

Implementasi Pemrograman Terstruktur C++: Pengembangan Aplikasi Menu Interaktif Sebagai Media Pembelajaran Konsep Dasar Pemrograman

Zoyanito Tsaqif¹, Christopher Ruben Ouji Raharjo², Devananda Jusuf Kurniawan³

^{1,2,3}Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Kota Semarang

Artikel Info

Kata kunci:

pemrograman terstruktur
bahasa C++
menu interaktif
media pembelajaran
konsep dasar pemrograman

ABSTRAK

Perkembangan teknologi informasi menuntut penguasaan keterampilan pemrograman yang terstruktur dan sistematis. Bahasa pemrograman C++ tetap menjadi pilihan utama dalam pembelajaran pemrograman karena kemampuannya dalam membangun fondasi pemahaman logika dan struktur program. Penelitian ini bertujuan untuk mengembangkan aplikasi menu interaktif berbasis C++ yang mengintegrasikan konsep-konsep fundamental pemrograman terstruktur sebagai media pembelajaran bagi pemula. Program yang dikembangkan terdiri dari tiga fitur utama yaitu penjualan, pembelian, dan sewa yang diimplementasikan menggunakan konsep fungsi, percabangan switch-case, perulangan do-while, dan struktur data array. Metode pengembangan dilakukan secara sistematis melalui tahapan analisis kebutuhan, perancangan algoritma, implementasi kode, pengujian, dan dokumentasi. Hasil pengembangan menunjukkan bahwa program berhasil mengimplementasikan konsep-konsep dasar pemrograman C++ dalam aplikasi yang interaktif dan fungsional. Program mampu melakukan perhitungan transaksi dengan akurat, menampilkan menu secara berulang hingga pengguna memilih untuk keluar, dan mengelola data menggunakan array dengan efisien. Struktur modular program memudahkan pemahaman logika dan memungkinkan pengembangan lebih lanjut. Program ini terbukti efektif sebagai media pembelajaran untuk melatih logika pemrograman dan memahami penerapan konsep-konsep teoritis dalam konteks aplikasi praktis. Pengembangan selanjutnya dapat difokuskan pada penambahan validasi input, sistem penyimpanan data, dan peningkatan antarmuka pengguna untuk menghasilkan aplikasi yang lebih robust dan profesional.

Penulis Korespondensi :

Zoyanito Tsaqif,
Program Studi Teknik Informatika
Fakultas Ilmu Komputer
Universitas Dian Nuswantoro, Kota Semarang, 50131
Email: 111202516206@mhs.dinus.ac.id

1. PENDAHULUAN

Perkembangan teknologi informasi yang sangat pesat pada era digital saat ini telah membawa perubahan signifikan dalam berbagai aspek kehidupan, termasuk dalam dunia pendidikan [1]. Integrasi teknologi dalam proses pembelajaran tidak lagi bersifat pelengkap, melainkan telah menjadi kebutuhan utama dalam membentuk sumber daya manusia yang kompeten dan adaptif terhadap perubahan zaman. Transformasi digital mendorong institusi pendidikan untuk menyesuaikan kurikulum agar selaras dengan kebutuhan industri dan perkembangan teknologi global. Kemajuan teknologi ini menuntut para pelajar dan mahasiswa untuk menguasai berbagai keterampilan teknologi, khususnya kemampuan pemrograman yang baik dan terstruktur [2]. Pemrograman tidak hanya menjadi kebutuhan bagi mereka yang mengambil jurusan teknologi informasi, tetapi juga menjadi keterampilan fundamental yang diperlukan dalam berbagai bidang keilmuan di era modern ini [3], [4].

Bahasa pemrograman C++ masih menjadi salah satu bahasa yang paling banyak digunakan sebagai dasar pembelajaran pemrograman di berbagai institusi pendidikan [5]. Hal ini dikarenakan C++ memiliki karakteristik yang membantu pembelajar dalam memahami sintaks, logika, dan struktur program secara sistematis [6]. Sebagai bahasa pemrograman tingkat menengah, C++ menyediakan keseimbangan antara kemudahan pembelajaran dan kemampuan kontrol yang mendalam terhadap sistem komputer [7]. Struktur sintaks yang tegas dan mekanisme pengelolaan memori yang eksplisit membantu mahasiswa memahami bagaimana program berinteraksi dengan perangkat keras. Selain itu, C++ mendukung berbagai paradigma pemrograman, baik prosedural maupun berorientasi objek, sehingga memberikan fleksibilitas dalam pengembangan aplikasi. Penguasaan C++ sebagai fondasi awal mempermudah mahasiswa untuk mempelajari bahasa pemrograman lain yang memiliki struktur serupa, sekaligus memperdalam pemahaman terhadap konsep algoritma dan struktur data. Penguasaan C++ memberikan fondasi yang kuat bagi pembelajar untuk mengembangkan kemampuan pemrograman mereka ke tingkat yang lebih dalam serta mempermudah adaptasi terhadap bahasa pemrograman lainnya [7].

Dalam konteks pembelajaran pemrograman, pendekatan praktis melalui pembuatan program aplikasi sederhana terbukti efektif untuk membantu pemahaman konsep-konsep fundamental [8]. Program menu interaktif merupakan salah satu bentuk aplikasi yang ideal untuk tujuan pembelajaran karena dapat mengintegrasikan berbagai konsep dasar pemrograman dalam satu kesatuan yang utuh [9]. Melalui implementasi program menu yang mencakup fitur-fitur seperti penjualan, pembelian, dan sewa, pembelajar dapat memahami dan mempraktikkan konsep-konsep penting seperti *input-output*, percabangan, perulangan, serta pengelolaan data dalam konteks yang aplikatif.

Program menu interaktif yang dikembangkan dalam penelitian ini dirancang khusus sebagai media pembelajaran bagi pemula dalam pemrograman C++. Program ini menyajikan antarmuka sederhana namun fungsional yang memungkinkan pengguna untuk berinteraksi dengan berbagai fitur yang telah diimplementasikan. Melalui program ini, pembelajar dapat melatih logika pemrograman mereka secara bertahap, memahami alur kerja program yang terstruktur, serta menerapkan konsep-konsep dasar bahasa C++ dalam situasi yang menyerupai aplikasi nyata. Penguasaan terhadap konsep-konsep dasar ini menjadi bekal penting untuk pengembangan program yang lebih kompleks di masa mendatang.

