

Transformasi Digital dalam Manajemen Barang: Membangun Program Efisien dengan C++

Daffa Syauqi Raihan (4111422034)

Prodi Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Semarang

E-mail : daffasyauqi2705@students.unnes.ac.id

ABSTRAK

Program yang diimplementasikan menggunakan C++ mencerminkan transformasi digital dalam manajemen barang. Dengan fokus pada efisiensi, program ini memberikan solusi terintegrasi untuk toko material bangunan. Pendefinisian struktur data, tampilan yang terstruktur, kemampuan penambahan dan pencarian barang, serta fitur penghapusan memastikan program dapat digunakan dengan fleksibilitas dan responsibilitas terhadap perubahan dalam persediaan. Dalam era di mana informasi cepat diperlukan, program ini membantu mengoptimalkan manajemen barang dengan pendekatan yang efisien dan terkomputerisasi.

Keyword : Transformasi Digital, Manajemen Barang, C++ (Bahasa Pemrograman)

Pendahuluan

Meskipun teknologi komputer telah ada selama beberapa dekade, konsep transformasi digital relatif baru. Konsep ini hadir pada 1990-an dengan diperkenalkannya internet mainstream. Sejak itu, kemampuan untuk mengubah bentuk tradisional media (seperti dokumen dan foto) menjadi satu dan nol telah memudar di tengah pentingnya hal yang dibawa oleh teknologi digital kepada masyarakat. Hari ini, digitalisasi menyentuh setiap bagian dari kehidupan kita,

memengaruhi cara kita bekerja, berbelanja, bepergian, mendidik, mengelola, dan hidup (Wijayanto & Harsadi, 2017).

Praktik transformasi digital biasanya digunakan dalam konteks bisnis. Pengenalan teknologi digital telah memicu penciptaan model bisnis baru dan aliran pendapatan. Teknologi yang muncul seperti kecerdasan buatan (AI), cloud computing dan Internet of Things (IoT) mempercepat transformasi, sementara teknologi dasar seperti manajemen data dan analitik diperlukan untuk menganalisis sejumlah besar data yang dihasilkan dari transformasi digital. Transformasi digital bukan hanya tentang teknologi. Itu terjadi di persimpangan orang, bisnis dan teknologi dan dipandu oleh strategi bisnis yang luas. Sukses hadir ketika organisasi dapat secara efektif menggunakan yang dibuat oleh atau melalui teknologi dengan cara yang memungkinkan perubahan bisnis terjadi secara dinamis (Wijayanto & Harsadi, 2017).

Dalam era digital yang terus berkembang, transformasi digital menjadi suatu keharusan, terutama dalam mengoptimalkan proses manajemen barang. Salah satu langkah konkrit dalam mencapai hal tersebut adalah dengan mengembangkan program berbasis C++ yang mampu memberikan solusi efisien dan terintegrasi. Artikel ini akan membahas secara mendalam langkah-langkah implementasi program tersebut dan bagaimana hal tersebut dapat mendukung manajemen barang yang efisien.

Manajemen barang yang baik sangat penting dalam menjaga kelancaran operasional suatu bisnis, terutama di sektor material bangunan. Seiring dengan kemajuan teknologi, penggunaan program komputer dapat membantu menciptakan sistem manajemen barang yang lebih terstruktur dan terorganisir.

Transformasi digital dalam konteks manajemen barang melibatkan penerapan teknologi informasi untuk meningkatkan efisiensi, ketepatan, dan keterukuran dalam pengelolaan stok dan proses logistik. Ini dapat mencakup otomatisasi, analisis data, dan integrasi sistem. Otomatisasi proses inventaris menggunakan sensor, RFID, dan teknologi terkini adalah inti dari transformasi

digital. Penerapan ini dapat mengurangi kesalahan manusia, meningkatkan akurasi, dan memastikan pembaruan real-time (Lee & Lapira, 2019).

