instansi pendidikan langsung didatangi banyak pendaftar ini menimbulkan layanan akan proses pendataan sedikit terhambat dengan banyaknya pendaftar, serta jumlah petugas pendaftaran yang sangat sedikit akan kesulitan dalam melakukan pendataan calon siswa baru. (Arif Ranu Wicaksono, 2007)

Kualitas pelayanan adalah hal yang penting untuk diperhatikan, karena hal tersebut akan dipersepsikan oleh konsumen setelah konsumen menggunakan barang atau jasa. Persaingan yang semakin ketat akhir-akhir ini menuntut sebuah lembaga penyedia jasa atau layanan untuk selalu memanjakan konsumen dengan memberikan pelayanan terbaik. Para pelanggan akan mencari produk berupa barang atau jasa dari perusahaan yang dapat memberikan pelayanan yang terbaik kepadanya (Assauri, 2008).

SMP Wacana (Swasta) Surakarta merupakan yayasan pendidikan yang berdiri sejak tahun 1996. Meskipun tergolong baru, SMP Wacana Surakarta sudah menjadi SMP favorit. Setiap waktu pendaftaran tiba, loket pendaftaran selalu dipenuhi calon siswa baru setiap harinya. Sehingga petugas pendaftaran selalu kewalahan melayani pendaftaran tersebut.

Petugas yang kewalahan melayani pendaftaran membuat pendataan manual jadi tidak nyaman, dan ini menyebabkan pendataan jadi lambat dan tidak efisien. Sehingga pendaftar mengantri terlalu lama dan menyebabkan pelayanan tidak optimal.

Berdasarkan permasalahan diatas, maka penulis akan menganalisis dan merancang pengolahan data siswa yang berkaitan dengan data Pendaftaran dan Penerimaan siswa, yang mana akan menghasilkan laporan Inputan Data diri Siswa serta keterangan diterima tidak menjadi siswa. Semuanya dituangkan dalam bentuk Penerapan Konsep Algoritma Dan Struktur Data dengan mengimplementasikan dengan menggunakan bahasa C atau family C.

B. RUMUSAN MASALAH

- a. Bagaimana menganalisis dan merancang algoritma sistem penerimaan siswa SMP Wacana.
- b. Bagaimana Implementasi Sistem Pendaftaran Siswa SMP Wacana dengan menerapkan konsep struktur data menggunakan bahasa C / Lainnya.

C. BATASAN ALGORITMA DAN KONSEP

Dalam Perancangan Sistem Pendaftaran Siswa SMP Wacana penulis membatasi pembuatan aplikasi, agar sistem dapat mengatasi permasalahan diatas. Untuk itu penulis membatasi konsep algoritma adalah sebagai berikut :

- a. Perancangan Algoritma khusus untuk system Pendaftaran Siswa SMP.
- b. Sistem ini mengolah data Siswa, data nilai, dan data keterangan siswa.
- c. Konsep struktur data yang digunakan adalah (konsep algoritma runtutan percabangan perulangan, konsep LinkedList, konsep searching, dan konsep sorting).
- d. Implementasi Sistem menggunakan bahasa pemrograman C++ atau C family.

D. Tujuan

- a. Dapat menganalisis dan merancang algoritma sistem pendaftaran siswa SMP Wacana.
- b. Dapat mengimplementasi konsep struktur untuk sistem pendaftaran siswa SMP Wacana dalam bahasa pemrograman C.

BAB II

PEMBAHASAN

A. Permasalahn Studi Kasus

Disebuah Pendidikan SMP Wacana sedang mengadakan pendaftaran siswa baru. Untuk mempermudah dan mempersingkat waktu guru dalam mengolah data siswa baru, sekolah tersebut memerlukan program untuk mengelola pendaftaran siswa dan mampu menghasilkan keterangan pendaftaran maupun melihat data siswa sekaligus. Yang mempunyai keterangan sebagai berikut :

Ada 6 Fitur, yaitu :

- Input Data Siswa baru : Untuk memasukkan data siswa pendaftar dengan mengumpulkan data nisl, nama, tempat tanggal lahir, jenis kelamin, alamat nilai UN dan asal SD.
- Ubah Data Siswa baru : Untuk mengubah Data siswa yang sudah mendaft.
- Delete Data Siswa Baru : Untuk menghapus data siswa yang sudah mendaft.
- Tampilkan data pendaftar: Untuk menampilkan data siswa yang sudah mendaft dan akan muncul informasi keterangan penerimaan siswa.
- Sorting data : untuk mengurutkan data siswa yang sudah mendaft bisa berdasarkan nama, dan nilai akhir UN.
- Searching data : Untuk mencari data siswa yang sudah mendaft, bisa berdasarkan nama siswa, nisl, dan nilai akhir.

Ada ketentuan sebagai berikut :