Pembuatan program menu interaktif ini memiliki beberapa tujuan spesifik yang ingin dicapai. Tujuan pertama adalah menerapkan konsep fungsi (*function*) dalam bahasa pemrograman C++ untuk modularisasi kode program dan meningkatkan efisiensi serta keterbacaan program. Tujuan kedua adalah mengimplementasikan struktur percabangan *switch-case* sebagai mekanisme pemilihan menu yang memungkinkan program untuk menjalankan fungsi-fungsi berbeda berdasarkan pilihan pengguna. Tujuan ketiga adalah menggunakan struktur perulangan *do-while* untuk menciptakan sistem menu yang terus ditampilkan hingga pengguna memilih untuk mengakhiri program, sehingga meningkatkan interaktivitas aplikasi. Selain sebagai media latihan teknis, pengembangan program ini juga berfungsi untuk memperkuat pemahaman terhadap prinsip modularitas dalam pemrograman. Penggunaan fungsi (*function*) memungkinkan pemisahan tugas program ke dalam bagian-bagian yang lebih kecil dan terorganisir, sehingga meningkatkan keterbacaan serta kemudahan pemeliharaan kode. Lebih lanjut, implementasi algoritma perhitungan sederhana, seperti kalkulasi harga, diskon, dan total transaksi, memberikan gambaran konkret mengenai penerapan logika matematis dalam pemrograman. Mahasiswa tidak hanya mempelajari sintaks bahasa, tetapi juga memahami bagaimana algoritma dirancang untuk menyelesaikan permasalahan transaksi secara sistematis dan akurat. Dengan demikian, proses pengembangan program ini tidak hanya berorientasi pada hasil akhir berupa aplikasi yang berjalan, tetapi juga pada proses pembelajaran yang menekankan pemahaman konseptual dan penguatan logika komputasional.

Selain itu, program ini juga bertujuan untuk mengimplementasikan struktur data *array* pada fitur menu pembelian untuk mengelola data produk dan transaksi secara terorganisir. Tujuan terakhir adalah melatih dan mengembangkan kemampuan logika pemrograman melalui implementasi algoritma perhitungan sederhana yang diterapkan dalam berbagai fitur program, seperti kalkulasi harga, diskon, dan total transaksi. Secara keseluruhan, penelitian ini bertujuan untuk merancang dan mengimplementasikan program menu interaktif berbasis C++ sebagai sarana pembelajaran yang aplikatif dan kontekstual. Melalui integrasi berbagai konsep fundamental pemrograman, diharapkan program ini dapat membantu pemula dalam memahami struktur dan alur kerja program secara lebih komprehensif. Selain itu, penelitian ini juga memberikan kontribusi dalam pengembangan metode pembelajaran pemrograman yang lebih sistematis, terarah, dan relevan dengan kebutuhan kompetensi digital di era modern.

2. METODE

Bab ini menjelaskan metodologi yang digunakan dalam pengembangan program menu interaktif berbasis C++. Pembahasan dimulai dengan tinjauan pustaka yang menguraikan konsep-konsep teoritis yang menjadi landasan pengembangan program, meliputi bahasa pemrograman C++, pemrograman terstruktur, dan konsep-konsep dasar yang diterapkan. Selanjutnya, dijelaskan alat dan bahan yang digunakan dalam proses pengembangan, metode pengembangan program yang diterapkan secara sistematis, serta desain program yang menjadi acuan implementasi. Pemahaman terhadap aspek-aspek metodologis ini penting untuk memastikan program dikembangkan secara terstruktur dan memenuhi tujuan pembelajaran yang telah ditetapkan.

2.1 Bahasa Pemrograman C++

Bahasa pemrograman C++ merupakan pengembangan dari bahasa C yang dikembangkan oleh Bjarne Stroustrup pada tahun 1979 di Bell Laboratories [10]. C++ dirancang sebagai bahasa pemrograman yang mendukung paradigma pemrograman berorientasi objek (*object-oriented programming*) sekaligus mempertahankan efisiensi dan fleksibilitas dari bahasa C [10]. Sebagai bahasa pemrograman tingkat menengah, C++ menyediakan kontrol tingkat rendah terhadap memori dan perangkat keras, namun tetap menawarkan abstraksi tingkat tinggi yang memudahkan pengembangan aplikasi kompleks [11]. C++ memiliki beberapa karakteristik utama yang menjadikannya bahasa yang ideal untuk pembelajaran pemrograman. Pertama, C++ menerapkan konsep *strong typing* yang membantu programmer memahami pentingnya tipe data dalam pemrograman [12]. Kedua, C++ mendukung berbagai paradigma pemrograman termasuk prosedural, berorientasi objek, dan *generic programming*, sehingga memberikan fleksibilitas dalam pendekatan penyelesaian masalah [12]. Ketiga, C++ memiliki sintaks yang jelas dan terstruktur yang membantu pembelajar memahami logika pemrograman secara sistematis [13].

2.2 Pemrograman Terstruktur

Pemrograman terstruktur adalah paradigma pemrograman yang menekankan pada penggunaan struktur kontrol yang jelas dan terorganisir untuk meningkatkan kualitas, kejelasan, dan waktu pengembangan program [14]. Konsep pemrograman terstruktur pertama kali dipopulerkan oleh Edsger Dijkstra pada tahun 1960-an sebagai respons terhadap kompleksitas kode program yang tidak terstruktur [15]. Pemrograman terstruktur menggunakan tiga struktur kontrol dasar: sekuensial (berurutan), seleksi (percabangan), dan iterasi (perulangan) [15]. Dalam konteks pembelajaran ini, pemrograman terstruktur membantu mahasiswa dan pelajar untuk mengembangkan kemampuan berpikir logis dan sistematis. Dengan menerapkan prinsip-prinsip pemrograman terstruktur, program menjadi lebih mudah dipahami, dipelihara, dan dikembangkan. Konsep modularisasi melalui fungsi juga merupakan bagian penting dari pemrograman terstruktur yang memungkinkan pemecahan masalah kompleks menjadi bagian-bagian yang lebih sederhana dan dapat dikelola.

