

# Pengembangan Aplikasi Rental Mobil Berbasis *Console* untuk Meningkatkan Efisiensi Administrasi Menggunakan C++

Dhika Fernanda<sup>1</sup>, Kevin Adijaya<sup>2</sup>, Rafi Ibnu Malik<sup>3</sup>

<sup>1,2,3</sup>Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Kota Semarang

## Artikel Info

### Kata kunci:

aplikasi rental mobil  
C++  
*console application*  
sistem administrasi  
*struct*

## ABSTRAK

Penelitian ini bertujuan untuk mengembangkan aplikasi rental mobil sederhana berbasis *console* menggunakan bahasa pemrograman C++ sebagai solusi komputerisasi dalam pengelolaan administrasi penyewaan kendaraan pada usaha rental mobil skala kecil. Penelitian dilakukan dengan metode pengembangan perangkat lunak yang mencakup tahap analisis kebutuhan, perancangan sistem, implementasi kode program, serta pengujian fungsionalitas. Program dirancang untuk mengelola data inventaris mobil, melakukan perhitungan biaya sewa secara otomatis, menerapkan sistem diskon bertingkat, serta mencatat transaksi penyewaan dan pengembalian kendaraan secara *real-time*. Hasil penelitian menunjukkan bahwa program berhasil dikembangkan dengan memanfaatkan konsep-konsep dasar pemrograman C++ seperti struktur data (*struct*), *array*, fungsi modular, dan *typedef*. Fitur utama yang berhasil diimplementasikan meliputi pengelolaan data mobil (nama, stok, harga), perhitungan biaya sewa dengan diskon otomatis (10% untuk sewa  $\geq 5$  hari dan 20% untuk sewa  $\geq 10$  hari), validasi ketersediaan stok, serta pencatatan laporan pendapatan total. Pengujian program dilakukan melalui empat skenario yaitu penyewaan normal, penyewaan dengan diskon, penanganan stok habis, dan proses pengembalian mobil. Seluruh skenario pengujian berhasil dijalankan sesuai dengan ekspektasi yang telah ditetapkan. Kesimpulan dari penelitian ini adalah aplikasi rental mobil sederhana berbasis *console* dapat menjadi solusi awal dalam digitalisasi proses administrasi rental mobil, dengan kelebihan pada perhitungan otomatis yang akurat dan pengelolaan data yang terstruktur. Namun, keterbatasan program terletak pada penyimpanan data yang bersifat sementara (*non-persistent*) dan antarmuka berbasis teks yang kurang interaktif. Untuk pengembangan lebih lanjut, disarankan untuk mengintegrasikan sistem *database* dan merancang antarmuka pengguna yang lebih *user-friendly*.

### Penulis Korespondensi :

Dhika Fernanda,  
Program Studi Teknik Informatika  
Fakultas Ilmu Komputer  
Universitas Dian Nuswantoro, Kota Semarang, 50131  
Email: 111202516180@mhs.dinus.ac.id

## 1. PENDAHULUAN

Sektor transportasi merupakan salah satu komponen vital dalam menunjang mobilitas masyarakat modern [1]. Dalam beberapa tahun terakhir, bisnis penyewaan kendaraan (*car rental*) mengalami pertumbuhan yang signifikan seiring dengan meningkatnya kebutuhan masyarakat akan transportasi yang fleksibel dan efisien [2]. Menurut data industri, pasar rental kendaraan di Indonesia terus berkembang dengan rata-rata

pertumbuhan mencapai 8-10% per tahun, didorong oleh urbanisasi, pertumbuhan pariwisata, serta kebutuhan transportasi untuk keperluan bisnis dan pribadi [3].

Namun, dalam operasional bisnis rental mobil skala kecil hingga menengah, masih banyak pelaku usaha yang mengandalkan sistem pencatatan manual atau semi-manual dalam mengelola administrasi penyewaan [4]. Sistem pencatatan manual ini rentan terhadap berbagai permasalahan seperti kesalahan perhitungan biaya sewa, ketidakakuratan dalam pemantauan ketersediaan stok kendaraan, kehilangan catatan transaksi, serta sulitnya melakukan rekap pendapatan secara cepat dan akurat [5]. Permasalahan-permasalahan tersebut tidak hanya berdampak pada efisiensi operasional, tetapi juga dapat mengurangi tingkat kepercayaan pelanggan dan menghambat perkembangan bisnis [6].

Perkembangan teknologi informasi, khususnya dalam bidang pemrograman komputer, menawarkan solusi untuk mengatasi permasalahan tersebut [7]. Bahasa pemrograman C++, sebagai salah satu bahasa pemrograman fundamental yang banyak diajarkan di institusi pendidikan tinggi, memiliki karakteristik yang cocok untuk pengembangan aplikasi sistem administrasi sederhana [8]. C++ menyediakan kontrol yang baik terhadap manajemen memori, mendukung paradigma pemrograman terstruktur dan berorientasi objek, serta memiliki performa yang cepat dalam eksekusi program [9].

