
Perancangan dan Implementasi Sistem Informasi Toko Buku Sederhana Menggunakan Bahasa C++

Joel Justin Adrian¹, Jonsahdo Saragih², Atika Dwi Rahma Dani³, Mandasari Kusuma Dyah Tantri⁴

Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Kota Kediri
Jl. Penanggungan No.41 a, Bandar Lor, Kec.Kota, Kota Kediri

Abstrak

Transformasi digital telah menjadi kebutuhan utama dalam pengelolaan usaha, termasuk pada sektor perdagangan toko buku. Sistem manual yang masih banyak digunakan menimbulkan berbagai permasalahan seperti kehilangan data, kesalahan pencatatan stok, dan lambatnya proses transaksi. Permasalahan ini tidak hanya memperlambat operasional toko, tetapi juga berdampak pada kepuasan pelanggan. Untuk mengatasi hal tersebut, diperlukan sistem informasi sederhana yang mampu mengelola data buku, mempermudah pencarian informasi, serta mempercepat proses transaksi. Penelitian ini bertujuan untuk merancang dan mengimplementasikan program toko buku berbasis bahasa pemrograman C++ sebagai solusi awal dalam komputerisasi usaha kecil. Metode yang digunakan adalah pendekatan pemrograman terstruktur dengan penggunaan array dan fungsi untuk menyimpan, mengelola, serta memproses data buku dan transaksi. Hasil implementasi menunjukkan bahwa program mampu menyimpan data buku secara digital, melakukan pencarian berdasarkan judul atau kategori, dan menghitung total transaksi secara otomatis. Program ini juga meminimalisir kesalahan manusia (human error) serta meningkatkan efisiensi kerja. Dengan adanya sistem ini, pengelolaan toko buku menjadi lebih tertata dan profesional meskipun dalam skala kecil. Kesimpulannya, penggunaan program komputerisasi sederhana dapat menjadi solusi efektif dalam menunjang pengelolaan usaha toko buku dan merupakan langkah awal yang penting menuju digitalisasi usaha mikro dan menengah.

Kata kunci: Toko buku, Komputerisasi, Sistem Informasi, Bahasa C++, Efisiensi

Abstract

Digital transformation has become a primary necessity in business management, including in the book retail sector. Manual systems that are still widely used often lead to various issues such as data loss, stock recording errors, and slow transaction processes. These problems not only hinder store operations but also affect customer satisfaction. To address these challenges, a simple information system is needed to manage book data, facilitate information retrieval, and accelerate transaction processes. This study aims to design and implement a bookstore program using the C++ programming language as an initial solution to computerize small-scale businesses. The method used is a structured programming approach involving arrays and functions to store, manage, and process book data and transactions. The implementation results show that the program is capable of storing book data digitally, performing searches based on titles or categories, and automatically calculating transaction totals. This program also minimizes human error and improves work efficiency. With this system, bookstore management becomes more organized and professional, even on a small scale. In conclusion, the use of a simple computerized program can be an effective solution in supporting bookstore business management and represents an important first step toward the digitalization of micro and small enterprises.

Keywords: Bookstore, computerization, information system, C++ language, efficiency

PENDAHULUAN

Di era digital saat ini, penggunaan teknologi informasi telah menjadi bagian yang tidak terpisahkan dari berbagai aspek kehidupan termasuk dalam dunia bisnis dan perdagangan. Salah satu bentuk usaha yang terdampak oleh kemajuan teknologi adalah toko buku [1], [2], [3]. Toko buku tidak lagi mengandalkan pencatatan manual dalam mengelola stok dan transaksi melainkan mulai

beralih ke sistem berbasis komputer yang lebih efisien dan terstruktur [4].

Pengelolaan toko buku secara manual seringkali menghadapi berbagai kendala seperti data buku yang tercecer, kesalahan dalam pencatatan stok, kesulitan dalam pencarian judul buku tertentu, dan proses transaksi yang lambat [5]. Permasalahan ini tidak hanya memperlambat operasional toko tetapi juga berdampak pada kepuasan



pelanggan. Oleh karena itu, dibutuhkan sistem informasi

sederhana yang dapat membantu pemilik toko dalam mengelola buku, melakukan pencarian, dan memproses pembelian dengan lebih cepat dan akurat [6], [7], [8].