2.3 Konsep Dasar Pemrograman C++

Beberapa konsep dasar yang diterapkan dalam program menu interaktif ini meliputi fungsi, percabangan, perulangan, dan struktur data *array* [16]. Fungsi dalam C++ merupakan blok kode yang dirancang untuk melakukan tugas tertentu dan dapat dipanggil berulang kali, sehingga meningkatkan efisiensi dan modularitas kode [17]. Percabangan *switch-case* merupakan struktur kontrol yang memungkinkan program untuk memilih salah satu dari beberapa blok kode untuk dieksekusi berdasarkan nilai dari sebuah ekspresi, sangat efektif untuk implementasi sistem menu [17]. Perulangan *do-while* adalah struktur kontrol yang mengeksekusi blok kode setidaknya sekali, kemudian mengulangnya selama kondisi tertentu terpenuhi [18]. Struktur ini sangat cocok untuk implementasi menu interaktif yang memerlukan tampilan berulang hingga pengguna memilih untuk keluar. *Array* adalah struktur data yang menyimpan kumpulan elemen dengan tipe data yang sama dalam lokasi memori yang berurutan, memungkinkan pengelolaan data yang efisien dalam program [19].

2.4 Alat dan Material

Pengembangan program menu interaktif ini menggunakan beberapa alat dan bahan yang diperlukan untuk proses pemrograman. Perangkat lunak yang digunakan adalah kompiler C++ yang dapat berupa Dev-C++, Code::Blocks dan Visual Studio yang mendukung standar C++ [20]. Sistem operasi yang digunakan adalah berupa Windows, Linux, dan macOS yang kompatibel dengan kompiler yang dipilih. Perangkat keras yang digunakan adalah komputer atau laptop dengan spesifikasi minimal yang mampu menjalankan kompiler C++ dan menampilkan antarmuka konsol.

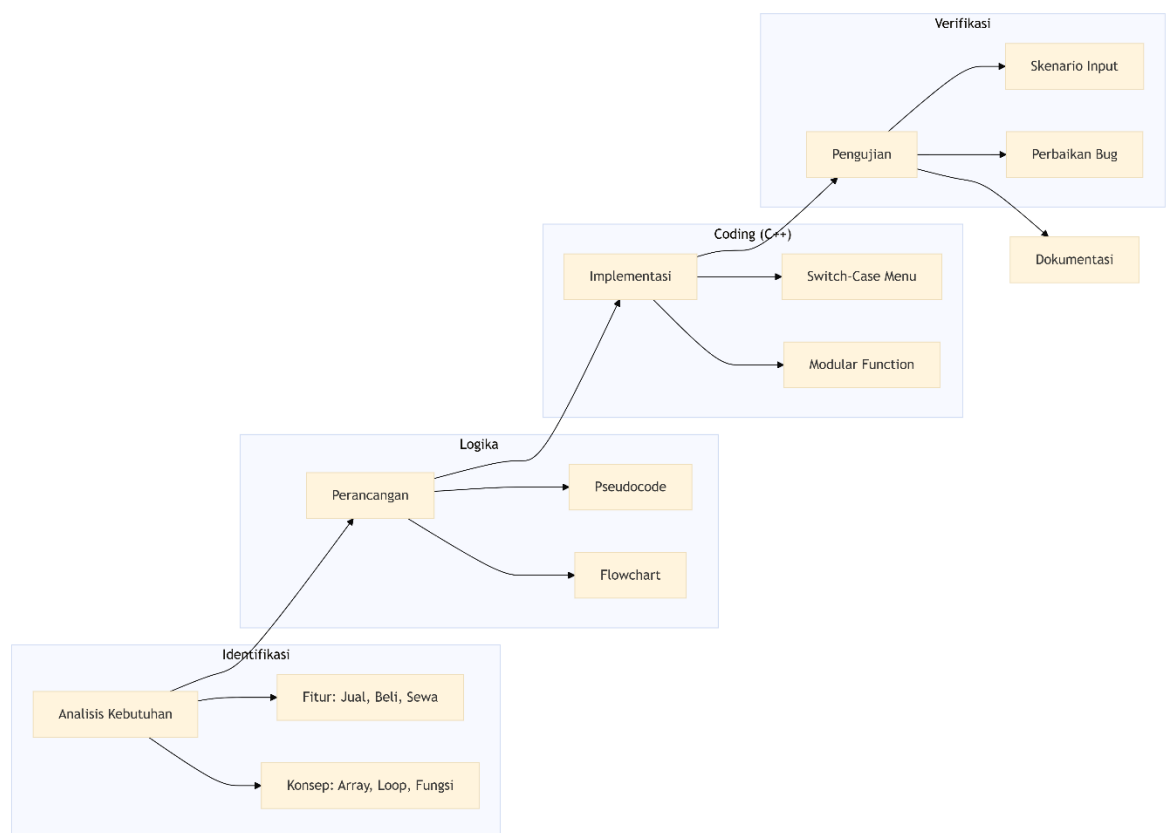


Gambar 1 Logo C++

Bahan yang digunakan dalam pengembangan program ini meliputi kode sumber (*source code*) yang ditulis dalam bahasa C++ dengan ekstensi file .cpp, serta dokumentasi program yang menjelaskan algoritma dan logika program. Selain itu, diperlukan juga literatur pendukung berupa buku teks, tutorial *online*, dan dokumentasi resmi C++ sebagai referensi dalam proses pengembangan program.

2.5 Metode Pengembangan Program

Pengembangan program menu interaktif ini menggunakan metode pengembangan yang sistematis dan terstruktur yang ditunjukkan pada Gambar 2.

