

IMPLEMENTASI LARAVEL PADA APLIKASI MONITORING DOKUMEN BERBASIS *WEBSITE* DENGAN METODE EXTREM PROGRAMMING

Khusnul Rizka¹, Nizirwan Anwar²

FAKULTAS ILMU KOMPUTER
UNIVERSITAS ESA UNGGUL

Email: khusnul120601@gmail.com¹, nizirwan.anwar@esaunggul.ac.id²

Abstrak

Dalam menghadapi tuntutan era teknologi informasi yang terus berkembang, sistem manajemen dokumen berbasis web menjadi elemen krusial untuk mendukung berbagai sektor, termasuk pendidikan, bisnis, dan pemerintahan. Kurangnya aplikasi yang memenuhi kebutuhan pengguna menjadi tantangan utama yang dihadapi, dengan aplikasi yang sudah ada seringkali tidak sepenuhnya memenuhi harapan dan kebutuhan mereka. Oleh karena itu, penelitian ini bertujuan untuk mengimplementasikan sistem monitoring dokumen berbasis website dengan menggunakan kerangka kerja Laravel dan menerapkan Metode Extreme Programming (XP) dalam pengembangannya. Masalah utama yang diidentifikasi meliputi kebutuhan akan tingkat keamanan yang tinggi, performa aplikasi yang optimal, partisipasi aktif pengguna selama pengembangan, integrasi dengan sistem lain, pengujian dan debugging yang efektif, serta skalabilitas aplikasi. Dengan fokus pada penggunaan Laravel sebagai kerangka kerja yang efisien, penelitian ini berusaha membangun solusi yang dapat mengatasi tantangan tersebut. Dengan menggunakan pendekatan XP, pengguna terlibat secara aktif dalam pengembangan aplikasi, memastikan pemahaman mendalam terhadap kebutuhan pengguna dan menerima umpan balik secara berkala. Integrasi dengan sistem lain diatasi untuk memastikan kelancaran operasional, sementara pengujian dan debugging menjadi komponen kunci dalam pengembangan aplikasi yang kompleks. Hasil dari penelitian ini diharapkan dapat mengatasi kurangnya aplikasi monitoring dokumen berbasis website yang memenuhi kebutuhan pengguna dengan baik. Dengan implementasi Laravel dan penerapan XP, diharapkan aplikasi ini tidak hanya efektif dan handal, tetapi juga responsif terhadap perubahan dan kebutuhan pengguna. Penelitian ini berkontribusi pada pengelolaan dokumen yang lebih efisien dan dapat diandalkan di berbagai sektor industri.

Kata kunci : Laravel, Sistem Monitoring Dokumen, Metode Extreme Programming (XP), Pengembangan Perangkat Lunak

IMPLEMENTATION OF LARAVEL IN A WEBSITE-BASED DOCUMENT MONITORING APPLICATION USING EXTREME PROGRAMMING METHODOLOGY

Abstract

In response to the demands of the ever-evolving information technology era, web-based document management systems have become crucial elements to support various sectors, including education, business, and government. The lack of applications that meet user needs is a major challenge, with existing applications often falling short of user expectations and requirements. Therefore, this research aims to implement a web-based document monitoring system using the Laravel framework and applying the Extreme Programming (XP) method in its development. The main identified issues include the need for a high level of security, optimal application performance, active user participation during development, integration with other systems, effective testing and debugging, and application scalability. With a focus on using Laravel as an efficient framework, this research seeks to build solutions to address these challenges. Using the XP approach, users actively engage in the application development process, ensuring a deep understanding of user needs and receiving regular feedback. Integration with other systems is addressed to ensure smooth operation, while testing and debugging are key components in developing a complex application. The results of this research are expected to overcome the lack of web-based document monitoring applications that effectively meet user needs. With the implementation of Laravel and the application of XP, it is anticipated that this application will not only be effective and reliable but

also responsive to changes and user needs. This research contributes to more efficient and reliable document management across various industrial sectors.

Keyword : *Laravel, Document Monitoring System, Web-Based Application, Extreme Programming (XP), Software Development*

1. PENDAHULUAN

Di era teknologi informasi yang semakin berkembang, sistem manajemen dokumen menjadi semakin penting untuk berbagai bidang, termasuk pendidikan, bisnis, dan pemerintahan. Untuk mengelola, menyimpan, dan mengakses dokumen dengan mudah dan efisien, aplikasi pengawasan dokumen berbasis web telah menjadi pilihan yang bagus.

Salah satu masalah utama yang perlu ditangani dalam penelitian ini adalah kurangnya aplikasi yang memenuhi kebutuhan pengguna. Aplikasi yang sudah ada seringkali tidak memenuhi harapan pengguna dan tidak memenuhi kebutuhan mereka. Akibatnya, tujuan penelitian ini adalah untuk membangun sistem yang lebih sesuai dan menawarkan solusi yang tepat bagi pengguna.

Penelitian ini perlu memastikan bahwa tingkat keamanan yang tinggi diterapkan untuk melindungi dokumen dari akses yang tidak sah, khususnya ketika data sensitif harus disimpan dan diakses oleh berbagai pihak.

