

Aplikasi Sistem Kasir dan Manajemen Stok *Bakery* Terintegrasi

Yunida Aulia Khoirun Nisa¹, Miftachul Jannah Wigno Febriyani²

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

Artikel Info

Kata kunci:

C++
Sistem Kasir
Manajemen Stok
Array Multidimensi
Struct

ABSTRAK

Dalam industri kuliner khususnya usaha toko roti (*bakery*), kecepatan dan akurasi dalam proses transaksi serta pengelolaan stok merupakan faktor krusial. Sistem pencatatan manual yang masih banyak diterapkan seringkali menimbulkan risiko kesalahan perhitungan (*human error*), ketidakakuratan data stok, dan lambatnya pelayanan kepada pelanggan. Tujuan dari proyek akhir ini adalah merancang sebuah aplikasi sistem kasir berbasis komputer yang mampu mengotomatisasi proses penjualan dan pemantauan stok barang secara sederhana namun efisien. Aplikasi ini dikembangkan menggunakan bahasa pemrograman C++. Implementasi kode program menerapkan konsep-konsep dasar pemrograman meliputi penggunaan tipe data terstruktur (*Struct*) untuk merepresentasikan data produk dan pesanan, serta *Array Multidimensi* untuk mensimulasikan tata letak rak penyimpanan stok secara visual. Selain itu, program ini memanfaatkan fungsi (*function*) modular untuk memisahkan logika perhitungan aritmatika dan tampilan antarmuka. Hasil pengujian menunjukkan bahwa program mampu memproses *input* pesanan pelanggan, melakukan pengurangan stok secara otomatis (*real-time*), menghitung subtotal, pajak (PPN), dan biaya layanan antar (*delivery*), serta mencetak struk pembayaran yang valid. Dengan adanya sistem ini, proses transaksi menjadi lebih transparan, akurat, dan memudahkan kasir dalam memantau ketersediaan produk di rak.

Penulis Korespondensi :

Yunida Aulia Khoirun Nisa,
Program Studi Teknik Informatika
Fakultas Ilmu Komputer
Universitas Dian Nuswantoro, Kota Semarang, 50131
Email: 111202516194@mhs.dinus.ac.id

1. PENDAHULUAN

Perkembangan teknologi informasi saat ini telah merambah ke berbagai sektor, termasuk sektor bisnis kuliner dan Usaha Mikro, Kecil, dan Menengah (UMKM) [1]. Dalam persaingan bisnis yang semakin ketat, kecepatan pelayanan dan akurasi data menjadi kunci utama kepuasan pelanggan [2]. Salah satu jenis usaha yang memerlukan manajemen transaksi yang cepat dan tepat adalah toko roti atau *bakery* [3]. Industri *bakery* yang terus berkembang menuntut adanya sistem yang mampu mengakomodasi kebutuhan operasional secara efisien, mulai dari pencatatan penjualan hingga pengelolaan inventaris produk [4], [5].

Pada sistem operasional toko roti konvensional, proses pencatatan transaksi sering kali masih dilakukan secara manual menggunakan nota tulis tangan atau kalkulator sederhana [6]. Metode ini memiliki banyak kelemahan yang dapat menghambat produktivitas dan merugikan usaha [7]. Di antaranya adalah tingginya risiko kesalahan perhitungan akibat faktor manusia (*human error*), lambatnya proses pelayanan ketika menghadapi antrean panjang pelanggan, serta kesulitan dalam memantau ketersediaan stok barang di rak secara *real-time* [8]. Sering kali terjadi ketidaksesuaian antara data stok di catatan dengan kondisi fisik barang yang sebenarnya ada di rak, yang dapat menyebabkan kerugian finansial bagi pemilik usaha, baik karena kehilangan peluang penjualan maupun pemborosan akibat produk yang tidak terjual [9].

Selain masalah operasional yang dihadapi oleh pelaku usaha, dari sisi pengembangan perangkat lunak, pembuatan sistem kasir merupakan studi kasus yang sangat relevan untuk menerapkan berbagai konsep logika dasar pemrograman [10]. Dalam ilmu komputer, pengelolaan data barang yang memiliki atribut majemuk

seperti nama produk, harga jual, dan jumlah stok tersedia memerlukan struktur data yang tepat dan terorganisir, salah satunya adalah penggunaan *Struct* (struktur data terstruktur) [11]. Sementara itu, representasi rak penyimpanan fisik ke dalam bentuk digital yang dapat diproses oleh komputer membutuhkan pemahaman yang kuat mengenai konsep *Array* Multidimensi, yang memungkinkan penyimpanan data dalam format baris dan kolom layaknya rak sesungguhnya [12].

Berdasarkan permasalahan tersebut, penulis merancang sebuah aplikasi sederhana berjudul "Sistem Kasir *Bakery* Integrasi" menggunakan bahasa pemrograman C++. Program ini dirancang tidak hanya untuk melakukan perhitungan total bayar dan kembalian secara otomatis dan akurat, tetapi juga mengintegrasikan fitur manajemen stok yang akan berkurang secara otomatis setiap kali transaksi penjualan terjadi. Dengan demikian, data inventaris akan selalu *ter-update* secara *real-time* tanpa perlu *input* manual yang rentan kesalahan. Selain itu, program ini juga mengakomodasi berbagai jenis layanan pelanggan seperti *Dine In* (makan di tempat), *Pick Up* (ambil sendiri), dan *Delivery* (pengiriman) yang sangat relevan dengan model bisnis toko roti modern saat ini, di mana pelanggan menginginkan fleksibilitas dalam cara memperoleh produk.

Pembuatan proyek ini juga ditujukan untuk memenuhi tugas besar mata kuliah Dasar Pemrograman, sebagai bentuk implementasi nyata dari berbagai materi yang telah dipelajari selama perkuliahan. Proyek ini mencakup penerapan tipe data dasar dan kompleks, struktur kontrol seperti perulangan (*looping*) dan percabangan (*branching*), hingga penggunaan fungsi atau *modular programming* untuk memecahkan masalah yang kompleks menjadi bagian-bagian yang lebih sederhana dan mudah dipelihara. Dengan pendekatan ini, diharapkan mahasiswa dapat memahami bagaimana teori pemrograman diterapkan dalam menyelesaikan permasalahan dunia nyata. Berdasarkan latar belakang yang telah diuraikan di atas, rumusan masalah dari proyek ini adalah sebagai berikut. Pertama, bagaimana cara membuat sistem kasir yang dapat menyimpan dan mengelola data roti secara terstruktur, meliputi nama produk, harga jual, dan jumlah stok yang tersedia. Kedua, bagaimana menerapkan konsep *Array* Multidimensi untuk merepresentasikan rak roti secara digital sehingga memudahkan visualisasi dan pengelolaan posisi produk. Ketiga, bagaimana menghitung subtotal pembelian, pajak yang dikenakan, dan ongkos kirim secara otomatis menggunakan fungsi-fungsi yang telah didefinisikan, sehingga proses perhitungan menjadi lebih efisien dan terhindar dari kesalahan manual.