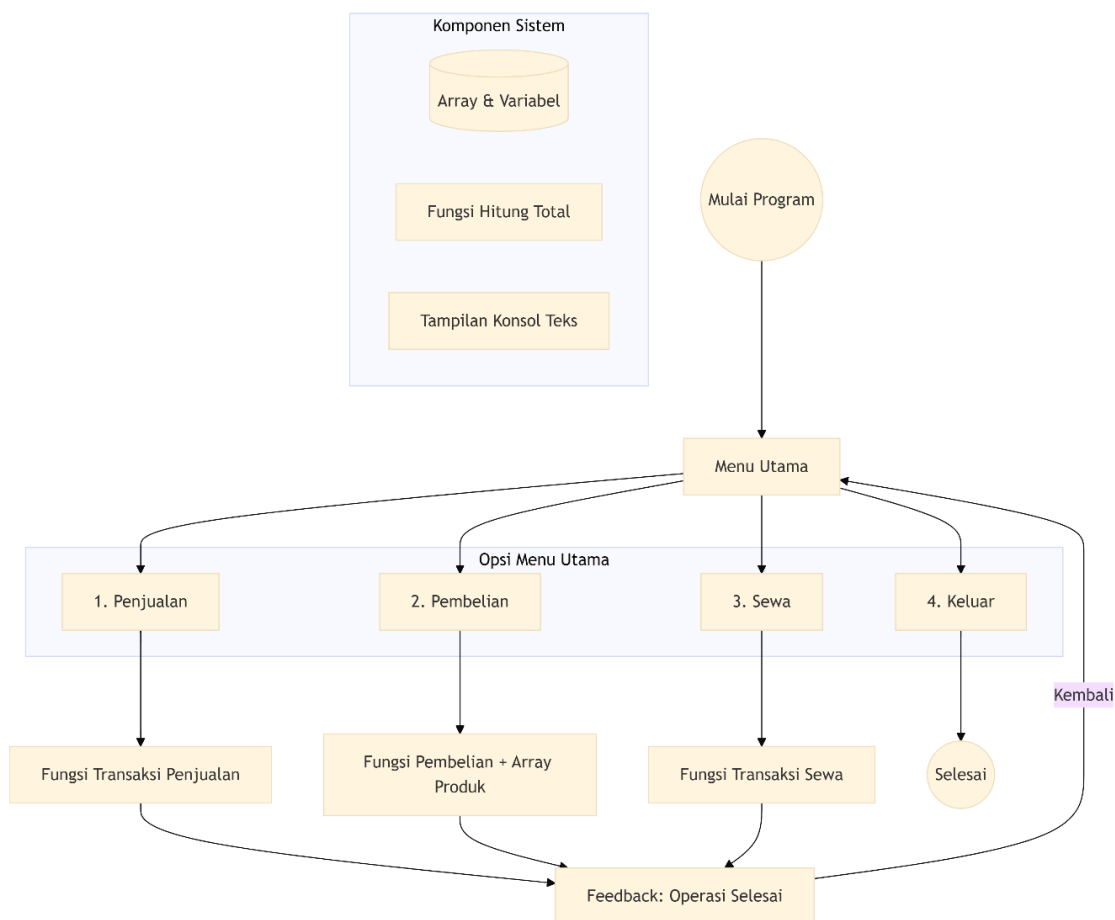


Gambar 2 Diagram Alur Pengembangan Program

Tahap pertama adalah analisis kebutuhan, di mana dilakukan identifikasi fitur-fitur yang akan diimplementasikan dalam program, yaitu menu penjualan, pembelian, dan sewa. Pada tahap ini juga ditentukan konsep-konsep pemrograman yang akan diterapkan, seperti fungsi, percabangan, perulangan, dan *array*. Tahap kedua adalah perancangan algoritma dan struktur program. Pada tahap ini dirancang alur kerja program secara keseluruhan, termasuk struktur menu utama, alur percabangan untuk setiap pilihan menu, dan algoritma perhitungan yang diperlukan pada setiap fitur. Perancangan dilakukan dengan menggunakan *pseudocode* atau *flowchart* untuk memvisualisasikan logika program sebelum diterjemahkan ke dalam kode C++. Tahap ketiga adalah implementasi kode program, di mana algoritma yang telah dirancang diterjemahkan ke dalam bahasa pemrograman C++. Pada tahap ini dilakukan penulisan kode untuk fungsi-fungsi utama, implementasi struktur percabangan *switch-case* untuk menu, penerapan perulangan *do-while* untuk sistem menu berulang, dan penggunaan *array* untuk pengelolaan data produk. Setiap fungsi ditulis secara modular untuk memudahkan pengembangan dan pemeliharaan kode. Tahap keempat adalah pengujian dan *debugging*, di mana program yang telah dibuat diuji untuk memastikan setiap fitur berfungsi dengan baik sesuai dengan rancangan. Pengujian dilakukan dengan mencoba berbagai skenario *input* untuk memverifikasi kebenaran *output* dan mengidentifikasi potensi kesalahan atau *bug* dalam program. Jika ditemukan kesalahan, dilakukan perbaikan kode hingga program berjalan dengan baik. Tahap terakhir adalah dokumentasi, di mana dilakukan pencatatan seluruh proses pengembangan program, penjelasan algoritma, dan panduan penggunaan program. Dokumentasi ini penting sebagai referensi untuk pemahaman program dan pengembangan lebih lanjut di masa mendatang.

2.6 Desain Program

Program menu interaktif ini dirancang dengan struktur hierarkis yang terdiri dari menu utama dan submenu untuk setiap fitur. Menu utama menampilkan pilihan-pilihan yang tersedia, yaitu penjualan, pembelian, sewa, dan keluar dari program. Setiap pilihan menu akan mengarahkan pengguna ke fungsi-fungsi spesifik yang telah didefinisikan.



Gambar 3 Prototype Antarmuka Sistem

Desain antarmuka program menggunakan tampilan berbasis teks pada konsol yang sederhana namun informatif. Setiap menu dilengkapi dengan nomor pilihan yang jelas dan instruksi penggunaan yang mudah dipahami. Program dirancang untuk memberikan umpan balik (*feedback*) kepada pengguna setelah setiap transaksi atau operasi selesai dilakukan, kemudian mengembalikan pengguna ke menu utama untuk melanjutkan atau mengakhiri program. Struktur data yang digunakan dalam program ini meliputi variabel-variabel untuk menyimpan *input* pengguna, *array* untuk menyimpan data produk pada fitur pembelian, serta variabel-variabel untuk menyimpan hasil perhitungan. Setiap fungsi dirancang untuk melakukan tugas spesifik, seperti fungsi untuk menghitung total harga, fungsi untuk menampilkan menu, dan fungsi-fungsi untuk setiap fitur transaksi.

