

**PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN LANJUT**  
**SEMESTER GENAP T.A. 2024/2025**  
**LAPORAN PROYEK AKHIR**



**DISUSUN OLEH:**

**NAMA/NIM : Muhammad Naufal Alfarisi / 123240182**  
**KELAS/PLUG : IF-J**  
**NAMA ASISTEN : Azkal Azkiya Akbar / 124220085**  
**Muh. Raditya Haikal Mumtaz / 123210062**

**PROGRAM STUDI INFORMATIKA**  
**JURUSAN INFORMATIKA**  
**FAKULTAS TEKNIK INDUSTRI**  
**UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"**  
**YOGYAKARTA**  
**2025**

**HALAMAN PENGESAHAN**  
**LAPORAN PROYEK AKHIR**

Disusun oleh :

Muhammad Naufal Alfarisi 123240182

-

Telah Diperiksa dan Disetujui oleh Asisten Praktikum Algoritma dan Pemrograman  
Pada Tanggal : ...

**Asisten Praktikum**

**Asisten Praktikum**

Azkal Azkiya Akbar  
124220085

Muh. Raditya Haikal Mumtaz  
123210062

## **KATA PENGANTAR**

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa yang senantiasa mencurahkan rahmat dan hidayah-Nya sehingga kami dapat menyelesaikan Praktikum Pemrograman Lanjut serta laporan akhir Pemrograman Lanjut. Adapun laporan ini berisi tentang proyek akhir semester dari hasil pembelajaran selama praktikum berlangsung.

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

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

Yogyakarta, 02 Juni 2025

Penyusun

Muhammad Naufal Alfarisi

123240182

## DAFTAR ISI

<b>HALAMAN PENGESAHAN .....</b>	<b>1</b>
<b>KATA PENGANTAR .....</b>	<b>2</b>
<b>DAFTAR ISI .....</b>	<b>3</b>
<b>BAB I .....</b>	<b>1</b>
<b>BAB II.....</b>	<b>2</b>
<b>BAB III .....</b>	<b>3</b>
3.1. Implementasi Materi.....	3
3.2. Penjelasan Fitur .....	3
3.3. Screenshot Program .....	3
3.4. Source Code Lengkap.....	3
<b>BAB IV .....</b>	<b>4</b>
4.1. Jadwal Pengerjaan .....	4
4.2. Pembagian Tugas.....	4
<b>BAB V .....</b>	<b>5</b>
5.1 Kesimpulan.....	5
5.2 Saran .....	5
<b>DAFTAR PUSTAKA .....</b>	<b>6</b>

## **BAB I**

### **Aplikasi Manajemen Toko**

Program yang dikembangkan adalah aplikasi manajemen toko berbasis teks menggunakan bahasa pemrograman C++. Aplikasi ini bertujuan untuk membantu pengguna dalam mengelola data stok barang, termasuk penambahan, pencarian, pengurutan, pengeditan, penghapusan, dan penampilan data item. Konsep utama yang digunakan dalam program ini adalah implementasi *struct* sebagai representasi dari data item, penggunaan array statik untuk menyimpan data dalam memori sementara, serta operasi file untuk menyimpan dan membaca data secara permanen.

Program ini menyajikan menu interaktif yang dapat digunakan pengguna untuk memilih berbagai operasi yang tersedia. Setiap fitur telah dirancang dengan mempertimbangkan efisiensi dan kemudahan penggunaan. Selain itu, aplikasi ini mendukung sistem penyimpanan berbasis file eksternal (.dat) untuk menjaga keberlangsungan data meskipun program ditutup.

Topik-topik yang diterapkan meliputi: penggunaan *struct*, manipulasi file menggunakan *fstream* dan fungsi *fprintf/fscanf*, algoritma pengurutan (insertion sort dan selection sort), algoritma pencarian (sequential search), serta manipulasi array.

## **BAB II**

### **TUJUAN PROYEK AKHIR**

Tujuan utama dari proyek akhir ini adalah untuk mengimplementasikan seluruh materi yang telah diajarkan selama praktikum Algoritma dan Pemrograman Lanjut dalam bentuk aplikasi nyata. Dengan aplikasi manajemen toko ini, mahasiswa dapat memahami bagaimana konsep pemrograman terapan seperti manipulasi data, penggunaan struktur data, file handling, dan logika kontrol dapat digabungkan menjadi solusi perangkat lunak yang bermanfaat.

Secara khusus, aplikasi ini dibuat untuk:

1. Mengelola data item secara terstruktur menggunakan *struct*.
2. Melakukan proses CRUD (Create, Read, Update, Delete).
3. Mengimplementasikan teknik pencarian dan pengurutan data.
4. Menerapkan file eksternal untuk menyimpan data secara persist.
5. Meningkatkan kemampuan logika dan struktur program secara keseluruhan.

