

Sistem Informasi Pengelolaan Keuangan Di Instansi X Dengan Metode *Rapid Application Development*

DOI: <http://dx.doi.org/10.35889/jutisi.v15i4.3948>

Creative Commons License 4.0 (CC BY – NC)



Shania Yuliana¹, Agnes Novita Ida Safitri^{2*}, Mercurius Broto Legowo³, Deden Prayitno⁴
Sistem Informasi, Institut Keuangan Perbankan dan Informatika Asia (IKPIA) Perbanas, Jakarta,
Indonesia

*email Corresponding Author: agnes.novita@perbanas.id

Abstract

This study aims to develop a Financial Management Information System for Institution X to overcome manual financial recording, data redundancy, and delays in financial reporting. The system was developed using the Rapid Application Development (RAD) method integrated with the Model-View-Controller (MVC) architecture through the Laravel Framework, enabling rapid development, modular design, and easier maintenance. The development process consisted of four phases: Requirements Planning, User Design, Construction, and Cutover. The implemented system includes user authentication, master data management, financial transaction processing, budgeting, and automated reporting. Functional testing using Black Box Testing confirmed that all system features met user requirements. The results demonstrate that the system improves the efficiency and accuracy of financial management, minimizes data redundancy, and accelerates report generation. This study contributes by integrating the RAD method with the MVC architecture in Laravel to develop a more adaptive, maintainable, and scalable financial management information system.

Keywords: *Black Box Testing; Financial Management Information System; Laravel Framework; Model-View-Controller; Rapid Application Development*

Abstrak

Penelitian ini bertujuan mengembangkan Sistem Informasi Pengelolaan Keuangan pada Instansi X untuk mengatasi pencatatan keuangan yang masih dilakukan secara manual, redundansi data, dan keterlambatan penyusunan laporan keuangan. Sistem dikembangkan menggunakan metode *Rapid Application Development* (RAD) yang diintegrasikan dengan arsitektur *Model-View-Controller* (MVC) melalui *Laravel Framework*, sehingga mendukung proses pengembangan yang cepat, desain yang modular, serta kemudahan dalam pemeliharaan sistem. Proses pengembangan meliputi empat tahapan, yaitu *Requirements Planning*, *User Design*, *Construction*, dan *Cutover*. Sistem yang dihasilkan mencakup fitur autentikasi pengguna, pengelolaan data master, transaksi keuangan, penganggaran, dan pelaporan otomatis. Pengujian fungsional menggunakan *Black Box Testing* menunjukkan bahwa seluruh fitur sistem telah memenuhi kebutuhan pengguna. Hasil penelitian menunjukkan bahwa sistem mampu meningkatkan efisiensi dan akurasi pengelolaan keuangan, mengurangi redundansi data, serta mempercepat penyusunan laporan keuangan. Penelitian ini memberikan kontribusi melalui integrasi metode RAD dengan arsitektur MVC pada *Laravel* untuk menghasilkan sistem informasi pengelolaan keuangan yang lebih adaptif, mudah dipelihara, dan skalabel.

Kata kunci: *Black Box Testing; Laravel Framework; Model-View-Controller; Rapid Application Development; Sistem Informasi Pengelolaan Keuangan.*

1. Pendahuluan

Perkembangan teknologi informasi telah menjadi faktor penting dalam meningkatkan kualitas dan efektivitas pengelolaan organisasi melalui penerapan sistem yang terintegrasi [1]. Namun, masih banyak instansi yang mengelola data keuangan secara manual atau

menggunakan aplikasi yang belum terintegrasi, sehingga proses pencatatan, pelaporan, dan pengendalian keuangan menjadi kurang efektif. Oleh karena itu, pengembangan sistem informasi diperlukan untuk mengintegrasikan pengelolaan transaksi keuangan, meningkatkan efisiensi operasional, serta menyediakan informasi dan laporan yang akurat, tepat waktu, dan andal sebagai dasar pengambilan keputusan.

Instansi X masih menghadapi berbagai kendala dalam pengelolaan keuangan karena proses pencatatan, pelaporan belanja pegawai, belanja barang, dan realisasi anggaran masih dilakukan secara manual atau menggunakan spreadsheet yang belum terintegrasi. Kondisi tersebut menyebabkan duplikasi data, kesalahan input, keterlambatan pelaporan, serta menyulitkan proses audit dan evaluasi. Oleh karena itu, diperlukan pengembangan sistem informasi keuangan yang mampu mengintegrasikan proses perencanaan, pencatatan, pelaksanaan, dan pelaporan anggaran secara efektif, transparan, dan akuntabel. Sistem informasi ini diharapkan dapat meningkatkan efisiensi, akurasi data, ketepatan waktu pelaporan, serta mendukung proses audit dan pengambilan keputusan yang lebih baik [2].

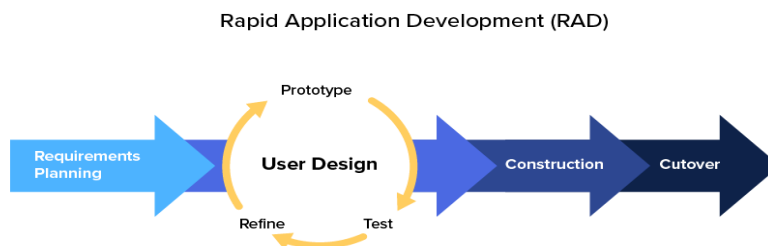
Beberapa penelitian terdahulu telah menerapkan metode *Rapid Application Development* (RAD) dalam pengembangan sistem informasi keuangan pada berbagai organisasi. Panja dan Manongga [3] Mengembangkan sistem informasi keuangan berbasis web pada Gereja Kristen Sumba menggunakan metode RAD. Selanjutnya, Apiscad dkk [4] Menerapkan RAD untuk mengembangkan aplikasi pencatatan keuangan di PT Samsriwi Adi Megah, sedangkan Saidah dkk. [5] Mengembangkan sistem informasi manajemen keuangan koperasi siswa menggunakan Laravel Framework yang terbukti meningkatkan akurasi pencatatan dan kualitas sistem. Penelitian berikutnya oleh Ramadan dkk.[6], Rizky dkk. [7]. Yuwanda dan Setiawan[8], serta Crisvin dkk.[9] Juga menunjukkan bahwa penerapan RAD dan Laravel mampu meningkatkan efisiensi pengelolaan keuangan, mempercepat pengembangan sistem, meningkatkan akurasi data, serta menghasilkan sistem yang mudah dipelihara.