Penerapan sistem komputerisasi toko buku, meskipun sederhana dapat menjadi solusi awal untuk mengatasi permasalahan tersebut. Sistem ini mampu menyimpan data buku dalam bentuk digital, melakukan pencarian buku berdasarkan judul atau kategori, dan memproses transaksi pembelian dengan perhitungan otomatis. Selain meningkatkan efisiensi kerja, sistem ini juga meminimalisir risiko human error [9], [10], [11].

Bahasa pemrograman C++ dipilih dalam perancangan sistem ini karena merupakan bahasa yang cukup populer, efisien, dan memiliki struktur logika yang kuat untuk membangun aplikasi berbasis teks. Melalui makalah ini, kami akan merancang dan mengimplementasikan program sederhana toko buku berbasis C++ sebagai model sistem yang dapat dikembangkan lebih lanjut untuk kebutuhan nyata [12], [13].

Dengan adanya sistem toko buku sederhana ini, diharapkan dapat memberikan pemahaman awal mengenai pentingnya komputerisasi dalam dunia usaha dan mendorong pemanfaatan teknologi dalam pengelolaan data secara profesional meskipun dimulai dari skala kecil [11], [14], [15].

METODE

2.3.1 Deklarasi Global

```
Const:
    JUMLAH_BUKU ← 7
Array:
    judul[JUMLAH_BUKU] ← ["C++", "Struktur", "Algoritma", "Web",
"Basis", "DesignPattern", "Photoshop"]
    genre[JUMLAH_BUKU] ← ["Pemrograman", "Pemrograman",
"Pemrograman", "Desain", "Database", "Pemrograman", "Desain"]
    deskripsi[JUMLAH_BUKU] ← ["Panduan lengkap bahasa C++",
"Belajar struktur data modern",
"Teknik algoritma dasar dan lanjutan",
"Membuat web dari nol",
"Dasar-dasar basis data relasional",
"Pola desain perangkat lunak OOP",
"Edit foto profesional dengan Photoshop"]
```

Gambar 1. Deklarasi Global

2.3.2 Prosedur Garis Menu

```
ULANGI 2 KALI
    COUT "=" sebanyak 41 kali di satu baris
    COUT newline
SELESAI
END
```

Gambar 2. Prosedur Garis Menu

2.3.3 Prosedur Garis

```
ULANGI 2 KALI
    COUT "=" sebanyak 41 kali di satu baris
    COUT newline
SELESAI
END
```

Gambar 3. Prosedur Garis

2.3.4 Fungsi HitungTotalBayar(Jika tanpa diskon)

```
INT total,hargaBuku, Jumlah;
    total ← hargaBuku * jumlah
RETURN total
END
```

Gambar 4. Fungsi HitungTotalBayar

**2.3.5 Fungsi Beli Buku**

```

INT jumlah, uang;
FUNGSI beliBuku(namaBuku, hargaBuku)
    INPUT jumlah
    PANGGIL garis

    JIKA jumlah ≤ 0 MAKA
        COUT "Jumlah tidak valid"
        RETURN 0
    ENDIF
    totalBayar ← hitungTotalBayar(hargaBuku, jumlah)
    COUT "Total bayar: " totalBayar
    INPUT uang

    PANGGIL garis

    JIKA uang ≥ totalBayar MAKA
        COUT data pembelian:
        judul, jumlah, totalBayar, uang, kembalian
        PANGGIL garis
        COUT "Terima kasih telah berbelanja"
        COUT "Cari buku lagi? (1 = Ya, 2 = Tidak)"
        INPUT pilih
        JIKA pilih = 1 MAKA
            RETURN 1
        ELSE
            RETURN 0
        ENDIF
    ELSE
        COUT "Uang tidak cukup"
        COUT "1. Cari buku lain, 2. Kembali ke menu"
        INPUT ulang
        JIKA ulang = 1 MAKA
            RETURN 1
        ELSE
            RETURN 0
        ENDIF
    ENDIF
END