Performa dan kinerja aplikasi monitoring dokumen juga menjadi fokus penting dalam penelitian ini. Dengan jumlah data yang besar dan pengguna yang simultan, aplikasi harus dapat menangani beban kerja dengan baik dan memberikan performa yang optimal. Dalam konteks ini, penerapan Laravel sebagai kerangka kerja PHP yang efisien dan handal menjadi pilihan untuk memastikan kinerja yang baik.

Metode Extreme Programming (XP) dalam pengembangan perangkat lunak mengharuskan pengguna terlibat secara aktif selama proses pengembangan. Oleh karena itu, masalah lain yang harus diselesaikan adalah memastikan partisipasi pengguna yang tepat untuk memahami secara menyeluruh kebutuhan pengguna dan menerima umpan balik secara teratur.

Selain itu, dalam lingkungan organisasi atau perusahaan, aplikasi monitoring dokumen mungkin memerlukan integrasi dengan sistem lain, seperti basis data, aplikasi lain, atau sistem manajemen lainnya. Tantangan dalam integrasi ini harus diatasi untuk memastikan kesesuaian dan kelancaran operasional.

Dalam pengembangan aplikasi yang kompleks, pengujian dan debugging juga merupakan komponen penting yang harus diperhatikan. Metode pengujian yang efektif harus dikembangkan untuk menemukan dan mengatasi kesalahan kode secara menyeluruh.

Terakhir, skalabilitas aplikasi sangat penting untuk memastikan bahwa aplikasi dapat dengan mudah berkembang sesuai dengan kebutuhan bisnis atau organisasi. Penelitian ini harus mencari solusi untuk merancang arsitektur yang skalabel agar aplikasi tetap efisien meskipun ada peningkatan data dan pengguna.

Penelitian ini berharap dapat menyelesaikan masalah-masalah di atas dan membuat aplikasi pengawasan dokumen berbasis web yang efektif dan dapat diandalkan yang menggunakan metode Extreme Programming. Aplikasi ini harus memenuhi kebutuhan dan harapan pengguna dan memberikan kontribusi positif dalam pengelolaan dokumen di berbagai industri.

2. METODE PENELITIAN

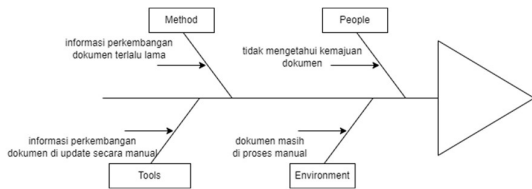
Dalam suatu studi, perhatian utama difokuskan pada subjek penelitian karena berperan sebagai tujuan yang ingin dicapai guna mendapatkan respons dan resolusi terhadap tantangan yang tengah dihadapi (Hamdani, 2016). Dalam konteks riset ini, entitas yang menjadi sorotan adalah sistem yang dipakai untuk mengelola dokumen di salah satu organisasi.

2.1 Analisis Fishbone

Analisis fishbone digunakan untuk mengidentifikasi berbagai faktor yang menyebabkan timbulnya masalah dalam suatu situasi tertentu (efek). Dalam konteks proses bisnis yang berjalan, fokus analisis ditujukan pada akibat dari ketiadaan sistem pemantauan dokumen yang bisa mendukung kegiatan operasional administrasi dokumen. Untuk menemukan solusi yang optimal terhadap masalah ini, langkah awalnya adalah mengidentifikasi akar penyebabnya. Setelah sejumlah penyebab teridentifikasi, solusi dapat ditemukan dengan mengatasi akar permasalahan tersebut.

Pelbagai faktor yang berkontribusi pada masalah ini akan diuraikan melalui diagram Fishbone. Penyebab masalah akan dikelompokkan dalam beberapa kategori penyebab utama, yakni Tools, Metode, Pelaku, dan Lingkungan. Kategori Tools akan mengelompokkan faktor-faktor penyebab berdasarkan metrik umum seperti alat yang digunakan, Kategori Metode akan mengelompokkan faktor-faktor penyebab berdasarkan prosedur-prosedur yang diterapkan, Kategori Pelaku akan mengelompokkan faktor-faktor penyebab berdasarkan individu-individu yang

terlibat, dan Kategori Lingkungan akan mengelompokkan faktor-faktor penyebab berdasarkan kondisi lingkungan terkait.



Gambar 1 Analisis Fishbone pada Sistem Monitoring Dokumen

Gambar diatas akan dijelaskan didalam tabel, tabel tersebut juga akan menentukan seberapa besar prioritas dari penyebab-penyebab yang diidentifikasi. Berikut adalah tabel penjelasan diagram fishbone tersebut :

| Cause | Effect | Priority |
|--|---|----------|
| People | | |
| Tidak mengetahui kemajuan dokumen | Karena karyawan dan pihak terkait tidak mengetahui perkembangan dokumen, menyebabkan proses bisnis lainnya menjadi lebih lama | Medium |
| Environment | | |
| Dokumen masih di proses manual | Pihak yang membutuhkan persetujuan dokumen harus secara berkala meminta informasi langsung ke divisi yang menangani dokumen | Medium |
| Methods | | |
| Informasi perkembangan dokumen terlalu lama | Informasi perkembangan dokumen menjadi lama dan berdampak ke bagian lainnya. | High |
| Tools | | |
| Perkembangan dokumen di update secara manual | Kemungkinan terjadi kesalahan menjadi lebih besar. | High |