Meskipun demikian, sebagian besar penelitian tersebut hanya berfokus pada penerapan metode RAD atau Laravel secara terpisah serta menitikberatkan pada pemenuhan kebutuhan fungsional sistem. Belum banyak penelitian yang mengintegrasikan metode RAD dengan arsitektur *Model-View-Controller* (MVC) pada Laravel Framework dalam pengembangan sistem informasi pengelolaan keuangan di lingkungan instansi. Oleh karena itu, penelitian ini mengisi kesenjangan tersebut dengan mengembangkan Sistem Informasi Pengelolaan Keuangan di Instansi X melalui integrasi RAD dan MVC pada Laravel Framework, sehingga menghasilkan sistem yang tidak hanya cepat dikembangkan, tetapi juga lebih modular, mudah dipelihara, aman, dan siap dikembangkan lebih lanjut.

Penelitian ini bertujuan mengembangkan Sistem Informasi Pengelolaan Keuangan di Instansi X menggunakan metode *Rapid Application Development* (RAD) untuk mengatasi permasalahan seperti duplikasi data, kesalahan input, keterlambatan pelaporan, serta kendala dalam proses audit dan evaluasi[2]. Pengembangan sistem mencakup tahap analisis kebutuhan, perancangan, implementasi, hingga pengujian aplikasi secara iteratif agar sesuai dengan kebutuhan pengguna. Sistem dibangun menggunakan framework Laravel dengan arsitektur *Model-View-Controller* (MVC) guna menghasilkan aplikasi yang terintegrasi, aman, modular, dan mudah dikembangkan[10], sehingga mampu meningkatkan efisiensi pengelolaan keuangan, menyajikan informasi yang akurat dan tepat waktu, serta mendukung pengambilan keputusan organisasi. Hasil penelitian ini diharapkan tidak hanya memberikan manfaat bagi Instansi X, tetapi juga menjadi referensi dalam pengembangan sistem informasi keuangan pada instansi lain yang memiliki karakteristik serupa.

2. Metodologi

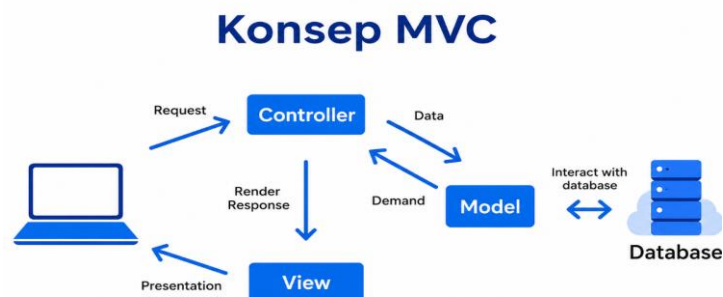
Metode pengembangan sistem yang digunakan adalah *Rapid Application Development* (RAD). RAD adalah metode pengembangan perangkat lunak yang menekankan kecepatan pengembangan melalui pendekatan prototipe dan keterlibatan pengguna secara aktif pada setiap tahapan, sehingga sistem yang dihasilkan lebih sesuai dengan kebutuhan pengguna. Metode RAD dipilih berdasarkan karakteristik kebutuhan sistem yang memerlukan pengembangan singkat, fleksibilitas terhadap perubahan kebutuhan, serta keterlibatan aktif pengguna[11]. Metode RAD yang diterapkan diilustrasikan pada Gambar 1.



Gambar 1. Metode Rapid Application Development

RAD terdiri atas empat tahapan utama, yaitu *requirements planning*, *user design*, *construction*, dan *cutover*. Dibandingkan dengan *Waterfall* yang bersifat linier, RAD lebih fleksibel terhadap perubahan kebutuhan, sementara dibandingkan dengan *Agile*, RAD lebih sederhana untuk diterapkan pada sistem berskala kecil hingga menengah karena tidak memerlukan siklus sprint berkelanjutan[12]. Penerapan RAD pada perancangan sistem informasi keuangan berbasis web juga dilaporkan mampu menghasilkan prototipe yang cepat disetujui pengguna pada studi kasus organisasi non-profit maupun instansi pemerintah[13]. Pendekatan RAD dapat mempercepat proses pengembangan, tetapi kurang ideal untuk proyek berisiko tinggi[14].

Laravel merupakan *framework* berbasis bahasa pemrograman PHP yang menerapkan pola arsitektur *Model-View-Controller* (MVC), sehingga memisahkan logika bisnis, tampilan antarmuka, dan pengelolaan data secara terstruktur. Penerapan arsitektur MVC pada sistem informasi berbasis Laravel terbukti memudahkan proses pengembangan dan pemeliharaan sistem pemantauan produksi [11]. Studi perbandingan kinerja *framework* PHP juga menunjukkan bahwa Laravel memiliki keunggulan pada sisi keamanan bawaan (*built-in security*) dan efisiensi pengelolaan data dibandingkan *CodeIgniter* untuk aplikasi berskala menengah [9].

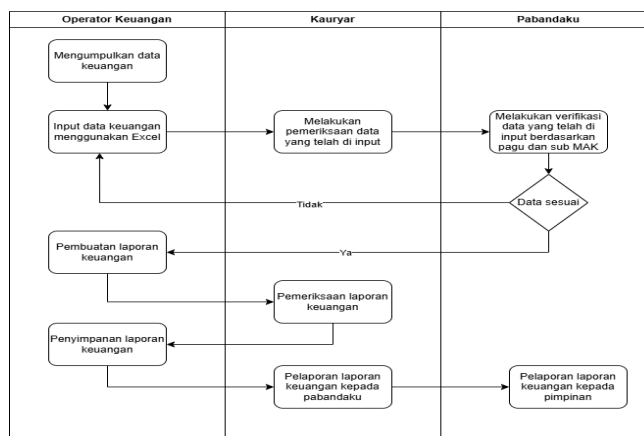


Gambar 2. Konsep MVC Framework Laravel[10]

Deskripsi metode **tidak** bersifat teoritis/normatif, namun menyajikan prosedur atau tahapan-tahapan riil yang dilalui dalam upaya pencapaian tujuan akhir yang diharapkan dalam riset/pengkajian.