## BAB III

### PEMBAHASAN

#### 3.1. Implementasi Materi

##### 3.1.1 File Handling

File handling diterapkan untuk membaca, menulis, menyimpan, menyunting dan menghapus data.

```
ifstream inFile("data00.dat");
if (!inFile) {
    cerr << "Terjadi Kesalahan saat Membaca File!" << endl;
    cerr << "Membuat File Baru...";
    fstream newfile;
    newfile.open("data00.dat", ios::out);
    newfile.close();
    cerr << "\nSilakan Mulai Ulang Program.";
    getch();
    return 1;
}
string line;
while (getline(inFile, line) && count < 500) {
    stringstream ss(line);
    getline(ss, _data[count].id, ',');
    getline(ss, _data[count].name, ',');
    string stockstr, pricestr;
    getline(ss, stockstr, ',');
    getline(ss, pricestr, ',');
    _data[count].stock = stoi(stockstr);
    _data[count].price = stoi(pricestr);
    count++;
}
inFile.close();

...
FILE* outdata;
outdata = fopen("data00.dat", "a");
if (outdata == NULL){
    cout<<"Terjadi Kesalahan saat Membaca File!";
    getch();
    main();
}
fprintf(outdata, "%s,%s,%d,%d\n", temp[i].id.c_str(),
temp[i].name.c_str(), temp[i].stock, temp[i].price);
fclose(outdata);

...
ifstream infile("data00.dat");
ofstream tempfile("temp00.dat");
```

```

        if(!infile || !tempfile) {
            cerr << "Terjadi Kesalahan saat Membuka File.\n";
            getch();
            main();
        }
        ostringstream oss;
        oss << temp.id << "," << temp.name << "," << temp.stock << "," <<
temp.price;
        if(outp01 == 'y' || outp01 == 'Y'){
            while (getline(infile, line)) {
                if (line.find(id_) != string::npos) {
                    line = oss.str();
                }
                tempfile << line << '\n';
            }
        }
        infile.close();
        tempfile.close();
        remove("data00.dat");
        rename("temp00.dat", "data00.dat");
        ...
    ifstream infile("data00.dat");
    ofstream tempfile("temp00.dat");
    if(!infile || !tempfile) {
        cerr << "Terjadi Kesalahan saat Membuka File.\n";
        getch();
        main();
    }
    while (getline(infile, line)) {
        if (line.find(id_) != string::npos) {
            continue;
        }
        tempfile << line << '\n';
    }
    infile.close();
    tempfile.close();
    remove("data00.dat");
    rename("temp00.dat", "data00.dat");

```



## 3.2. Penjelasan Fitur

### 3.2.1 Sorting

Fitur sorting digunakan untuk mengurutkan data.

```
void sort_by_name(data00 _data[], int count){
    int i, j;
    data00 temp;
    cout<<"\033c";
    header();
    if (count == 0){
        cerr << "Data tidak ada!";
        getch();
        main();
    }
    cout<<"Daftar Barang Sebelum Diurutkan : \n";
    data_output(_data, count);
    getch();
    for (i = 1; i < count; i++){
        temp = _data[i];
        j = i - 1;
        while (j >= 0 && _data[j].name > temp.name){
            _data[j + 1] = _data[j];
            j = j - 1;
        }
        _data[j + 1] = temp;
    }
    cout<<"\033c";
    header();
    cout<<"Daftar Barang Setelah Diurutkan : \n";
    data_output(_data, count);
    getch();
}

void sort_by_price(data00 _data[], int count){
    cout<<"\033c";
    header();
    if (count == 0){
        cerr << "Data tidak ada!";
        getch();
        main();
    }
    cout<<"Daftar Barang Sebelum Diurutkan : \n";
    data_output(_data, count);
    getch();
    for (int i = 0; i < count - 1; i++){
        int awal = i;
        for(int j = i + 1; j < count; j++){
            if (_data[j].price < _data[awal].price){
                awal = j;
            }
        }
        swap(_data[awal], _data[i]);
    }
```

```
}  
cout<<"\033c";  
header();  
cout<<"Daftar Barang Setelah Diurutkan : \n";  
data_output(_data, count);  
getch();  
}
```

### 3.3. Screenshot Program

#### 3.3.1. Main Menu

```
-----  
|          STORE MANAGER          |  
-----  
Menu :  
1. Tambahkan Item  
2. Cari Item  
3. Urutkan Item  
4. Tampilkan Item  
5. Edit Item  
6. Hapus Item  
7. Keluar  
Pilih : 
```