2.2 Alur Metode Extreme Programming

Pada pembuatan “Aplikasi Monitoring Berbasis Website” metode pengembangan aplikasi ini menggunakan metode Extreme Programming, dimana tahap ini terdapat 4 tahapan utama. Tahapan tersebut sebagai berikut :

- a. Tahap Planning
- b. Tahap Design
- c. Tahap Coding
- d. Tahap Testing
- e. Tahap Release

2.2.1 Tahap Planning (Perencanaan)

Tahap perencanaan merupakan langkah awal dalam pengembangan aplikasi menggunakan metode Extreme Programming (XP) pada implementasi Laravel. Pada tahap ini, peneliti melakukan pengumpulan data dengan menggunakan teknik wawancara. Wawancara dilakukan untuk mengetahui permasalahan yang dialami saat melakukan monitoring dokumen dan untuk mendapatkan data yang di perlukan dalam penelitian ini.

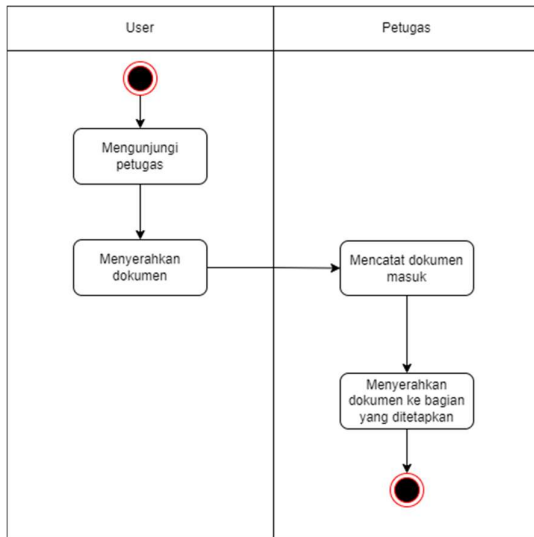
| No | Pertanyaan |
|----|---|
| 1 | Bagaimana Anda mengidentifikasi kebutuhan khusus perusahaan dalam hal monitoring dokumen dan bagaimana rencana Anda untuk mengintegrasikan kebutuhan tersebut ke dalam aplikasi? |
| 2 | Apa tantangan utama yang dihadapi perusahaan terkait monitoring dokumen saat ini, dan bagaimana aplikasi yang Anda rencanakan akan mengatasi atau meningkatkan proses tersebut? |
| 3 | Bagaimana Anda merencanakan antarmuka pengguna aplikasi monitoring dokumen agar sesuai dengan kebutuhan dan kemampuan pengguna di perusahaan ini? |
| 4 | Apakah ada integrasi atau keterkaitan yang perlu dipertimbangkan dengan sistem atau aplikasi lain yang sudah ada di perusahaan dalam proses pembuatan aplikasi monitoring dokumen? |
| 5 | Bagaimana rencana implementasi dan pelatihan pengguna akan diatur untuk memastikan adopsi yang efektif dari aplikasi monitoring dokumen di seluruh departemen atau tim di perusahaan? |

2.2.1.1 Analisa Kebutuhan

Setelah tahap planning yang menggunakan metode wawancara telah dilakukan maka berikutnya adalah menganalisa kebutuhan aplikasi monitoring dokumen yang akan di bangun. Analisa kebutuhan meliputi 2 hal yaitu analisa fungsional dan analisa non fungsional yang sebelumnya diperlukan sistem yang berjalan dan sistem yang di usulkan.

1. Sistem Berjalan

Sistem berjalan yang dilakukan oleh bagian administrasi dokumen akan di jelaskan melalui *activity diagram* sebagai berikut :



Gambar 2. Activity Diagram Sistem Berjalan

Pada *activity diagram* diatas aktifitas pertama adalah user mengunjungi petugas yang menangani dokumen berikutnya menyerahkan dokumen kepada petugas, petugas menerima dokumen tersebut dan memprosesnya.

2. Sistem Usulan

Pada sistem yang diusulkan ada tiga aktor yang terlibat yaitu aktor admin, petugas dan user. Penjelasan dari aktor tersebut di jelaskan pada table di bawah ini :

| | |
|----------------|--|
| User | Pengunjung mampu mencari dan memantau perkembangan dokumen yang telah diserahkan |
| Petugas | Dapat melakukan login Dapat menginput data berkas Dapat merubah status perkembangan berkas |
| Admin | Admin dapat melakukan login Admin dapat mengolah data petugas Admin dapat mengolah data dokumen Admin dapat mengolah data laporan |

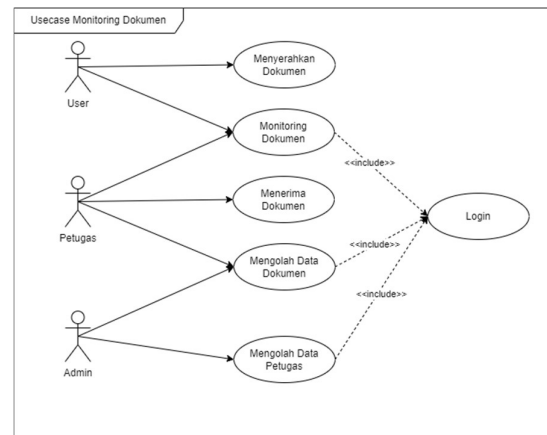
2.2.2 Tahap Design (Perancangan)

Setelah tahap perencanaan, peneliti akan memasuki tahap desain. Pada tahap ini, fokus utama

adalah merancang struktur aplikasi secara keseluruhan. Peneliti akan membuat desain arsitektur sistem, merinci struktur database, dan merancang antarmuka pengguna. Dalam konteks MVC, desain juga mencakup pemilihan model, view, dan controller yang akan digunakan untuk setiap fitur yang telah direncanakan sebelumnya.