```

Gambar 5. Fungsi Beli Buku**2.3.6 FUNGSI cariBuku**

```

Deklarasi: cariGenre, ditemukan[], jumlahDitemukan, pilihBuku, aksi, idx,
hasil
WHILE TRUE
    COUT daftar genre: Pemrograman, Desain, Database
    INPUT cariGenre

    JIKA cariGenre = "0" MAKA
        RETURN 0
    ENDIF

    PANGGIL garis
    Deklarasi ditemukan[], jumlahDitemukan ← 0

    FOR i ← 0 s/d JUMLAH_BUKU - 1
        JIKA genre[i] = cariGenre MAKA
            ditemukan[jumlahDitemukan] ← i
            jumlahDitemukan ← jumlahDitemukan + 1
        ENDIF
    ENDFOR

    JIKA jumlahDitemukan = 0 MAKA
        COUT "Buku tidak ditemukan"
        COUT "1. Cari lagi, 2. Kembali ke menu"
        INPUT aksi
        PANGGIL garis
        JIKA aksi = 2 MAKA
            RETURN 0
        ENDIF
    ELSE
        COUT daftar judul dari genre yang sesuai
        INPUT pilihBuku

        JIKA pilihBuku = 0 MAKA
            CONTINUE (ulang input genre)
        ENDIF

        JIKA pilihBuku < 1 ATAU > jumlahDitemukan MAKA
            COUT "Pilihan tidak valid"
            COUT
        ENDIF

        idx ← ditemukan[pilihBuku - 1]
        COUT detail buku[idx]
        INPUT aksi
    ENDIF

```

```

    JIKA aksi = 1 MAKA
        hasil ← beliBuku(judul[idx], harga[idx])
        RETURN hasil
    ELSE IF aksi = 2 MAKA
        RETURN (kembali ke genre)
    ELSE
        COUT "Pilihan tidak valid"
        CONTINUE
    ENDIF
ENDIF
ENDWHILE
END

```

Gambar 6. Fungsi cariBuku

2.3.7 PROGRAM UTAMA

```

    menu ← 0

    WHILE menu ≠ 2
        PANGGIL garisMenu
        COUT "SELAMAT DATANG DI TOKO BUKU DINUSPRO"
        PANGGIL garisMenu

        COUT Menu:

```

```

        1. Cari Buku
        2. Keluar
    INPUT menu
    PANGGIL garis

    IF menu = 1 THEN
        hasil ← cariBuku()
        IF hasil = 1 THEN
            menu ← 1
        ELSE
            menu ← 0
        ENDIF
    ELSE IF menu = 2 THEN
        COUT "Terima kasih telah menggunakan aplikasi"
    ELSE
        COUT "Pilihan tidak valid"
        menu ← 0
        RETURN
    ENDIF
ENDWHILE
END

```

Gambar 7. Program Utama

Program dimulai dengan menampilkan menu utama kepada pengguna berupa pilihan untuk mencari buku atau keluar dari aplikasi. Jika pengguna memilih untuk mencari buku, program meminta input genre buku (Pemrograman, Desain, atau Database), lalu menampilkan daftar buku yang sesuai. Pengguna dapat memilih salah satu buku untuk melihat detailnya dan melanjutkan proses pembelian. Dalam proses pembelian, pengguna akan diminta jumlah buku dan pembayaran; sistem menghitung total dan mengembalikan kembalian jika cukup, lalu menanyakan apakah ingin mencari buku lain atau kembali ke menu utama. Jika uang tidak mencukupi, pengguna juga diberi pilihan untuk kembali mencari buku atau ke menu. Program akan terus berjalan hingga pengguna memilih keluar dari aplikasi.

PEMBAHASAN

Pada tahun 1967, seorang ilmuwan komputer dari universitas Cambridge yang bernama Martin Richards menciptakan Bahasa BCPL (Basic Combined Programming Language) yang menjadi titik awal bagi perkembangan dan pembentukan Bahasa C++. BCPL dibuat untuk menghilangkan fitur lengkap dalam bahasa CPL (Combined Programming Language) yang menyulitkan proses kompilasi, yang dikembangkan pada awal tahun 1960-an. CPL dirancang di Laboratorium Matematika di Universitas Cambridge sebagai Bahasa Pemrograman Cambridge.