#### 3.3.2. Menu Masukan

```
-----  
|          STORE MANAGER          |  
-----  
Masukkan jumlah item yang ingin ditambahkan : 2  
Item #1  
ID   : KJHG  
Nama : Item00  
Stock : 456  
Price : 20000  
Item #2  
ID   : JNMH  
Nama : Item01  
Stock : 585  
Price : 41111  
Apakah Anda ingin menambahkan data tersebut? (y/n) y  
Data berhasil ditambahkan!
```

#### 3.3.3. Menu Pencarian Data

```
-----  
|          STORE MANAGER          |  
-----  
"Item01" ditemukan pada index #1  
-----  
|ID   |Nama Produk           |QTY |Harga  |  
-----  
|JNMH |Item01                 |585 |41111  |  
-----
```

### 3.3.4. Menu Pengurutan Data

STORE MANAGER				
Daftar Barang Setelah Diurutkan :				
ID	Nama Produk	QTY	Harga	
KJHG	Item00	456	20000	
JNMH	Item01	585	41111	

### 3.3.5. Menu Tampilan Data

STORE MANAGER				
ID	Nama Produk	QTY	Harga	
KJHG	Item00	456	20000	
JNMH	Item01	585	41111	

### 3.3.6. Menu Penyuntingan Data

```
-----  
| STORE MANAGER |  
-----  
ID : JMJN  
Nama : Edited  
Stock : 654  
Price : 3248  
Ubah item ini? (y/n) y  
Data Berhasil Diubah.
```

### 3.3.7. Menu Penghapusan Data

```
-----  
| STORE MANAGER |  
-----  
Masukkan ID Item yang akan dihapus : KJHG  
Data Berhasil Dihapus.
```

### 3.4.Source Code Lengkap

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

struct data00{
    string id;
    string name;
    int stock;
    int price;
};

void header(){
    cout << setfill('-') << setw(35) << "-" << endl;
    cout << "|" << setfill(' ') << setw(23) << "STORE MANAGER" <<
setw(12) << "|\\n";
    cout << setfill('-') << setw(35) << "-" << endl;
}

void data_input(data00 _data[], int &count);

void data_output(data00 _data[], int qty);

void sort_by_name(data00 _data[], int count);

void sort_by_price(data00 _data[], int count);

void sequential_search(data00 _data[], int count);

void data_edit(data00 _data[], int &count);

void data_delete(data00 _data[], int &count);

int main(){
    data00 _data[500]; int count = 0, menu00;
    char exit_c;
    ifstream inFile("data00.dat");
    if (!inFile) {
        cerr << "Terjadi Kesalahan saat Membaca File!" << endl;
        cerr << "Membuat File Baru...";
        fstream newfile;
        newfile.open("data00.dat", ios::out);
        newfile.close();
        cerr << "\\nSilakan Mulai Ulang Program.";
        getch();
        return 1;
    }
    string line;
```



```

                break;
            default: cout<<"Input Error!";
                    getch();
                    break;
        }
    } while (menu_c2 != 1 && menu_c2 != 2);
    break;

case 4: system("cls");
        header();
        data_output(_data, count);
        getch();
        break;

case 5: data_edit(_data, count);
        break;

case 6: data_delete(_data, count);
        break;

case 7: cout<<"Terima Kasih!";
        getch();
        exit(1);
        break;

default: cout<<"Input Invalid!";
         getch();
         break;
    }
}while (menu00 != 7);
}

void data_input(data00 _data[], int &count){
    int qty;
    char confirm;
    system("cls");
    do{
        cout<<"\033c";
        header();
        cout<<"Masukkan jumlah item yang ingin ditambahkan : ";
        cin>>qty;
        cin.ignore();
        if (qty < 1){
            cout<<"Input Invalid!";
            getch();
        }
    } while (qty < 1);
    data00 temp[qty];
    for(int i = 0; i < qty; i++){
        cout<<"Item #"<<i + 1;
        cout<<"\nID    : "; cin>>temp[i].id; cin.ignore();
        cout<<"Nama  : ";getline(cin, temp[i].name);
        cout<<"Stock : ";cin>>temp[i].stock;
    }
}

```