2.2.2.1 Usecase Diagram

Usecase Diagram merupakan salah satu gambaran untuk fungsionalitas dari sistem yang diharapkan. Fungsi utama dari usecase ini adalah untuk mengetahui fungsi apa saja yang terdapat didalam sistem yang akan dibangun dan siapa saja yang berhak untuk menggunakan fungsi-fungsi tersebut. Berikut ini adalah tampilan Usecase Diagram yang terdiri dari tiga aktor yaitu admin, petugas dan user pada Aplikasi Monitoring Dokumen



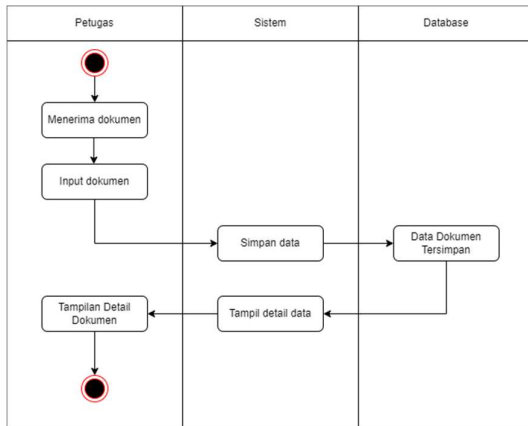
Gambar 3 Use Case Diagram

Pada usecase diagram diatas terdapat 3 aktor, yaitu user, petugas dan admin. User adalah aktor yang bisa menyerahkan dan memantau dokumen yang telah diserahkan. Selanjutnya adalah admin yang mempunyai *use case* untuk mengolah data dokumen dan juga data petugas. Terakhir adalah actor petugas yang mana actor ini adalah pihak yang bertugas untuk menerima dokumen dan mengolahnya.

2.2.2.2 Activity Diagram

Bagian perancangan ditahap pembuatan *activity diagram* menggambarkan alur dari setiap aktivitas yang ada pada sistem yang telah dirancang. Berikut ini merupakan perancangan *activity diagram* pada pengembangan aplikasi monitoring dokumen, adalah sebagai berikut :

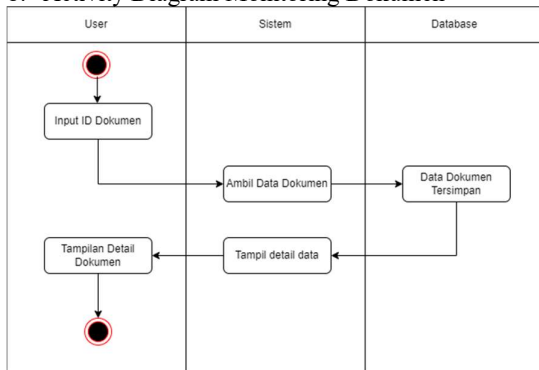
- a. Activity Diagram Penyerahan Dokumen



Gambar 4 Activity Diagram Penyerahan Dokumen

Activity diagram diatas menggambarkan runtutan aktifitas yang terjadi saat user menyerahkan dokumen dan diterima oleh petugas. Setelah petugas menerima dokumen dari user, petugas menginputkan informasi dokumen ke melalui form yang di berikan oleh sistem dan submit form yang telah di isi, berikutnya sistem akan menyimpan data dokumen tersebut dan menampilkan data dokumen ke petugas.

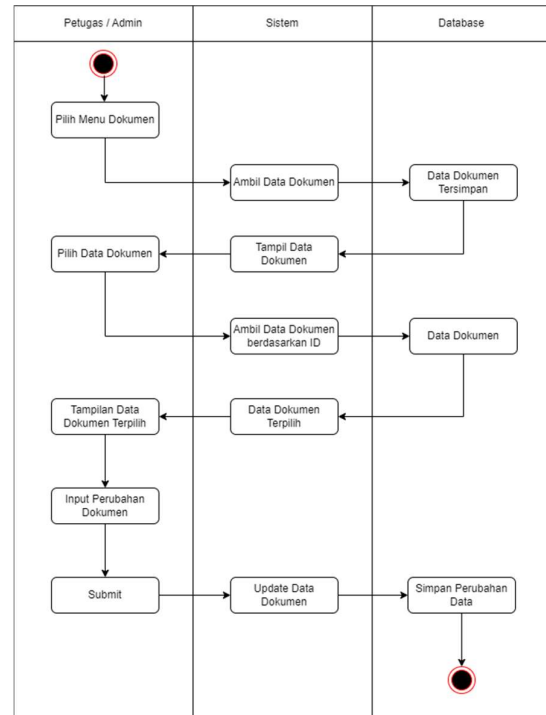
b. Activity Diagram Monitoring Dokumen



Gambar 5 Activity Diagram Monitoring Dokumen

Activity Diagram diatas menggambarkan runtutan aktifitas yang terjadi saat user melakukan monitoring dokumen. Aktifitas dimulai dari user menginputkan ID dokumen dan men-submitnya, berikutnya sistem akan mencari data dokumen berdasarkan ID yang telah di inputkan oleh user tersebut dan menampilkannya ke user.