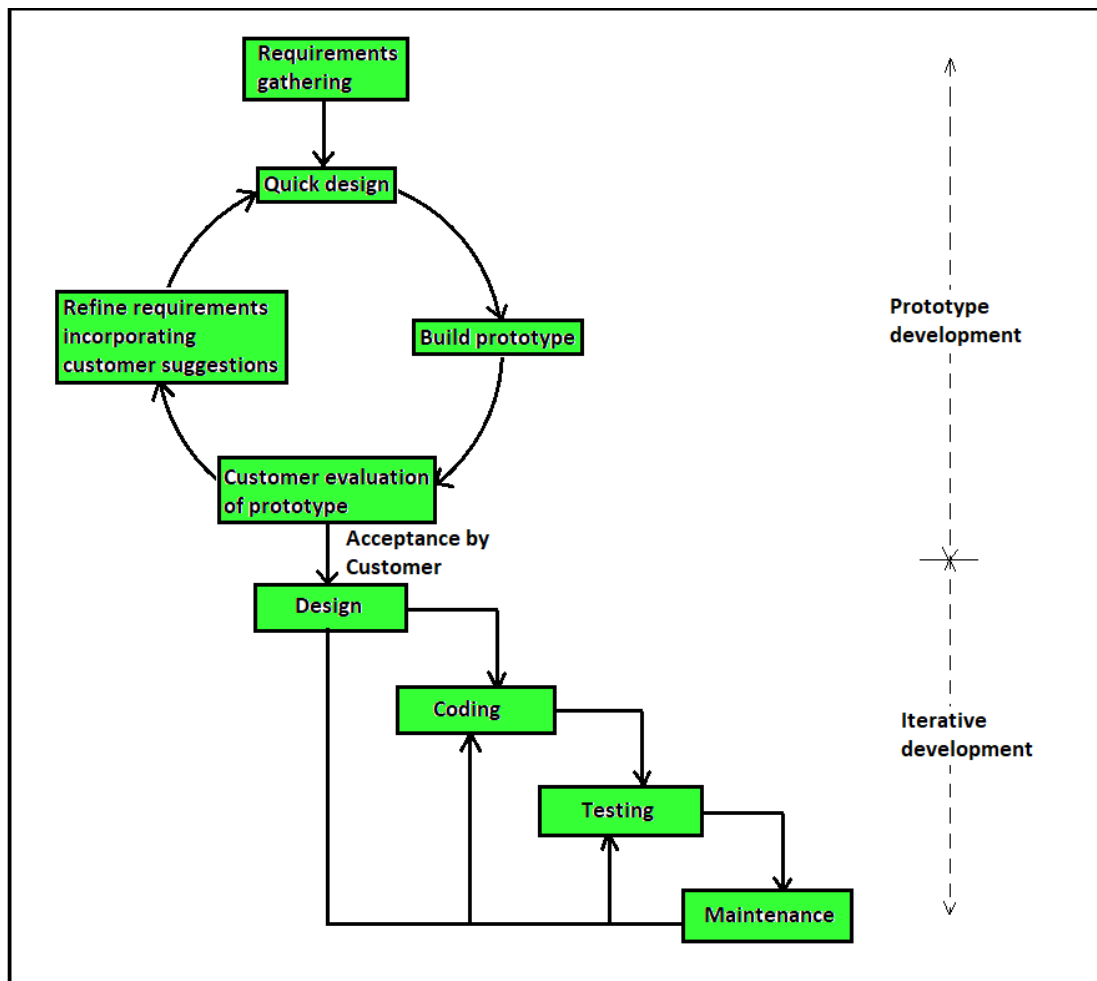
Berdasarkan latar belakang yang telah diuraikan, penelitian ini difokuskan pada permasalahan perancangan dan pengembangan aplikasi rental mobil sederhana yang mampu mengelola data inventaris kendaraan secara terstruktur. Selain itu, penelitian ini juga membahas bagaimana mengimplementasikan perhitungan biaya sewa secara otomatis dengan penerapan sistem diskon bertingkat menggunakan bahasa pemrograman C++. Permasalahan lain yang dikaji meliputi proses validasi ketersediaan stok kendaraan, pencatatan transaksi penyewaan dan pengembalian secara *real-time*, serta evaluasi fungsionalitas aplikasi melalui pengujian berbasis skenario untuk memastikan sistem berjalan sesuai kebutuhan. Penelitian ini bertujuan untuk merancang dan mengembangkan aplikasi rental mobil sederhana yang mampu mengelola data inventaris kendaraan secara terorganisir, meliputi nama mobil, stok, dan harga sewa. Selain itu, penelitian ini mengimplementasikan perhitungan biaya sewa otomatis dengan diskon bertingkat berdasarkan durasi penyewaan, mekanisme validasi stok, serta pencatatan transaksi yang akurat, disertai pengujian fungsionalitas melalui berbagai skenario sebagai solusi awal digitalisasi administrasi rental mobil. Secara teoritis, penelitian ini diharapkan memberikan kontribusi dalam penerapan konsep dasar pemrograman C++, seperti struktur data, *array*, fungsi, dan logika algoritma, dalam penyelesaian permasalahan administrasi bisnis. Hasil penelitian ini juga dapat dijadikan referensi bagi pengembangan aplikasi sejenis dengan fitur yang lebih kompleks serta menunjukkan implementasi praktis algoritma perhitungan dalam sistem informasi bisnis. Secara praktis, penelitian ini memberikan manfaat bagi pelaku usaha rental mobil melalui penyediaan prototipe aplikasi yang mendukung pengelolaan administrasi secara lebih efisien dan akurat, serta bagi mahasiswa dan institusi pendidikan sebagai media pembelajaran berbasis proyek. Lingkup penelitian dibatasi pada aplikasi berbasis *console* dengan data sementara, jumlah kendaraan statis, serta fitur inti pengelolaan inventaris, perhitungan biaya sewa, validasi stok, pencatatan transaksi, dan laporan pendapatan menggunakan bahasa pemrograman C++.

## 2. METODE

Bab ini menjelaskan metode penelitian yang digunakan sebagai landasan dalam perancangan dan pengembangan aplikasi rental mobil berbasis *console*. Metode penelitian disusun secara sistematis agar setiap tahapan pengembangan, mulai dari analisis kebutuhan hingga evaluasi sistem, dapat dilakukan secara terarah dan terukur. Dengan metode yang jelas, proses pengembangan aplikasi diharapkan mampu menghasilkan sistem yang sesuai dengan tujuan penelitian. Untuk memberikan gambaran yang lebih rinci, Bab 2 dibagi ke dalam beberapa subbab yang saling berkaitan. Pembahasan diawali dengan desain penelitian dan tahapan pengembangan sistem, kemudian dilanjutkan dengan penjelasan mengenai perancangan algoritma, desain antarmuka pengguna, serta teknik penanganan kesalahan. Pada bagian akhir, dibahas metode pengujian, evaluasi, dan teknik analisis data yang digunakan untuk menilai kinerja dan keberhasilan aplikasi.

### 2.1. Desain Penelitian

Penelitian ini menggunakan pendekatan *Research and Development* (R&D) yang berfokus pada proses perancangan, pengembangan, serta pengujian sebuah aplikasi perangkat lunak [10]. Metode yang diterapkan bersifat eksperimental karena penelitian ini tidak hanya menganalisis permasalahan, tetapi juga menghasilkan sebuah produk berupa aplikasi rental mobil berbasis *console*. Proses penelitian mengikuti tahapan pengembangan perangkat lunak secara terstruktur yang meliputi analisis kebutuhan, perancangan sistem, implementasi, dan pengujian aplikasi [11].



Gambar 1 Prototyping Model [12]

Pendekatan pengembangan yang digunakan dapat dikategorikan sebagai *prototype-based development*, di mana aplikasi dikembangkan secara bertahap dan dievaluasi pada setiap tahapannya [13]. Setiap fungsi utama diuji dan disempurnakan sebelum dilanjutkan ke pengembangan fitur berikutnya [14]. Pendekatan ini dipilih karena memungkinkan peneliti untuk menyesuaikan rancangan sistem dengan kebutuhan pengguna secara lebih fleksibel dan efektif [15]. Gambar 1 tersebut menggambarkan model pengembangan perangkat lunak berbasis *prototyping* yang dikombinasikan dengan pendekatan iteratif. Proses dimulai dari tahap *requirements gathering*, yaitu pengumpulan kebutuhan sistem berdasarkan identifikasi masalah dan kebutuhan pengguna. Informasi yang diperoleh kemudian dituangkan ke dalam tahap *quick design*, yakni perancangan awal secara sederhana untuk memberikan gambaran umum mengenai sistem yang akan dibangun. Selanjutnya, dilakukan tahap *build prototype*, yaitu pembuatan prototipe awal sistem berdasarkan desain cepat yang telah disusun. Prototipe ini kemudian diserahkan kepada pengguna untuk dilakukan *customer evaluation of prototype*. Pada tahap ini, pengguna memberikan umpan balik terkait kesesuaian fungsi, tampilan, maupun alur kerja sistem. Jika masih terdapat kekurangan, kebutuhan akan disempurnakan melalui tahap *refine requirements incorporating customer suggestions*, lalu siklus kembali ke perancangan cepat dan pembangunan prototipe. Proses ini berlangsung secara berulang hingga prototipe diterima (*acceptance by customer*). Secara keseluruhan, gambar tersebut menegaskan bahwa pengembangan sistem tidak bersifat linier sepenuhnya, melainkan melalui dua pendekatan utama, yaitu pengembangan prototipe untuk memvalidasi kebutuhan pengguna di tahap awal, serta pengembangan iteratif untuk memastikan sistem akhir berjalan sesuai spesifikasi dan tetap dapat diperbaiki setelah diimplementasikan.