3. PEMBAHASAN HASIL

Bab ini menyajikan pembahasan mendalam mengenai hasil pengembangan program menu interaktif berbasis C++. Pembahasan dimulai dengan deskripsi umum program yang menjelaskan fungsi dan alur kerja program secara keseluruhan. Selanjutnya, diuraikan struktur program yang mencakup fungsi-fungsi utama yang diimplementasikan beserta perannya dalam sistem. Bagian inti dari bab ini adalah penjelasan detail mengenai setiap komponen program, meliputi implementasi fungsi menu, penggunaan struktur percabangan *switch-case*, penerapan perulangan *do-while*, serta implementasi fungsi-fungsi spesifik untuk penjualan, pembelian, dan sewa. Setiap komponen dianalisis dari segi logika pemrograman, algoritma yang diterapkan, dan kontribusinya terhadap pencapaian tujuan pembelajaran.

3.1 Deskripsi Program

Program menu interaktif ini dirancang sebagai aplikasi berbasis konsol yang menyediakan tiga fitur utama transaksi, yaitu penjualan, pembelian, dan sewa. Ketika program dijalankan, pengguna akan disajikan dengan menu utama yang menampilkan empat pilihan, yaitu menu Penjualan, menu Pembelian, menu Sewa, dan menu Keluar. Setiap pilihan menu diberi nomor identifikasi yang memudahkan pengguna untuk melakukan seleksi sesuai dengan kebutuhan transaksi yang diinginkan. Sistem kerja program didasarkan pada interaksi antara pengguna dengan antarmuka konsol melalui *input* dan *output*. Pengguna diminta untuk memasukkan nomor pilihan menu yang diinginkan, kemudian program akan mengeksekusi fungsi yang sesuai dengan pilihan tersebut. Setelah suatu transaksi selesai diproses dan hasilnya ditampilkan, program secara otomatis akan kembali menampilkan menu utama, memberikan kesempatan kepada pengguna untuk melakukan transaksi lain atau memilih untuk mengakhiri program. Program akan terus berjalan dalam mode interaktif hingga pengguna secara eksplisit memilih menu Keluar dengan memasukkan nomor pilihan yang sesuai. Desain ini memastikan bahwa pengguna memiliki kontrol penuh terhadap durasi penggunaan program dan dapat melakukan beberapa transaksi dalam satu sesi eksekusi program tanpa perlu menjalankan ulang aplikasi. Pendekatan ini mencerminkan prinsip *user-friendly* dalam desain antarmuka aplikasi.

3.2 Struktur Program

Program menu interaktif ini dibangun dengan struktur modular yang terdiri dari beberapa fungsi utama yang masing-masing memiliki tugas dan tanggung jawab spesifik. Pendekatan modular ini diterapkan untuk meningkatkan keterbacaan kode, memudahkan proses *debugging*, serta mempermudah pengembangan dan pemeliharaan program di masa mendatang. Struktur program secara hierarkis terdiri dari lima fungsi utama yang saling berinteraksi membentuk sistem yang kohesif. Fungsi pertama adalah `Menu()`, yang bertugas untuk menampilkan daftar pilihan menu kepada pengguna. Fungsi ini bersifat independen dan dapat dipanggil berulang kali untuk menampilkan antarmuka menu utama. Dengan memisahkan tampilan menu ke dalam fungsi tersendiri, modifikasi terhadap tampilan menu dapat dilakukan dengan mudah tanpa mempengaruhi bagian program lainnya. Fungsi kedua adalah `Penjualan()`, yang mengimplementasikan logika perhitungan untuk transaksi penjualan barang. Fungsi ini menerima *input* dari pengguna berupa harga satuan barang dan jumlah barang yang dijual, kemudian melakukan kalkulasi untuk menghasilkan total nilai penjualan. Hasil perhitungan ditampilkan kepada pengguna dengan format yang jelas dan informatif.

Tabel 1 Cuplikan Kode Struktur Utama Program

main.cpp
<pre>void menu() { cout << "\n===== MENU UTAMA =====" << endl; cout << "1. Penjualan" << endl;</pre>

```

    cout << "2. Pembelian" << endl;
    cout << "3. Sewa" << endl;
    cout << "4. Keluar" << endl;
}

void penjualan() {
    int harga, jumlah, total;

    cout << "\n--- MENU PENJUALAN ---" << endl;
    cout << "Masukkan harga barang : ";
    cin >> harga;
    cout << "Masukkan jumlah barang: ";
    cin >> jumlah;

    total = harga * jumlah;

    cout << "Total penjualan: Rp " << total << endl;
}

void pembelian() {
    int harga[10];
    int jumlah, total = 0;

    cout << "\n--- MENU PEMBELIAN ---" << endl;
    cout << "Masukkan jumlah barang yang dibeli: ";
    cin >> jumlah;

    for (int i = 0; i < jumlah; i++) {
        cout << "Masukkan harga barang ke-" << i + 1 << ": ";
        cin >> harga[i];
        total += harga[i];
    }

    cout << "Total pembelian: Rp " << total << endl;
}

void sewa() {
    int lama;
    const int hargaPerHari = 50000;
    int total;

    cout << "\n--- MENU SEWA ---" << endl;
    cout << "Masukkan lama sewa (hari): ";
    cin >> lama;

    total = lama * hargaPerHari;

    cout << "Total biaya sewa: Rp " << total << endl;
}