Proyek ini memiliki beberapa tujuan yang hendak dicapai. Pertama, untuk memenuhi persyaratan tugas besar atau proyek akhir mata kuliah Dasar Pemrograman sebagai bagian dari evaluasi pembelajaran. Kedua, untuk mengimplementasikan struktur data *Struct* dan *Array* dalam studi kasus nyata berupa sistem kasir toko roti, sehingga pemahaman terhadap konsep-konsep tersebut tidak hanya bersifat teoretis namun juga aplikatif. Ketiga, untuk membuat program aplikasi yang mampu melakukan transaksi penjualan secara komprehensif, termasuk pengurangan stok secara otomatis setiap kali ada barang terjual, serta pencetakan struk belanja yang informatif bagi pelanggan. Dengan tercapainya tujuan-tujuan ini, diharapkan program yang dihasilkan dapat memberikan manfaat praktis bagi pelaku usaha *bakery* sekaligus menjadi pembelajaran berharga bagi pengembang dalam menguasai dasar-dasar pemrograman.

2. METODE

Bab ini menjelaskan metodologi yang digunakan dalam pengembangan Sistem Kasir *Bakery* Integrasi, mencakup kerangka konseptual sistem, alat dan bahasa pemrograman yang digunakan, struktur data untuk representasi entitas, teknik pemrograman yang diterapkan, serta struktur kontrol yang mengatur alur logika program. Pembahasan metodologi ini disusun secara sistematis untuk memberikan pemahaman komprehensif mengenai landasan teknis dan pendekatan yang digunakan dalam membangun sistem yang efektif dan efisien.

2.1 Kerangka Konseptual Sistem

Penelitian ini menggunakan pendekatan pengembangan perangkat lunak berbasis konsep *Point of Sale* (POS) yang diintegrasikan dengan manajemen inventaris [13]. Sistem *Point of Sale* atau sistem kasir merupakan sistem yang digunakan untuk melakukan transaksi penjualan ritel secara terstruktur dan terkomputerisasi [14]. Dalam operasional bisnis modern, POS tidak hanya berfungsi sebagai alat hitung layaknya kalkulator konvensional, tetapi juga berperan sebagai alat pencatat data transaksi secara komprehensif dan pemantau stok barang secara *real-time* [15]. Penerapan sistem POS berbasis komputerisasi bertujuan untuk meminimalisir kesalahan yang disebabkan oleh faktor manusia (*human error*), mempercepat proses pelayanan pelanggan, serta meningkatkan akurasi data keuangan dan inventaris yang sangat penting bagi pengambilan keputusan bisnis [16].

2.2 Alat dan Bahasa Pemrograman

Implementasi sistem dalam penelitian ini menggunakan bahasa pemrograman C++ sebagai *tools* utama pengembangan. C++ merupakan bahasa pemrograman tingkat tinggi (*high-level language*) yang merupakan pengembangan dan perluasan dari bahasa pemrograman C [17]. Bahasa ini mendukung berbagai paradigma pemrograman, termasuk pemrograman prosedural dan pemrograman berorientasi objek (*Object-*

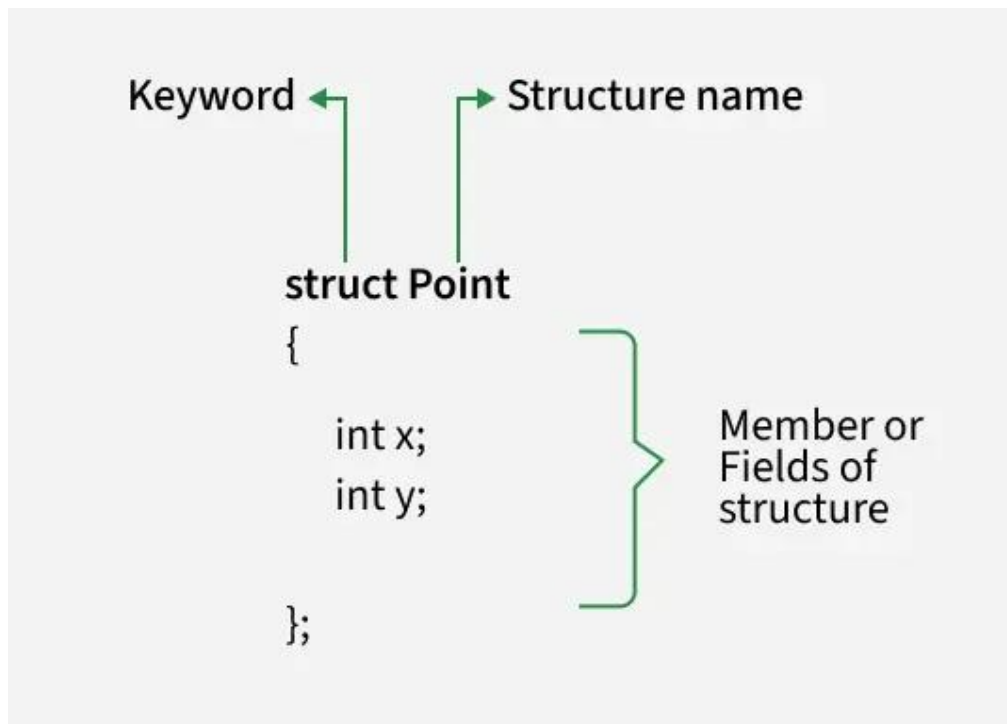
Oriented Programming) [18]. C++ dikenal luas di kalangan pengembang perangkat lunak karena efisiensi dan kecepatan tinggi dalam eksekusi program, menjadikannya pilihan ideal untuk aplikasi yang membutuhkan performa optimal [19]. Dalam penelitian ini, C++ digunakan dengan pendekatan prosedural, memanfaatkan pustaka standar (*standard library*) yang tersedia. Pustaka *iostream* digunakan untuk operasi *input-output* standar seperti *cin* dan *cout*, sementara pustaka *string* digunakan untuk manipulasi dan pengolahan data teks. Pemilihan C++ juga didasarkan pada fleksibilitasnya dalam mengimplementasikan struktur data kompleks yang diperlukan dalam sistem kasir terintegrasi.

2.3 Struktur Data dan Representasi Entitas

Untuk merepresentasikan entitas bisnis dan data operasional dalam sistem, diperlukan struktur data yang tepat dan efisien. Sistem kasir *bakery* yang dikembangkan melibatkan berbagai jenis data dengan karakteristik yang berbeda, mulai dari data produk dengan atribut majemuk hingga representasi spasial rak penyimpanan. Oleh karena itu, penelitian ini menerapkan dua pendekatan struktur data utama yang saling melengkapi: tipe data struktur (*struct*) untuk mengelola entitas dengan atribut kompleks, dan *array* multidimensi untuk merepresentasikan posisi dan *layout* fisik penyimpanan produk. Kedua pendekatan ini dipilih berdasarkan kesesuaiannya dengan kebutuhan fungsional sistem dan kemampuannya dalam menyediakan akses data yang cepat dan terorganisir.