Mapel UN	Nilai
Bahasa Indonesia	66,6
Matematika	66,6
IPA	66,6
<hr/>	
Jumlah	200

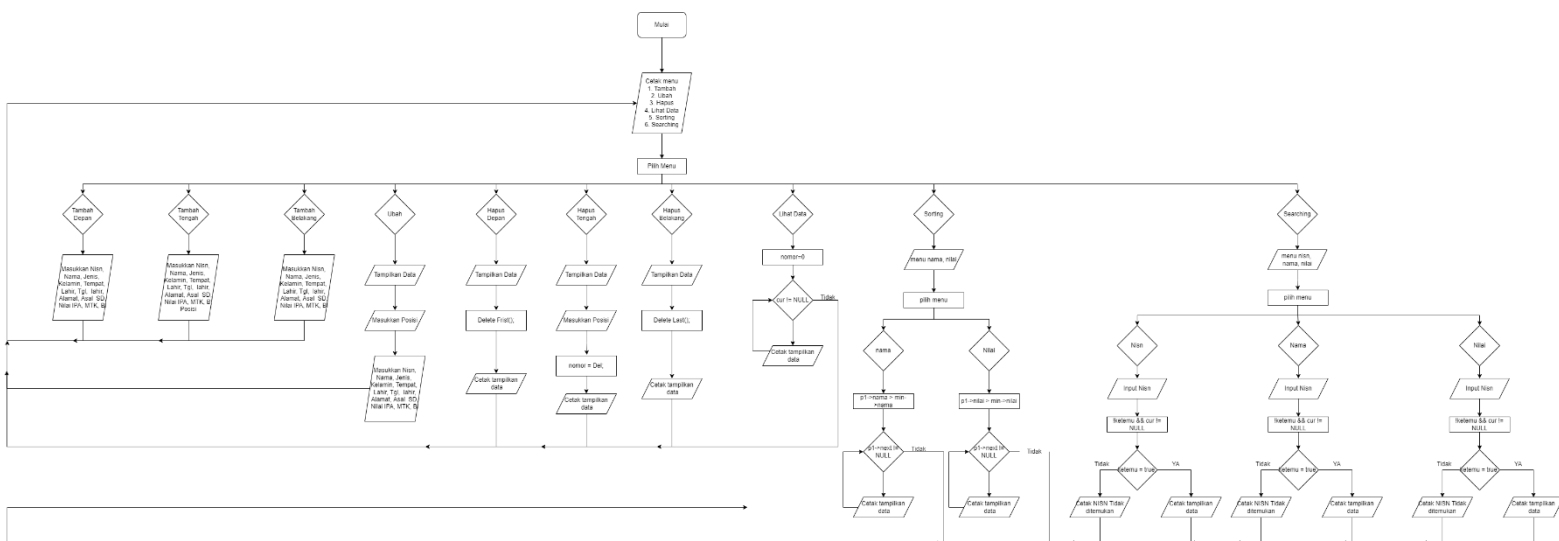
Ketentuan Penerimaan :

- Jika Jumlah nilai Mapel lebih dari 200 → Mendapat Keterangan Diterima.
- Jika Jumlah nilai Mapel kurang dari 200 → Mendapat Keterangan Tidak Diterima.

B. Konsep Algoritma dan Struktur Data

- Runtunan digunakan untuk mengisi data siswa yang mendaftar dan juga mengubah data siswa yang mendaftar.
- Percabangan digunakan dalam menu utama untuk memilih action (daftar, hapus, ubah, lihat data, sorting, dan searching), untuk memilih menu sorting berdasarkan (nama, dan nilai akhir) dan terakhir digunakan untuk memilih menu searching berdasarkan (nisd, nama, nilai akhir).
- Konsep perulangan digunakan untuk Kembali ke menu utama setelah menginputkan data/ubah data/hapus data/lihat data/sorting dan searching.
- Konsep linked list digunakan untuk menyimpan data pada node Siswa yang berisi nisd, nama siswa, jenis kelamin, tempat tanggal lahir, alamat, asal sd, dan nilai UN.
- Konsep Serching digunakan untuk mencari data siswa berdasarkan nama, nisd dan nilai akhir siswa.
- Konsep Shorting digunakan untuk mengurutkan data siswa berdasarkan nama, dan nilai akhir siswa tertinggi.

C. Flowchart



D. Implementasi Bahasa Pemrograman

- Nama Varibel dan Tipe Variabel

```
Struct Siswa{
    string nisl;
    string nama, jk, tempat, tgllahir, alamat, asal_sd, keterangan;
    float nilai1, nilai2, nilai3, hasil;
    Siswa *next;}

Int posisi, pilih, pil, pilih_sorting;
Int temp, nomor=0;
string strtemp;
float strtemp2;
```

- Listingan Program Dan Hasil Compiler Program

```
#include <iostream>
#include <iomanip>
#include <conio.h>
using namespace std;

// deklarasi single linked list
struct Siswa{
    string nisl;
    string nama, jk, tempat, tgllahir, alamat, asal_sd, keterangan;
    float nilai1, nilai2, nilai3, hasil;
    Siswa *next;
};
Siswa *head, *tail, *cur, *newNode, *del, *before;

void printSingleLinkedList();

// create single linked list
void createSingleLinkedList(string nisl, string nama, string jk, string tempat, string
tgllahir, string alamat, string asal_sd, string keterangan,
                                float nilai1, float nilai2, float
nilai3, float hasil)
{
    head = new Siswa();
    head->nisl = nisl;
    head->nama = nama;
    head->jk = jk;
    head->tempat = tempat;
    head->tgllahir = tgllahir;
    head->alamat = alamat;
    head->asal_sd = asal_sd;
    head->nilai1 = nilai1;
    head->nilai2 = nilai2;
    head->nilai3 = nilai3;
    head->hasil = nilai1 + nilai2 + nilai3;
    if(head->hasil >= 200){
        keterangan = "Diterima";
    }

    else if(head->hasil <= 199){
        keterangan = "Tidak Diterima";
    }
    head->keterangan = keterangan;

    head->next = NULL;
    tail = head;
}
```

```

// print single linked list
int countSingleLinkedList(){
    cur = head;
    int jumlah = 0;
    while( cur != NULL ){
        jumlah++;
        cur = cur->next;
    }
    return jumlah;
}

// tambahAwal
void addFirst(string nispn, string nama, string jk, string tempat, string tgllahir, string
alamat, string asal_sd, string keterangan,
float nilai1, float nilai2, float
nilai3, float hasil){
    if (head == NULL) //jika head kosong maka kita buat singlelist dulu
    {
        head = new Siswa();
        head->nispn = nispn;
        head->nama = nama;
        head->jk = jk;
        head->tempat = tempat;
        head->tgllahir = tgllahir;
        head->alamat = alamat;
        head->asal_sd = asal_sd;
        head->nilai1 = nilai1;
        head->nilai2 = nilai2;
        head->nilai3 = nilai3;
        head->hasil = nilai1 + nilai2 + nilai3;
        if(head->hasil >= 200){
            keterangan = "Diterima";
        }

        else if(head->hasil <= 199){
            keterangan = "Tidak Diterima";
        }
        head->keterangan = keterangan;

        head->next = NULL;
        tail = head;
    }

    else
    {
        newNode = new Siswa();
        newNode->nispn = nispn;
        newNode->nama = nama;
        newNode->jk = jk;
        newNode->tempat = tempat;
        newNode->tgllahir = tgllahir;
        newNode->alamat = alamat;
        newNode->asal_sd = asal_sd;
        newNode->nilai1 = nilai1;
        newNode->nilai2 = nilai2;
        newNode->nilai3 = nilai3;
        newNode->hasil = nilai1 + nilai2 + nilai3;
        if(newNode->hasil >= 200){
            keterangan = "Diterima";
        }

        else if(newNode->hasil <= 199){
            keterangan = "Tidak Diterima";
        }
    }
}