Landasan Teori

Manajemen barang yang efisien menjadi krusial dalam menghadapi persaingan bisnis yang semakin ketat. Perusahaan perlu mengintegrasikan sistem yang dapat memberikan visibilitas maksimal terhadap inventaris, meminimalkan risiko kesalahan manusia, dan meningkatkan respons terhadap perubahan pasar. Transformasi digital dalam manajemen barang tidak hanya berarti menggantikan proses manual dengan teknologi, tetapi juga menciptakan ekosistem di mana data dapat bergerak dengan mulus antara berbagai sistem. Hal ini mencakup penggunaan perangkat keras dan perangkat lunak terbaru, seperti RFID, sensor, dan program komputer yang efisien (Syamil dkk, 2023).

Database management system atau Sistem Manajemen Basis Data adalah sebuah sistem perangkat lunak yang memungkinkan pengguna untuk dapat mendefinisikan, membuat, memelihara dan menyediakan akses kontrol ke dalam database. Teknologi Sistem Manajemen Basis Data sangat beragam, contohnya Oracle RDBMS, MySQL, SQLite, SQL Server, dan lainnya (Hamdallah dkk, 2020). Penggunaan sistem manajemen basis data (DBMS) membantu menyimpan dan mengelola data inventaris dengan efisien. C++ menyediakan dukungan kuat untuk interaksi dengan basis data, seperti MySQL atau SQLite (Elmasri & Navathe, 2016).

C++ adalah bahasa pemrograman yang kuat dan efisien yang digunakan dalam berbagai aplikasi, mulai dari pembuatan sistem operasi hingga perangkat lunak bisnis. Dengan dukungan terhadap pemrograman berorientasi objek, C++ memungkinkan pengguna untuk memodelkan dunia nyata ke dalam kode program yang terstruktur dan modular (cplusplus.com, n.d.).

Kemampuan C++ untuk pemrosesan real-time mendukung pembangunan program yang memberikan informasi terkini tentang status inventaris,

memungkinkan pengambilan keputusan yang lebih cepat. Membangun program efisien dengan menggunakan bahasa pemrograman C++ menjadi langkah kritis dalam mewujudkan transformasi digital dalam manajemen barang. C++ memberikan keunggulan dalam pemrosesan real-time, manajemen memori yang efisien, dan dukungan yang kuat untuk pengembangan aplikasi desktop (Schach, 2014). Selain itu, C++ dapat digunakan untuk mengimplementasikan fitur-fitur keamanan seperti enkripsi data dan manajemen akses (Stallings, 2017).

Pemilihan struktur data dan algoritma yang tepat dalam pengembangan program sangat penting untuk meningkatkan kinerja. C++ menyediakan berbagai struktur data dan algoritma yang dapat dioptimalkan. Pembangunan antarmuka pengguna yang intuitif dapat meningkatkan pengalaman pengguna. C++ dapat digunakan untuk mengembangkan aplikasi desktop dengan antarmuka grafis yang ramah pengguna (Eckel, 2003).

C++ dapat digunakan untuk mengintegrasikan berbagai sistem dan menggunakan API untuk memastikan aliran data yang mulus antara komponen sistem yang berbeda (Josuttis, 2007). Menggunakan analisis data dan teknik pembelajaran mesin dapat membantu dalam meramalkan permintaan, mengidentifikasi tren, dan meningkatkan kebijakan persediaan (James et, al., 2013).

Struktur Program

Program ini menggunakan bahasa pemrograman C++ dan mengadopsi struktur data dalam bentuk array dari struktur Barang. Setiap elemen array mewakili suatu barang dengan atribut seperti kode barang, nama barang, jumlah stok, dan lokasi penyimpanan. Program ini memanfaatkan beberapa fungsi utama:

1. **displayData():** Menampilkan data barang dalam bentuk tabel. Pengguna memiliki opsi untuk mengurutkan data berdasarkan kode, nama, atau jumlah stok.

2. **tambahBarang()**: Menambahkan data barang baru ke dalam array. Jika barang sudah ada, program akan menambahkan stoknya.
3. **carinama()**: Mencari barang berdasarkan nama dan menampilkan hasil pencarian.
4. **hapusBarang()**: Menghapus data barang berdasarkan kode barang.
5. **main()**: Menampilkan menu utama yang memungkinkan pengguna untuk melakukan update data barang, menampilkan data, mencari data, menghapus data, atau keluar dari program.