2.3.1 Tipe Data Struktur (*Struct*)

Structure atau yang biasa disingkat *struct* merupakan tipe data bentukan (*user-defined data type*) yang memungkinkan pemrogram untuk mengelompokkan beberapa variabel dengan tipe data yang berbeda di bawah satu nama atau identitas tunggal [19]. *Struct* sangat berguna untuk merepresentasikan objek atau entitas nyata yang memiliki atribut majemuk dan saling berkaitan [20]. Dalam konteks pengembangan program kasir pada penelitian ini, *struct* digunakan untuk mendefinisikan dua entitas utama.



Gambar 1 Sintaksis *Struct* Beserta Anggotanya [21]

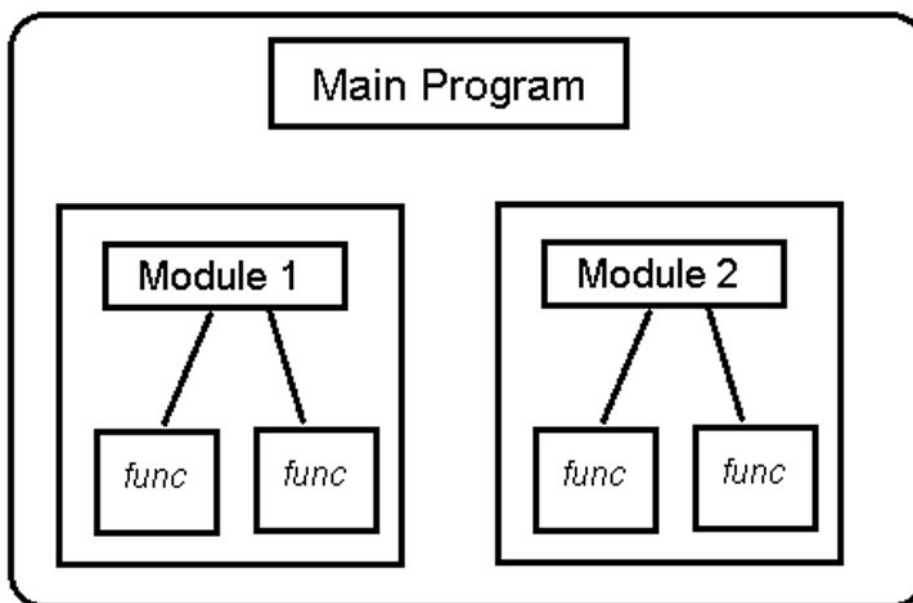
Pertama, entitas Roti yang mengelompokkan tiga atribut penting, yaitu nama produk dengan tipe data *string* untuk menyimpan identitas roti, harga dengan tipe data integer untuk menyimpan nilai jual per unit, dan stok dengan tipe data integer untuk menyimpan jumlah ketersediaan produk. Kedua, entitas Pesanan yang mengelompokkan atribut-atribut terkait identitas pelanggan dan keranjang belanja, memungkinkan sistem untuk melacak setiap transaksi secara individual dan terstruktur. Penggunaan *struct* memudahkan pengelolaan data yang kompleks dan meningkatkan keterbacaan kode program.

2.3.2 Array Multidimensi untuk Representasi Spasial

Array merupakan struktur data fundamental yang menyimpan sekumpulan elemen dengan tipe data yang sama secara berurutan dalam memori komputer [22]. *Array* multidimensi, khususnya *array* dua dimensi (2D *array*), sering digambarkan sebagai sebuah tabel atau matriks yang memiliki dimensi baris dan kolom, mirip dengan struktur *spreadsheet* [23]. Pada penelitian ini, *array* dua dimensi digunakan untuk memvisualisasikan dan merepresentasikan rak penyimpanan roti secara digital [24]. Indeks pertama dari *array* merepresentasikan posisi baris rak dalam *layout* fisik toko, sedangkan indeks kedua merepresentasikan posisi kolom rak pada baris tersebut. Pendekatan ini memungkinkan sistem untuk mengakses data stok berdasarkan koordinat lokasi fisik barang, sehingga memudahkan proses pencarian produk, pembaruan stok, dan visualisasi ketersediaan barang dalam format yang intuitif dan mudah dipahami. Representasi spasial ini juga memfasilitasi pengembangan fitur tambahan seperti optimasi penempatan produk berdasarkan frekuensi penjualan.

2.4 Teknik Pemrograman Modular

Pemrograman modular merupakan teknik pengembangan perangkat lunak yang memecah program besar dan kompleks menjadi sub-program yang lebih kecil, spesifik, dan independen yang disebut fungsi (*function*) [25]. Penggunaan fungsi bertujuan untuk membuat struktur kode program lebih rapi dan terorganisir, mudah dibaca dan dipahami oleh pengembang lain, serta menghindari penulisan kode yang berulang (*code reusability*) sehingga meningkatkan efisiensi pengembangan dan pemeliharaan program [26].



Gambar 2 Visualisasi *Modular Programming* [27]

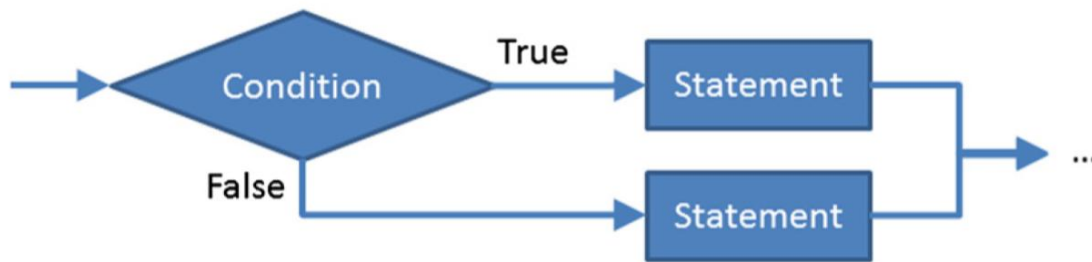
Dalam sistem yang dikembangkan pada penelitian ini, diterapkan berbagai fungsi dengan tipe pengembalian nilai (*return type*) yang berbeda sesuai kebutuhan fungsional. Fungsi dengan tipe *void* digunakan untuk prosedur yang tidak mengembalikan nilai, seperti fungsi *tampilkanRak* yang bertugas menampilkan visualisasi rak penyimpanan ke layar. Sementara itu, fungsi dengan *return type* tertentu digunakan untuk operasi perhitungan yang menghasilkan nilai, seperti fungsi *hitungSubtotal* yang mengembalikan nilai total harga barang sebelum pajak, dan fungsi *hitungPajak* yang menghitung dan mengembalikan nilai pajak yang harus dibayarkan berdasarkan persentase yang telah ditentukan. Modularisasi ini memungkinkan setiap fungsi untuk diuji secara independen dan memudahkan proses *debugging*.