2.1. Requirements Planning

Proses bisnis pengelolaan keuangan yang berjalan saat ini di Instansi X masih dilakukan secara manual, mulai dari pencatatan pagu anggaran, belanja pegawai dan belanja barang, hingga penyusunan laporan realisasi anggaran menggunakan berkas fisik dan spreadsheet yang terpisah. Alur kerja tersebut digambarkan pada flowchart sistem berjalan berikut.



Gambar 1. Flowchart Sistem yang Sedang Berjalan

Berdasarkan hasil wawancara dan observasi, teridentifikasi kebutuhan fungsional utama yang harus dipenuhi oleh sistem yang dikembangkan, sebagaimana dirangkum pada Tabel 1.

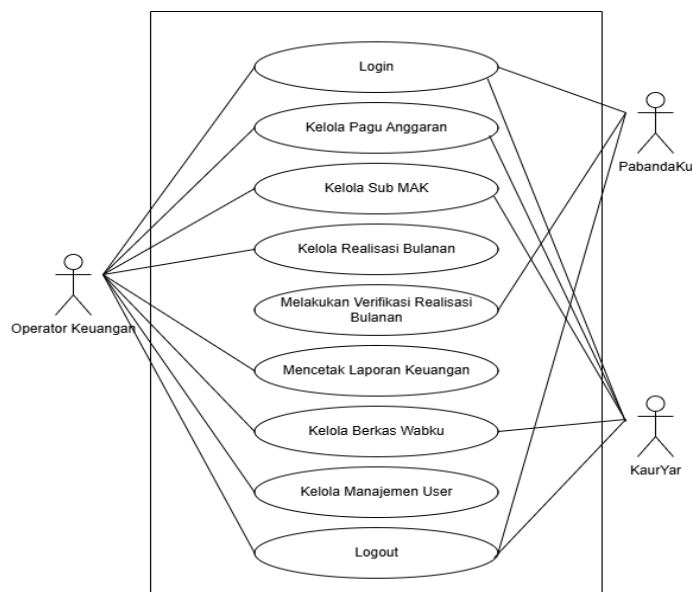
Tabel 1. Kebutuhan Fungsional Sistem

No	Kebutuhan Fungsional
1	Login pengguna berdasarkan peran
2	Pengelolaan pagu anggaran dan sub Mata Anggaran Keluaran (MAK)
3	Pencatatan realisasi belanja bulanan (pegawai dan barang)
4	Verifikasi berjenjang atas realisasi anggaran yang diinput
5	Pencetakan laporan keuangan dalam format PDF dan Excel
6	Pengarsipan berkas pelaksanaan anggaran
7	Manajemen data pengguna (hak akses sistem)

Selain kebutuhan fungsional, penelitian ini juga mengidentifikasi kebutuhan non-fungsional sistem, meliputi keamanan (autentikasi berbasis peran dan penyimpanan data pada lingkungan lokal), kegunaan (antarmuka yang sederhana dan mudah dipahami oleh operator keuangan), keandalan (validasi input untuk mencegah kesalahan data), serta performa (waktu respons sistem yang cepat pada proses pencatatan dan pencetakan laporan). Kebutuhan nonfungsional ini menjadi acuan tambahan pada tahap perancangan sistem agar aspek keamanan dan kenyamanan pengguna tetap terjaga di samping pemenuhan fitur fungsional.

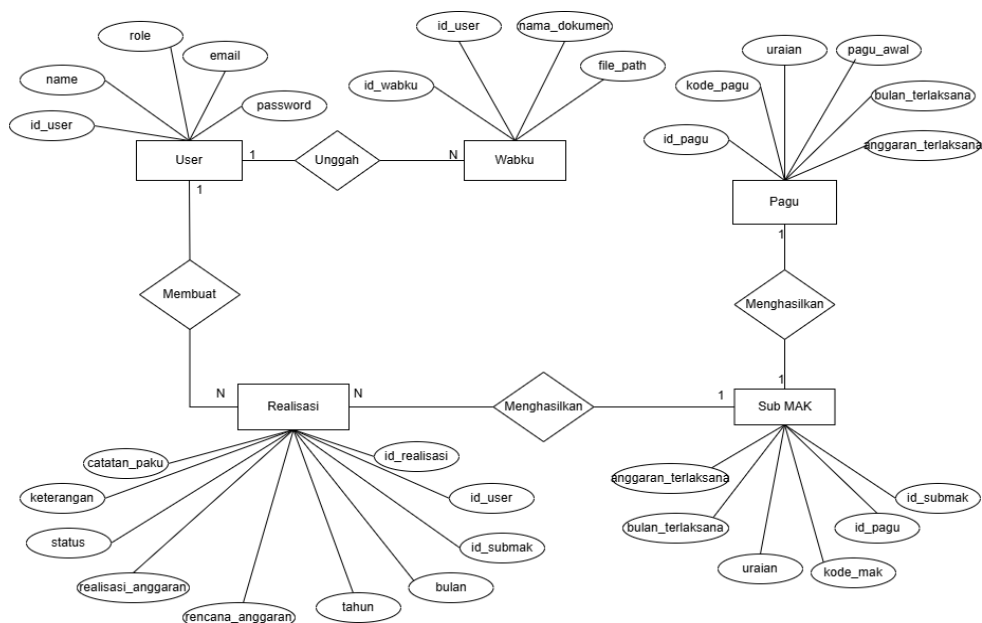
2.2. Perancangan Sistem

Berdasarkan hasil analisis kebutuhan, rancangan interaksi antara aktor dan sistem digambarkan melalui use case diagram pada Gambar 2. Aktor yang teridentifikasi meliputi Operator Keuangan, Kauryar (Kepala Urusan Pembayaran), dan Pabandaku (Perwira Pembantu Muda Keuangan), masing-masing dengan hak akses yang berbeda sesuai perannya dalam alur verifikasi anggaran berjenjang.