Langkah-Langkah Implementasi

1. Pendefinisian Struktur Data

Pada langkah pertama ini, dilakukan pendefinisian struktur data yang disebut "Barang". Struktur data ini memiliki empat atribut yaitu kode, nama, stock, dan loker. Tujuannya adalah untuk menyediakan kerangka kerja yang jelas untuk menyimpan informasi setiap barang yang ada di dalam program.

Struktur data ini memungkinkan program untuk menyimpan detail-detail penting dari setiap barang, seperti kode unik, nama barang, jumlah stok, dan lokasi penyimpanan.

```

const int MAX_BARANG = 100;

struct Barang {
    string kode;
    string nama;
    int stock;
    string loker;
};

int i, pil_urutan;
int n = 17;

Barang daftarBarang[MAX_BARANG] = {
    {"p45", "kaca polos", 9, "A"},
    {"n88", "kaca tempered", 7, "H"},
    {"j12", "kaca antirecah", 18, "U"},
    {"j765", "batako Trass", 90, "E"},
    {"o612", "gypsum knauf", 111, "A"},
    {"i12", "batako Press", 222, "J"},
    {"m64", "blockboard", 3, "J"},
    {"h666", "melaminto", 12, "K"},
    {"h812", "partikel board", 2, "O"},
    {"w472", "bata merah", 34, "O"},
    {"s421", "bata roster", 54, "A"},
    {"s45", "batako Beton", 33, "U"},
    {"w124", "gypsum alabaster", 12, "A"},
    {"x827", "basik putih", 233, "I"},
    {"s200", "cat epoxy", 45, "O"},
    {"c800", "batatan", 75, "A"},
    {"g28", "marmex", 7, "O"}
};

```

2. Tampilan Data Barang

Langkah kedua mencakup pembuatan fungsi displayData(), yang bertanggung jawab untuk menampilkan data barang dalam bentuk tabel. Pengguna dapat memilih opsi pengurutan yang diinginkan, sehingga informasi ditampilkan sesuai dengan preferensi mereka.

Fungsi ini memudahkan pengguna untuk dengan cepat melihat daftar barang yang tersedia beserta detailnya. Dengan opsi pengurutan, pengguna dapat mengakses informasi dengan lebih terstruktur dan sesuai dengan kebutuhan mereka.

```
void displayData() {
    bool inginUrutkan=true;
    do {
        cout << "Daftar barang: " << endl;
        cout << setw(10) << "Kode" << setw(20) << "Nama Barang" << setw(10) << "Stock" << setw(15) << "Lokasi" << endl;
        for (int i = 0; i < n; i++) {
            cout << setw(10) << daftarBarang[i].kode << setw(20) << daftarBarang[i].nama << setw(10) << daftarBarang[i].stock << setw(15) << daftarBarang[i].loker << endl;
        }
        cout << endl;

        // Meminta input dari user
        cout << "Apakah anda mauurutkan data? (1:Ya / 0:Tidak) ";

        cin >> inginUrutkan;

        if (inginUrutkan) {
            // Meminta pilihan pengurutan dari user
            int pil_urutan;
            cout << "Urut berdasarkan : " << endl;
            cout << "1. Kode" << endl;
            cout << "2. Nama Barang" << endl;
            cout << "3. Stock" << endl;
            cout << "Pilihan : ";
            cin >> pil_urutan;

            // Proses pengurutan
            switch (pil_urutan) {
                case 1:
                    for (int i = 0; i < n; i++) {
                        for (int j = i + 1; j < n; j++) {
                            if (daftarBarang[i].kode > daftarBarang[j].kode) {
                                // Tukar nilai menggunakan variabel temp
                                Barang temp = daftarBarang[i];
                                daftarBarang[i] = daftarBarang[j];
                                daftarBarang[j] = temp;
                            }
                        }
                    }
                    break;

                case 2:
                    for (int i = 0; i < n; i++) {
                        for (int j = i + 1; j < n; j++) {
                            if (daftarBarang[i].nama > daftarBarang[j].nama) {
                                // Tukar nilai menggunakan variabel temp
                                Barang temp = daftarBarang[i];
                                daftarBarang[i] = daftarBarang[j];
                                daftarBarang[j] = temp;
                            }
                        }
                    }
                    break;

                case 3:
                    for (int i = 0; i < n; i++) {
                        for (int j = i + 1; j < n; j++) {
                            if (daftarBarang[i].stock > daftarBarang[j].stock) {
                                // Tukar nilai menggunakan variabel temp
                                Barang temp = daftarBarang[i];
                                daftarBarang[i] = daftarBarang[j];
                                daftarBarang[j] = temp;
                            }
                        }
                    }
                    break;

                default:
                    cout << "Pilihan salah!" << endl;
                    return;
            }
        }
        else {
            // User tidak ingin mengurutkan data
            cout << "Tidak melakukan pengurutan data." << endl;
            break;
        }
    } while (!inginUrutkan);
}
```