2.5 Struktur Kontrol Program

Untuk mengatur alur logika dan pengambilan keputusan dalam program, penelitian ini menerapkan dua jenis struktur kontrol utama yang merupakan elemen fundamental dalam pemrograman prosedural.

2.5.1 Percabangan (*Branching*)

Struktur percabangan menggunakan konstruksi *if-else* untuk mengimplementasikan logika kondisional dalam berbagai aspek sistem [28]. Pertama, percabangan digunakan untuk validasi ketersediaan stok, yaitu memeriksa apakah jumlah stok produk yang diminta pelanggan mencukupi atau sudah habis, sehingga sistem dapat memberikan respons yang sesuai.

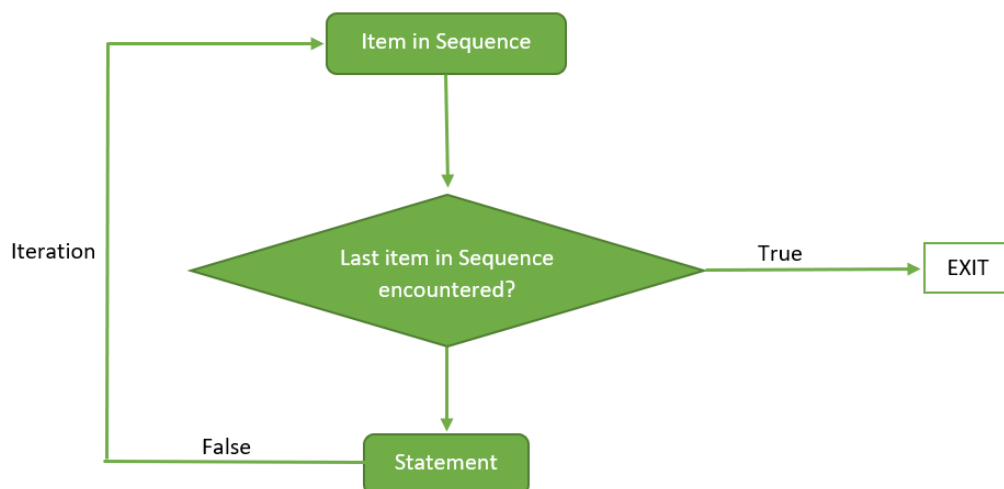


Gambar 3 Alur Percabangan Pada Pemrograman [29]

Kedua, percabangan diterapkan dalam pemilihan jenis layanan pelanggan, memungkinkan sistem untuk membedakan antara opsi *Dine In*, *Pick Up*, dan *Delivery*, yang masing-masing mungkin memiliki perlakuan perhitungan biaya yang berbeda. Ketiga, validasi pembayaran menggunakan percabangan untuk memastikan bahwa jumlah uang yang dibayarkan pelanggan mencukupi total tagihan sebelum transaksi diproses lebih lanjut.

2.5.2 Perulangan (Looping)

Struktur perulangan menggunakan konstruksi *for loop* untuk melakukan iterasi atau pengulangan operasi tertentu [28]. Dalam sistem ini, *for loop* digunakan untuk menampilkan seluruh isi rak penyimpanan secara berulang dengan cara mengiterasi setiap elemen dalam *array* multidimensi, memungkinkan visualisasi lengkap *layout* rak.



Gambar 4 Alur Perulangan Pada Pemrograman [30]

Selain itu, perulangan juga diterapkan untuk melakukan iterasi pada *array* keranjang belanja ketika menghitung total harga pembelian, di mana sistem menjumlahkan harga setiap item yang telah dipilih pelanggan. Kombinasi antara struktur percabangan dan perulangan ini memungkinkan sistem untuk menangani berbagai skenario transaksi dengan logika yang konsisten dan efisien.

2.6 Alur Kerja Sistem

Metodologi pengembangan sistem mengikuti alur kerja terstruktur yang dimulai dari inisialisasi data produk dan rak penyimpanan, dilanjutkan dengan proses interaksi dengan pengguna untuk pemilihan produk dan jumlah, validasi ketersediaan stok menggunakan struktur kontrol percabangan, perhitungan otomatis menggunakan fungsi-fungsi modular, pembaruan data inventaris secara *real-time*, hingga pencetakan struk sebagai *output* akhir transaksi. Setiap tahapan dirancang untuk memastikan integritas data dan akurasi perhitungan dalam seluruh proses transaksi.

3. PEMBAHASAN HASIL

Bab ini menyajikan pembahasan mengenai implementasi dan hasil pengembangan Sistem Kasir Bakery Integrasi yang telah dirancang menggunakan bahasa pemrograman C++. Pembahasan mencakup struktur program secara keseluruhan, penjelasan setiap komponen utama beserta fungsinya, mekanisme kerja sistem, serta analisis fitur-fitur yang telah diimplementasikan. Setiap bagian dijelaskan secara detail untuk memberikan pemahaman komprehensif tentang bagaimana sistem bekerja dan bagaimana konsep-konsep pemrograman dasar diterapkan dalam menyelesaikan permasalahan nyata.

3.1 Arsitektur dan Struktur Program

Program Sistem Kasir Bakery Integrasi dibangun dengan arsitektur modular yang terdiri dari beberapa komponen utama. Struktur program dimulai dengan direktif preprosesor yang digunakan untuk mengimpor pustaka-pustaka yang diperlukan, yaitu `<iostream>` untuk operasi input-output standar dan `<string>` untuk manipulasi data teks. Deklarasi using namespace std ditambahkan untuk menyederhanakan penulisan kode dengan menghilangkan kebutuhan menuliskan prefix std:: sebelum setiap fungsi standar seperti cin dan cout.

Tabel 1 Direktif Preprocessor untuk Impor Pustaka

main.cpp
#include <iostream> // Library untuk input/output (cin, cout, endl) #include <string> // Library untuk tipe data string (teks)
using namespace std; // Agar tidak perlu menulis std:: sebelum cin/cout

Arsitektur program dirancang dengan pendekatan top-down, di mana fungsi utama main() bertindak sebagai pengendali alur program utama, sementara fungsi-fungsi pendukung seperti inisialisasiData(), tampilkanRak(), hitungSubtotal(), dan hitungPajak() menangani tugas-tugas spesifik. Pendekatan ini memudahkan pemeliharaan kode, debugging, dan pengembangan fitur tambahan di masa mendatang.

3.2 Implementasi Struktur Data

Implementasi struktur data dalam program memiliki peran penting untuk mengelola dan menyimpan data secara terorganisir serta efisien. Struct Roti dirancang sebagai tipe data bentukan yang merepresentasikan entitas produk roti dengan tiga atribut utama: nama bertipe string yang menyimpan identitas produk seperti "Roti Tawar" atau "Croissant", harga bertipe int yang menyimpan harga jual per unit dalam satuan Rupiah, dan stok bertipe int yang menyimpan jumlah ketersediaan produk yang dapat dijual. Penggunaan struct ini memungkinkan pengelompokan data yang saling berkaitan ke dalam satu kesatuan logis, sehingga akses data menjadi lebih terstruktur dan intuitif melalui notasi dot seperti rak[0][0].nama untuk mengakses nama produk di posisi baris 0 kolom 0.