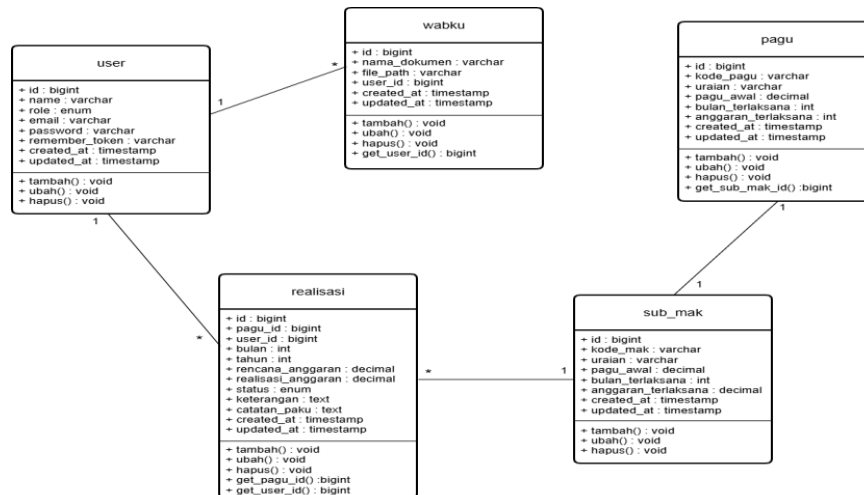
Gambar 2. Use Case Diagram

Entity-Relationship Diagram (ERD) merupakan teknik pemodelan konseptual yang digunakan pada tahap awal pengembangan basis data untuk mendefinisikan entitas, atribut, serta relasi antarentitas sebelum proses implementasi database dilakukan[15]. *Entity Relationship Diagram* (ERD) di sini digunakan untuk memodelkan entitas pengguna, pagu anggaran, sub-MAK, realisasi bulanan, verifikasi, dan berkas arsip beserta relasinya, sebagaimana ditampilkan pada Gambar 5. Rancangan ini memastikan integritas data antar-entitas serta mendukung proses pelaporan yang terintegrasi.



Gambar 3. Entity Relationship Diagram (ERD)

Struktur kelas sistem yang menggambarkan objek, atribut, dan relasi antarkelas dalam implementasi kode program turut dirancang menggunakan class diagram, sebagaimana ditampilkan pada Gambar 4. Class diagram ini menjadi acuan bagi pengembang pada tahap konstruksi untuk memastikan konsistensi antara model data dan logika bisnis yang diterapkan pada arsitektur MVC *framework Laravel*.



Gambar 4. Class Diagram

2.3 Konstruksi Sistem

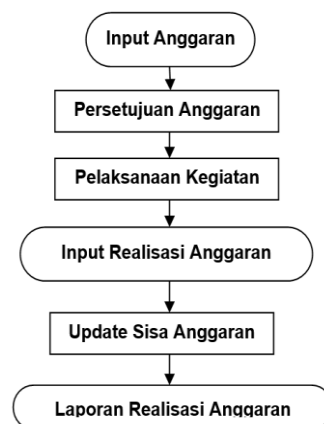
Berikut adalah konstruksi aplikasi sistem (prototype) pengelolaan realisasi anggaran menggunakan framework Laravel dengan arsitektur *Model-View-Controller* (MVC) yang sesuai untuk penelitian ini.

Pembangunan Prototype Pengelolaan Realisasi Anggaran

Tujuan Prototype: Prototype ini digunakan untuk membantu bendahara atau bagian keuangan dalam mencatat realisasi penggunaan anggaran; membandingkan anggaran dengan realisasi secara otomatis; mengetahui sisa anggaran setiap kegiatan; dan menghasilkan laporan realisasi anggaran secara real-time.

Proses Bisnis:

Proses bisnis sangat penting dalam pengembangan prototipe sistem informasi karena menjadi dasar untuk merancang sistem yang sesuai dengan kebutuhan pengguna, mendukung alur kerja organisasi, dan meningkatkan efisiensi serta efektivitas proses operasional.



Gambar 7. Proses Bisnis Pengelolaan Realisasi Anggaran

Struktur MVC Laravel

Struktur *Model-View-Controller* (MVC) pada Laravel memberikan kerangka yang terstruktur dan terorganisir untuk pengembangan website[16]. Contohnya adalah MVC untuk model anggaran dan model realisasi.[10]

Model.

Anggaran.php
Realisasi.php

Controller

AnggaranController.php
RealisasiController.php

View

anggaran/
index.blade.php
create.blade.php

realisasi/
index.blade.php
create.blade.php
laporan.blade.php

Tabel 2. Pengembangan Iteratif Menggunakan RAD

Iterasi	Prototype	Hasil
Iterasi 1	Input Data Anggaran	Data pagu anggaran tersimpan
Iterasi 2	Input Realisasi	Pencatatan realisasi anggaran
Iterasi 3	Monitoring Sisa Anggaran	Perhitungan otomatis realisasi dan sisa anggaran
Iterasi 4	Dashboard Pimpinan	Ringkasan realisasi dan indikator kinerja
Iterasi 5	Laporan Realisasi Anggaran	Laporan siap cetak (PDF/Excel)
Iterasi 6	Penyempurnaan	Validasi, hak akses pengguna, notifikasi, dan audit trail

Prototype ini menunjukkan bagaimana RAD dan framework Laravel, dengan arsitektur *Model-View-Controller* (MVC), saling melengkapi. Setiap iterasi menghasilkan modul yang dapat langsung diuji oleh pengguna, sehingga umpan balik dapat segera diintegrasikan. Arsitektur MVC [10]. Untuk memastikan logika bisnis (Model), alur aplikasi (Controller), dan antarmuka pengguna (View) tetap terpisah dengan baik, sehingga sistem lebih mudah dipelihara, dikembangkan, dan diskalakan untuk kebutuhan pengelolaan keuangan instansi. Hal ini menjadikan proses pengembangan lebih cepat, kualitas aplikasi lebih baik, dan implementasi lebih sesuai dengan kebutuhan pengguna.

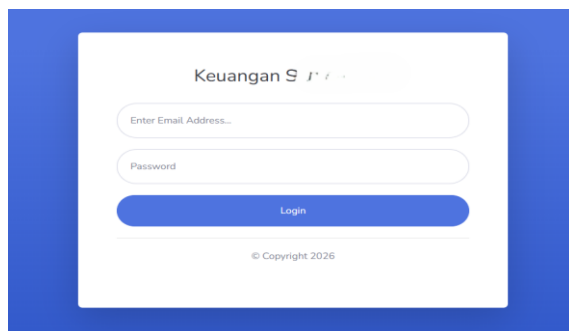
2.4 Pengujian Sistem

Pengujian sistem dilakukan menggunakan metode *black box testing*. *Black Box Testing* merupakan teknik pengujian fungsional yang digunakan untuk memverifikasi apakah setiap fungsi perangkat lunak telah bekerja sesuai spesifikasi, dengan menitikberatkan pada perilaku sistem tanpa mempertimbangkan implementasi atau struktur internal program [17]. Pengujian ini dilakukan dengan mengevaluasi kesesuaian antara data masukan (input) dan keluaran (output) yang dihasilkan berdasarkan kebutuhan sistem.