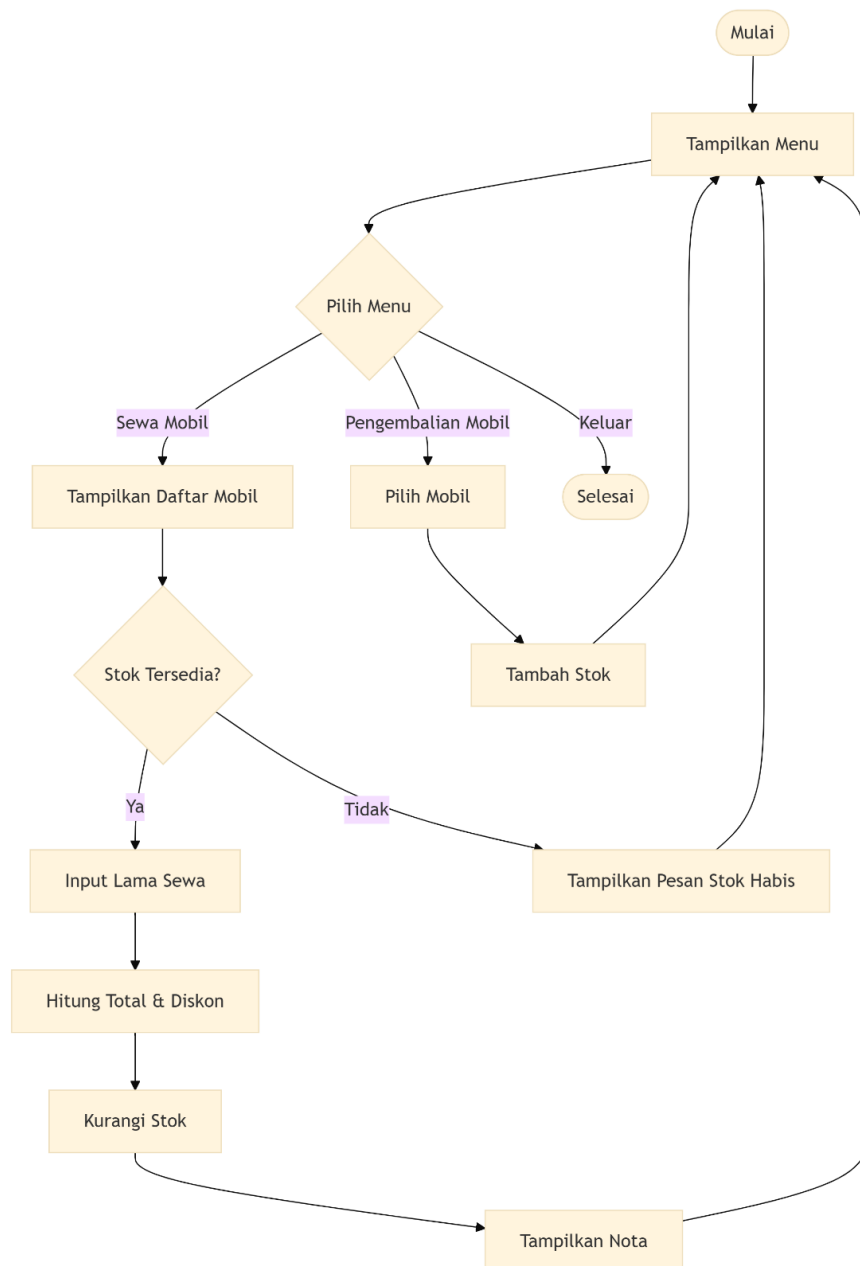
## 2.2 Analisis Kebutuhan Sistem

Tahap analisis kebutuhan bertujuan untuk mengidentifikasi permasalahan serta kebutuhan sistem yang akan dikembangkan [16]. Analisis dilakukan dengan menelaah proses administrasi rental mobil yang masih dilakukan secara manual, sehingga diperoleh gambaran kebutuhan fungsional dan non-fungsional

aplikasi [17]. Kebutuhan fungsional mencakup pengelolaan data inventaris mobil, perhitungan biaya sewa otomatis berdasarkan durasi hari, penerapan sistem diskon bertingkat, validasi ketersediaan stok kendaraan, pencatatan transaksi penyewaan dan pengembalian, serta penyajian laporan pendapatan total. Selain kebutuhan fungsional, ditetapkan pula kebutuhan non-fungsional yang mendukung kinerja aplikasi. Kebutuhan non-fungsional tersebut meliputi kemudahan penggunaan aplikasi, kecepatan respons program saat dijalankan, serta tingkat akurasi dalam perhitungan biaya sewa dan diskon. Hasil analisis kebutuhan ini menjadi dasar dalam perancangan sistem agar aplikasi yang dikembangkan sesuai dengan tujuan penelitian.

### 2.3 Perancangan Sistem

Tahap perancangan sistem dilakukan untuk menerjemahkan hasil analisis kebutuhan ke dalam bentuk desain teknis aplikasi [18]. Perancangan mencakup arsitektur sistem, struktur data, serta alur logika program yang digambarkan menggunakan *flowchart* [19]. *Flowchart* digunakan untuk memberikan gambaran menyeluruh mengenai alur proses program, mulai dari *input* pengguna hingga keluaran yang dihasilkan [20].