```

```

    }
    newNode->keterangan = keterangan;

    newNode->next = head;
    head = newNode;
}

// tambahAkhir
void addLast(string nispn, string nama, string jk, string tempat, string tgllahir, string
alamat, string asal_sd, string keterangan,
float nilai1, float nilai2, float
nilai3, float hasil){
    if (head == NULL) //jika head kosong maka kita buat singlelist dulu
    {
        head = new Siswa();
        head->nispn = nispn;
        head->nama = nama;
        head->jk = jk;
        head->tempat = tempat;
        head->tgllahir = tgllahir;
        head->alamat = alamat;
        head->asal_sd = asal_sd;
        head->nilai1 = nilai1;
        head->nilai2 = nilai2;
        head->nilai3 = nilai3;
        head->hasil = nilai1 + nilai2 + nilai3;
        if(head->hasil >= 200){
            keterangan = "Diterima";
        }

        else if(head->hasil <= 199){
            keterangan = "Tidak Diterima";
        }
        head->keterangan = keterangan;

        head->next = NULL;
        tail = head;
    }

    else
    {
        newNode = new Siswa();
        newNode->nispn = nispn;
        newNode->nama = nama;
        newNode->jk = jk;
        newNode->tempat = tempat;
        newNode->tgllahir = tgllahir;
        newNode->alamat = alamat;
        newNode->asal_sd = asal_sd;
        newNode->nilai1 = nilai1;
        newNode->nilai2 = nilai2;
        newNode->nilai3 = nilai3;
        newNode->hasil = nilai1 + nilai2 + nilai3;
        if(newNode->hasil >= 200){
            keterangan = "Diterima";
        }

        else if(newNode->hasil <= 199){
            keterangan = "Tidak Diterima";
        }
        newNode->keterangan = keterangan;

        newNode->next = NULL;
        tail->next = newNode;
        tail = newNode;
    }
}

```

```

    }

}

// tambah tengah
void addMiddle(string nispn, string nama, string jk, string tempat, string tgllahir, string
alamat, string asal_sd, string keterangan,
float nilai1, float nilai2, float
nilai3, float hasil, int posisi){

    if(head == NULL) //jika head kosong maka kita buat singlelist dulu
    {
        head = new Siswa();
        head->nispn = nispn;
        head->nama = nama;
        head->jk = jk;
        head->tempat = tempat;
        head->tgllahir = tgllahir;
        head->alamat = alamat;
        head->asal_sd = asal_sd;
        head->nilai1 = nilai1;
        head->nilai2 = nilai2;
        head->nilai3 = nilai3;
        head->hasil = nilai1 + nilai2 + nilai3;
        if(head->hasil >= 200){
            keterangan = "Diterima";
        }

        else if(head->hasil <= 199){
            keterangan = "Tidak Diterima";
        }
        head->keterangan = keterangan;

        head->next = NULL;
        tail = head;
    }

    else{
        if( posisi < 1 || posisi > countSingleLinkedList() ) {
            cout << "Posisi diluar jangkauan" << endl;
        }
        else if( posisi == 1 ) {
            cout << "Posisi bukan posisi tengah" << endl;
        }
        else{
            newNode = new Siswa();
            newNode->nispn = nispn;
            newNode->nama = nama;
            newNode->jk = jk;
            newNode->tempat = tempat;
            newNode->tgllahir = tgllahir;
            newNode->alamat = alamat;
            newNode->asal_sd = asal_sd;
            newNode->nilai1 = nilai1;
            newNode->nilai2 = nilai2;
            newNode->nilai3 = nilai3;
            newNode->hasil = nilai1 + nilai2 + nilai3;
            if(newNode->hasil >= 200){
                keterangan = "Diterima";
            }

            else if(newNode->hasil <= 199){
                keterangan = "Tidak Diterima";
            }
            newNode->keterangan = keterangan;