Dua tahun kemudian tepatnya pada tahun 1969, ilmuwan komputer bernama Kenneth Thompson dan Dennis MacAlister Ritchie membuat Sistem Operasi Unix di Bell Laboratories. Di tahun yang sama Kenneth Thompson juga membuat bahasa B yang merupakan pengembangan dari bahasa BCPL (Basic Combined Programming Language).

Pada tahun 1978 di Bell Laboratories, Brian Wilson Kernighan dan Dennis MacAlister Ritchie melakukan pengembangan terhadap bahasa B. Dalam pengembangan ini menghasilkan bahasa baru yaitu bahasa C yang populer hingga saat ini. Tahun 1989 merupakan tahun penting

setelah dikeluarkannya standar bahasa C oleh ANSI (American National Standards Institute) dan kemudian dikenal dengan nama ANSI C.

Awal tahun 1980, Bjarne Stroustrup seorang ilmuwan komputer yang bekerja di Bell Laboratories melakukan pengembangan bahasa C. Setelah melakukan pengembangan bahasa C selama 5 tahun, tepatnya pada tahun 1985 tercipta bahasa pemrograman baru yaitu bahasa C++. Bahasa C++ mengalami dua tahap evolusi, bahasa C++ pertama dinamakan cfront yang dirilis oleh Bell Laboratories. Cfront hanya berupa kompiler penerjemah bahasa C++ menjadi bahasa C.

Struktur Bahasa Pemrograman C++

Bahasa Pemrograman C++ adalah bahasa pemrograman tingkat tinggi (High Level Language) yang dapat digunakan untuk bahasa tingkat rendah (Low Level Language) dan juga dapat digunakan untuk bahasa assembly (mesin).

Struktur bahasa pemrograman C++ ada 3 bagian, yaitu:

1. Header/Kepala

Header adalah bagian yang berisi deklarasi konstanta, fungsi, namespace dan sebagainya. Header merupakan bagian yang mengatur kompilasi karena header itu sendiri adalah pengarah kompiler. Bagian-bagian pada header:

a. #include

Tanda kress (#) adalah indikasi kompiler untuk menyertakan header file standard iostream atau library yang digunakan.

b. using namespace std;

Using namespace std merupakan perintah untuk kompiler bahwa seluruh berkas atau fungsi menjadi bagian namespace std.

2. int main()

Fungsi main adalah mengeksekusi seluruh program C++. Main diikuti oleh sepasang tanda kurung () yang dapat berisi argumen.

Dibawah ini adalah struktur header pada bahasa C++ :

```
#include <iostream> //Header

using namespace std; //Header

int main()
```

Gambar 2. Struktur Header pada Bahasa C++

Implementasi Code Program C++



```

1 #include <iostream>
2 #include <string>
3 using namespace std;
4 //1. Deklarasi Global
5 const int JUMLAH_BUKU = 7;
6
7 // Data buku (judul, genre, deskripsi, harga)
8 string judul[JUMLAH_BUKU] = {
9     "C++", "Struktur", "Algoritma", "Web", "Basis", "DesignPattern", "Photoshop";
10
11 string genre[JUMLAH_BUKU] = {
12     "Pemrograman", "Pemrograman", "Pemrograman", "Desain", "Database", "Pemrograman", "Desain";
13
14 string deskripsi[JUMLAH_BUKU] = {
15     "Panduan lengkap bahasa C++",
16     "Belajar struktur data modern",
17     "Teknik algoritma dasar dan lanjutan",
18     "Membuat web dari nol",
19     "Dasar-dasar basis data relasional",
20     "Pola desain perangkat lunak OOP",
21     "Edit foto profesional dengan Photoshop";
22
23 int harga[JUMLAH_BUKU] = {
24     50000, 60000, 55000, 70000, 65000, 72000, 48000;
25