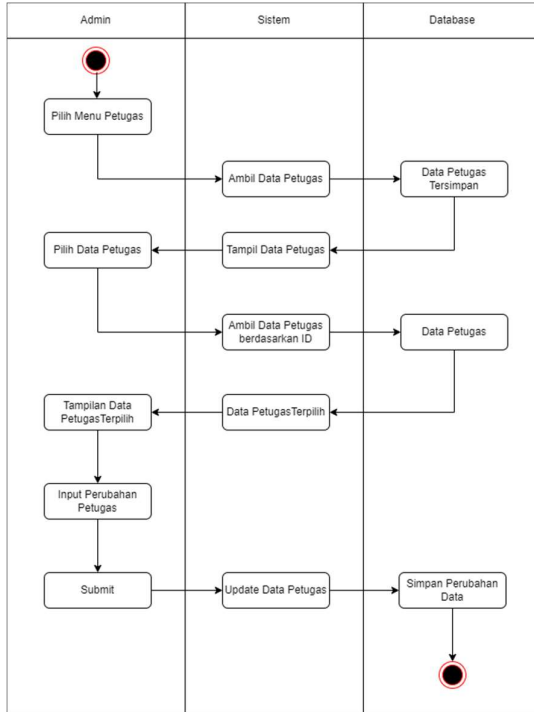
c. Activity Diagram Kelola Data Dokumen



Gambar 6 Activity Diagram Kelola Data Dokumen

Activity Diagram diatas menggambarkan urutan aktifitas ketika petugas atau admin melakukan pengelolaan dokumen. Aktifitas dimulai dari admin atau petugas melakukan login ke dalam sistem dan memilih menu dokumen. Berikutnya sistem akan mengambil data dokumen dan petugas memilih data dokumen mana yang akan di Kelola, seperti melakukan update atau pun delete data dokumen.

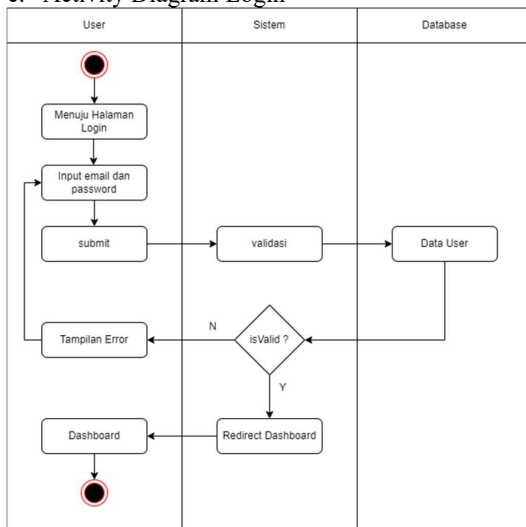
d. Activity Diagram Kelola Data Petugas



Gambar 7 Activity Diagram Kelola Data Petugas

Diagram Aktifitas diatas menggambarkan urutan aktifitas yang terjadi pada saat admin mengelola petugas. Diagram tersebut menggambarkan proses dimana admin harus memilih data petugas yang akan di edit yang berikutnya sistem akan menampilkan data petugas berdasarkan yang dipilih oleh admin.

e. Activity Diagram Login



Gambar 8 Activity Diagram Login

Diagram aktifitas diatas menggambarkan aktifitas yang terjadi ketika user baik yang memiliki aktor admin, petugas maupun user melakukan login ke dalam sistem. Sistem akan memvalidasi inputan

yang dikirim oleh pengguna, apabila tervalidasi maka akan di arahkan ke halaman dashboard.

2.2.2.3 Class Diagram

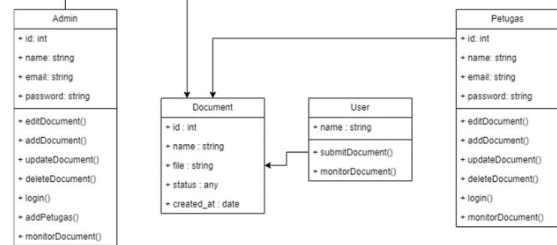
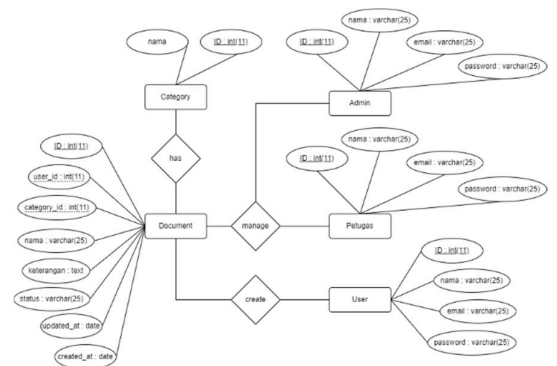