3. Hasil dan Pembahasan

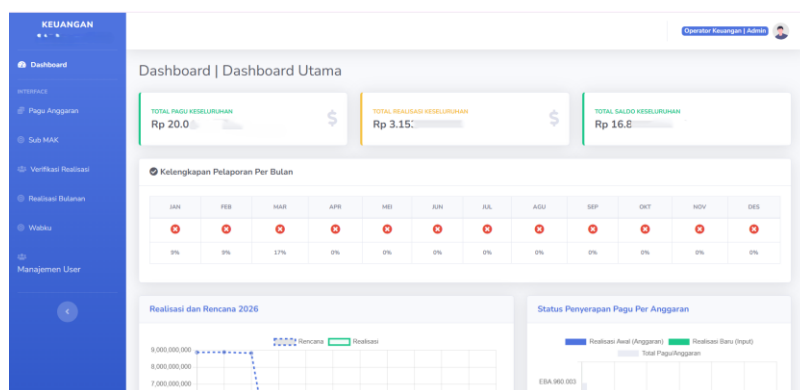
3.1 Hasil Pengembangan Sistem

Pengembangan sistem informasi menghasilkan sejumlah antarmuka pengguna yang dikembangkan menggunakan framework Laravel versi 10. Ini menggunakan bahasa pemrograman PHP versi 8 dan pengelolaan database MySQL. Adapun metode yang digunakan adalah metode MVC (Model, View, dan Controller). MVC merupakan salah satu konsep pemrograman yang membagi aplikasi menjadi tiga bagian, yaitu model yang mengelola interaksi antara aplikasi dan database, view yang mengelola urusan logika pemrograman, dan controller yang mengatur interaksi antara view dan model [18]. Halaman login, sebagaimana ditampilkan pada Gambar 8, menyediakan autentikasi berbasis peran sehingga setiap pengguna hanya dapat mengakses menu sesuai kewenangannya.



Gambar 8. Antarmuka Halaman Login

Setelah proses autentikasi berhasil, pengguna diarahkan ke halaman utama (*dashboard*) sesuai perannya. Gambar 9 menampilkan halaman utama untuk peran Operator Keuangan, yang menyediakan ringkasan pagu anggaran, status realisasi bulanan, serta akses cepat ke menu pencatatan dan pelaporan.



Gambar 9. Antarmuka Halaman Utama Operator Keuangan

Selain kedua halaman tersebut, sistem juga menyediakan menu pengelolaan pagu anggaran dan sub-MAK, verifikasi berjenjang, serta manajemen pengguna sistem. Salah satu menu utama, yaitu halaman kelola realisasi anggaran, ditampilkan pada Gambar 10.

Kode Anggaran	Uraian	Pagu Awal	Total Terpakai	Sisa Saldo
EBA.99	Dukungan Eksekutif / Pertahanan	Rp 312.000	Rp 14.400	Rp 297.600
EBA.99	Operasional / Pemeliharaan	Rp 10.000	Rp 0	Rp 10.000
EBA.99	Gaji dan Tunjangan	Rp 8.000	Rp 3.100	Rp 4.900

Status	Nama Sub	Bulan Pelaporan	Catatan Paku	Rencana	Realisasi	Selisih
REVISI	Gaji dan Tunjangan	Mei	-	Rp 8.000	Rp 1.500	Rp 6.500
ACC	Gaji dan Tunjangan	Februari	-	Rp 8.000	Rp 0	Rp 8.000
ACC	Gaji dan Tunjangan	Januari	-	Rp 8.000	Rp 804	Rp 7.196

Gambar 10. Antarmuka Halaman Kelola Realisasi Anggaran

Halaman ini berfungsi sebagai antarmuka utama bagi operator keuangan untuk memantau dan menginput serapan anggaran secara berkala, dengan menampilkan kode anggaran, pagu awal, total terpakai, dan sisa saldo secara otomatis.

Seluruh antarmuka tersebut dikembangkan mengikuti hasil rancangan pada tahap user design, dengan fitur pencetakan laporan keuangan dalam format PDF dan Excel, pengarsipan berkas pelaksanaan anggaran, serta manajemen pengguna sistem.

3.2 Hasil Konstruksi Sistem melalui Penerapan Konsep MVC dan Framework Laravel.

Pada konstruksi dalam implementasi system ini:

- (1) Model (Realisasi dan Anggaran) bertugas mengelola data serta relasi antartabel melalui Eloquent ORM.
- (2) Controller (RealisasiController) menangani permintaan dari pengguna, mengambil data dari model, kemudian meneruskannya ke tampilan.
- (3) View (index.blade.php) hanya bertanggung jawab menampilkan data dalam bentuk tabel menggunakan Blade Template Engine, tanpa memuat logika bisnis.

Pemisahan tanggung jawab ini merupakan inti dari arsitektur *Model–View–Controller* (MVC) dan memberikan manfaat berupa kode yang lebih terstruktur, mudah diuji, mudah dipelihara, serta mendukung pengembangan modul secara bertahap sesuai pendekatan *Rapid Application Development* (RAD). Berikut hasil konstruksi sistem melalui penerapan konsep MVC dan framework Laravel sebagai berikut:

1. Model (Realisasi.php)



```

app/Models/Realisasi.php

<? PHP
<?php

namespace App\Models;
use Illuminate\Database\Eloquent\Model;

class Realisasi extends Model
{
    protected $table = 'realisasi';

    protected $fillable = [
        'anggaran_id',
        'tanggal',
        'nominal',
        'uraian',
        'bukti_transaksi'
    ];

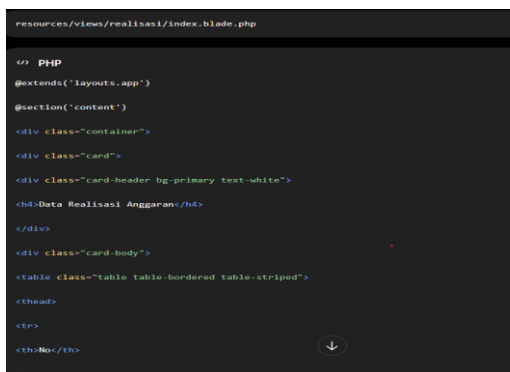
    public function anggaran()
    {
        return $this->belongsTo(Anggaran::class);
    }
}