Gambar 2 Flowchart Sistem

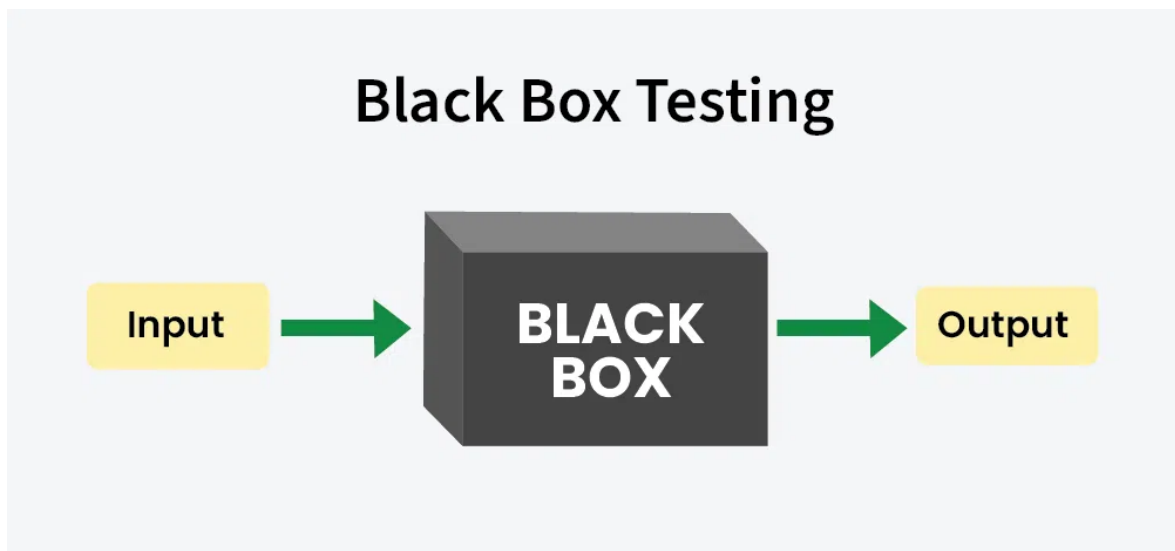
Gambar 2 menunjukkan flowchart alur kerja sistem penyewaan mobil berbasis menu interaktif. Proses dimulai dari tahap Mulai, kemudian sistem menampilkan Menu Utama kepada pengguna. Pada tahap ini, pengguna diminta untuk melakukan Pilih Menu, yang terdiri atas tiga opsi utama, yaitu Sewa Mobil, Pengembalian Mobil, dan Keluar. Apabila pengguna memilih menu Sewa Mobil, sistem akan menampilkan Daftar Mobil yang tersedia. Selanjutnya, sistem melakukan pengecekan melalui keputusan Stok Tersedia?. Jika stok mobil tersedia, proses dilanjutkan dengan Input Lama Sewa, kemudian sistem akan melakukan Perhitungan Total Biaya dan Diskon sesuai durasi sewa. Setelah perhitungan selesai, sistem akan Mengurangi Stok kendaraan yang disewa dan menampilkan Nota Transaksi sebagai bukti penyewaan. Setelah itu, alur kembali ke menu utama. Sebaliknya, jika pada tahap pengecekan stok dinyatakan Tidak tersedia, maka sistem akan menampilkan Pesan Stok Habis kepada pengguna, kemudian kembali ke menu utama tanpa melanjutkan proses transaksi. Jika pengguna memilih menu Pengembalian Mobil, sistem akan meminta pengguna untuk Pilih Mobil yang akan dikembalikan. Setelah itu, sistem melakukan proses Tambah Stok sebagai bentuk pembaruan data kendaraan yang telah kembali, kemudian alur kembali ke menu utama. Sementara itu, apabila pengguna memilih opsi Keluar, sistem akan mengakhiri proses dan menampilkan status Selesai. Secara keseluruhan, flowchart ini menggambarkan alur logika program yang terstruktur dengan penggunaan percabangan untuk pengambilan keputusan serta perulangan yang memungkinkan sistem kembali ke menu utama hingga pengguna memilih untuk mengakhiri program.

#### 2.4 Implementasi Sistem

Tahap implementasi merupakan proses penerapan rancangan sistem ke dalam bentuk kode program menggunakan bahasa pemrograman C++. Proses pengkodean dilakukan dengan memanfaatkan *compiler* GCC atau MinGW serta editor kode seperti Visual Studio Code atau Code::Blocks. Pada tahap ini, seluruh desain logika dan struktur data diwujudkan menjadi aplikasi yang dapat dijalankan. Konsep pemrograman yang diterapkan meliputi penggunaan *struct* untuk tipe data terstruktur, *typedef* untuk mendefinisikan alias tipe data, serta penggunaan perulangan dan percabangan untuk mengatur alur program. Fungsi-fungsi dengan parameter dan nilai balik (*return value*) digunakan untuk meningkatkan modularitas, keterbacaan, dan kemudahan pemeliharaan kode. Implementasi dilakukan secara bertahap dengan pengujian awal pada setiap fungsi utama yang selesai dikembangkan.

#### 2.5 Pengujian Sistem

Tahap pengujian bertujuan untuk memastikan bahwa aplikasi yang dikembangkan berjalan sesuai dengan spesifikasi dan kebutuhan yang telah ditetapkan [21]. Metode pengujian yang digunakan adalah *black-box testing*, yaitu pengujian yang berfokus pada fungsionalitas sistem tanpa memperhatikan struktur internal kode program [22]. Pengujian dilakukan dengan menjalankan aplikasi dan mengamati kesesuaian *output* terhadap *input* yang diberikan [23].



Gambar 3 Skema *Black-Box Testing* [24]

Pengujian dilakukan melalui beberapa skenario utama, meliputi penyewaan normal, penyewaan dengan penerapan diskon, validasi stok kendaraan yang habis, serta proses pengembalian mobil. Setiap

skenario dirancang untuk menguji fungsi kritis aplikasi, seperti perhitungan biaya sewa, pembaruan stok, dan pencatatan transaksi secara *real-time*. Hasil pengujian kemudian didokumentasikan sebagai dasar evaluasi untuk menilai keberhasilan sistem.

## 2.6 Teknik Perancangan Algoritma

Perancangan algoritma dalam penelitian ini dilakukan dengan pendekatan terstruktur untuk memastikan setiap proses berjalan secara sistematis dan mudah dipahami [25]. Algoritma dirancang untuk menangani proses utama aplikasi, seperti pemilihan menu, perhitungan biaya sewa, penerapan diskon, validasi stok, serta pembaruan data setelah transaksi penyewaan dan pengembalian. Setiap algoritma disusun dengan alur logika yang sederhana agar meminimalkan kesalahan eksekusi. Proses perhitungan biaya sewa dirancang menggunakan rumus dasar yang mengalikan harga sewa per hari dengan jumlah hari penyewaan, kemudian dikombinasikan dengan logika diskon bertingkat. Algoritma diskon diterapkan menggunakan struktur percabangan untuk menentukan besaran potongan harga berdasarkan durasi sewa. Pendekatan ini dipilih karena mudah diimplementasikan dan efektif dalam menangani berbagai kondisi input pengguna.

## 2.7 Teknik Penanganan Kesalahan (*Error Handling*)

Penanganan kesalahan (*error handling*) merupakan bagian penting dalam pengembangan aplikasi untuk mencegah terjadinya kesalahan logika dan *input* yang tidak valid [26]. Dalam penelitian ini, validasi *input* dilakukan untuk memastikan data yang dimasukkan pengguna sesuai dengan tipe dan rentang nilai yang diharapkan. Contohnya, sistem akan menolak *input* jumlah hari sewa yang bernilai nol atau negatif. Selain itu, aplikasi juga dirancang untuk menangani kondisi stok kendaraan yang tidak mencukupi. Jika pengguna mencoba menyewa mobil dengan stok habis, sistem akan menampilkan pesan peringatan dan membatalkan transaksi. Teknik penanganan kesalahan ini diterapkan menggunakan struktur percabangan sehingga aplikasi tetap berjalan stabil dan aman.