```

```

        // tranversing
        cur = head;
        int nomor = 1;
        while( nomor < posisi - 1 ){
            cur = cur->next;
            nomor++;
        }
        newNode->next = cur->next;
        cur->next = newNode;
    }
}

void change(string nispn, string nama, string jk, string tempat, string tgllahir, string
alamat, string asal_sd, string keterangan,
float nilai1, float nilai2, float
nilai3, float hasil, int posisi){

    if(head == NULL) //jika head kosong maka kita buat singlelist dulu
    {
        head = new Siswa();
        head->nispn = nispn;
        head->nama = nama;
        head->jk = jk;
        head->tempat = tempat;
        head->tgllahir = tgllahir;
        head->alamat = alamat;
        head->asal_sd = asal_sd;
        head->nilai1 = nilai1;
        head->nilai2 = nilai2;
        head->nilai3 = nilai3;
        head->hasil = nilai1 + nilai2 + nilai3;
        if(head->hasil >= 200){
            keterangan = "Diterima";
        }

        else if(head->hasil <= 199){
            keterangan = "Tidak Diterima";
        }
        head->keterangan = keterangan;

        head->next = NULL;
        tail = head;
    }
    else{
        if( posisi < 1 || posisi > countSingleLinkedList() ){
            cout << "Posisi diluar jangkauan" << endl;
        }
        else if( posisi == 1 || posisi == countSingleLinkedList() ){
            cout << "Posisi bukan posisi tengah" << endl;
        }
        else{
            cur = head;
            int nomor = 1;

            while( nomor < posisi ){
                cur = cur->next;
                nomor++;
            }

            cur->nispn = nispn;
            cur->nama = nama;
            cur->jk = jk;

```

```

        cur->tempat = tempat;
        cur->tgllahir = tgllahir;
        cur->alamat = alamat;
        cur->asal_sd = asal_sd;
        cur->nilai1 = nilai1;
        cur->nilai2 = nilai2;
        cur->nilai3 = nilai3;
        cur->hasil = nilai1 + nilai2 + nilai3;
        if(cur->hasil >= 200){
            keterangan = "Diterima";
        }

        else if(cur->hasil <= 199){
            keterangan = "Tidak Diterima";
        }
        cur->keterangan = keterangan;
    }
}

// Remove First
void removeFirst(){
    del = head;
    head = head->next;
    delete del;
}

// Remove Last
void removeLast(){
    del = tail;
    cur = head;
    while( cur->next != tail ){
        cur = cur->next;
    }
    tail = cur;
    tail->next = NULL;
    delete del;
}

// remove middle
void removeMiddle(int posisi){
    if( posisi < 1 || posisi > countSingleLinkedList() ){
        cout << "Posisi diluar jangkauan" << endl;
    }else if( posisi == 1){
        cout << "Posisi bukan posisi tengah" << endl;
    }else{
        int nomor = 1;
        cur = head;
        while( nomor <= posisi ){
            if( nomor == posisi-1 ){
                before = cur;
            }
            if( nomor == posisi ){
                del = cur;
            }
            cur = cur->next;
            nomor++;
        }
        before->next = cur;
        delete del;
    }
}

```

```

void sorting_nilai()
{
    int temp;
    string strtemp;
    float strtemp2;
    Siswa *p1, *p2, *min;
    p1 = new Siswa();
    p2 = new Siswa();
    min = new Siswa();

    if (head != NULL)
    {
        p1 = head;
        while (p1->next != NULL)
        {
            p2 = p1;
            min = p1;
            while (p2 != NULL)
            {
                if (p2->hasil > min->hasil)
                {
                    min = p2;
                }
                p2 = p2->next;
            }
            if (p1->hasil < min->hasil)
            {
                temp = p1->hasil;
                p1->hasil = min->hasil;
                min->hasil = temp;

                strtemp = p1->nisn;
                p1->nisn = min->nisn;
                min->nisn = strtemp;

                strtemp = p1->nama;
                p1->nama = min->nama;
                min->nama = strtemp;

                strtemp = p1->jk;
                p1->jk = min->jk;
                min->jk = strtemp;

                strtemp = p1->tempat;
                p1->tempat = min->tempat;
                min->tempat = strtemp;

                strtemp = p1->tgllahir;
                p1->tgllahir = min->tgllahir;
                min->tgllahir = strtemp;

                strtemp = p1->alamat;
                p1->alamat = min->alamat;
                min->alamat = strtemp;

                strtemp = p1->asal_sd;
                p1->asal_sd = min->asal_sd;
                min->asal_sd = strtemp;

                strtemp2 = p1->nilai1;
                p1->nilai1 = min->nilai1;
                min->nilai1 = strtemp2;

                strtemp2 = p1->nilai2;
                p1->nilai2 = min->nilai2;
                min->nilai2 = strtemp2;
            }
        }
    }
}

```

```

        strtemp2 = p1->nilai3;
        p1->nilai3 = min->nilai3;
        min->nilai3 = strtemp2;

        strtemp = p1->keterangan;
        p1->keterangan = min->keterangan;
        min->keterangan = strtemp;
    }
    p1 = p1->next;
}
else
{
    cout << "Data Masih Kosong!!!" << endl;
}
}

void sorting_nama()
{
    string temp;
    string strtemp;
    float strtemp2;
    Siswa *p1, *p2, *min;
    p1 = new Siswa();
    p2 = new Siswa();
    min = new Siswa();

    if (head != NULL)
    {
        p1 = head;
        while (p1->next != NULL)
        {
            p2 = p1;
            min = p1;
            while (p2 != NULL)
            {
                if (p2->nama < min->nama)
                {
                    min = p2;
                }
                p2 = p2->next;
            }
            if (p1->nama > min->nama)
            {
                temp = p1->nama;
                p1->nama = min->nama;
                min->nama = temp;

                strtemp = p1->nisn;
                p1->nisn = min->nisn;
                min->nisn = strtemp;

                strtemp2 = p1->hasil;
                p1->hasil = min->hasil;
                min->hasil = strtemp2;

                strtemp = p1->jk;
                p1->jk = min->jk;
                min->jk = strtemp;

                strtemp = p1->tempat;
                p1->tempat = min->tempat;
                min->tempat = strtemp;
            }
        }
    }
}