```

        cout<<"Price : ";cin>>temp[i].price;
    }
    cout<<"Apakah Anda ingin menambahkan data tersebut? (y/n) ";
    cin>>confirm;
    if (confirm == 'y' || confirm == 'Y'){
        for (int i = 0; i < qty; i++){
            FILE* outdata;
            outdata = fopen("data00.dat", "a");
            if (outdata == NULL){
                cout<<"Terjadi Kesalahan saat Membaca File!";
                getch();
                main();
            }
            fprintf(outdata, "%s,%s,%d,%d\n", temp[i].id.c_str(),
temp[i].name.c_str(), temp[i].stock, temp[i].price);
            fclose(outdata);
            _data[count].id = temp[i].id;
            _data[count].name = temp[i].name;
            _data[count].stock = temp[i].stock;
            _data[count].price = temp[i].price;
            count++;
        }
        cout<<"Data berhasil ditambahkan!";
    }
    getch();
}

void data_output(data00 _data[], int qty){
    cout << setiosflags (ios::left);
    cout << setfill ('-') << setw (72) << "-" << endl;
    cout<<"|"
    <<setfill(' ')
    <<setw(6) << "ID" << "|"
    <<setw(45) << "Nama Produk" << "|"
    <<setw(5) << "QTY" << "|"
    <<setw(11) << "Harga" << "|" << endl;
    cout<<setfill('-') << setw(72) << "-" << endl;
    if(qty == 0){
        cout << "|" << setfill(' ') << setw(25) << " "
        << setw(45) << "Tidak ada data!" << "|" << endl;
    }
    for (int i = 0; i < qty; i++) {
        cout <<"|"
        << setfill(' ')
        << setw (6) << _data[i].id << "|"
        << setw (45) << _data[i].name << "|"
        << setw (5) << _data[i].stock << "|"
        << setw (11) << _data[i].price << "|" << endl;
    }
    cout << setfill ('-') << setw (72) << "-" << endl;
    cout << resetiosflags (ios::left);
}

```



```

void sort_by_name(data00 _data[], int count){
    int i, j;
    data00 temp;
    cout<<"\033c";
    header();
    if (count == 0){
        cerr << "Data tidak ada!";
        getch();
        main();
    }
    cout<<"Daftar Barang Sebelum Diurutkan : \n";
    data_output(_data, count);
    getch();
    for (i = 1; i < count; i++){
        temp = _data[i];
        j = i - 1;
        while (j >= 0 && _data[j].name > temp.name){
            _data[j + 1] = _data[j];
            j = j - 1;
        }
        _data[j + 1] = temp;
    }
    cout<<"\033c";
    header();
    cout<<"Daftar Barang Setelah Diurutkan : \n";
    data_output(_data, count);
    getch();
}

void sort_by_price(data00 _data[], int count){
    cout<<"\033c";
    header();
    if (count == 0){
        cerr << "Data tidak ada!";
        getch();
        main();
    }
    cout<<"Daftar Barang Sebelum Diurutkan : \n";
    data_output(_data, count);
    getch();
    for (int i = 0; i < count - 1; i++){
        int awal = i;
        for(int j = i + 1; j < count; j++){
            if (_data[j].price < _data[awal].price){
                awal = j;
            }
        }
        swap(_data[awal], _data[i]);
    }
    cout<<"\033c";
    header();
    cout<<"Daftar Barang Setelah Diurutkan : \n";
    data_output(_data, count);
}

```

```

    getch();
}

void sequential_search(data00 _data[], int count){
    string name_;
    int price_;
    bool found = false;
    cout<<"\033c";
    char outp;
    header();
    if (count == 0){
        cerr << "Data tidak ada!";
        getch();
        main();
    }
    cout << "Tampilkan Produk? (y/n) "; cin >> outp;
    if (outp == 'y' || outp == 'Y'){
        cout << "\033c";
        header();
        data_output(_data, count);
        getch();
    }
    cout<<"\033c";
    header();
    int menu_c1;
    cout<<"Cari Item Berdasarkan : \n";
    cout<<"1. Nama Item\n";
    cout<<"2. Harga Item\n";
    cout<<"Pilih : "; cin>>menu_c1;
    do{
        switch (menu_c1){
            case 1: cout << "Masukkan Nama Item : "; cin.ignore();
getline(cin, name_);
break;
            case 2: cout << "Masukkan Harga Item : "; cin>> price_;
break;
            default: cout<<"Input Invalid!";
break;
        }
    } while (menu_c1 != 1 && menu_c1 != 2);
    int i = 0;
    while ((i < count) && (!found)){
        if (_data[i].name == name_ || _data[i].price == price_)
        {
            found = true;
        } else{
            i = i + 1;
        }
    }
    if (found) {
        cout<<"\033c";
        header();
        if (menu_c1 == 1){

```