Tabel 2 Definisi Struct dan Deklarasi Array

main.cpp
typedef struct { string nama; // Menyimpan nama roti, contoh: "Roti Tawar" int harga; // Menyimpan harga satuan, contoh: 12000 int stok; // Menyimpan jumlah tersedia, contoh: 10 } Roti; typedef struct { string nama_pelanggan; // Menyimpan nama pembeli string tanggal; // Menyimpan tanggal transaksi int keranjang[3][2]; // Array 2D untuk menyimpan jumlah pesanan } Pesanan; Roti rak[3][2]; // Deklarasi array 2D untuk rak roti

Struct Pesanan dirancang untuk menyimpan informasi transaksi pelanggan dengan tiga atribut penting: nama_pelanggan bertipe string untuk identitas pembeli, tanggal bertipe string untuk waktu transaksi, dan keranjang[3][2] berupa array dua dimensi bertipe int yang menyimpan jumlah pesanan untuk setiap posisi

produk di rak. Array keranjang dirancang dengan dimensi yang sama dengan array rak (3 baris $\times$ 2 kolom), sehingga terdapat korespondensi langsung antara posisi produk di rak fisik dengan posisi di keranjang belanja. Setiap elemen keranjang[i][j] menyimpan jumlah unit produk yang dipesan dari posisi rak[i][j], memungkinkan sistem melacak dengan akurat produk mana saja yang dibeli dan dalam jumlah berapa. Array global rak[3][2] bertipe Roti digunakan untuk merepresentasikan layout fisik rak penyimpanan di toko bakery. Array dua dimensi ini divisualisasikan sebagai tabel dengan 3 baris dan 2 kolom, menghasilkan total 6 slot penyimpanan produk. Indeks baris (i) dari 0 hingga 2 merepresentasikan posisi vertikal rak dari atas ke bawah, sementara indeks kolom (j) dari 0 hingga 1 merepresentasikan posisi horizontal dari kiri ke kanan. Representasi spasial ini memberikan keuntungan: memudahkan visualisasi posisi produk dalam format intuitif, mempermudah akses dan pembaruan data berdasarkan koordinat fisik, memfasilitasi validasi ketersediaan stok secara real-time, dan memungkinkan ekspansi sistem untuk rak dengan dimensi lebih besar tanpa perubahan fundamental pada logika program.

3.3 Fungsi-Fungsi Utama Sistem

Setelah struktur data diterapkan untuk mengelola dan menyimpan informasi secara terorganisir, tahap selanjutnya adalah mengimplementasikan fungsi-fungsi utama yang mengatur alur kerja sistem. Fungsi inisialisasiData() bertugas mengisi data awal produk ke dalam array rak saat program pertama kali dijalankan. Fungsi ini tidak menerima parameter dan tidak mengembalikan nilai (void function). Implementasi menggunakan sintaksis inisialisasi struct dengan tanda kurung kurawal seperti rak[0][0] = {"Roti Tawar", 12000, 10} yang secara berurutan mengisi atribut nama, harga, dan stok. Meskipun data saat ini bersifat hardcoded, desain fungsi ini memudahkan pengembangan di masa depan untuk membaca data dari file eksternal atau database. Fungsi tampilkanRak() bertugas menampilkan status terkini seluruh rak penyimpanan dalam format visual yang terstruktur menggunakan nested loop untuk mengiterasi seluruh elemen dalam array dua dimensi. Loop luar mengiterasi baris dari 0 hingga 2, sementara loop dalam mengiterasi kolom dari 0 hingga 1 untuk setiap baris. Implementasi mencakup fitur formatting yang sophisticated: sistem menghitung panjang string nama produk menggunakan method length() dan menambahkan spasi tambahan hingga mencapai panjang standar 13 karakter; untuk produk dengan harga di bawah Rp 10.000 dan stok kurang dari 10 unit ditambahkan spasi ekstra untuk konsistensi visual. Setiap item ditampilkan dengan format [baris][kolom] Nama_Produk (Rp Harga | Stok:Jumlah), memberikan informasi lengkap tentang posisi, identitas, harga, dan ketersediaan produk.

Tabel 3 Potongan Kode Fungsi Utama Sistem

main.cpp
<pre> void inisialisasiData() { // Mengisi data roti ke setiap slot rak // Format: {nama, harga, stok} // Baris 0 (Baris paling atas) rak[0][0] = {"Roti Tawar", 12000, 10}; // Slot kiri atas rak[0][1] = {"Roti Coklat", 5000, 20}; // Slot kanan atas // Baris 1 (Baris tengah) rak[1][0] = {"Roti Keju ", 6000, 15}; // Spasi untuk formatting rak[1][1] = {"Bomboloni ", 8000, 8}; // Spasi untuk alignment // Baris 2 (Baris paling bawah) rak[2][0] = {"Croissant ", 15000, 5}; rak[2][1] = {"Bagelen ", 10000, 12}; // Tiga spasi untuk formatting } long long hitungSubtotal(Pesanan p) { long long total = 0; for (int i = 0; i < 3; i++) { for (int j = 0; j < 2; j++) { if (p.keranjang[i][j] > 0) { total += (long long)p.keranjang[i][j] * rak[i][j].harga; } } } } </pre>

```

    }
    return total;
}

```

Fungsi `hitungSubtotal()` bertipe `long long` menerima parameter berupa objek Pesanan dan mengembalikan nilai total harga semua produk yang dipesan sebelum ditambah pajak dan biaya tambahan. Fungsi menggunakan nested loop untuk mengiterasi seluruh elemen dalam array keranjang, menghitung total harga dengan rumus `p.keranjang[i][j] * rak[i][j].harga` untuk setiap posisi yang memiliki nilai lebih besar dari 0. Penggunaan tipe data `long long` penting untuk menghindari integer overflow pada transaksi bernilai besar yang mungkin melebihi batas maksimal tipe `int` (2.147.483.647). Type casting (`long long`) diterapkan pada operand sebelum perkalian untuk memastikan operasi aritmatika dilakukan dalam presisi `long long` sejak awal. Fungsi `hitungPajak()` bertipe `long long` menerima parameter subtotal dan mengembalikan nilai pajak yang dihitung sebesar 5% dari subtotal menggunakan rumus `subtotal * 5 / 100`. Penggunaan integer division menghasilkan nilai pajak yang dibulatkan ke bawah, praktik umum dalam sistem kasir untuk menghindari pecahan sen. Desain fungsi ini sangat modular dan fleksibel, memudahkan modifikasi persentase pajak di masa depan tanpa mengubah bagian lain dari program.