3. Penambahan Barang

Fungsi `tambahBarang()` merupakan langkah ketiga, yang memberikan kemampuan kepada pengguna untuk menambahkan barang baru atau menambah stok barang yang sudah ada. Proses ini dirancang untuk memberikan fleksibilitas dalam mengelola perubahan dalam stok barang.

Dengan adanya fitur ini, program dapat memperbarui dan menyesuaikan stok barang sesuai dengan kebutuhan dan perubahan dalam persediaan.

```
void tambahBarang() {
    if (n >= MAX_BARANG) {
        cout << "Data barang sudah penuh!" << endl;
        return;
    }
    Barang newBarang;
    cout << endl;
    cout << "Masukkan Data Barang:" << endl;
    cout << "Input Kode Barang : ";
    cin >> newBarang.kode;
    cin.ignore();
    cout << "Input Nama Barang : ";
    getline(cin, newBarang.nama);
    cout << "Input Lokasi Barang : ";
    getline(cin, newBarang.loker);

    bool barangSudahAda = false;
    for (int i = 0; i < n; i++) {
        if (daftarBarang[i].kode == newBarang.kode &&
            daftarBarang[i].nama == newBarang.nama &&
            daftarBarang[i].loker == newBarang.loker) {

            // Jika barang sudah ada, tambahkan stok
            cout << "Input tambahan stok barang: ";
            int tambahanStok;
            cin >> tambahanStok;
            daftarBarang[i].stock += tambahanStok;
            barangSudahAda = true;
            break;
        }
    }

    if (!barangSudahAda) {
        // Jika barang belum ada, tambahkan barang baru
        cout << "Input Stock Barang : ";
        cin >> newBarang.stock;
        daftarBarang[n] = newBarang;
        n++;
    }
    cout << endl;
}
```

4. Pencarian Barang

Langkah keempat melibatkan fungsi `carinama()`, yang memungkinkan pengguna untuk mencari barang berdasarkan nama dan menampilkan hasil pencarian. Fungsi ini dirancang untuk memberikan efisiensi dalam menemukan barang tertentu di antara daftar yang mungkin sudah cukup panjang.

Fitur pencarian ini mendukung upaya pengguna untuk menemukan barang dengan cepat dan tanpa harus melalui seluruh daftar barang.

```
void carinama() {
    bool stop = false;
    do {
        string keyword;
        cout << "Masukkan Nama barang yang ingin dicari: ";
        cin.ignore();
        getline(cin, keyword);
        bool ketemu = false;

        cout<<endl;
        cout<< "Daftar Barang : "<<endl;
        cout << setw(10) << "Kode" << setw(20) << "Nama Barang" << setw(10) << "Stock" << setw(15) << "Lokasi" << endl;

        for (int i = 0; i < n; i++) {
            if (daftarBarang[i].nama.find(keyword) != string::npos) {
                ketemu = true;
                cout << setw(10) << daftarBarang[i].kode << setw(20) << daftarBarang[i].nama << setw(10) << daftarBarang[i].stock << setw(15) << daftarBarang[i].loker << endl;
            }
        }
        if (!ketemu) {
            cout << "Barang tidak ditemukan" << endl;
        }
        bool again;
        cout << "Cari lagi? (1/0): ";
        cin >> again;
        stop = !again; // berhenti jika user memilih 0
    } while (!stop);
}
```

5. Penghapusan Barang

Fungsi hapusBarang() pada langkah kelima memberikan kemampuan kepada pengguna untuk menghapus data barang berdasarkan kode barang. Proses ini bermanfaat untuk mengelola stok dan informasi barang yang tidak relevan atau tidak diperlukan lagi.