```



```

        strtemp = p1->tgllahir;
        p1->tgllahir = min->tgllahir;
        min->tgllahir = strtemp;

        strtemp = p1->alamat;
        p1->alamat = min->alamat;
        min->alamat = strtemp;

        strtemp = p1->asal_sd;
        p1->asal_sd = min->asal_sd;
        min->asal_sd = strtemp;

        strtemp2 = p1->nilai1;
        p1->nilai1 = min->nilai1;
        min->nilai1 = strtemp2;

        strtemp2 = p1->nilai2;
        p1->nilai2 = min->nilai2;
        min->nilai2 = strtemp2;

        strtemp2 = p1->nilai3;
        p1->nilai3 = min->nilai3;
        min->nilai3 = strtemp2;

        strtemp = p1->keterangan;
        p1->keterangan = min->keterangan;
        min->keterangan = strtemp;
    }
    p1 = p1->next;
}
}
else
{
    cout << "Data Masih Kosong!!!" << endl;
}
}

void print_sorting(){
    void printSingleLinkedList();
    int pilih_sorting;

    cout << "+++++\n";
    cout << "|1.| Sorting Berdasarkan Nama      |\n";
    cout << "|2.| Sorting Berdasarkan Nilai Tertinggi |\n";
    cout << "|                                           |\n";
    cout << "|3.| Kembali Ke Menu                      |\n";
    cout << "+++++\n";
    cout << "Masukkan Pilihan Anda = ";   cin >> pilih_sorting;

    if (pilih_sorting == 1){
        system("cls");
        cout << "Sebelum Di sorting\n";
        printSingleLinkedList();
        sorting_nama();
        cout << "\nSetelah Di Sorting Berdasarkan Nama Siswa\n";
        printSingleLinkedList();
        getch();
    }
    else if(pilih_sorting == 2){
        system("cls");
        cout << "Sebelum Di sorting\n";
        printSingleLinkedList();
        sorting_nilai();
    }
}

```



```

    }
    getch();
}

else if (pi == 2)
{
    cout << "\nMasukkan NISN : ";          cin >> nisn;
    while (!ketemu && cur != NULL)
    {
        if (nisn == cur->nisn)
        {
            ketemu = true;
        }
        else
        {
            cur = cur->next;
        }
    }
    if (ketemu)
    {
        cout << "Data NISN = " << nisn << " Ditemukan" << endl;
        print_urutan();
    }
    else
    {
        cout << "Data NISN = " << nisn << " Tidak Ditemukan" << endl;
    }
    getch();
}

else if (pi == 3)
{
    cout << "\nMasukkan Nilai Akhir : "; cin >> hasil;
    while (!ketemu && cur != NULL)
    {
        if (hasil == cur->hasil)
        {
            ketemu = true;
        }
        else
        {
            cur = cur->next;
        }
    }
    if (ketemu)
    {
        cout << "Nilai Akhir Siswa = " << hasil << " Ditemukan" << endl;
        print_urutan();
    }
    else
    {
        cout << "Nilai Akhir Siswa = " << hasil << " Tidak Ditemukan" << endl;
    }
    getch();
}
}

int main(){
    addFirst("2101030", "Liye", "perempuan", "karanganyar", "02-03-2003",
"Botok,Kerjo,Karanganyar", "SDN1", "-", 90, 80, 80, 0);
    addFirst("2245866", "Ades", "laki-laki", "surakarta", "23-05-2003",
"Botok,Kerjo,Karanganyar", "SDN5", "-", 65, 80, 69, 0);
    addFirst("2754866", "Mineral", "laki-laki", "jatim", "18-05-2003",
"Botok,Kerjo,Karanganyar", "SDN8", "-", 75, 50, 66, 0);
}