```

Fungsi ketiga adalah `Pembelian()`, yang dirancang untuk mengelola transaksi pembelian dengan memanfaatkan struktur data *array*. Fungsi ini memungkinkan pengguna untuk memasukkan informasi pembelian beberapa barang sekaligus, di mana setiap harga barang disimpan dalam elemen *array*. Penggunaan *array* memungkinkan pengelolaan data yang lebih terorganisir dan efisien, terutama ketika jumlah item yang dibeli cukup banyak. Fungsi keempat adalah `Sewa()`, yang mengimplementasikan algoritma perhitungan biaya sewa berdasarkan durasi waktu penyewaan. Fungsi ini menghitung total biaya dengan mengalikan lama sewa dalam satuan hari dengan tarif sewa per hari yang telah ditentukan. Implementasi fungsi ini mendemonstrasikan penerapan operasi aritmatika sederhana dalam konteks aplikasi nyata. Fungsi kelima dan yang paling penting

adalah `Main()`, yang berfungsi sebagai kontroler utama yang mengatur alur eksekusi keseluruhan program. Fungsi `main()` mengintegrasikan semua fungsi lainnya dan mengimplementasikan logika percabangan serta perulangan yang mengatur interaksi antara pengguna dengan berbagai fitur program. Fungsi ini bertindak sebagai inti koordinasi yang memastikan setiap komponen program bekerja secara harmonis.

3.3 Penjelasan Detail Komponen Program

Program menu interaktif ini dibangun berdasarkan beberapa konsep fundamental pemrograman C++ yang diimplementasikan secara sistematis. Setiap komponen program dirancang untuk mendemonstrasikan penerapan konsep-konsep tersebut dalam konteks aplikasi praktis yang dapat dipahami oleh pembelajar pemula.

3.3.1 Implementasi Fungsi Menu

Fungsi `Menu()` merupakan komponen penting yang bertanggung jawab untuk menampilkan antarmuka utama program kepada pengguna. Fungsi ini didesain sebagai fungsi `void` yang tidak menerima parameter dan tidak mengembalikan nilai, melainkan berfokus pada tugas menampilkan *output* ke layar konsol. Penggunaan fungsi untuk menampilkan menu memberikan beberapa keuntungan signifikan dalam hal efisiensi dan pemeliharaan kode program. Dengan memisahkan tampilan menu ke dalam fungsi tersendiri, kode untuk menampilkan menu tidak perlu dituliskan berulang kali di berbagai bagian program. Fungsi `Menu()` dapat dipanggil kapan saja ketika program memerlukan tampilan menu, sehingga menerapkan prinsip DRY (*Don't Repeat Yourself*) dalam pemrograman. Jika di masa mendatang diperlukan perubahan pada tampilan menu, modifikasi hanya perlu dilakukan di satu tempat yaitu dalam fungsi `Menu()` itu sendiri, tanpa harus mencari dan mengubah kode di berbagai lokasi. Implementasi fungsi `Menu()` juga mendemonstrasikan konsep modularisasi dalam pemrograman terstruktur. Dengan memecah program menjadi fungsi-fungsi yang lebih kecil dan fokus pada tugas spesifik, kompleksitas program secara keseluruhan dapat dikurangi. Setiap fungsi menjadi lebih mudah untuk dipahami, diuji, dan diperbaiki jika terjadi kesalahan. Pendekatan modular ini merupakan praktik terbaik yang penting untuk dikuasai oleh pembelajar pemrograman.

3.3.2 Implementasi Perulangan *Do-While* dan Percabangan *Switch-Case*

Perulangan *do-while* dipilih sebagai struktur kontrol untuk mengimplementasikan sistem menu yang berulang dalam program ini. Karakteristik utama dari perulangan *do-while* adalah bahwa blok kode di dalamnya akan dieksekusi minimal satu kali sebelum kondisi perulangan dievaluasi, berbeda dengan perulangan *while* biasa yang mengevaluasi kondisi terlebih dahulu. Dalam konteks program menu, karakteristik ini sangat ideal karena menu harus ditampilkan setidaknya sekali kepada pengguna ketika program pertama kali dijalankan, terlepas dari kondisi apapun. Setelah eksekusi pertama, perulangan akan terus berlanjut selama kondisi yang ditetapkan bernilai *true*, yaitu selama pengguna belum memilih menu Keluar. Struktur percabangan *switch-case* diintegrasikan dalam perulangan *do-while* sebagai mekanisme seleksi untuk menentukan tindakan yang akan dieksekusi berdasarkan pilihan menu yang dimasukkan oleh pengguna. Pemilihan struktur *switch-case* dibandingkan dengan alternatif percabangan *if-else* didasarkan pada keterbacaan kode yang lebih rapi dan terorganisir ketika harus menangani banyak kondisi percabangan yang berbeda. Setiap *case statement* dengan jelas menunjukkan aksi yang akan dilakukan untuk setiap pilihan menu tertentu, dan penggunaan *case default* untuk menangani *input* yang tidak valid mendemonstrasikan praktik pemrograman yang robust dengan validasi *input*. Penggunaan kombinasi *do-while* dan *switch-case* dalam konteks ini mengajarkan konsep penting tentang kontrol alur program dan manajemen state aplikasi. Pembelajar dapat memahami bagaimana perulangan dapat digunakan untuk menciptakan aplikasi yang responsif dan interaktif, di mana program tetap aktif dan siap menerima *input* hingga pengguna secara eksplisit memutuskan untuk mengakhiri sesi. Konsep ini fundamental dalam pengembangan aplikasi interaktif yang user-friendly, di mana *switch-case* memiliki performa yang lebih baik untuk percabangan dengan banyak kondisi karena kompiler dapat mengoptimalkan eksekusinya menggunakan *jump table*.