Fitur ini memungkinkan pengguna untuk secara efektif mengelola daftar barang dengan menghilangkan barang yang sudah tidak relevan atau tidak lagi tersedia.

```
void hapusBarang() {
    string cari;
    int ada = 0;

    cout << "Input Kode Barang yang akan dihapus : ";
    cin >> cari;

    for (int i = 0; i < n; i++) {
        if (daftarBarang[i].kode == cari) {
            ada = 1;
            for (int j = i; j < n - 1; j++) {
                daftarBarang[j] = daftarBarang[j + 1];
            }
            n--;
            break;
        }
    }

    if (ada == 0) {
        cout << "Data barang tidak ditemukan" << endl;
    } else {
        cout << "Data barang berhasil dihapus" << endl;
    }
}
```


6. Menu Utama

Langkah terakhir, yaitu fungsi main(), merupakan menu utama yang memberikan opsi kepada pengguna untuk melakukan berbagai fungsi di atas. Fungsi ini berperan sebagai pengendali utama program, memandu pengguna untuk menggunakan dan memanfaatkan fitur-fitur yang telah diimplementasikan.

Menu utama ini memberikan kejelasan dan keteraturan dalam penggunaan program, memastikan pengguna dapat dengan mudah menjalankan fungsi yang diinginkan sesuai dengan kebutuhan mereka.

```
int main()
{
    int choice;
    do {
        cout << "\t\t\t\t\t =====" << endl;
        cout << "\t\t\t\t\t |      InventoBuild: Sistem Inventory Toko Material Bangunan      |" << endl;
        cout << "\t\t\t\t\t |                                TB. LUCKY TWO                                |" << endl;
        cout << "\t\t\t\t\t |                                PUSAT MATERIAL BANGUNAN                                |" << endl;
        cout << "\t\t\t\t\t =====" << endl;
        cout << "Silahkan pilih fitur - fitur yang tersedia" << endl << endl;
        cout << "1. Update Data Barang" << endl;
        cout << "2. Tampilkan Data Barang" << endl;
        cout << "3. Cari Data Barang" << endl;
        cout << "4. Hapus Data Barang" << endl;
        cout << "5. Keluar " << endl << endl;
        cout << "Pilihan: ";
        cin >> choice;
        switch (choice) {
            case 1:
                tambahBarang();
                break;
            case 2:
                displayData();
                break;
            case 3:
                carinama();
                break;
            case 4:
                hapusBarang();
                break;
            case 5:
                cout << endl << "Terima kasih telah menggunakan program ini." << endl;
                return 0; // Keluar dari program
            default:
                cout << "Pilihan salah! Silahkan pilih kembali." << endl;
        }
    } while (true);
    return 0;
}
```

Transformasi Digital dalam Manajemen Barang

Program ini mewakili transformasi digital dalam manajemen barang dengan menyediakan solusi berbasis teknologi untuk mengelola dan mengoptimalkan stok barang. Dengan menggunakan bahasa pemrograman C++,

program ini dapat memberikan kecepatan dan efisiensi dalam pemrosesan data, mendukung upaya digitalisasi pada tingkat operasional.

Pembahasan

Pembahasan ini akan mengeksplorasi konsep transformasi digital dalam manajemen barang melalui implementasi program efisien menggunakan bahasa pemrograman C++. Program ini dirancang untuk membantu toko material bangunan dalam mengelola stok barang, mencari informasi dengan cepat, dan memastikan efisiensi operasional. Berikut adalah beberapa poin penting yang perlu dibahas:

1. Pendefinisian Struktur Data

Pendefinisian struktur data "Barang" memberikan fondasi yang kokoh untuk menyimpan informasi setiap barang. Dengan menggunakan struktur data, program dapat secara terorganisir menyimpan atribut penting seperti kode, nama, stok, dan lokasi. Ini memfasilitasi penyimpanan data yang efisien dan aksesibilitas informasi yang cepat.