Diagram diatas menggambarkan kelas yang berinteraksi di dalam sistem monitoring dokumen. Kelas tersebut adalah :

- a. Admin
Kelas admin merepresentasikan entitas yang mengelola sebagian besar data yang ada pada sistem, seperti data petugas, dokumen dan laporan.
- b. Petugas
Kelas petugas merepresentasikan entitas yang melakukan pengelolaan dokumen yang dikirimkan user.
- c. User
Kelas user merepresentasikan entitas yang mengirimkan dokumen ke petugas yang selanjutnya akan menjadi objek dari monitoring.
- d. Document
Kelas dokumen merepresentasikan entitas yang di jadikan objek utama di dalam sistem, dimana entitas ini harus di bawa oleh user dan menyerahkannya ke petugas.

2.2.2.4 ERD



Gambar 9 Entity Relational Diagram

Gambar diatas menunjukkan relasi antar entitas yang ada pada aplikasi monitoring dokumen berbasis website. Terdapat 4 entitas yang berinteraksi di dalam aplikasi. Entitas tersebut adalah :

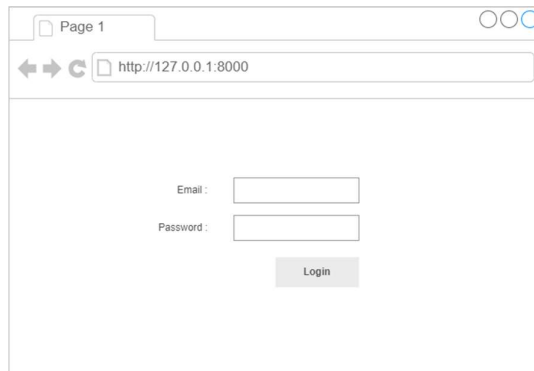
- a. category
- b. document

- c. admin
- d. petugas
- e. user

2.2.2.5 Design Antar Muka

Design Antar muka adalah salah satu tahapan dimana peneliti merancang antar muka yang akan digunakan oleh aplikasi monitoring dokumen. Antar muka yang akan digunakan meliputi antar muka login, dashboard, Kelola anggota, Kelola dokumen dan monitor dokumen.

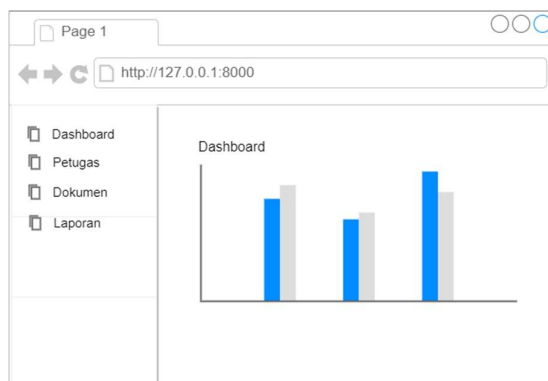
1. Wireframe Login



Gambar 10 Wireframe Login

Wireframe diatas adalah rancangan antar muka yang akan digunakan untuk membuat halaman login pada aplikasi monitoring dokumen. Design diatas terdiri dari 2 input yaitu input email dan password. Selain 2 input terdapat button login yang digunakan oleh user untuk mengirimkan kredensialnya ke sistem.

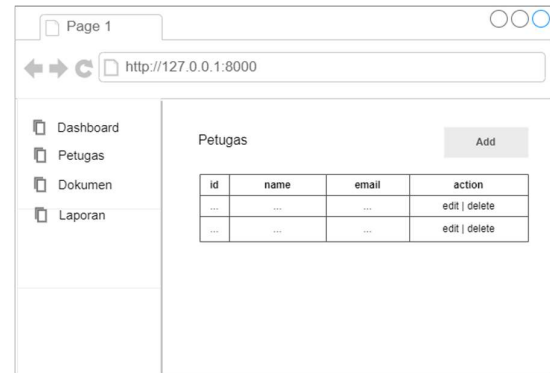
2. Wireframe Dashboard



Gambar 11 Wireframe Dashboard

Wireframe diatas menggambarkan rancangan yang akan di gunakan untuk halaman dashoard, dimana pada rancangan tersebut terdapat grafik yang menunjukkan data dokumen yang di proses di dalam sistem.

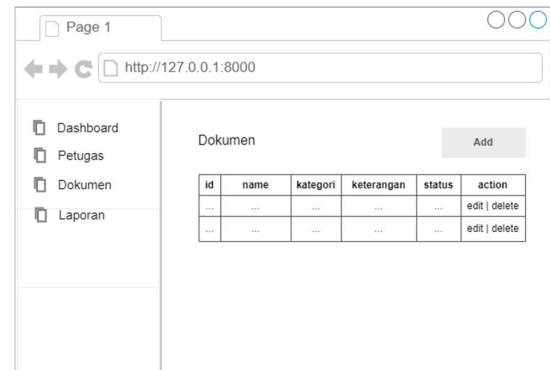
3. Wireframe Kelola Petugas



Gambar 12 Wireframe Kelola Petugas

Wireframe diatas adalah rancangan antar muka yang digunakan untuk membuat halaman Kelola petugas. Fitur ini nantinya akan digunakan oleh admin guna mengelola data petugas yang bertanggung jawab untuk mengelola dokumen yang di kirimkan oleh user.