```

```

        addFirst("2210368", "Aqua", "laki-laki", "pekalongan", "18-06-2003",
"Botok,Kerjo,Karanganyar", "SDN4", "-", 45, 90, 59, 0);
        addLast("5223653", "Dee", "perempuan", "klaten", "12-12-2003",
"Botok,Kerjo,Karanganyar", "SDN5", "-", 90, 88, 70, 0);
        addLast("2343534", "Fiersa Besari", "laki-laki", "smg", "21-08-2003",
"Botok,Kerjo,Karanganyar", "SDN2", "-", 57, 52, 70, 0);
        addLast("5464334", "Dee", "perempuan", "solo", "26-04-2003",
"Botok,Kerjo,Karanganyar", "SDN2", "-", 68, 55, 30, 0);
        addLast("3244563", "des", "perempuan", "pacitan", "25-12-2003",
"Botok,Kerjo,Karanganyar", "SDN1", "-", 78, 97, 92, 0);

int pilih;
//untuk linkedlist
int posisi;
string nsn;
string nama, jk, tempat, tgllahir, alamat, asal_sd, keterangan;
float nilai1, nilai2, nilai3, hasil;

menu:
system ("cls");
cout << "+-----+\n";
cout << "|      Sistem Penerimaan Siswa Baru SMP Wacana      |\n";
cout << "+-----+\n";
cout << "|1. Insert Data Depan                                |\n";
cout << "|2. Insert Data Tengah                              |\n";
cout << "|3. Insert Data Belakang                            |\n";
cout << "|                                                    |\n";
cout << "|4. Ubah Data Siswa                                |\n";
cout << "|                                                    |\n";
cout << "|5. Delete Data Atas/Awal                           |\n";
cout << "|6. Delete Data Tengah                             |\n";
cout << "|7. Delete Data Bawah/Akhir                         |\n";
cout << "|                                                    |\n";
cout << "|8. Tampil Data Siswa yang Sudah Mendaftar          |\n";
cout << "|9. Sorting Data Siswa                             |\n";
cout << "|10. Searching Data Siswa                           |\n";
cout << "|11. Keluar                                          |\n";
cout << "+-----+\n";
cout << "Silahkan Masukkan Pilihan Anda :";
cin >> pilih;
cout << "\n";

//Tambah Awal
if (pilih == 1)
{
    system("cls");
    cout << "Masukkan Nsn Siswa    = "; cin >> nsn;
    cout << "Masukkan Nama Siswa    = "; fflush(stdin);      getline (cin,
nama);
    cout << "Masukkan Jenis Kelamin = "; fflush(stdin);      getline (cin,
jk);
    cout << "Masukkan Tempat Lahir = "; fflush(stdin);      getline (cin,
tempat);
    cout << "Masukkan Tgl lahir   = "; fflush(stdin);      getline (cin,
tgllahir);
    cout << "Masukkan Alamat      = "; fflush(stdin);      getline (cin,
alamat);
    cout << "Masukkan Asal SD      = "; fflush(stdin);      getline (cin,
asal_sd);
    cout << "Masukkan Nilai IPA    = "; cin >> nilai1;
    cout << "Masukkan Nilai MTK    = "; cin >> nilai2;
    cout << "Masukkan Nilai BI     = "; cin >> nilai3;
    addFirst(nsn, nama, jk, tempat, tgllahir, alamat, asal_sd, keterangan,
nilai1, nilai2, nilai3, hasil);
    goto menu;
}

```

```

    }

    //Tambah Tengah
    else if (pilih == 2)
    {
        system("cls");
        cout << "Masukkan Nisn Siswa    = "; cin >> nisn;
        cout << "Masukkan Nama Siswa    = "; fflush(stdin);      getline (cin,
nama);
        cout << "Masukkan Jenis Kelamin = "; fflush(stdin);      getline (cin,
jk);
        cout << "Masukkan Tempat Lahir  = "; fflush(stdin);      getline (cin,
tempat);
        cout << "Masukkan Tgl lahir     = "; fflush(stdin);      getline (cin,
tgllahir);
        cout << "Masukkan Alamat        = "; fflush(stdin);      getline (cin,
alamat);
        cout << "Masukkan Asal SD       = "; fflush(stdin);      getline (cin,
asal_sd);
        cout << "Masukkan Nilai IPA     = "; cin >> nilai1;
        cout << "Masukkan Nilai MTK     = "; cin >> nilai2;
        cout << "Masukkan Nilai BI      = "; cin >> nilai3;
        cout << "\nMasukkan Posisi Data ke= ";      cin >> posisi;
        addMiddle(nisn, nama, jk, tempat, tgllahir,alamat, asal_sd, keterangan,
nilai1, nilai2, nilai3, hasil, posisi);
        goto menu;
    }

    //Tambah Akhir
    else if (pilih == 3)
    {
        system("cls");
        cout << "Masukkan Nisn Siswa    = "; cin >> nisn;
        cout << "Masukkan Nama Siswa    = "; fflush(stdin);      getline (cin,
nama);
        cout << "Masukkan Jenis Kelamin = "; fflush(stdin);      getline (cin,
jk);
        cout << "Masukkan Tempat Lahir  = "; fflush(stdin);      getline (cin,
tempat);
        cout << "Masukkan Tgl lahir     = "; fflush(stdin);      getline (cin,
tgllahir);
        cout << "Masukkan Alamat        = "; fflush(stdin);      getline (cin,
alamat);
        cout << "Masukkan Asal SD       = "; fflush(stdin);      getline (cin,
asal_sd);
        cout << "Masukkan Nilai IPA     = "; cin >> nilai1;
        cout << "Masukkan Nilai MTK     = "; cin >> nilai2;
        cout << "Masukkan Nilai BI      = "; cin >> nilai3;
        addLast(nisn, nama, jk, tempat, tgllahir,alamat, asal_sd, keterangan,
nilai1, nilai2, nilai3, hasil);
        goto menu;
    }

    //ubah data
    else if (pilih == 4)
    {
        system("cls");
        printSingleLinkedList();
        cout << "\n\n";
        cout << "Masukkan Posisi Data ke- "; cin >> posisi;
        cout << "\nMasukkan Nisn Siswa  = "; cin >> nisn;
        cout << "Masukkan Nama Siswa    = "; fflush(stdin);      getline (cin,
nama);
        cout << "Masukkan Jenis Kelamin = "; fflush(stdin);      getline (cin,
jk);