```

Gambar 5. Tampil Kode untuk Model dalam penerapan MVC

Gambar 5 menunjukkan implementasi Realisasi.php sebagai komponen Model pada framework Laravel yang menerapkan arsitektur *Model-View-Controller* (MVC). Model ini berfungsi sebagai representasi tabel realisasi pada basis data dengan mendefinisikan nama tabel melalui properti `$table` serta atribut yang dapat diisi secara massal (`$fillable`), yaitu `anggaran_id`, `tanggal`, `nominal`, `uraian`, dan `bukti_transaksi`. Selain itu, fungsi `anggaran()` menggunakan relasi `belongsTo` (`Anggaran::class`) untuk menyatakan bahwa setiap data realisasi anggaran terhubung dengan satu data anggaran. Dengan memanfaatkan Eloquent ORM Laravel, model ini memungkinkan proses penyimpanan, pengambilan, pembaruan, dan pengelolaan relasi data dilakukan secara lebih sederhana tanpa harus menuliskan perintah SQL secara langsung, sehingga kode menjadi lebih terstruktur, mudah dipelihara, dan sesuai dengan prinsip pengembangan aplikasi berbasis MVC.

2. Views (Realisasi.php)



```

resources/views/realisasi/index.blade.php

<? PHP
@extends('layouts.app')

@section('content')
<div class="container">
<div class="card">
<div class="card-header bg-primary text-white">
<h2>Data Realisasi Anggaran</h2>
</div>
<div class="card-body">
<table class="table table-bordered table-striped">
<thead>
<tr>
<th>No</th>

```

Gambar 6. Tampil Kode untuk View dalam penerapan MVC

Gambar 6 menampilkan implementasi View `index.blade.php` pada framework Laravel yang merupakan bagian dari arsitektur *Model-View-Controller* (MVC). View ini menggunakan

Blade Template Engine dengan mewarisi layout utama melalui perintah `@extends('layouts.app')` dan mendefinisikan isi halaman menggunakan `@section('content')`. Tampilan dirancang menggunakan komponen Bootstrap seperti container, card, dan table untuk menyajikan data Realisasi Anggaran dalam bentuk tabel yang rapi dan responsif. Pada komponen MVC, View hanya bertanggung jawab menampilkan data yang dikirimkan oleh **Controller** tanpa mengandung logika bisnis maupun akses langsung ke basis data, sehingga pemisahan antara antarmuka pengguna, logika aplikasi, dan pengelolaan data dapat terjaga dengan baik.

2. Controller (Realisasi php)

Pada Gambar 7 ditampilkan implementasi *RealisasiController.php* pada Laravel yang berfungsi sebagai penghubung antara *Model* dan *View* dalam arsitektur *Model–View–Controller (MVC)*. Melalui fungsi *index()*, controller mengambil data realisasi beserta relasi anggaran menggunakan *eager loading* (`with('anggaran')`), mengurutkannya berdasarkan tanggal terbaru (`orderBy('tanggal','desc')`), kemudian mengirimkan data ke halaman *realisasi.index*. Mekanisme ini membuat pengelolaan data lebih efisien, mengurangi *query* berulang, serta memisahkan logika bisnis dari antarmuka pengguna sesuai prinsip MVC.

```
app/Http/Controllers/RealisasiController.php

<?php

namespace App\Http\Controllers;

use App\Models\Realisasi;

class RealisasiController extends Controller
{
    public function index()
    {
        $realisasi = Realisasi::with('anggaran')
            ->orderBy('tanggal','desc')
            ->get();

        return view('realisasi.index', compact('realisasi'));
    }
}
```

Gambar 7 Tampil Kode untuk Controller dalam penerapan MVC

3.3 Hasil Pengujian Sistem

Pengujian sistem dilakukan menggunakan metode *black box testing* yang berfokus pada kesesuaian keluaran sistem terhadap data masukan tanpa memeriksa struktur kode program [2]. Pengujian dilakukan bersama pengguna pada seluruh fitur utama sistem. Ringkasan hasil pengujian ditampilkan pada Tabel 4.

Tabel 4. Ringkasan Hasil *Black Box Testing*

Fitur	Skenario Pengujian	Hasil yang Diharapkan	Kesimpulan
Login sistem	Input username dan password sesuai peran	Berhasil masuk ke dashboard sesuai peran	Sesuai
Kelola pagu anggaran	Tambah dan ubah data pagu anggaran	Data tersimpan dan tampil pada daftar pagu	Sesuai
Realisasi bulanan	Input data realisasi belanja bulanan	Data realisasi tersimpan dan terhitung otomatis	Sesuai
Verifikasi realisasi	Verifikasi data realisasi oleh Kauryar/Pabandaku	Status realisasi berubah menjadi terverifikasi	Sesuai
Cetak laporan	Cetak laporan realisasi anggaran	Laporan terunduh dalam format PDF/Excel	Sesuai
Manajemen pengguna	Tambah, ubah, dan hapus akun pengguna	Data pengguna diperbarui sesuai input	Sesuai

Hasil pengujian menunjukkan bahwa seluruh fitur utama sistem berjalan sesuai dengan skenario yang diharapkan tanpa ditemukan kesalahan fungsional. Temuan ini sejalan dengan penelitian yang menyatakan bahwa pengujian *black box* efektif memvalidasi kesesuaian fungsi sistem berbasis web terhadap kebutuhan pengguna[17].

3.3 Pembahasan

Penelitian ini berfokus pada pengembangan Sistem Informasi Pengelolaan Keuangan di Instansi X menggunakan metode *Rapid Application Development (RAD)* dengan implementasi *Model-View-Controller (MVC)* melalui *Laravel Framework*. Pendekatan tersebut menjadi kebaruan penelitian karena tidak hanya menekankan percepatan proses pengembangan perangkat lunak, tetapi juga mengintegrasikan arsitektur aplikasi modern yang mendukung modularitas, kemudahan pemeliharaan (*maintainability*), keamanan, dan skalabilitas sistem. Penggunaan metode RAD terbukti mampu mempercepat proses pengembangan melalui pendekatan iteratif dan keterlibatan aktif pengguna dalam setiap tahapan pengembangan sistem [19]. Selain itu, penerapan *Laravel Framework* dengan arsitektur MVC menghasilkan struktur aplikasi yang lebih terorganisasi, mudah dikembangkan, serta mendukung pemeliharaan sistem secara berkelanjutan [5],[19],[20]. Dengan demikian, integrasi RAD dan MVC pada *Laravel Framework* memberikan pendekatan yang lebih adaptif dan efektif dalam pengembangan sistem informasi pengelolaan keuangan berbasis web..