4. Wireframe Kelola Dokumen



Gambar 12 Wireframe Kelola Dokumen

Wireframe diatas adalah rancangan antar muka yang digunakan untuk membuat halaman Kelola dokumen. Fitur ini nantinya akan digunakan oleh admin atau petugas guna mengelola data dokumen yang di kirimkan oleh user.

2.2.3 Tahap Coding (Penulisan Kode Program)

Tahap pengkodean adalah implementasi nyata dari desain yang telah dibuat sebelumnya. Tim akan mulai menulis kode untuk setiap fitur dan modul aplikasi menggunakan Laravel. Pada tahap ini, penerapan prinsip-prinsip XP. Penggunaan fitur-fitur khusus Laravel, seperti Eloquent ORM dan Blade templating, akan mempermudah pengembangan pada tahap ini. Tahapan ini akan di jelaskan secara lebih rinci pada bab 4.

2.2.4 Tahap Testing (Pengujian)

Setelah kode telah ditulis, tahap pengujian dimulai. Tim akan melakukan pengujian unit untuk memastikan bahwa setiap modul berfungsi seperti yang diharapkan. Selain itu, pengujian integrasi akan memastikan bahwa semua bagian aplikasi dapat berinteraksi dengan baik. Metode pengujian manual akan digunakan untuk memastikan kualitas dan keandalan aplikasi. Tahapan ini akan di jelaskan secara lebih rinci pada bab berikutnya.

2.2.5 Tahap Release (Peluncuran)

Setelah berhasil melewati tahap pengujian, aplikasi siap untuk dirilis. Pada tahap ini, tim akan mempublikasikan aplikasi monitoring dokumen berbasis website yang telah diimplementasikan dengan Laravel. Proses rilis melibatkan pemasangan aplikasi di server yang sesuai dan memastikan bahwa semua konfigurasi yang diperlukan telah diatur dengan baik.

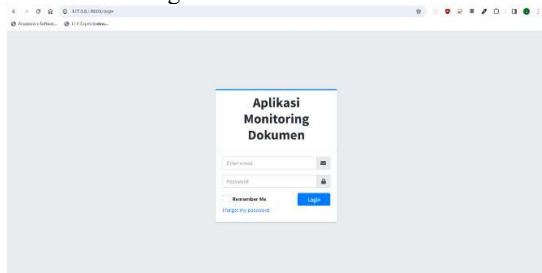
3. HASIL DAN PEMBAHASAN

3.1 Hasil Tahap Design

Hasil dari Aplikasi Monitoring Dokumen adalah sebuah aplikasi yang dapat dilakukan untuk memantau dokumen, dan mudahnya admin atau petugas untuk mengelola data yang terkait oleh dokumen. Aplikasi Monitoring Dokumen dengan menerapkan pengembangan Perangkat Lunak Extreme Programming.

3.1.1 Tampilan Admin

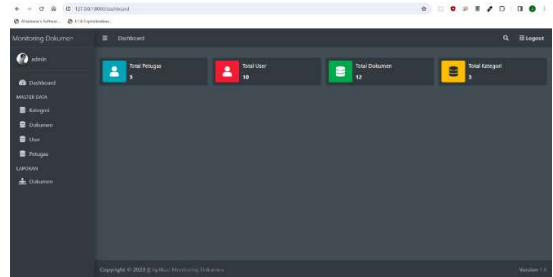
a. Halaman Login



Gambar 13 Halaman *Login*

Antar muka diatas adalah tampilan yang digunakan oleh pengguna untuk melakukan login ke dalam sistem. Pada antar muka tersebut terdapat form dengan dua inputan yaitu email dan password juga terdapat 1 button untuk submit kredensial yang telah di inputkan oleh pengguna.

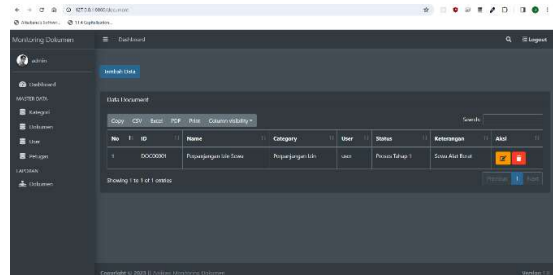
b. Halaman Dashboard



Gambar 14 Halaman *Dashboard*

Antar muka diatas adalah tampilan dashboard ketika pengguna yang berperan sebagai admin berhasil masuk ke dalam sistem. Pada tampilan diatas terdapat informasi jumlah petugas, user, dokumen dan kategori dari dokumen yang di Kelola.

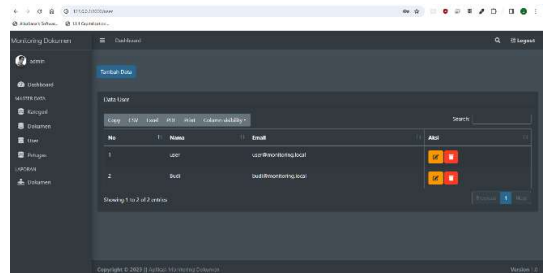
c. Halaman Kelola Dokumen



Gambar 15 Halaman Kelola Dokumen

Antar muka diatas adalah tampilan yang digunakan oleh admin untuk mengelola dokumen yang ada. Pada tampilan tersebut terdapat table yang berisi data dokumen, tombol tambah. Pada setiap data dokumen yang ditampilkan terdapat button untuk edit dan hapus.