```

        cout << "\"" << name_ << "\"" ditemukan pada index #" <<
i<<endl;
    } else if (menu_c1 == 2){
        cout << "\"" << price_ << "\"" ditemukan pada index #" <<
i<<endl;
    }
    cout << setiosflags (ios::left);
    cout << setfill ('-') << setw (72) << "-" << endl;
    cout<<"|"
        <<setfill(' ')
        <<setw(6) << "ID" << "|"
        <<setw(45) << "Nama Produk" << "|"
        <<setw(5) << "QTY" << "|"
        <<setw(11) << "Harga" << "|" << endl;
    cout<<setfill('-') << setw(72) << "-" << endl;
    cout <<"|"
        << setfill(' ')
        << setw (6) << _data[i].id << "|"
        << setw (45) << _data[i].name << "|"
        << setw (5) << _data[i].stock << "|"
        << setw (11) << _data[i].price << "|" << endl;
    cout << setfill ('-') << setw (72) << "-" << endl;
    cout << resetiosflags (ios::left);
} else {
    if(menu_c1 == 1){
        cout<<"Nilai \""<<name_<<"\" tidak ditemukan:(";
    } else if(menu_c1 == 2){
        cout<<"Nilai \""<<price_<<"\" tidak ditemukan:(";
    }
}
getch();
}

void data_edit(data00 _data[], int &count){
    char outp, outp01;
    string id_;
    string line;
    data00 temp;
    cout<<"\033c";
    header();
    if (count == 0){
        cerr << "Data tidak ada!";
        getch();
        main();
    }
    cout << "Tampilkan Produk? (y/n) "; cin >> outp;
    if (outp == 'y' || outp == 'Y'){
        cout << "\033c";
        header();
        data_output(_data, count);
        getch();
    }
    cout<<"\033c";

```

```

    header();
    cout<<"Masukkan ID Item yang akan diubah : ";
    cin>>id_;
    ifstream infile("data00.dat");
    ofstream tempfile("temp00.dat");
    if(!infile || !tempfile) {
        cerr << "Terjadi Kesalahan saat Membuka File.\n";
        getch();
        main();
    }
    cout<<"\033c";
    header();
    cout<<"ID      : "; cin>>temp.id; cin.ignore();
    cout<<"Nama    : ";getline(cin, temp.name);
    cout<<"Stock  : ";cin>>temp.stock;
    cout<<"Price  : ";cin>>temp.price;
    cout<<"Ubah item ini? (y/n) ";
    cin>>outp01;
    ostringstream oss;
    oss << temp.id << "," << temp.name << "," << temp.stock << "," <<
temp.price;
    if(outp01 == 'y' || outp01 == 'Y'){
        while (getline(infile, line)) {
            if (line.find(id_) != string::npos) {
                line = oss.str();
            }
            tempfile << line << '\n';
        }
    }
    infile.close();
    tempfile.close();
    remove("data00.dat");
    rename("temp00.dat", "data00.dat");
    cout<<"Data Berhasil Diubah.";
    getch();
    main();
}

void data_delete(data00 _data[], int &count){
    char outp;
    string id_;
    string line;
    cout<<"\033c";
    header();
    if (count == 0){
        cerr << "Data tidak ada!";
        getch();
        main();
    }
    cout << "Tampilkan Produk? (y/n) "; cin >> outp;
    if (outp == 'y' || outp == 'Y'){
        cout << "\033c";
        header();

```

```

        data_output(_data, count);
        getch();
    }
    cout<<"\033c";
    header();
    cout<<"Masukkan ID Item yang akan dihapus : ";
    cin>>id_;
    ifstream infile("data00.dat");
    ofstream tempfile("temp00.dat");
    if(!infile || !tempfile) {
        cerr << "Terjadi Kesalahan saat Membuka File.\n";
        getch();
        main();
    }
    while (getline(infile, line)) {
        if (line.find(id_) != string::npos) {
            continue;
        }
        tempfile << line << '\n';
    }
    infile.close();
    tempfile.close();
    remove("data00.dat");
    rename("temp00.dat", "data00.dat");
    cout<<"Data Berhasil Dihapus.";
    getch();
    main();
}

```

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Aplikasi manajemen toko ini berhasil mengimplementasikan berbagai materi praktikum Algoritma dan Pemrograman Lanjut. Fitur-fitur dasar seperti pencarian, pengurutan, penambahan, pengeditan, dan penghapusan data dapat dijalankan dengan baik. Proyek ini membuktikan bahwa kombinasi dari berbagai konsep dasar dapat membentuk program yang fungsional dan berguna.

**DAFTAR PUSTAKA**

Nopalinhu. 2025. Projek123240182. <https://github.com/nopalinhu/Projek123240182/>. 2025.