```

```

tempat);        cout << "Masukkan Tempat Lahir = "; fflush(stdin);        getline (cin,
tgl_lahir);      cout << "Masukkan Tgl lahir    = "; fflush(stdin);        getline (cin,
alamat);         cout << "Masukkan Alamat      = "; fflush(stdin);        getline (cin,
asal_sd);        cout << "Masukkan Asal SD       = "; fflush(stdin);        getline (cin,
                cout << "Masukkan Nilai IPA    = "; cin >> nilai1;
                cout << "Masukkan Nilai MTK     = "; cin >> nilai2;
                cout << "Masukkan Nilai BI      = "; cin >> nilai3;
                change(nisn, nama, jk, tempat, tgl_lahir, alamat, asal_sd, keterangan,
nilai1, nilai2, nilai3, hasil, posisi);
                goto menu;
        }

//Hapus Awal
else if (pilih == 5)
{
    system("cls");
    cout << "Sebelum Di Hapus Awal/Atas\n";
    printSingleLinkedList();
    cout << "Tekan Untuk Lanjut....";
    getch();
    removeFirst();
    cout << "\n\nSetelah Di Hapus Data Awal/Atas\n";
    printSingleLinkedList();
    getch();
    goto menu;
}

//Hapus Tengah
else if (pilih == 6)
{
    system("cls");
    cout << "Sebelum Di Hapus Data Tengah\n";
    printSingleLinkedList();
    cout << "Masukkan No Data Yang Ingin Di Hapus = "; cin >> posisi;
    removeMiddle(posisi);
    cout << "\n\nSetelah Di Hapus Data Tengah\n";
    printSingleLinkedList();
    getch();
    goto menu;
}

//Hapus Akhir
else if (pilih == 7)
{
    system("cls");
    cout << "Sebelum Di Hapus Akhir/Bawah\n";
    printSingleLinkedList();
    cout << "Tekan Untuk Lanjut....";
    getch();
    removeLast();
    cout << "\n\nSetelah Di Hapus Data Akhir/Bawah\n";
    printSingleLinkedList();
    getch();
    goto menu;
}

//lihat data pendaftar
else if (pilih == 8)
{
    system("cls");
    printSingleLinkedList();
    getch();
}

```

```

        goto menu;
    }

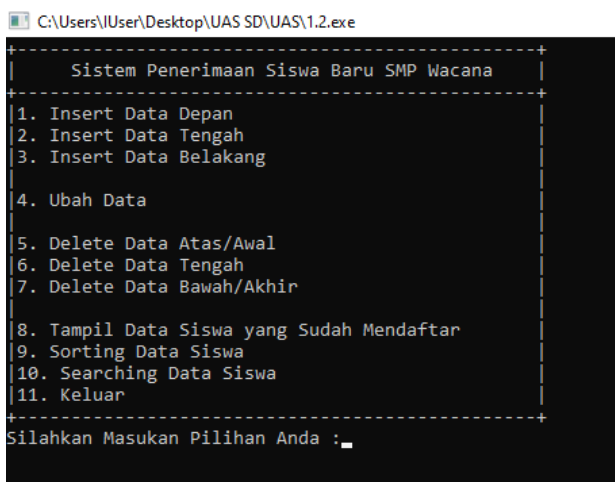
    //sorting data siswa
    else if (pilih == 9)
    {
        system("cls");
        print_sorting();
        goto menu;
    }

    //searching data siswa
    else if (pilih == 10)
    {
        system("cls");
        searching();
        goto menu;
    }

    //Menu Keluar
    else if (pilih == 11)
    {
        system("cls");
        cout << "Nama : Bachrul Rizal    || Muhammad Lintang Pamungkas\n";
        cout << "NIM : 210103009      ||      210103028\n";
        cout << "Kelas: 21 TIA1\n";
        cout << "\n\n\n\n\n\n\n\n";
    }
}

```

- Tampilan Hasil Compiler



Gambar 1 Tampilan Utama



Gambar 2 Tampilan Masukkan Data


```

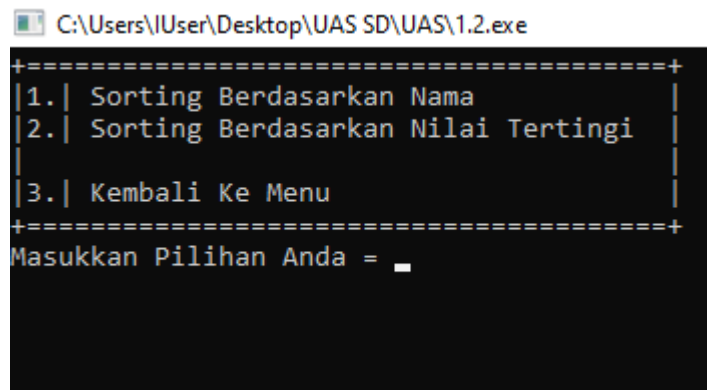
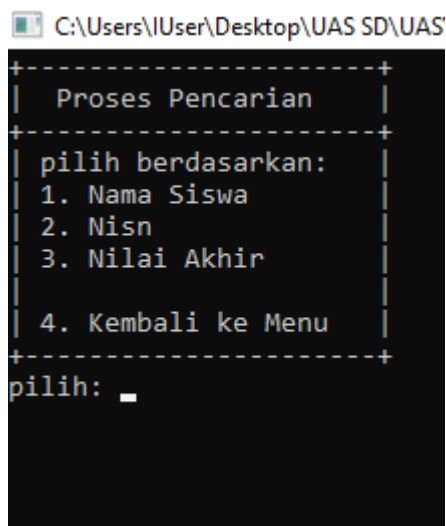
C:\Users\User\Desktop>UAS SD\UAS\1.2.exe
Jumlah data ada : 9
-----
|No.|Nisn|Nama|Jenis Kelamin|Tempat Tanggal Lahir|Alamat|Asal SD|Nilai IPA|Nilai MTK|BI|Jumlah|Keterangan|
-----
|1.|21010302SMuhammad Lintang|laki-laki|karanganyar|03-12-2003|jl.yos sudarso no4, bogor|SDN 01 Bogor|90|80|85|255|Diterima| |
|2.|2210368|Aqua|laki-laki|pekalongan|18-06-2003|Botok,Kerjo,Karanganyar|SDN4|45|90|59|194|Tidak Diterima|
|3.|2754866|Mineral|laki-laki|jatim|18-05-2003|Botok,Kerjo,Karanganyar|SDN8|75|50|66|191|Tidak Diterima|
|4.|2245866|Ades|laki-laki|surakarta|23-05-2003|Botok,Kerjo,Karanganyar|SDN5|65|80|69|214|Diterima|
|5.|2101030|Liye|perempuan|karanganyar|02-03-2003|Botok,Kerjo,Karanganyar|SDN1|90|80|80|259|Diterima|
|6.|5223653|Dee|perempuan|klaten|12-12-2003|Botok,Kerjo,Karanganyar|SDN5|90|88|70|248|Diterima|
|7.|2243534|Fiensa Besar|laki-laki|smg|21-08-2003|Botok,Kerjo,Karanganyar|SDN2|57|52|70|179|Tidak Diterima|
|8.|5464334|Dee|perempuan|solo|26-04-2003|Botok,Kerjo,Karanganyar|SDN2|68|55|30|153|Tidak Diterima|
|9.|3244563|des|perempuan|pacitan|25-12-2003|Botok,Kerjo,Karanganyar|SDN1|78|97|92|267|Diterima|
-----

Masukkan Posisi Data ke- 5

Masukkan Nisn Siswa = 212213
Masukkan Nama Siswa = nama baru
Masukkan Jenis Kelamin = laki-laki
Masukkan Tempat Lahir = bekasi
Masukkan Tgl lahir = 15-05-2003
Masukkan Alamat = bekasi timur
Masukkan Asal SD = sdn 01 rawa bebek
Masukkan Nilai IPA = 78
Masukkan Nilai MTK = 88
Masukkan Nilai BI = 93