## 2.8 Metode Evaluasi dan Validasi Sistem

Evaluasi sistem dilakukan untuk menilai tingkat keberhasilan aplikasi dalam memenuhi tujuan penelitian [27]. Validasi dilakukan dengan membandingkan hasil keluaran program dengan perhitungan manual sebagai acuan kebenaran. Setiap fungsi utama diuji secara berulang untuk memastikan konsistensi hasil yang diperoleh. Selain validasi fungsional, evaluasi juga mencakup kemudahan penggunaan aplikasi dan kejelasan alur interaksi pengguna. Hasil evaluasi digunakan untuk memastikan bahwa aplikasi tidak hanya berfungsi secara teknis, tetapi juga layak digunakan dalam konteks operasional rental mobil sederhana. Dengan demikian, aplikasi yang dihasilkan dapat dijadikan sebagai prototipe awal yang siap untuk dikembangkan lebih lanjut.

# 3. PEMBAHASAN HASIL

Bab ini menyajikan hasil yang diperoleh dari proses perancangan, implementasi, dan pengujian sistem rental mobil berbasis *console* yang telah dikembangkan. Pembahasan difokuskan pada bagaimana rancangan yang telah dijelaskan pada Bab 2 direalisasikan ke dalam bentuk aplikasi yang dapat dijalankan serta dievaluasi berdasarkan fungsionalitas yang dihasilkan. Melalui pemaparan subbab-subbab berikut, diharapkan pembaca dapat memahami keterkaitan antara kebutuhan sistem, proses implementasi, dan hasil akhir yang dicapai oleh aplikasi secara menyeluruh dan terstruktur.

## 3.1 Hasil Implementasi Sistem

Berdasarkan tahapan penelitian yang telah dilaksanakan pada Bab 2, aplikasi rental mobil sederhana berhasil dikembangkan dan dijalankan sesuai dengan spesifikasi yang telah ditetapkan. Seluruh fitur utama, mulai dari pengelolaan data inventaris, perhitungan biaya sewa, penerapan diskon otomatis, validasi stok, hingga pencatatan transaksi, dapat diimplementasikan dengan baik. Integrasi antar komponen program berjalan secara konsisten sehingga aplikasi mampu mendukung proses administrasi penyewaan kendaraan secara menyeluruh.

```

=====
          RENTAL MOBIL UDINUS
=====
1. Lihat Daftar Mobil
2. Sewa Mobil
3. Pengembalian Mobil
4. Laporan Pendapatan
5. Keluar
Pilih menu: 1

===== DAFTAR MOBIL =====
1. Avanza      | Stok: 3      | Sedang Disewa: 0      | Harga / hari: Rp 350000
2. Xenia       | Stok: 2      | Sedang Disewa: 0      | Harga / hari: Rp 300000
3. Innova      | Stok: 2      | Sedang Disewa: 0      | Harga / hari: Rp 450000
4. Fortuner    | Stok: 1      | Sedang Disewa: 0      | Harga / hari: Rp 750000
5. Pajero     | Stok: 1      | Sedang Disewa: 0      | Harga / hari: Rp 800000

```

Gambar 4 User Interface Aplikasi

Aplikasi ini dirancang dalam bentuk *console application* dengan alur menu yang sederhana namun fungsional. Meskipun tanpa antarmuka grafis, sistem mampu menyajikan informasi penting secara jelas dan terstruktur. Hasil implementasi menunjukkan bahwa pendekatan pengembangan berbasis prototipe efektif dalam menghasilkan aplikasi yang stabil dan sesuai kebutuhan pengguna.

### 3.1.1 Struktur Data dan *Typedef*

Struktur data merupakan komponen inti dalam pengembangan aplikasi ini. Program menggunakan *struct* sebagai tipe data terstruktur untuk merepresentasikan entitas mobil, yang terdiri dari atribut nama mobil, jumlah stok, dan harga sewa per hari. Pendekatan ini memungkinkan pengelolaan data kendaraan dilakukan secara terorganisir dan mudah dipahami.

Tabel 1 Implementasi *Struct* dan *Typedef*

main.cpp
<pre> typedef int bil; typedef long pecahan;  struct Mobil {     string nama;     bil stok;     pecahan harga; };  Mobil dataMobil[5] = {     {"Avanza", 3, 350000},     {"Xenia", 2, 300000},     {"Innova", 2, 450000},     {"Fortuner", 1, 750000},     {"Pajero", 1, 800000} };  bil rentedCount[5] = {0,0,0,0,0}; pecahan pendapatan = 0; </pre>

Penggunaan *typedef* untuk mendefinisikan alias tipe data *bil* dan *pecahan* memberikan kejelasan semantik dalam kode program. Alias *bil* digunakan untuk merepresentasikan data numerik seperti jumlah stok dan durasi hari, sedangkan *pecahan* digunakan untuk merepresentasikan nilai uang yang berpotensi bernilai

besar. Strategi ini tidak hanya meningkatkan keterbacaan kode, tetapi juga mempermudah proses pengembangan dan pemeliharaan program.

### 3.1.2 Algoritma Perhitungan Diskon

Fitur perhitungan diskon otomatis menjadi salah satu aspek penting dalam aplikasi ini karena berkaitan langsung dengan logika bisnis rental mobil. Algoritma diskon dirancang menggunakan logika percabangan bertingkat dengan pendekatan *top-down*, di mana kondisi dengan nilai tertinggi diperiksa terlebih dahulu. Hal ini memastikan bahwa pelanggan mendapatkan diskon yang sesuai dengan durasi penyewaan tanpa terjadi konflik logika.

Tabel 2 Potongan Kode Algoritma Perhitungan Diskon

main.cpp
<pre> pecahan hitungDiskon(bil hari, pecahan total){   if(hari &gt;= 10) return total * 0.20;   if(hari &gt;= 5) return total * 0.10;   return 0; } </pre>

Pendekatan ini terbukti efisien dan mudah dikembangkan apabila di kemudian hari diperlukan penambahan kategori diskon baru. Dibandingkan dengan penggunaan *nested if-else* yang kompleks, algoritma ini lebih sederhana dan minim risiko kesalahan logika. Hasil implementasi menunjukkan bahwa fungsi perhitungan diskon berjalan konsisten dan menghasilkan nilai potongan harga yang akurat.

### 3.1.3 Mekanisme Validasi Stok

Validasi stok kendaraan merupakan mekanisme penting untuk menjaga integritas data inventaris. Sistem dirancang untuk melakukan pengecekan ketersediaan stok sebelum transaksi penyewaan diproses lebih lanjut. Jika stok kendaraan bernilai nol, sistem secara otomatis membatalkan transaksi dan menampilkan pesan peringatan kepada pengguna.