2. Tampilan Data Barang

Fungsi **displayData()** memberikan pengguna kemampuan untuk melihat data barang dalam bentuk tabel yang terstruktur. Opsi pengurutan menambahkan fleksibilitas, memungkinkan pengguna untuk mengakses informasi sesuai dengan kebutuhan mereka. Tampilan yang terorganisir membantu dalam pengambilan keputusan yang informasional dan cepat.

3. Penambahan Barang

Fungsi **tambahBarang()** memberikan pengguna cara untuk menambahkan barang baru atau menambah stok barang yang sudah ada. Ini mencerminkan responsibilitas program dalam mengakomodasi perubahan dalam stok barang. Kemampuan untuk menyesuaikan jumlah stok memberikan fleksibilitas yang diperlukan untuk mengelola persediaan.

4. Pencarian Barang

Fungsi **carinama()** memfasilitasi pencarian barang berdasarkan nama. Fitur ini sangat berguna dalam situasi di mana pengguna perlu mencari atau memverifikasi keberadaan suatu barang. Pencarian yang cepat dan efisien menjadi kunci dalam manajemen barang yang adaptif.

5. Penghapusan Barang

Fungsi **hapusBarang()** memberikan kemampuan untuk menghapus data barang berdasarkan kode barang. Proses penghapusan ini membantu dalam menjaga kebersihan dan relevansi data. Dengan menghapus barang yang sudah tidak diperlukan, program mendukung upaya untuk menjaga keakuratan informasi stok.

6. Menu Utama

Fungsi **main()** sebagai menu utama memberikan pengguna pemandu yang jelas untuk menjalankan berbagai fungsi. Ini menciptakan antarmuka yang mudah digunakan, memastikan bahwa pengguna dapat mengakses fitur-fitur program dengan lancar dan efisien.

Kesimpulan

Implementasi program manajemen barang menggunakan C++ adalah langkah positif dalam mewujudkan transformasi digital dalam bisnis material bangunan. Dengan memanfaatkan teknologi, toko material dapat mengelola stok barang, melakukan pencarian, dan menghapus data dengan lebih efisien. Transformasi digital semacam ini menjadi kunci untuk tetap kompetitif dan adaptif di era bisnis yang terus berkembang dan terdigitalisasi.

Daftar Pustaka

- C++ Language - cplusplus.com - The C++ Resources Network. (n.d.). Retrieved from cplusplus.com: <http://www.cplusplus.com/doc/tutorial/>
- Eckel, B. (2003). *"Thinking in C++, Volume 1: Introduction to Standard C++."* Prentice Hall.
- Elmasri, R., & Navathe, S. B. (2016). *"Fundamentals of Database Systems."* Pearson.
- Hamdallah, F., Husain, A., Wijaya, L. A., & Widiyanto, S. R. (2020, February). Sistem Manajemen Basis Data pada Sistem Perpustakaan (Studi Kasus: SMK Al-Wafa). In *Seminar Nasional Teknologi Komputer & Sains (SAINTEKS)* (Vol. 1, No. 1, pp. 30-32).
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *"An Introduction to Statistical Learning."* Springer.
- Josuttis, N. M. (2007). *"The C++ Standard Library: A Tutorial and Reference."* Addison-Wesley.
- Lee, L., & Lapira, E. (2019). *"Internet of Things for Smart Logistics: Technologies and Security Issues."* IEEE Access, 7, 4725-4746.
- Schach, S. R. (2014). *"Object-Oriented and Classical Software Engineering."* McGraw-Hill Education.
- Stallings, W. (2017). *"Cryptography and Network Security: Principles and Practice."* Pearson.
- Syamil, A., Danial, R. D. M., Saori, S., Waty, E., Fahmi, M. A., Hartati, V., ... & Haryadi, R. M. (2023). *Buku Ajar Manajemen Rantai Pasok.* PT. Sonpedia Publishing Indonesia.
- Wijayanto, H., & Harsadi, P. (2017). *Transformasi Digital.* Surakarta: STMIK Sinar Nusantara.

Link Video Penjelasan Artikel dan Tutorial :

<https://youtu.be/IQTxCuDBQL4?si=oxpweg-AUt4aTGIIH>