```

No.	Nisn	Nama	Jenis Kelamin	Tempat Lahir	Tanggal	Alamat	Asal SD	Nilai IPA MTK BI	Jumlah	Keterangan
1.	210103028	Muhammad Lintang	laki-laki	karanganyar	03-12-2003	jl.yos sudarso no4, bogor	SDN 01 Bogor	90 80 85	255	Diterima
2.	2210368	Aqua	laki-laki	pekalongan	18-06-2003	Botok, Kerjo, Karanganyar	SDN4	45 90 59	194	Tidak Diterima
3.	2754866	Mineral	laki-laki	jatim	18-05-2003	Botok, Kerjo, Karanganyar	SDN8	75 50 66	191	Tidak Diterima
4.	2245866	Ades	laki-laki	surakarta	23-05-2003	Botok, Kerjo, Karanganyar	SDN5	65 80 69	214	Diterima
5.	212213	nama baru	laki-laki	bekasi	15-05-2003	bekasi timur	sdn 01 rawa bebek	78 88 93	259	Diterima
6.	5223653	Dee	perempuan	klatan	12-12-2003	Botok, Kerjo, Karanganyar	SDN5	90 88 70	248	Diterima
7.	2343534	Fiersa Besari	laki-laki	smg	21-08-2003	Botok, Kerjo, Karanganyar	SDN2	57 52 70	179	Tidak Diterima
8.	5464334	Dee	perempuan	solo	26-04-2003	Botok, Kerjo, Karanganyar	SDN2	68 55 30	153	Tidak Diterima
9.	2344563	des	perempuan	pacitan	25-12-2003	Botok, Kerjo, Karanganyar	SDN1	78 97 92	267	Diterima



BAB III PENUTUP

A. KESIMPULAN

Perkembangan teknologi yang semakin meningkat kini diterapkan oleh berbagai bidang dalam pembangunan, termasuk bidang system akademik. Penyelenggaraan pelayanan pendaftaran siswa mempunyai karakteristik dan organisasi yang sangat kompleks. Pendaftaran siswa SMP Wacana sebagai salah satu pelayanan pendidikan yang mempunyai karakteristik dan organisasi yang sangat kompleks, seperti pendaftaran data diri siswa, melihat hasil seleksi, melihat kuota siswa, dan sebagainya membutuhkan Aplikasi untuk mengoptimalkan pelayanan yang diberikan kepada masyarakat. Sistem Pendaftaran Siswa SMP Wacana adalah sebuah sistem yang digunakan untuk akademik dimana dalam aplikasi ini memungkinkan aliran data dari sebuah siswa bisa dilakukan secara elektronik, sehingga pelayanan kepada peserta didik baru dapat dilakukan dengan lebih cepat, akurat dan transparan yang pada akhirnya bisa memberikan kepuasan kepada siswa baru.

B. SARAN

Sistem Pendaftaran dan penerimaan siswa ini dapat dikembangkan agar lebih kompleks dan lebih mudah digunakan oleh admin. Data yang digunakan pun diharapkan lebih detail dan lebih lengkap dan juga disertai dengan penggunaan suatu algoritma pemrograman yang lebih baik .

DAFTAR PUSTAKA

SAIFUDIN. 2015. “Latar Belakang Pendaftaran Siswa”.

<http://eprints.umpo.ac.id/1135/4/BAB%20I.pdf>.

Pintarkom. 2019. “Single Linked List pada C++”. <https://pintarkom.com/single-linked-list-pada-c-plus/>.

Ketutrare. 2018. “*Metode Searching dan Sorting Double Linked List*”.

<https://www.ketutrare.com/2018/07/metode-searching-dan-sorting-double.html>.

Study With Student. 2021. “*Single linked list c++*”.

<https://www.youtube.com/watch?v=VVemCxif9vg&t=5s>.

Study With Student. 2021. “*source code Single linked list c++*”.

<https://github.com/yunusfebriansyah/struktur-data>.