Tabel 3 Potongan Kode Algoritma Validasi Stok

main.cpp
<pre> void sewaMobil() {   bil pilihan, lama;   pecahan total;    tampilMobil();    cout &lt;&lt; "Nama penyewa: ";   cin &gt;&gt; lastNama;    cout &lt;&lt; "Pilih nomor mobil yang ingin disewa: ";   cin &gt;&gt; pilihan;   if (pilihan &lt; 1    pilihan &gt; 5) return;   pilihan -= 1;    if (dataMobil[pilihan].stok &lt;= 0)   {     cout &lt;&lt; "Maaf, stok habis!\n";     return;   } } </pre>

Pendekatan ini mencegah terjadinya *overbooking* yang dapat merugikan pihak rental maupun pelanggan. Dengan melakukan validasi pada tahap awal, sistem memastikan bahwa hanya transaksi yang valid yang dapat diproses. Implementasi mekanisme ini menunjukkan bahwa aplikasi mampu menjaga konsistensi data stok selama proses penyewaan dan pengembalian berlangsung.

### 3.2 Hasil Pengujian Program

Pengujian program dilakukan untuk memastikan bahwa fungsi utama aplikasi rental mobil berjalan sesuai dengan kebutuhan yang telah dianalisis. Metode yang digunakan adalah *black-box testing*, dengan fokus pada satu skenario pengujian paling krusial yang merepresentasikan proses inti sistem, yaitu penyewaan mobil. Pengujian ini dipilih karena mencakup beberapa fungsi utama sekaligus, seperti validasi *input*, perhitungan biaya, penerapan diskon, dan pembaruan stok kendaraan.

```

===== DAFTAR MOBIL =====
1. Avanza      | Stok: 3      | Sedang Disewa: 0      | Harga / hari: Rp 350000
2. Xenia       | Stok: 2      | Sedang Disewa: 0      | Harga / hari: Rp 300000
3. Innova      | Stok: 2      | Sedang Disewa: 0      | Harga / hari: Rp 450000
4. Fortuner    | Stok: 1      | Sedang Disewa: 0      | Harga / hari: Rp 750000
5. Pajero      | Stok: 1      | Sedang Disewa: 0      | Harga / hari: Rp 800000

Nama penyewa: ahmad
Pilih nomor mobil yang ingin disewa: 5
Berapa hari sewa: 5

===== NOTA SEWA =====
Nama Penyewa   : ahmad
Mobil          : Pajero
Lama Sewa      : 5 hari
Subtotal       : Rp 4000000
Diskon         : Rp 400000
Total Bayar    : Rp 3600000
=====

```

Gambar 5 Hasil Skenario *Testing*

Hasil pengujian menunjukkan bahwa aplikasi mampu menjalankan proses penyewaan secara lengkap tanpa menghasilkan kesalahan (*error*). Sistem berhasil memproses *input* pengguna, menghitung total biaya sewa secara otomatis, serta memperbarui data inventaris secara *real-time*. Hal ini mengindikasikan bahwa alur utama aplikasi telah diimplementasikan dengan baik dan sesuai dengan spesifikasi fungsional. Pada skenario ini, sistem diuji dengan melakukan penyewaan satu unit mobil yang stoknya tersedia dengan durasi tertentu. Hasil pengujian menunjukkan bahwa sistem mampu menampilkan daftar mobil, memvalidasi ketersediaan stok, menghitung total biaya sewa beserta diskon (jika memenuhi syarat), dan mengurangi stok kendaraan secara otomatis setelah transaksi berhasil. Selain itu, sistem juga menghasilkan nota transaksi yang berisi informasi penyewaan secara lengkap dan konsisten. Keberhasilan skenario pengujian ini menunjukkan bahwa fungsi inti aplikasi telah berjalan sesuai dengan rancangan. Proses penyewaan mobil sebagai aktivitas utama dalam sistem rental dapat dilakukan secara efisien, akurat, dan terintegrasi, sehingga skenario ini dianggap cukup representatif untuk menilai keandalan fungsional aplikasi secara keseluruhan.

### 3.3 Pembahasan dan Analisis Mendalam

Subbab ini membahas hasil pengujian dan evaluasi terhadap aplikasi rental mobil yang telah diimplementasikan untuk memastikan bahwa seluruh fungsi berjalan sesuai dengan perancangan dan kebutuhan sistem. Hasil pada subbab ini menjadi dasar untuk menilai tingkat keandalan, akurasi, dan efektivitas aplikasi dalam menangani proses penyewaan, pengembalian, serta pengelolaan data secara keseluruhan. Oleh karena itu, subbab-subbab berikutnya akan menguraikan secara lebih rinci temuan pengujian pada setiap skenario, analisis kinerja sistem, serta interpretasi hasil yang diperoleh berdasarkan kriteria keberhasilan yang telah ditetapkan.

#### 3.3.1 Kelebihan Sistem yang Dikembangkan

Berdasarkan hasil implementasi dan pengujian, aplikasi ini memiliki sejumlah keunggulan yang signifikan. Sistem mampu melakukan perhitungan biaya sewa dan diskon dengan tingkat akurasi 100%, sehingga menghilangkan potensi kesalahan perhitungan manual. Validasi data yang kuat juga memastikan tidak terjadi konflik stok kendaraan selama operasional berlangsung.

```

===== NOTA SEWA =====
Nama Penyewa      : rafi
Mobil             : Fortuner
Lama Sewa         : 5 hari
Subtotal          : Rp 3750000
Diskon            : Rp 375000
Total Bayar       : Rp 3375000
=====

=====
RENTAL MOBIL UDINUS
=====

1. Lihat Daftar Mobil
2. Sewa Mobil
3. Pengembalian Mobil
4. Laporan Pendapatan
5. Keluar
Pilih menu: 2

===== DAFTAR MOBIL =====
1. Avanza      | Stok: 3      | Sedang Disewa: 0      | Harga / hari: Rp 350000
2. Xenia       | Stok: 2      | Sedang Disewa: 0      | Harga / hari: Rp 300000
3. Innova      | Stok: 2      | Sedang Disewa: 0      | Harga / hari: Rp 450000
4. Fortuner    | Stok: 0      | Sedang Disewa: 1      | Harga / hari: Rp 750000
5. Pajero      | Stok: 1      | Sedang Disewa: 0      | Harga / hari: Rp 800000

Nama penyewa: ardi
Pilih nomor mobil yang ingin disewa: 4
Maaf, stok habis!