3.4 Alur Kerja Program Utama

Setelah fungsi-fungsi utama sistem dirancang dan diimplementasikan, langkah berikutnya adalah menyusun alur kerja program utama secara terstruktur. Eksekusi program dimulai dengan tahap inisialisasi pada fungsi `main()` yang mendeklarasikan konstanta `harga_ongkir` dengan nilai 8000, memanggil fungsi `inisialisasiData()` untuk mengisi array `rak`, dan mendeklarasikan objek Pesanan bernama `p`. Sistem melakukan inisialisasi array keranjang dengan mengatur semua elemennya menjadi 0 menggunakan nested loop untuk mencegah garbage value. Setelah inisialisasi selesai, program menampilkan header aplikasi dan meminta input data pelanggan berupa nama dan tanggal transaksi menggunakan fungsi `getline()` yang dapat membaca string dengan spasi. Setelah data pelanggan terinput, sistem memasuki loop utama pemesanan yang diimplementasikan sebagai infinite loop (`for(;;)`). Dalam setiap iterasi, sistem menampilkan status rak terkini dengan memanggil fungsi `tampilkanRak()`, kemudian meminta input posisi produk dalam bentuk indeks baris (0-2) dan kolom (0-1). Proses validasi input dilakukan secara bertingkat dengan tiga level pemeriksaan. Validasi level pertama memeriksa apakah posisi berada dalam rentang valid array (`b >= 0 && b < 3 && k >= 0 && k < 2`). Validasi level kedua memeriksa apakah stok produk masih tersedia (`rak[b][k].stok > 0`). Validasi level ketiga memeriksa apakah jumlah yang diinputkan positif dan tidak melebihi stok (`jumlah > 0 && jumlah <= rak[b][k].stok`). Hanya setelah ketiga validasi berhasil, sistem mengurangi stok di rak (`rak[b][k].stok -= jumlah`) dan menambahkan jumlah pesanan ke keranjang (`p.keranjang[b][k] += jumlah`). Pendekatan validasi multi-level ini penting untuk menjaga integritas data dan mencegah error seperti array out of bounds, pembelian produk yang tidak tersedia, atau transaksi dengan jumlah negatif.

Tabel 4 Potongan Kode Alur Kerja Utama Program

```

main.cpp
int main() {
    int harga_ongkir = 8000;
    inisialisasiData();
    Pesanan p;
    // Inisialisasi keranjang
    for(int i=0; i<3; i++) {
        for(int j=0; j<2; j++) {
            p.keranjang[i][j] = 0;
        }
    }

    if (layanan == 1) {
        jenisLayanan = "Dine In";
        biayaOngkir = 0;
    }
    else if (layanan == 2) {
        jenisLayanan = "Pick Up";
        biayaOngkir = 0;
    }
}

```

```

else if (layanan == 3) {
    jenisLayanan = "Delivery";
    biayaOngkir = harga_ongkir; // Tambah biaya delivery
}
else {
    jenisLayanan = "Take Away"; // Default jika input salah
    biayaOngkir = 0;
}

```

Setelah pengguna selesai melakukan pemesanan dan memilih keluar dari loop (input 'n' atau 'N'), sistem memasuki tahap pemilihan jenis layanan dengan menampilkan tiga opsi: Dine In, Pick Up, dan Delivery. Sistem menggunakan struktur percabangan if-else untuk menentukan jenis layanan dan biaya ongkir yang sesuai, dengan nilai default "Take Away" untuk input tidak valid. Sistem kemudian melakukan perhitungan komprehensif dengan memanggil fungsi `hitungSubtotal(p)` dan `hitungPajak(subtotal)`, lalu menghitung total pembayaran: $\text{totalBayar} = \text{subtotal} + \text{pajak} + \text{biayaOngkir}$. Sistem memasuki loop validasi pembayaran yang terus berjalan hingga pengguna memasukkan jumlah uang yang mencukupi, kemudian menghitung kembalian dan keluar dari loop. Tahap final adalah pencetakan struk pembayaran komprehensif yang dimulai dengan header "STRUCK PEMBAYARAN" diikuti informasi transaksi dasar. Bagian detail pesanan menampilkan semua item yang dibeli dengan mengiterasi array keranjang dalam format Nama Jumlah $\times$ Harga = Total. Penggunaan flag boolean `adaPesanan` memungkinkan sistem mendeteksi apakah ada item yang dibeli. Setelah detail pesanan, struk menampilkan rincian biaya (subtotal, pajak 5%, dan ongkir jika ada) serta ringkasan pembayaran (total yang harus dibayar, tunai yang diterima, dan kembalian), ditutup dengan pesan "Terima Kasih Telah Berbelanja".

3.5 Fitur-Fitur Unggulan Sistem

Setelah alur kerja program utama dijelaskan, tahap selanjutnya adalah mengidentifikasi fitur-fitur unggulan yang menjadi kelebihan sistem. Fitur manajemen stok real-time merupakan kemampuan sistem untuk secara otomatis mengurangi nilai atribut stok setiap kali pembelian berhasil diproses, memastikan data inventaris selalu ter-update dan akurat. Validasi stok dilakukan sebelum transaksi diproses dengan memeriksa `rak[b][k].stok > 0` dan `jumlah <= rak[b][k].stok`, mencegah overselling yang dapat menyebabkan masalah operasional dan ketidakpuasan pelanggan. Sistem validasi multi-tahap mengimplementasikan validasi input yang sangat komprehensif pada berbagai tahap interaksi: validasi posisi rak untuk mencegah array out of bounds error, validasi ketersediaan stok, validasi jumlah pembelian untuk memastikan nilai positif dan tidak melebihi stok, serta validasi pembayaran untuk memastikan uang mencukupi. Setiap jenis validasi dilengkapi dengan pesan error yang spesifik dan informatif, meningkatkan robustness sistem melalui pendekatan defensive programming.

Tabel 5 Cuplikan Kode Pengurangan Stok Secara *Real-Time*

main.cpp
<pre> rak[b][k].stok = rak[b][k].stok - jumlah; p.keranjang[b][k] = p.keranjang[b][k] + jumlah; </pre>

Fleksibilitas opsi layanan disediakan melalui tiga pilihan (Dine In, Pick Up, dan Delivery) dengan implementasi menggunakan struktur percabangan yang menyesuaikan variabel `jenisLayanan` dan `biayaOngkir` berdasarkan pilihan pengguna. Desain modular ini memudahkan penambahan opsi layanan baru atau modifikasi biaya di masa depan, dengan default option untuk input tidak valid yang memastikan program tetap stabil. Perhitungan otomatis dan akurat dilakukan menggunakan fungsi-fungsi yang telah didefinisikan, dengan penggunaan tipe data `long long` untuk semua variabel yang menyimpan nilai uang guna mencegah overflow. Type casting eksplisit ke `long long` dilakukan pada operasi perkalian untuk memastikan presisi perhitungan, menjaga akurasi finansial dan mencegah kerugian akibat kesalahan pembulatan atau overflow. Sistem juga secara otomatis menghitung kembalian dan menampilkan semua komponen biaya secara transparan dalam struk. User interface yang terstruktur diterapkan meskipun berbasis console/terminal, dengan penggunaan formatting konsisten, separator lines, spacing yang tepat, header dan footer yang jelas, serta petunjuk input yang eksplisit. Fungsi `tampilkanRak()` menyajikan visualisasi rak dalam format tabel intuitif yang memudahkan pengguna melihat posisi dan status setiap produk dengan cepat, mengurangi ambiguitas dan meminimalkan error input.