```

Gambar 2. Deklarasi Global

Bagian ini mendefinisikan konstanta JUMLAH_BUKU yang bernilai 7, serta 4 array yaitu judul, genre, deskripsi, dan harga yang masing-masing berisi data lengkap mengenai buku yang tersedia di toko, seperti nama buku, jenis genre, deskripsi singkat, dan harga setiap buku.

```

25 //2. Prosedur garis menu
26 void garisMenu()
27 {
28     for (int i = 0; i < 2; i++)
29     {
30         for (int j = 0; j < 41; j++)
31             cout << " ";
32         cout << endl;
33     }
34 }
35 //3. Prosedur garis biasa
36 void garis()
37 {
38     for (int i = 0; i < 2; i++)
39     {
40         for (int j = 0; j < 41; j++)
41             cout << " ";
42         cout << endl;
43     }
44 }

```

Gambar 3. Prosedur Garis Menu dan Garis Biasa

Tempatkan Keduanya digunakan untuk menampilkan dua baris garis pemisah sebanyak 41 karakter sebagai elemen visual. garisMenu() digunakan khusus untuk membingkai tampilan menu utama, sedangkan garis() digunakan secara umum untuk memisahkan bagian-bagian dalam program agar tampilan lebih rapi dan terstruktur.

```

45
46 //4. Fungsi hitung total bayar tanpa diskon
47 int hitungTotalBayar(int hargaBuku, int jumlah)
48 {
49     return hargaBuku * jumlah;
50 }
51

```

Gambar 4. Fungsi Hitung Total Bayar

Fungsi ini digunakan ketika menghitung total biaya pembelian buku tanpa diskon dengan cara mengalikan harga satuan buku dengan jumlah yang dibeli.

```

//5. Fungsi Beli buku
int beliBuku(string namaBuku, int hargaBuku)
{
    int jumlah, uang;

    cout << "\nMasukkan jumlah buku yang ingin dibeli: ";
    cin >> jumlah;
    garis();

    if (jumlah <= 0)
    {
        cout << "\nJumlah tidak valid.\n\n";
        return 0; // kembali ke menu utama
    }

    int totalBayar = hitungTotalBayar(hargaBuku, jumlah);

    cout << "Total bayar          : Rp " << totalBayar << endl;
    cout << "Masukkan uang anda       : Rp ";
    cin >> uang;
    garis();

    if (uang >= totalBayar)
    {
        cout << "\nBerhasil membeli buku:\n";
        cout << "Judul      : " << namaBuku << "\n";
        cout << "Jumlah     : " << jumlah << "\n";
        cout << "Total bayar : Rp " << totalBayar << "\n";
        cout << "Bayar      : Rp " << uang << "\n";
        cout << "Kembalian  : Rp " << uang - totalBayar << "\n\n";
        garis();
    }
}

```

```

// terimakasih sudah berbelanja
cout << "Terimakasih Sudah berbelanja di Toko Buku Dinus Pro\n";
<< endl;

cout << "Apakah Anda ingin mencari buku lagi?\n";
cout << "1. Ya\n2. Tidak (Menu)\nPilih: ";
int pilih;
cin >> pilih;
if (pilih == 1)
{
    return 1; // cari buku lagi
}
else
{
    return 0; // keluar ke menu utama
}
}
else
{
    cout << "Uang tidak cukup.\n";
    cout << "1. Cari buku lain\n";
    cout << "2. Kembali ke menu utama\nPilih: ";
    int ulang;
    cin >> ulang;
    if (ulang == 1)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}
}

```

Gambar 5. Fungsi Beli Buku

Fungsi ini mengatur proses pembelian buku, dimulai dari input jumlah buku yang ingin dibeli. Jika jumlah tidak



valid (≤ 0), maka pembelian dibatalkan dan fungsi mengembalikan 0. Bila jumlah valid, sistem menghitung total harga menggunakan `hitungTotalBayar()`, lalu meminta input uang dari pengguna. Jika uang cukup, ditampilkan detail transaksi berupa judul buku, jumlah, total bayar, uang bayar, dan kembalian.