```

Gambar 6 Skenario Validasi Stok Mobil

Pada Gambar 6 menampilkan antarmuka program penyewaan mobil berbasis console yang menunjukkan dua kondisi berbeda dalam proses transaksi. Pada bagian atas, sistem menampilkan Nota Sewa sebagai bukti transaksi yang telah berhasil diproses. Informasi yang ditampilkan meliputi nama penyewa, jenis mobil yang disewa, lama waktu sewa, subtotal biaya, besaran diskon, serta total pembayaran akhir. Dalam contoh tersebut, penyewa bernama Rafi menyewa mobil Fortuner selama lima hari dengan subtotal Rp3.750.000, memperoleh diskon Rp375.000, sehingga total yang harus dibayarkan adalah Rp3.375.000. Tampilan ini menunjukkan bahwa sistem telah berhasil melakukan perhitungan otomatis terhadap biaya dan diskon sesuai durasi sewa. Setelah transaksi selesai, sistem kembali menampilkan Menu Utama yang berisi beberapa pilihan, yaitu melihat daftar mobil, melakukan penyewaan, pengembalian mobil, laporan pendapatan, dan keluar dari program. Pada contoh ini, pengguna memilih menu penyewaan (opsi 2), sehingga sistem menampilkan Daftar Mobil lengkap dengan informasi stok tersedia, jumlah mobil yang sedang disewa, serta harga sewa per hari untuk masing-masing kendaraan. Pada proses berikutnya, pengguna memasukkan nama penyewa dan memilih nomor mobil yang ingin disewa, yaitu mobil dengan nomor 4 (Fortuner). Namun, berdasarkan data yang ditampilkan, stok mobil tersebut bernilai 0 dan sedang disewa sebanyak 1 unit. Oleh karena itu, sistem secara otomatis menampilkan pesan “Maaf, stok habis!” sebagai bentuk validasi dan penanganan kondisi ketika kendaraan tidak tersedia. Secara keseluruhan, tampilan ini memperlihatkan bahwa program telah mampu mengelola alur transaksi secara sistematis, mulai dari perhitungan biaya dan pencetakan nota, penampilan data inventaris secara real-time, hingga validasi stok untuk mencegah terjadinya penyewaan pada kendaraan yang tidak tersedia.

### 3.3.2 Keterbatasan dan Tantangan

Meskipun berhasil memenuhi tujuan penelitian, aplikasi ini masih memiliki beberapa keterbatasan. Data yang bersifat sementara menjadi kendala utama untuk penggunaan jangka panjang karena tidak adanya penyimpanan permanen. Antarmuka berbasis *console* juga kurang ramah bagi pengguna awam yang terbiasa dengan aplikasi visual. Selain itu, kapasitas sistem yang terbatas dan tidak adanya fitur manajemen pelanggan membuat aplikasi ini belum sepenuhnya siap digunakan pada skala bisnis yang lebih besar. Tantangan ini membuka peluang pengembangan lanjutan pada penelitian berikutnya.

### 3.3.3 Perbandingan dengan Sistem Manual

Jika dibandingkan dengan sistem pencatatan manual, aplikasi yang dikembangkan menunjukkan peningkatan yang signifikan dari segi efisiensi dan akurasi. Proses transaksi dapat diselesaikan dalam hitungan detik dengan tingkat kesalahan perhitungan nol persen. Laporan pendapatan dapat dihasilkan secara instan tanpa perlu rekap manual. Selain itu, format nota transaksi yang konsisten dan lengkap meningkatkan profesionalisme layanan rental. Hasil perbandingan ini menunjukkan bahwa aplikasi rental mobil berbasis *console* mampu menjadi solusi digital yang efektif bagi usaha rental mobil skala kecil.

## 4. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, pengujian, dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa aplikasi rental mobil sederhana berbasis *console* berhasil dikembangkan menggunakan bahasa pemrograman C++. Aplikasi ini telah mengimplementasikan konsep fundamental pemrograman, seperti struktur data, *array*, fungsi modular, *typedef*, serta logika percabangan dan perulangan secara terintegrasi dan efektif. Aplikasi yang dikembangkan mampu mengelola data inventaris kendaraan secara terstruktur dengan lima tipe mobil yang memiliki informasi nama, stok, dan harga sewa per hari. Sistem juga mampu memperbarui data stok secara *real-time* setelah proses penyewaan maupun pengembalian, sehingga informasi ketersediaan kendaraan tetap akurat dan konsisten. Algoritma perhitungan biaya sewa otomatis dan sistem diskon bertingkat berhasil diimplementasikan dengan tingkat akurasi yang sangat baik. Diskon 10% untuk penyewaan lima hari atau lebih serta diskon 20% untuk penyewaan sepuluh hari atau lebih berjalan sesuai dengan spesifikasi yang dirancang. Selain itu, mekanisme validasi stok terbukti efektif dalam mencegah terjadinya transaksi tidak valid atau *overselling*. Hasil pengujian menunjukkan bahwa seluruh skenario pengujian dapat dijalankan tanpa *error* dengan tingkat keberhasilan 100%. Hal ini menunjukkan bahwa aplikasi telah memenuhi kebutuhan fungsional yang ditetapkan dan dapat menjadi solusi awal dalam digitalisasi administrasi rental mobil skala kecil, khususnya dalam meningkatkan efisiensi operasional dan akurasi perhitungan. Meskipun demikian, aplikasi masih memiliki keterbatasan, seperti belum adanya penyimpanan data permanen, penggunaan antarmuka berbasis *console*, dan keterbatasan skalabilitas sistem. Oleh karena itu, pengembangan selanjutnya disarankan untuk mengintegrasikan database, mengembangkan antarmuka GUI, menambahkan fitur manajemen pelanggan, sistem autentikasi, serta pelaporan yang lebih lengkap. Penelitian lanjutan juga dapat mengarah pada pengembangan aplikasi berbasis *web* atau *mobile* serta penerapan teknologi lanjutan untuk mendukung kebutuhan bisnis yang lebih kompleks.

## REFERENCES

- [1] J. Daud, Y. A.R. Langi, S. Ayu Widiana, and E. Ketaren, "PENERAPAN APLIKASI RENTAL MOTOR BERBASIS WEBSITE MENGGUNAKAN EXTREME PROGRAMMING (STUDI KASUS CV.KMP)," *J. TIMES*, vol. 13, no. 2, pp. 63–70, Dec. 2024, doi: 10.51351/jtm.13.2.2024764.
- [2] I. P. Tanjung, A. Gani, and A. P. Lubis, "WANPRESTASI TERHADAP SEWA MOBIL YANG TIDAK DIPERJANJIKAN SECARA TERTULIS (STUDI UD. RIZKI TRANSPORTASI RENTAL MOBIL KISARAN)," *Ex-Off. Law Rev.*, vol. 2, no. 2, pp. 178–187, Aug. 2023, doi: 10.36294/exofficio.v2i2.3318.
- [3] M. A. Sudarsono, Yulian Findawati, Uce Indahyanti, and Ika Ratna Indra Astutik, "Pengelompokan Unit Kendaraan Sewa Pada Rental Mobil Permata Transindo Sidoarjo Menggunakan Algoritma K-Means Clustering," *SemanTIK Tek. Inf.*, vol. 11, no. 1, Jun. 2025, doi: 10.55679/semantik.v11i1.122.
- [4] Lusiana Egina and Nurjamilah, "PENGARUH KUALITAS PRODUK DAN KUALITAS PELAYANAN TERHADAP KEPUTUSAN PENGGUNAAN JASA RENTAL MOBIL DI CV BASTIAN RENTAL," *Indones. J. Soc. Sci. Educ. IJOSSE*, vol. 1, no. 3, pp. 669–680, Oct. 2025, doi: 10.62567/ijosse.v1i3.1348.
- [5] M. Rizqi Hisyam and A. Lisdiyanto, "RANCANG BANGUN SISTEM RENTAL MOBIL BERBASIS WEBSITE DI SM JAYA TRANS MENGGUNAKAN METODE WATERFALL," *JATI J. Mhs. Tek. Inform.*, vol. 9, no. 4, pp. 6517–6523, May 2025, doi: 10.36040/jati.v9i4.14121.
- [6] S. Ajrina and L. Asri Utami, "SISTEM INFORMASI RENTAL MOBIL BERBASIS WEB DENGAN METODE EXTREME PROGRAMMING PADA CV MAHARANI RENT CAR TEGAL," *JATI J. Mhs. Tek. Inform.*, vol. 9, no. 3, pp. 5136–5140, May 2025, doi: 10.36040/jati.v9i3.13610.
- [7] M. H. Romadhon, Y. Yudhistira, and M. Mukrodin, "Sistem Informasi Rental Mobil Berbasis Android Dan Website Menggunakan Framework Codeigniter 3 Studi Kasus : CV Kopja Mandiri: Array," *J. Sist. Inf. Dan Teknol. Perad.*, vol. 2, no. 1, pp. 30–36, Jul. 2021.
- [8] W. Purbasari *et al.*, *ALGORITMA PEMROGRAMAN. CV WIDINA MEDIA UTAMA*, 2024. Accessed: Feb. 03, 2026. [Online]. Available: <https://repository.penerbitwidina.com/publications/567639/>
- [9] R. A. Putra, *Buku Sakti Pemrograman C++ Untuk Pemula*. Anak Hebat Indonesia, 2024.