4. KESIMPULAN

Pengembangan Sistem Kasir Bakery Terintegrasi menggunakan bahasa pemrograman C++ telah berhasil menghadirkan solusi terkomputerisasi untuk menggantikan proses pencatatan transaksi manual pada toko roti. Sistem ini mampu mengintegrasikan proses transaksi penjualan dengan pengelolaan inventaris secara real-time, di mana setiap transaksi yang dilakukan secara otomatis memperbarui jumlah stok produk yang disimpan dalam struktur array multidimensi. Penerapan struktur data seperti struct untuk merepresentasikan entitas Roti dan Pesanan, serta penggunaan array dua dimensi untuk merepresentasikan tata letak rak fisik, menunjukkan bahwa konsep dasar pemrograman dapat diterapkan secara efektif dalam penyelesaian permasalahan bisnis nyata. Fitur-fitur utama yang diimplementasikan meliputi validasi input bertingkat, perhitungan otomatis subtotal dan pajak, penyediaan beberapa opsi layanan, serta pencetakan struk pembayaran yang informatif dan terstruktur. Dari sisi pembelajaran, proyek ini berhasil mencapai tujuan sebagai sarana penerapan materi pada mata kuliah Dasar Pemrograman. Berbagai konsep fundamental, seperti penggunaan tipe data (int, string, long long, dan bool), struktur kontrol (percabangan if-else dan perulangan for), pemrograman modular melalui fungsi, serta mekanisme input-output menggunakan cin, cout, dan getline(), telah diterapkan secara terpadu dalam sebuah sistem yang berfungsi dengan baik. Penerapan pemrograman modular terbukti efektif dalam mengelompokkan kode program yang kompleks ke dalam fungsi-fungsi yang lebih terstruktur, mudah dipahami, dan mudah dipelihara. Selain itu, sistem validasi bertingkat yang digunakan menegaskan pentingnya pendekatan defensive programming dalam menghasilkan aplikasi yang andal dan ramah pengguna. Meskipun sistem telah berjalan sesuai dengan tujuan awal, masih terdapat beberapa aspek yang dapat dikembangkan untuk meningkatkan fungsionalitas dan penerapan praktisnya. Penambahan mekanisme penyimpanan data melalui file handling atau integrasi basis data memungkinkan penyimpanan data produk dan riwayat transaksi secara permanen. Selain itu, penerapan sistem autentikasi multi-level, fitur pelaporan serta analisis penjualan, dan pengelolaan rak dengan kapasitas dinamis akan menjadikan sistem lebih komprehensif dan relevan untuk operasional bakery yang sesungguhnya. Pengembangan antarmuka grafis (GUI) juga dapat dipertimbangkan guna meningkatkan pengalaman pengguna dan mempermudah penggunaan sistem oleh pengguna non-teknis. Secara keseluruhan, proyek ini telah berhasil memenuhi tujuan pembelajaran dan memberikan dasar yang kuat bagi pengembangan sistem Point of Sale yang lebih lanjut di masa mendatang.

REFERENCES

- [1] F. Susila, N. M. W. Pramesti Ningsih, P. S. A. Jaya Kusuma, and A. A. I. Maha Diva, "Pengembangan UMKM Kuliner Tradisional melalui Proses Produksi dan Inovasi Produk pada PT Bakpia Juwara Satoe Yogyakarta," *J. Pengabd. UntukMu NegeRI*, vol. 9, no. 2, pp. 372–379, Aug. 2025, doi: 10.37859/jpumri.v9i2.9953.
- [2] G. Wijayanto, A. Arjang, D. E. Subroto, I. Merakati, T. P. Pertiwi, and I. P. A. D. Hita, "PERAN INOVASI PRODUK DAN STRATEGI PEMASARAN DIGITAL DALAM MENINGKATKAN PENJUALAN UMKM KULINER TRADISIONAL," *J. Rev. Pendidik. Dan Pengajaran*, vol. 8, no. 1, pp. 3602–3607, Mar. 2025, doi: 10.31004/jrpp.v8i1.43766.
- [3] Anggia Hasti Benjamin, "Efektivitas Instagram ADS dalam Meningkatkan Brand Awareness Umkm Kuliner (Studi Kasus: Kedai Kopi 'Janji Jiwa' Cabang Bandung)," *Arus J. Sos. Dan Hum.*, vol. 5, no. 3, pp. 5748–5755, Dec. 2025, doi: 10.57250/ajsh.v5i3.2240.
- [4] A. Munandar, L. Hakim, and M. A. Istimror, "STRATEGI E-COMMERCE UNTUK MENGEMBANGKAN USAHA PELAKU BISNIS UMKM KULINER," *Sigmagri*, vol. 5, no. 2, pp. 298–305, Dec. 2025, doi: 10.32764/sigmagri.v5i2.1659.
- [5] A. Muhammad, A. D. Subagja, D. As Shidik, and P. Rahayu, "Pengaruh Strategi Pemasaran Digital Dan Kualitas Produk Terhadap Penjualan UMKM Kuliner Di Kecamatan Kalijati Kabupaten Subang," *World Bus. Adm. J.*, Jul. 2025, doi: 10.37950/5jn7tb88.
- [6] H. Chaniago, Z. C. Paramita, Firgiawan Yudha Tama, Moch Helmy, Razan Wilar Mahinda, and Vivin Alifianti Rusyana, "Pendampingan Transformasi Digital sebagai Upaya Penguatan Keberlanjutan UMKM Kuliner Soto Madura Mas Al 14," *J. PKM Merah Putih*, vol. 1, no. 2, pp. 9–23, Nov. 2025, doi: 10.65661/jpkmmp.v1i2.37.
- [7] T. Tibrani, M. Junita, and M. A. Hadi, "PENGARUH PERSEPSI KEAMANAN, KEMUDAHAN, MANFAAT, DAN KEPERCAYAAN TERHADAP MINAT PELAKU UMKM KULINER PENGGUNA ALAT QUICK RESPONSE INDONESIA STANDARS (QRIS) DI KEC. BATU AJI KOTA BATAM," *Glob. Digit.*, vol. 1, no. 1, pp. 41–55, Jul. 2025, doi: 10.33373/gd.v1i1.7899.
- [8] R. Dwi Soraya, M. Hasanah, A. Setiawan, and M. F. N. G. Ratumbuang, "Efektivitas Pelatihan Digital Marketing Terhadap Peningkatan Volume Penjualan UMKM Kuliner di Kota Banjarmasin," *Econ. Educ. J. Ecoducation*, vol. 5, no. 3, pp. 424–444, Jan. 2024, doi: 10.33503/ecoducation.v5i3.3857.
- [9] S. Avriyanti, "PENGARUH INOVASI PRODUK DAN TEKNOLOGI INFORMASI TERHADAP PERKEMBANGAN USAHA (Studi Pada UMKM Bidang Kuliner Di Kabupaten Tabalong)," *PubBis J. Pemikir. Dan Penelit. Adm. Publik Dan Adm. Bisnis*, vol. 6, no. 1, pp. 61–73, Mar. 2022, doi: 10.35722/pubbis.v6i1.582.