Tabel 2 Implementasi Switch-Case

main.cpp
<pre>int main() { int pilihan; do { menu(); cout << "Pilih menu: "; cin >> pilihan;</pre>

```

switch (pilihan) {
    case 1:
        penjualan();
        break;
    case 2:
        pembelian();
        break;
    case 3:
        sewa();
        break;
    case 4:
        cout << "Terima kasih, program selesai." << endl;
        break;
    default:
        cout << "Pilihan tidak valid!" << endl;
}

} while (pilihan != 4);

return 0;
}

```

3.3.4 Implementasi Fungsi-Fungsi Transaksi

Fungsi `Penjualan()` mengimplementasikan logika perhitungan untuk transaksi penjualan barang dengan algoritma yang mudah namun mendemonstrasikan konsep-konsep penting dalam pemrograman. Proses dimulai dengan meminta pengguna memasukkan dua data *input*, yaitu harga satuan barang dan jumlah barang yang dijual, yang disimpan dalam variabel dengan tipe data yang sesuai. Setelah *input* diperoleh, fungsi melakukan operasi aritmatika perkalian dengan formula:

$$\text{Total Penjualan} = \text{Harga Satuan} \times \text{Jumlah Barang} \quad (1)$$

Fungsi kemudian menampilkan hasil kepada pengguna dalam format yang informatif seperti "Total Penjualan: Rp xxxxx". Implementasi *output* yang informatif ini mengajarkan pentingnya user experience dalam pengembangan aplikasi, bahkan untuk program sederhana sekalipun. Fungsi `Pembelian()` merupakan implementasi yang lebih kompleks karena melibatkan penggunaan struktur data *array* untuk mengelola data. Proses dimulai dengan meminta pengguna memasukkan jumlah barang yang akan dibeli, kemudian menggunakan struktur perulangan *for* untuk meminta pengguna memasukkan harga setiap barang satu per satu yang disimpan dalam elemen *array* berbeda dengan indeks yang bersesuaian. Setelah semua harga barang tersimpan, fungsi melakukan iterasi melalui seluruh elemen *array* untuk menghitung total pembelian menggunakan variabel akumulator yang diinisialisasi dengan nilai nol. Penggunaan *array* dalam fungsi ini memberikan pembelajaran penting tentang cara mendeklarasikan dan menggunakan *array*, mengakses elemen menggunakan indeks, serta aplikasi praktis *array* dalam pengelolaan data transaksi pembelian multiple items.

Fungsi `Sewa()` mengimplementasikan algoritma perhitungan biaya sewa berdasarkan durasi waktu penyewaan dengan tarif yang telah ditetapkan sebesar Rp 50.000 per hari. Penggunaan konstanta untuk menyimpan tarif sewa merupakan praktik pemrograman yang baik karena meningkatkan keterbacaan kode dan memudahkan pemeliharaan. Fungsi meminta pengguna memasukkan lama sewa dalam satuan hari yang disimpan dalam variabel integer, kemudian melakukan operasi perkalian untuk menghitung total biaya sewa dengan formula:

$$\text{Total Biaya} = \text{Lama Sewa (Hari)} \times \text{Harga Sewa per Hari} \quad (2)$$

Hasil perhitungan ditampilkan dengan format informatif yang mencakup detail lama sewa, tarif per hari, dan total biaya yang harus dibayarkan, memberikan transparansi dalam perhitungan dan mendemonstrasikan penerapan konsep konstanta, operasi aritmatika, dan *formatting output* dalam konteks aplikasi perhitungan biaya yang sering dijumpai dalam kehidupan sehari-hari.

4. KESIMPULAN

Program menu penjualan, pembelian, dan sewa ini merupakan contoh implementasi sederhana namun komprehensif dari berbagai konsep dasar bahasa pemrograman C++ yang telah diintegrasikan dalam satu aplikasi interaktif. Melalui program ini, diterapkan beberapa konsep fundamental pemrograman terstruktur, meliputi penggunaan fungsi untuk modularisasi kode, struktur percabangan *switch-case* untuk seleksi menu, perulangan *do-while* untuk menciptakan sistem menu yang berulang, serta implementasi *array* untuk pengelolaan data. Selain itu, program ini juga mendemonstrasikan penggunaan operasi *input-output* yang merupakan aspek esensial dalam interaksi antara program dengan pengguna. Kombinasi dari berbagai konsep ini menghasilkan sebuah aplikasi menu yang interaktif, terstruktur, dan fungsional yang dapat menjadi referensi pembelajaran bagi pemula dalam memahami bagaimana konsep-konsep teoritis diterapkan dalam konteks aplikasi nyata. Dari aspek fungsionalitas, program ini berhasil mengimplementasikan kemampuan perhitungan dasar pada setiap fitur menu yang tersedia dengan logika yang benar dan algoritma yang efisien. Pada menu penjualan, program mampu menghitung total nilai penjualan berdasarkan harga satuan dan jumlah barang dengan akurat. Pada menu pembelian, program dapat mengelola data dengan menggunakan struktur *array* dan menghitung total pembelian dari seluruh barang yang diinputkan. Sementara pada menu sewa, program mampu menghitung biaya sewa berdasarkan durasi waktu dan tarif yang telah ditetapkan. Struktur modular program yang membagi fungsionalitas ke dalam fungsi-fungsi terpisah memberikan keuntungan signifikan dalam hal keterbacaan kode, kemudahan pemahaman alur logika program, dan fleksibilitas untuk pengembangan lebih lanjut. Setiap fungsi memiliki tanggung jawab yang jelas dan spesifik, memudahkan proses *debugging* dan pemeliharaan kode di masa mendatang.

Meskipun program ini tergolong sederhana dalam hal kompleksitas dan fitur, program ini memiliki potensi pengembangan yang sangat luas untuk ditingkatkan menjadi aplikasi yang lebih *robust* dan profesional. Beberapa aspek yang dapat dikembangkan meliputi penambahan validasi *input* untuk memastikan data yang dimasukkan pengguna sesuai dengan format yang diharapkan, implementasi sistem penyimpanan data menggunakan *file* atau *database* untuk persistensi data transaksi, pengembangan antarmuka pengguna yang lebih menarik dan *user-friendly* dengan memanfaatkan *library* grafis atau *framework*, serta penambahan fitur-fitur tambahan seperti sistem diskon, perhitungan pajak, atau laporan transaksi. Terlepas dari kesederhanaannya, program ini telah memenuhi tujuan utamanya sebagai media pembelajaran yang efektif untuk memahami dan menguasai konsep-konsep dasar pemrograman C++. Secara keseluruhan, program ini sangat cocok digunakan sebagai titik mulai bagi pembelajar pemula untuk membangun fondasi pemahaman pemrograman terstruktur yang solid sebelum melangkah ke topik-topik pemrograman yang lebih mendalam.