- [10] H. Burkhardt and A. Schoenfeld, "Not just 'implementation': the synergy of research and practice in an engineering research approach to educational design and development," *ZDM – Math. Educ.*, vol. 53, no. 5, pp. 991–1005, Oct. 2021, doi: 10.1007/s11858-020-01208-z.
- [11] R. Wulandari, I. P. D. Amiza, and F. Annas, "Pengembangan Model Inovasi Pembelajaran Berbasis Research and Development di SMP N 3 Pasaman Barat," *J. Educ. Manag. Strategy*, vol. 3, no. 2, pp. 117–124, Dec. 2024, doi: 10.57255/jemast.v3i02.602.
- [12] "Software Engineering | Phases of Prototyping Model | Set - 2," GeeksforGeeks. Accessed: Feb. 03, 2026. [Online]. Available: <https://www.geeksforgeeks.org/software-engineering/software-engineering-phases-prototyping-model-set-2/>
- [13] T. Zhou and W. Wang, "Prototype-Based Semantic Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 10, pp. 6858–6872, Oct. 2024, doi: 10.1109/TPAMI.2024.3387116.
- [14] W. Yao *et al.*, "Rapid system prototype-based physical simulation platforms for power systems with high penetration of inverter-based resources," *High Volt.*, vol. 10, no. 1, pp. 3–16, 2025, doi: 10.1049/hve2.12520.
- [15] J. Wei, T. Dingler, and V. Kostakos, "Developing the Proactive Speaker Prototype Based on Google Home," in *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, in CHI EA '21. New York, NY, USA: Association for Computing Machinery, May 2021, pp. 1–6. doi: 10.1145/3411763.3451642.
- [16] N. Azis, *ANALISIS PERANCANGAN SISTEM INFORMASI*. CV WIDINA MEDIA UTAMA, 2022. Accessed: Feb. 03, 2026. [Online]. Available: <https://repository.penerbitwidina.com/publications/407171/>
- [17] S. F. Arief and Y. Sugiarti, "Literature Review: Analisis Metode Perancangan Sistem Informasi Akademik Berbasis Web," *J. Ilm. Ilmu Komput. Fak. Ilmu Komput. Univ. Al Asyariah Mandar*, vol. 8, no. 2, pp. 87–93, Sep. 2022, doi: 10.35329/jiik.v8i2.229.
- [18] N. Y. A. M. Kom S. Kom *et al.*, *Analisa Perancangan Sistem Informasi*. Cendikia Mulia Mandiri, 2022.
- [19] H. C. Noiija *et al.*, "Perancangan Sistem Informasi Akuntansi Untuk Analisis Siklus Pendapatan Pada Orantata Celullar Menggunakan DFD Dan Flowchart," *J. Bisnis Dan Manaj. JURBISMAN*, vol. 1, no. 2, pp. 577–592, Jul. 2023, doi: 10.61930/jurbisman.v1i2.188.
- [20] Z. Tuasamu *et al.*, "Analisis Sistem Informasi Akuntansi Siklus Pendapatan Menggunakan DFD dan Flowchart Pada Bisnis Porobico," *J. Bisnis Dan Manaj. JURBISMAN*, vol. 1, no. 2, pp. 495–510, May 2023, doi: 10.61930/jurbisman.v1i2.181.
- [21] M. Jibril, Zulrahmadi, and 3Muhammad Amin, "PENGUJIAN SISTEM INFORMASI E-MODUL PADA SMPN 1 TEMPULING MENGGUNAKAN BLACK BOX TESTING," *J. PERANGKAT LUNAK*, vol. 6, no. 2, pp. 327–332, Jun. 2024, doi: 10.32520/jupel.v6i2.3326.
- [22] A. C. Praniffa, A. Syahri, F. Sandes, U. Fariha, and Q. A. Giansyah, "Pengujian Sistem Informasi Parkir Berbasis Web Pada UIN SUSKA RIAU Menggunakan White Box dan Black Box Testing," *J. Test. Dan Implementasi Sist. Inf.*, vol. 1, no. 1, pp. 1–16, Mar. 2023, doi: 10.55583/jtisi.v1i1.321.
- [23] M. Zen, I. Irwan, H. Hafni, and M. D. P. Ananda, "Implementasi dan Pengujian Menggunakan Metode BlackBox Testing Pada Sistem Informasi Tracer Study," *Bull. Comput. Sci. Res.*, vol. 4, no. 4, pp. 327–340, Jun. 2024, doi: 10.47065/bulletincsr.v4i4.359.
- [24] "Black Box Testing - Software Engineering," GeeksforGeeks. Accessed: Feb. 03, 2026. [Online]. Available: <https://www.geeksforgeeks.org/software-testing/software-engineering-black-box-testing/>
- [25] I. N. Rambe, H. N. Nasution, and S. W. R. Nasution, "PERANCANGAN MODUL PRAKTIKUM BERBASIS WEB PADA MATERI ALGORITMA DAN PEMROGRAMAN DI SMA NEGERI 1 BATANGTORU," *J. VINERTEK Vokasional Inform. Edukasi Ris. Dan Teknol.*, vol. 4, no. 1, pp. 17–20, 2024, doi: 10.37081/vn.v4i1.1222.
- [26] A. Dimara *et al.*, "A Novel Real-Time PV Error Handling Exploiting Evolutionary-Based Optimization," *Appl. Sci.*, vol. 13, no. 23, p. 12682, Jan. 2023, doi: 10.3390/app132312682.
- [27] S. N. Chotimah, S. Nurvita, and V. N. Dewi, "Penggunaan Metode Evaluasi Sistem Informasi Kesehatan di Indonesia: Literature Review," *J. Rekam Medis Manaj. Infomasi Kesehat.*, vol. 3, no. 2, pp. 36–43, Dec. 2023, doi: 10.53416/jurmik.v3i2.231.