- [10] “Pemanfaatan Teknologi Informasi untuk Media Pemasaran di Daerah Wisata Kuliner (Studi Kasus: Kampoeng Kuliner) | RUBINSTEIN”, Accessed: Feb. 04, 2026. [Online]. Available: <https://jurnal.buddhidharma.ac.id/index.php/rubin/article/view/2655>
- [11] W. Wahjono and A. C. Ismanto, “Penggunaan Teknologi Informasi Dalam Mengakses Media Sosial Sebagai Upaya Penguatan Pemasaran Usaha Kuliner,” *J. Ilm. Infokam*, vol. 19, no. 2, pp. 111–118, Feb. 2024, doi: 10.53845/infokam.v19i2.354.
- [12] A. F. A.-A. Faizal, “ANALISIS PENERAPAN EKONOMI DIGITAL DALAM MENINGKATKAN PENDAPATAN PELAKU USAHA MIKRO KECIL MENENGAH (UMKM) KULINER DI KECAMATAN BULUKUMPA,” *Sipakainge Inov. Penelit. Karya Ilm. Dan Pengemb. Islam. Sci.*, vol. 2, no. 11, Nov. 2024, Accessed: Feb. 04, 2026. [Online]. Available: <https://ejurnal.iainpare.ac.id/index.php/sipakainge/article/view/8484>
- [13] N. Heryana, A. Maburri, and S. Pomalingo, “Design and Analysis of Point of Sales Information System for CV XYZ: A Case Study,” *J. Eng. Technol. Innov. JETI*, vol. 2, no. 01, pp. 26–31, Feb. 2023.
- [14] E. Van Droogenbroeck and L. Van Hove, “COVID-19 and point-of-sale payments in Belgium: How the older generation also learned to love contactless,” *J. Paym. Strategy Syst.*, vol. 16, no. 1, pp. 17–27, Mar. 2022.
- [15] A. Junef, “IMPLEMENTASI POINT OF SALE PADA UMKM KABUPATEN SOLOK SELATAN MENGGUNAKAN METODE PERSONAL EXTREME PROGRAMING BAHASA PEMOGGRAMAN PHP DAN DATABASE MySQL,” other, Universitas Putra Indonesia YPTK Padang, 2024. Accessed: Feb. 04, 2026. [Online]. Available: <http://repository.upiyptk.ac.id/13489/>
- [16] D. FRIYANSYAH, “APLIKASI CAFÉ POINT OF SALES (CAPOS) DENGAN Dynamic System Development Method (DSDM) (Study Kasus Ropang Loilo),” s1, Universitas Mercu Buana Jakarta, 2022. Accessed: Feb. 04, 2026. [Online]. Available: <https://repository.mercubuana.ac.id/70755/>
- [17] S. Lahti, M. Rintala, and T. D. Hämäläinen, “Leveraging Modern C++ in High-Level Synthesis,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 42, no. 4, pp. 1123–1132, Apr. 2023, doi: 10.1109/TCAD.2022.3193646.
- [18] K. Xu, G. L. Zhang, X. Yin, C. Zhuo, U. Schlichtmann, and B. Li, “HLSRewriter: Efficient Refactoring and Optimization of C/C++ Code with LLMs for High-Level Synthesis,” *ACM Trans Autom Electron Syst*, Jul. 2025, doi: 10.1145/3749986.
- [19] D. R. Kirk, *Deciphering Object-Oriented Programming with C++: A practical, in-depth guide to implementing object-oriented design principles to create robust code*. Packt Publishing Ltd, 2022.
- [20] B. Wang et al., “A Comprehensive Study of OOP-Related Bugs in C++ Compilers,” *IEEE Trans. Softw. Eng.*, vol. 51, no. 6, pp. 1762–1782, Jun. 2025, doi: 10.1109/TSE.2025.3566490.
- [21] “Structures in C++,” GeeksforGeeks. Accessed: Feb. 04, 2026. [Online]. Available: <https://www.geeksforgeeks.org/cpp/structures-in-cpp/>
- [22] S. N. Mohanty and P. K. Tripathy, *Data Structure and Algorithms Using C++: A Practical Implementation*. John Wiley & Sons, 2021.
- [23] P. Fatima, “Optimizing Algorithm Efficiency through Advanced Data Structures in C++: A Comparative Analysis of Performance, Scalability, and Complexity,” *Int. J. Comput. Inf. Manuf. IJCIM*, vol. 3, no. 2, pp. 66–72, Dec. 2023, doi: 10.54489/ijcim.v3i2.256.
- [24] J. Farrier, *Data Structures and Algorithms with the C++ STL: A guide for modern C++ practitioners*. Packt Publishing Ltd, 2024.
- [25] T. Gupta and A. Kembhavi, “Visual Programming: Compositional Visual Reasoning Without Training,” presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 14953–14962. Accessed: Feb. 04, 2026. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2023/html/Gupta_Visual_Programming_Compositional_Visual_Reasoning_Without_Training_CVPR_2023_paper.html
- [26] D. Suris, S. Menon, and C. Vondrick, “ViperGPT: Visual Inference via Python Execution for Reasoning,” presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 11888–11898. Accessed: Feb. 04, 2026. [Online]. Available: https://openaccess.thecvf.com/content/ICCV2023/html/Suris_ViperGPT_Visual_Inference_via_Python_Execution_for_Reasoning_ICCV_2023_paper.html
- [27] “what is a modularization?**,” HackMD. Accessed: Feb. 04, 2026. [Online]. Available: <https://hackmd.io/@alaa22/HJ3QOFpTi>
- [28] B. Stroustrup, *A Tour of C++*. Addison-Wesley Professional, 2022.
- [29] “Branching and If-constructs | Data Science with MATLAB.” Accessed: Feb. 04, 2026. [Online]. Available: <https://www.cdslab.org/matlab/notes/control-constructs/branching/index.html>
- [30] “Loops in R (for, while, repeat),” GeeksforGeeks. Accessed: Feb. 04, 2026. [Online]. Available: <https://www.geeksforgeeks.org/r-language/loops-in-r-for-while-repeat/>