Setelah transaksi berhasil, pengguna diberikan opsi untuk mencari buku lain (return 1) atau kembali ke menu utama (return 0). Namun jika uang tidak mencukupi, pengguna juga diberi dua opsi serupa. Dengan begitu, nilai 1 mengarahkan pengguna untuk kembali ke proses pencarian, sementara 0 akan membawa pengguna kembali ke menu utama.

```
// 6. Fungsi Cari Buku
int cariBuku()
{
    while (true)
    {
        cout << "Genre buku yang tersedia: " << endl;
        cout << "\n1. Pemrograman. ";
        cout << "\n2. Desain. ";
        cout << "\n3. Database." << endl;
        cout << "\nMasukkan genre buku yang tersedia atau 0 untuk kembali ke menu utama\n";
        string cariGenre;
        cin >> cariGenre;

        if (cariGenre == "0")
        {
            return 0; // langsung kembali ke menu utama
        }

        garis();

        int ditemukan[JUMLAH_BUKU];
        int jumlahDitemukan = 0;

        for (int i = 0; i < JUMLAH_BUKU; i++)
        {
            if (genre[i] == cariGenre)
            {
                ditemukan[jumlahDitemukan] = i;
                jumlahDitemukan++;
            }
        }

        if (jumlahDitemukan == 0)
        {
            cout << "Tidak ditemukan buku dengan genre \"" << cariGenre << "\".\n";
            cout << "1. Cari lagi\n2. Kembali ke menu utama\nPilih: ";
            int aksi;
            cin >> aksi;
            garis();
            if (aksi == 2)
            {
                return 0;
            }
        }
    }
}
```

```
{
    return 0; // kembali menu utama
}
// jika aksi == 1, ulang loop cari genre
}
else
{
    cout << "Buku dengan genre \"" << cariGenre << "\":\n";
    for (int i = 0; i < jumlahDitemukan; i++)
    {
        cout << i + 1 << ". " << judul[ditemukan[i]] << endl;
    }
    cout << "\nPilih nomor buku untuk lihat detail (0 = kembali cari genre): ";
    int pilihBuku;
    cin >> pilihBuku;

    if (pilihBuku == 0)
    {
        continue; // ulang input genre
    }

    if (pilihBuku < 1 || pilihBuku > jumlahDitemukan)
    {
        cout << "Pilihan tidak valid.\n";
        continue; // ulang input buku
    }

    int idx = ditemukan[pilihBuku - 1];

    cout << "\nDetail Buku:\n";
    cout << "Judul      : " << judul[idx] << "\n";
    cout << "Genre       : " << genre[idx] << "\n";
    cout << "Deskripsi   : " << deskripsi[idx] << "\n";
    cout << "Harga       : Rp " << harga[idx] << "\n";

    cout << "\n1. Beli Buku\n2. Kembali ke daftar genre\nPilih: ";
    int aksi;
    cin >> aksi;

    if (aksi == 1)
    {
        if (aksi == 1)
        {
            int hasil = beliBuku(judul[idx], harga[idx]);
            if (hasil == 1)
            {
                return 1; // cari buku lagi
            }
            else
            {
                return 0; // kembali menu utama
            }
        }
        else if (aksi == 2)
        {
            continue; // kembali ke input genre
        }
        else
        {
            cout << "Pilihan tidak valid.\n";
            continue; // ulang input detail buku
        }
    }
}
}
```

Gambar 6. Fungsi Cari Buku

Fungsi ini memungkinkan pengguna mencari buku berdasarkan genre yang tersedia, yaitu "Pemrograman", "Desain", dan "Database". Pengguna diminta memasukkan genre, lalu program mencocokkan input dengan data yang ada. Jika tidak ditemukan buku yang sesuai, pengguna akan diberikan pilihan untuk mencoba mencari lagi atau kembali ke menu utama. Namun jika ditemukan, sistem akan menampilkan daftar buku dari genre tersebut, lengkap dengan nomor urutnya.