Hasil penelitian ini memperkuat temuan beberapa penelitian terdahulu mengenai efektivitas metode *Rapid Application Development (RAD)* dalam pengembangan sistem informasi keuangan. Penelitian oleh Panja dan Manongga [3], Apiscad dkk. [4], Saidah dkk. [5], Ramadan dkk.[6], Rizky dkk. [7], Yuwanda dan Setiawan[8], serta Crisvin dkk.[9] Menunjukkan bahwa penerapan RAD maupun *Laravel Framework* mampu meningkatkan efisiensi pengelolaan keuangan, mempercepat proses pengembangan sistem, meningkatkan akurasi data, serta menghasilkan sistem yang memenuhi kebutuhan pengguna. Hasil penelitian ini sejalan dengan temuan tersebut, di mana sistem yang dikembangkan mampu mengotomatisasi proses pengelolaan keuangan, mengurangi redundansi data, meningkatkan akurasi pencatatan, dan mempercepat penyusunan laporan keuangan. Namun demikian, penelitian ini memiliki nilai tambah dibandingkan penelitian sebelumnya karena mengintegrasikan metode RAD dengan arsitektur *Model-View-Controller (MVC)* pada *Laravel Framework*. Integrasi ini menghasilkan struktur aplikasi yang lebih modular, mudah dipelihara (*maintainable*), lebih aman, dan lebih mudah dikembangkan untuk mendukung kebutuhan organisasi yang terus berkembang, sehingga memberikan kontribusi baru terhadap pengembangan sistem informasi pengelolaan keuangan.

Implementasi sistem dilakukan berdasarkan tahapan *Rapid Application Development (RAD)* yang meliputi *Requirements Planning*, *User Design*, *Construction*, dan *Cutover*. Pendekatan ini memungkinkan keterlibatan pengguna secara aktif pada setiap tahapan pengembangan sehingga kebutuhan sistem dapat diakomodasi dengan cepat melalui proses umpan balik yang berkelanjutan. Temuan ini sejalan dengan penelitian Ramadan dkk.[6], dan Hidayat dan Amin [19] Yang menyatakan bahwa metode RAD mampu mempercepat proses pengembangan sekaligus meningkatkan kesesuaian sistem dengan kebutuhan pengguna. Selain itu, penerapan arsitektur *Model-View-Controller (MVC)* pada *Laravel Framework* menghasilkan pemisahan yang jelas antara lapisan Model, View, dan Controller, sehingga logika bisnis, antarmuka pengguna, dan pengelolaan data dapat dikelola secara terpisah. Struktur ini menjadikan aplikasi lebih modular, mudah dipelihara (*maintainable*), serta memudahkan pengembangan fitur baru, sebagaimana dilaporkan oleh Saidah dkk. [5], dan Permana dan Santosa[20]. Hasil implementasi menunjukkan bahwa sistem mampu mengotomatisasi proses pengelolaan keuangan mulai dari pencatatan transaksi hingga penyusunan laporan, sehingga mengurangi redundansi data, meningkatkan akurasi pencatatan, dan mempercepat penyajian laporan keuangan. Hasil *Black Box Testing* menunjukkan seluruh fungsi sistem berjalan sesuai dengan kebutuhan pengguna[17], sehingga sistem dinyatakan layak untuk diimplementasikan. Temuan ini konsisten dengan penelitian Rizky dkk. [7] Yang melaporkan bahwa sistem informasi keuangan berbasis *Laravel* dengan metode RAD memenuhi seluruh kebutuhan fungsional dan mampu meningkatkan efektivitas pengelolaan keuangan.

Meskipun sistem yang dikembangkan telah memenuhi kebutuhan pengguna, penelitian ini masih memiliki beberapa keterbatasan. Implementasi sistem hanya dilakukan pada satu instansi sehingga hasilnya belum dapat digeneralisasikan pada organisasi lain. Selain itu, penelitian berfokus pada tahap pengembangan menggunakan metode RAD dan belum mengevaluasi keberhasilan implementasi jangka panjang, seperti kepuasan pengguna, penerimaan teknologi, dan dampaknya terhadap kinerja organisasi. Sistem juga masih berbasis web serta belum terintegrasi dengan aplikasi eksternal, seperti ERP, sistem perbankan, perpajakan, dan layanan pembayaran digital. Oleh karena itu, penelitian selanjutnya disarankan mengembangkan integrasi berbasis API, aplikasi mobile, serta memperkuat aspek keamanan dan evaluasi kualitas sistem.

4. Simpulan

Penelitian ini menghasilkan sistem informasi pengelolaan keuangan berbasis web yang dibangun menggunakan *framework Laravel* dengan metode *Rapid Application Development* (RAD) untuk mendukung pengelolaan keuangan di Instansi X. Penerapan empat tahapan RAD, yaitu *requirements planning*, *user design*, *construction*, dan *cutover*, terbukti mampu mempercepat proses pengembangan sistem sekaligus menghasilkan rancangan antarmuka dan alur kerja yang sesuai dengan kebutuhan pengguna di lapangan. Sistem yang dihasilkan mencakup fitur pengelolaan pagu anggaran, pencatatan realisasi bulanan, verifikasi berjenjang, pencetakan laporan keuangan, serta pengarsipan dokumen pelaksanaan anggaran, dan dijalankan di lingkungan lokal untuk menjaga keamanan data yang bersifat sensitif. Hasil pengujian menggunakan metode *black box testing* menunjukkan bahwa seluruh fitur sistem berjalan sesuai kebutuhan tanpa ditemukan kesalahan fungsional, sehingga sistem dinyatakan layak digunakan untuk mendukung transparansi dan akuntabilitas pengelolaan keuangan di Instansi X.

Penelitian selanjutnya disarankan untuk menambahkan fitur dashboard visualisasi data dan monitoring anggaran secara real-time, mengarahkan sistem agar dapat diakses melalui jaringan internal (intranet), serta membandingkan penerapan metode pengembangan lain guna meningkatkan kualitas sistem yang dihasilkan. Dengan berbagai capaian tersebut, penelitian ini diharapkan dapat menjadi rujukan bagi pengembangan sistem informasi keuangan serupa pada instansi lain yang memiliki kebutuhan pengelolaan anggaran secara terstruktur dan aman.