REFERENCES

- [1] I. Wahyuni Firlir Fangestu and H. Syahrizal, "Digitalisasi Lembaga Pendidikan dalam Menghadapi Perkembangan dan Kemajuan Teknologi Informasi Dunia Pendidikan," *Al-Zayn J. Ilmu Sos. Huk.*, vol. 1, no. 2, pp. 26–38, Dec. 2023, doi: 10.61104/alz.v1i2.89.
- [2] Y. Belo and R. S., "ETIKA KOMUNIKASI GURU PENDIDIKAN AGAMA KRISTEN DI TENGAH KEMAJUAN TEKNOLOGI INFORMASI DAN KOMUNIKASI," *BONAFIDE J. Teol. Dan Pendidik. Kristen*, vol. 4, no. 1, pp. 128–146, Jun. 2023, doi: 10.46558/bonafide.v4i1.157.
- [3] E. Fauzi, M. Visar Sinatrya, N. Daru Ramdhani, R. Ramadhan, and Z. Muhammad Rasid Safari, "Pengaruh kemajuan teknologi informasi terhadap perkembangan akuntansi," *J. Ris. Pendidik. Ekon.*, vol. 7, no. 2, pp. 189–197, Oct. 2022, doi: 10.21067/jrpe.v7i2.6877.
- [4] Dewi Fitriah Khusnul Khotimah and Hafidz, "Pengembangan Media Pembelajaran Kreatif Dan Interaktif Melalui Teknologi Informasi dan Komunikasi (TIK) Dalam Pendidikan," *J. IHSAN J. Pendidik. Islam*, vol. 3, no. 1, pp. 1–8, Jan. 2025, doi: 10.61104/ihsan.v3i1.433.
- [5] A. Kristiyanto *et al.*, "Pelatihan Pemrograman C++ Melalui Tinkercad Guna Meningkatkan Kemampuan Computational Thinking Siswa SMK 11 Kab. Tangerang," *J. Pengabd. Kpd. Masy. Nusantara*, vol. 5, no. 2, pp. 2776–2782, Jun. 2024, doi: 10.55338/jpkmn.v5i2.3362.
- [6] R. Firdaus, D. Rinaldi, H. D. Septama, and S. Maulana, "Pengembangan e-Modul Pemrograman Dasar C++ dalam Upaya Meningkatkan Level of Understanding Siswa," *J. PPS-TP J. Pendidik. Anal. Apl. Teori Dan Has. Penelit.*, vol. 12, no. 2, pp. 17–33, Oct. 2024.
- [7] R. A. Putra, *Buku Sakti Pemrograman C++ Untuk Pemula*. Anak Hebat Indonesia, 2024.
- [8] J. J. Siang, L. D. Krisnawati, Y. Lukito, L. Chrisantyo, and R. G. Santoso, "Pelatihan Algoritma Pemrograman Untuk Persiapan OSN Informatika Siswa SMA Kolose de Britto Yogyakarta," *Prima Abdika J. Pengabd. Masy.*, vol. 5, no. 4, pp. 637–643, Nov. 2025, doi: 10.37478/abdika.v5i4.6897.

-
- [9] N. I. Prasetya and S. Syidada, "Integrasi Serious Games dalam Pengembangan Platform Pembelajaran Pemrograman," *Decode J. Pendidik. Teknol. Inf.*, vol. 5, no. 1, pp. 12–26, Mar. 2025, doi: 10.51454/decode.v5i1.864.
 - [10] B. Stroustrup, *A Tour of C++*. Addison-Wesley Professional, 2022.
 - [11] A. Nazri, H. Liyana, I. Hasanah, and Z. Ibrahim, "Tales of C++ Worlds: C++ Programming Language Game-Based Learning," *Multidiscip. Appl. Res. Innov.*, vol. 5, no. 1, pp. 223–231, Jan. 2024.
 - [12] J. Horton, *Beginning C++ Game Programming: Learn C++ from scratch by building fun games*. Packt Publishing Ltd, 2024.
 - [13] "CodeToProtect©: C++ Programming Language Video Game for Teaching Higher Education Learners - IOPscience." Accessed: Feb. 05, 2026. [Online]. Available: <https://iopscience.iop.org/article/10.1088/1742-6596/1874/1/012064/meta>
 - [14] B. N. Chandrashekhar and H. A. Sanjay, "Performance Analysis of Sequential and Parallel Programming Paradigms on CPU-GPUs Cluster," in *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, Feb. 2021, pp. 1205–1213. doi: 10.1109/ICICV50876.2021.9388469.
 - [15] L. de Moura and S. Ullrich, "The Lean 4 Theorem Prover and Programming Language," in *Automated Deduction – CADE 28*, A. Platzer and G. Sutcliffe, Eds., Cham: Springer International Publishing, 2021, pp. 625–635. doi: 10.1007/978-3-030-79876-5_37.
 - [16] R. Logožar, M. Mikac, and D. Radošević, "Exploring the Access to the Static Array Elements via Indices and via Pointers — the Introductory C++ Case Expanded," *J. Inf. Organ. Sci.*, vol. 48, no. 1, pp. 49–80, Jun. 2024, doi: 10.31341/jios.48.1.3.
 - [17] S. Anam, I. Yanti, Z. Fitriah, and U. Habibah, *Cara Mudah Belajar Bahasa Pemrograman C++*. Universitas Brawijaya Press, 2021.
 - [18] M. Gregoire, *Professional C++*. John Wiley & Sons, 2021.
 - [19] J. Reinders, B. Ashbaugh, J. Brodman, M. Kinsner, J. Pennycook, and X. Tian, *Data Parallel C++: Mastering DPC++ for Programming of Heterogeneous Systems using C++ and SYCL*. Springer Nature, 2021. doi: 10.1007/978-1-4842-5574-2.
 - [20] H. Tu *et al.*, "Detecting C++ Compiler Front-End Bugs via Grammar Mutation and Differential Testing," *IEEE Trans. Reliab.*, vol. 72, no. 1, pp. 343–357, Mar. 2023, doi: 10.1109/TR.2022.3171220.