Setelah itu, pengguna dapat memilih salah satu buku untuk melihat detail informasi seperti judul, genre, deskripsi, dan harga. Dari sini, pengguna diberikan dua opsi: melanjutkan pembelian dengan memanggil fungsi `beliBuku()` atau kembali ke daftar genre untuk melakukan pencarian lain. Fungsi ini mengembalikan nilai 1 jika

pengguna ingin kembali mencari buku, dan 0 jika ingin kembali ke menu utama.

```

1 // 7. Main Program
2 int main()
3 {
4     int menu = 0;
5
6     while (menu != 2)
7     {
8         garisMenu();
9         cout << "||\n";
10        cout << "||          SELAMAT DATANG DI TOKO          ||\n";
11        cout << "||          BUKU DINUSPRO          ||\n";
12        cout << "||\n";
13        garisMenu();
14
15        cout << "\nSilahkan pilih Menu:\n";
16        cout << "1. Cari Buku\n";
17        cout << "2. Keluar\n";
18        cout << "Pilih: ";
19        cin >> menu;
20        garis();
21
22        if (menu == 1)
23        {
24            int hasil = cariBuku();
25
26            if (hasil == 1)
27            {
28                menu = 1; // cari buku lagi
29            }
30            else
31            {
32                menu = 0; // kembali ke menu utama
33            }
34        }
35        else if (menu == 2)
36        {
37            cout << "\nTerima kasih telah menggunakan aplikasi.\n";
38        }
39        else
40        {
41            cout << "Pilihan tidak valid.\n";
42            menu = 0; // ulang menu utama
43            return 0;
44        }
45    }
46 }

```

Gambar 7. Program Utama

Fungsi ini merupakan inti dari program yang menampilkan menu utama kepada pengguna dengan dua pilihan, yaitu mencari buku atau keluar dari aplikasi. Menu ditampilkan dalam perulangan while sehingga program akan terus berjalan hingga pengguna memilih keluar. Jika memilih mencari buku, program akan menjalankan fungsi cariBuku() dan menyesuaikan alur berdasarkan interaksi pengguna.

PENUTUP

Simpulan

Penerapan sistem toko buku berbasis program C++ terbukti dapat membantu dalam pengelolaan data buku,

memudahkan proses pencarian, dan mempercepat transaksi pembelian. Program ini dirancang secara sederhana namun mampu memberikan gambaran nyata mengenai bagaimana teknologi dapat dimanfaatkan untuk meningkatkan efisiensi operasional sebuah toko buku.

Dengan menggunakan bahasa pemrograman C++, sistem ini dapat menjalankan fungsi-fungsi seperti penambahan data buku, pencarian berdasarkan judul atau kategori, serta pemrosesan pembelian secara otomatis. Meskipun belum dilengkapi dengan fitur canggih seperti integrasi database atau tampilan antarmuka grafis, sistem ini sudah memberikan pondasi awal yang kuat untuk pengembangan ke arah yang lebih kompleks dan profesional.

Secara keseluruhan, sistem ini mendukung kegiatan usaha toko buku agar menjadi lebih terstruktur, cepat, dan minim kesalahan serta dapat memberikan pelayanan yang lebih baik kepada pelanggan.

Saran

Untuk pengembangan ke depan, sistem toko buku ini sebaiknya dilengkapi dengan penyimpanan data berbasis database agar data tidak hilang setiap kali program ditutup. Selain itu, penambahan antarmuka grafis juga sangat disarankan untuk meningkatkan kenyamanan pengguna dalam mengoperasikan sistem terutama bagi pengguna yang tidak terbiasa dengan tampilan berbasis teks.

Dari sisi fungsionalitas, program juga dapat dikembangkan dengan fitur tambahan seperti sistem pencatatan stok otomatis, laporan penjualan, dan sistem login pengguna untuk keamanan. Dengan pengembangan lebih lanjut, program ini berpotensi menjadi aplikasi toko buku yang handal dan dapat digunakan secara nyata dalam operasional sehari-hari.