Daftar Referensi

- [1] E. Supriyadi and M. Sofiana, "Strategi Pengembangan Sistem Informasi Berbasis E-KTP di Kantor Kelurahan Galur (Studi Kasus Kelurahan Galur Kota Jakarta Pusat)," *J. Sist. Inf. STMIK ANTAR BANGSA*, vol. X, no. 02, pp. 69–74, 2021.
- [2] M. M. Naufal, H. Hasugian, and H. Harmain, "Analisis Penerapan Pernyataan Standar Akuntansi Pemerintah (PSAP) No . 02 Tentang Laporan Realisasi Anggaran Dalam Mewujudkan Akuntabilitas dan Transparansi Pengelolaan Keuangan Daerah," *PERMANA J. Perpajakan, Manajemen, dan Akunt.*, vol. 16, no. 1123, pp. 1123–1137, 2025.
- [3] E. Panja and D. Manongga, "Perancangan Sistem Informasi Keuangan Berbasis Web pada GKS Mauluru Menggunakan Rapid Application Development," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 7, no. 1, pp. 579–584, 2023.
- [4] D. Apisca and N. Surojudin, "Aplikasi Pencatatan Keuangan Berbasis Website Dengan Metode Rapid Application Development Pada PT Samsriwi Adi Megah," *JTekSis J. Teknol. dan Sist. Inf. Bisnis*, vol. 6, no. 1, pp. 212–219, 2024.
- [5] F. N. Saidah, R. R. Isnanto, I. Fauzi, and H. Alim, "Pengembangan Sistem Informasi Manajemen Keuangan Usaha Koperasi Siswa SMK Negeri 2 Semarang Menggunakan Kerangka Kerja Laravel The Development of Financial Management Information System for Student Cooperative at SMK Negeri 2 Using the Laravel Framework," *J. Tek. Komput.*, vol. 3, no. 1, pp. 25–31, 2024, doi: 10.14710/jtk.v3i1.44284.
- [6] A. T. Ramadan, G. Saputra, M. F. Adha, and W. Haryono, "Developing a Web-Based Financial Management Information System for PAUDQU with Rapid Application Development (RAD) Method," *J. TECH-E*, vol. 1, no. 9, 2025, doi: 10.31253/te.v9i1.3944.
- [7] M. H. Rizky, V. Terisia, and M. Syamsu, "Web-Based Financial Management Information System for MSMEs Using RAD Method," *J. Comput. Sci. Adv.*, vol. 3, no. 6, pp. 321–329, 2025.
- [8] M. Yuwanda and R. R. Setiawan, "Implementasi Sistem Informasi Keuangan Berbasis

- Laravel Guna Mewujudkan Transparansi Pembayaran SPP dan Buku di MTs Al Fattah Juwana," *J. Abdi Masy. Indones.*, vol. 6, no. 4, pp. 3415–3424, 2026.
- [9] C. Crisvin, N. Mulyaningsih, V. Andita, W. Winanti, and B. Haryanto, "Implementasi Metode Rapid Application Development (RAD) Pada Sistem Informasi Pengelolaan Bank Sampah Berbasis Web Menggunakan Framework Laravel," *J. IPSIKOM*, vol. 14, no. 1, 2026.
- [10] S. Hasyrif and R. Rismayani, "Penerapan Konsep MVC Pada Aplikasi Web Menggunakan Framework Laravel," in *Prosiding Seminar Ilmiah Sistem Informasi dan Teknologi Informasi*, 2016, pp. 174–183.
- [11] I. Ichsandi, W. Yanto, H. Alhaq, R. S. Sari, and M. Juanda, "Implementasi UML dalam Desain Sistem Informasi Program Studi SI di Universitas Merangin," *Impr. J. Teknol. dan Inf.*, vol. 4, no. 2, pp. 224–236, 2025.
- [12] D. Murdiani and M. Sobirin, "Perbandingan Metodologi Waterfall Dan Rad (Rapid Application Development) Dalam Pengembangan Sistem Informasi," *JINTEKS (Jurnal Inform. Teknol. dan Sains)*, vol. 4, no. 4, pp. 302–306, 2022.
- [13] M. R. Pratiwi, I. D. Perwitasari, P. Studi, S. Komputer, U. Pembangunan, and P. Budi, "Rancang Bangun Sistem Informasi Manajemen Laporan Keuangan Dengan Menggunakan Metode Rapid Application Development (RAD)," *RESOLUSI Rekayasa Tek. Inform. dan Inf.*, vol. 5, no. 1, pp. 67–73, 2024, doi: 10.30865/resolusi.v5i1.2161.
- [14] S. N. Bakri, M. Irwan, P. Nasution, U. Islam, and N. Sumatera, "Penerapan Metodologi Rekayasa Perangkat Lunak untuk Efisiensi Pengembangan Sistem," *JSITIK J. Sist. Inf. dan Teknol. Inf. Komput.*, vol. 3, no. 1, pp. 53–66, 2024.
- [15] K. Afifah, Z. F. Azzahra, and A. D. Anggoro, "Analisis Teknik Entity-Relationship Diagram dalam Perancangan Database : Sebuah Literature Review," *Inform. DAN Teknol.*, vol. 3, no. 2, pp. 18–22, 2022.
- [16] L. Rahmawati and S. Sumarsono, "Desain Pengembangan Website dengan Arsitektur Model View Controller pada Framework Laravel," *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 6, no. 4, pp. 785–790, 2024.
- [17] S. R. Wicaksono, *Blackbox Testing: Teori dan Studi Kasus*. 2021.
- [18] F. A. Purba, "Sistem Informasi Penjualan Pakaian Berbasis Web Pada TrendyStore Fitri Menggunakan Framework Laravel," *JCBDJournal Comput. Digit. Bus.*, vol. 2, no. 2, pp. 57–64, 2023.
- [19] J. J. Hidayat, M. Dikaisa, and I. Amin, "Implementasi Sistem Web Inventory dengan Metode Rapid Application Development (RAD) dan Framework Laravel," *J. Teknol. Inf. Digit.*, vol. 1, no. 2, pp. 63–70, 2025.
- [20] D. Permana and K. Santosa, "Perancangan Sistem Informasi Administrasi Transaksi E-Commerce Menggunakan Metode RAD dan MVC," *OKTAL J. Ilmu Komput. dan Sci.*, vol. 3, no. 5, pp. 1303–1313, 2024.