DAFTAR PUSTAKA

- [1] Safarina Binashrillah, Tiara Aulia, Sherly Rindayanti, Risqi Choirunnisa, Vita Ariska, and Rian Abdul Aziz, “Sistem Informasi Penjualan Pada Toko Buku Jendela Dunia Berbasis Web,” *JURNAL PENELITIAN SISTEM INFORMASI (JPSI)*, vol. 1, no. 3, pp. 142–153, Aug. 2023, doi: 10.54066/jpsi.v1i3.684.
- [2] N. Nilkawati, M. Assidiq, and S. Syarli, “Sistem Informasi Penjualan Buku Berbasis Web Pada Toko Buku Monster Book Kabupaten Mamuju,” *Journal Pegguruang: Conference Series*, vol. 2, no. 1, p. 289, May 2020, doi: 10.35329/jp.v2i1.1060.
- [3] M. I. Fauzi and Y. Cahyono, “Jurnal Teknologi Sistem Informasi dan Aplikasi Rancang Bangun Sistem Penjualan Buku Berbasis Web dengan Metode OOAD (Studi Kasus: Toko Buku LP3ES),” vol. 4, no. 4, pp. 211–218, 2021, doi: 10.32493/jtsi.v4i4.13264.
- [4] P. I. Cahayani, R. Suppa, and B. Sulaeman, “RANCANG BANGUN SISTEM INFORMASI PENJUALAN BUKU ONLINE PADA TOKO BUKU AKALANKA BERBASIS ANDROID,” 2024. [Online]. Available: <https://ojs.unanda.ac.id/index.php/jutinda>
- [5] R. Jayana Purba, “SISTEM INFORMASI BERBASIS WEB PADA TOKO BUKU DI TOKO BATAM,” *JURNAL COMASIE*, 2021.
- [6] M. Rizqy *et al.*, “Sistem Informasi Manajemen Persediaan Barang Pada Toko Buku (Studi Kasus Pada Toko Buku).”
- [7] Budi Hartono, “Sistem Informasi Penjualan Berbasis Web Pada Toko Buku Putra Ilmu Semarang,” *Jurnal Ilmiah Sistem Informasi*, vol. 3, no. 3, pp. 11–18, Jan. 2025, doi: 10.51903/qbaxkk13.
- [8] Herlinah B, “PERANCANGAN SISTEM INFORMASI PENJUALAN PADA TOKO BUKU PELANGI ILMU MAKASSAR BERBASIS BORLAND DELPHI,” vol. 11, p. 1, 2016.
- [9] A. Septiansyah, S. Hasanah, V. Nita Permatasari, and A. Yulawati, “SISTEM INFORMASI OTOMATISASI PELAPORAN DATA PENJUALAN TOKO BUKU NAZWA YANG MASUK DAN YANG KELUAR,” doi: 10.37817/ikraith-informatika.v8i1.
- [10] R. Mardikaningsih and D. Darmawan, “PERANAN SISTEM INFORMASI PERSEDIAAN TERHADAP PERSEPSI KEMUDAHAN PENGGUNAAN, KEGUNAAN YANG DIRASAKAN, DAN KEPUASAN PENGUNJUNG TOKO BUKU,” *Realible Accounting Journal*, vol. 1, no. 1, pp. 43–57, Aug. 2021, doi: 10.36352/raj.v1i1.135.
- [11] M. Desy Ria and A. Budiman, “PERANCANGAN SISTEM INFORMASI TATA KELOLA TEKNOLOGI INFORMASI PERPUSTAKAAN,” *Jurnal Informatika dan Rekayasa Perangkat Lunak (JATIKA)*, vol. 2, no. 1, pp. 122–133, 2021, [Online]. Available: <http://jim.teknokrat.ac.id/index.php/informatika>
- [12] N. Ade Riyanti, A. Prasetyo, Mk. Program Study Teknik Informatika Politeknik Purbaya, J. Pancakarya No, and T. Kab Tegal, “PROGRAM KASIR SEDERHANA PADA TOKO SUSU DENGAN MENGGUNAKAN BAHASA C++ BERBASIS CODE BLOCKS.”
- [13] N. Ariapuri, A. Prasetyo, M. Kom, and I. P. Purbaya, “EFEKTIVITAS APLIKASI KASIR MENGGUNAKAN CODE BLOCKS BERBASIS C++.”
- [14] M. Mailasari, “SISTEM INFORMASI PERPUSTAKAAN MENGGUNAKAN METODE WATERFALL,” *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 8, no. 2, pp. 207–214, Aug. 2019, doi: 10.32736/sisfokom.v8i2.657.
- [15] K. Wakhidah, B. Budiman, and W. Winarti, “Rancangan Sistem Informasi Perpustakaan Berbasis Website Menggunakan Barcode Di Sekolah MA Raden Rahmat,” *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 5, no. 1, pp. 61–68, Feb. 2023, doi: 10.47233/jteksis.v5i1.740.