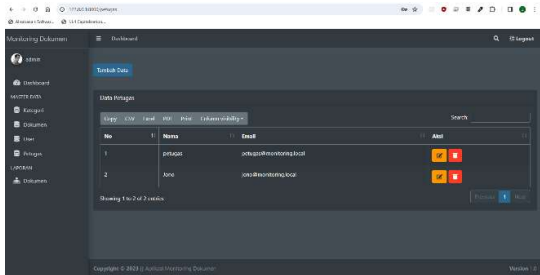
d. Halaman Kelola User



Gambar 16 Halaman Kelola *User*

Antar muka diatas adalah tampilan yang digunakan oleh admin untuk mengelola user yang ada. Pada tampilan tersebut terdapat table yang berisi data user dan tombol tambah. Pada setiap data user yang ditampilkan terdapat button untuk edit dan hapus.

e. Halaman Kelola Petugas

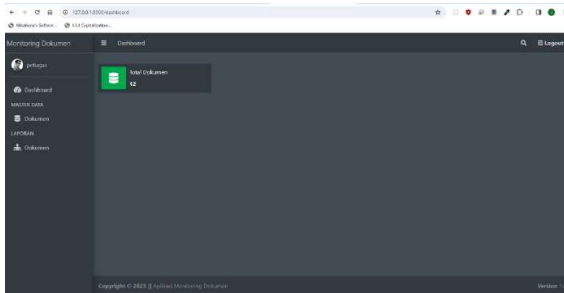


Gambar 17 Halaman Kelola Petugas

Antar muka diatas adalah tampilan yang digunakan oleh admin untuk mengelola data petugas yang ada. Pada tampilan tersebut terdapat table yang berisi data petugas dan tombol tambah. Pada setiap data petugas yang ditampilkan terdapat button untuk edit dan hapus.

3.1.2 Tampilan Petugas

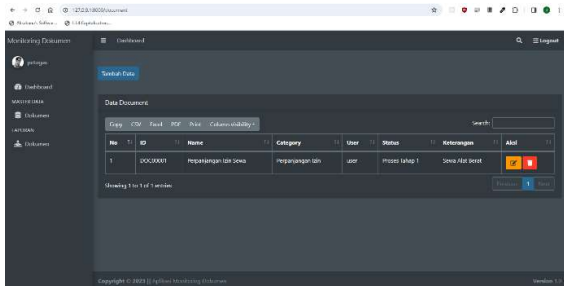
a. Halaman Dashboard



Gambar 18 Halaman Dashboard

Antar muka diatas adalah tampilan dashboard ketika pengguna yang berperan sebagai petugas berhasil masuk ke dalam sistem. Pada tampilan diatas terdapat informasi jumlah dokumen yang ada pada database.

b. Halaman Kelola Dokumen



Gambar 19 Halaman Kelola Dokumen

Antar muka diatas adalah tampilan yang digunakan oleh petugas untuk mengelola dokumen yang ada. Pada tampilan tersebut terdapat table yang berisi data dokumen, tombol tambah. Pada setiap data dokumen yang ditampilkan terdapat button untuk edit dan hapus.

3.2 Hasil Tahap Coding

Pada tahap ini peneliti akan memaparkan hasil dari tahapan coding untuk fitur lacak document yang dimiliki oleh aktif pengguna atau user dimana hasil ini dilakukan setelah tahap planning dan design yang peneliti jelaskan pada bab 3. Tahapan ini merupakan implementasi dari tahap sebelumnya dan menggunakan framework Laravel sebagai kerangka kerja dalam membangun aplikasi monitoring dokumen ini.

a. DocumentController

```
<?php

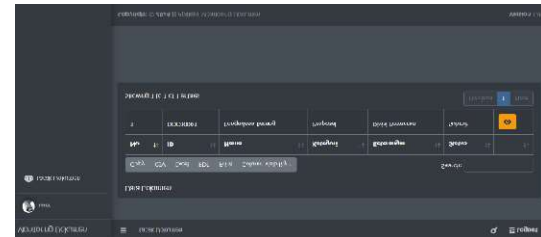
namespace App\Http\Controllers\User;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\Document;
use Illuminate\Support\Facades\Auth;

class DocumentController extends Controller
{
    public function index(){
        $user = Auth::user();
        $document = Document::where('user_id',
        $user->id)->get();

        return view('user.index',
        compact(['document']));
    }
}
```

b. Hasil View Lacak Document



Gambar 20 Halaman Lacak Dokumen

3.2.1 Model

Model adalah representasi dari entitas data yang di Kelola di dalam sistem. Model mewakili struktur data dan logika aplikasi. Model adalah bagian yang bertanggung jawab untuk mengelola data, serta berisi aturan dan operasi yang berkaitan dengan data tersebut. Model juga bertanggung jawab untuk memberitahu View ketika terjadi perubahan data. Pada aplikasi monitoring dokumen ini terdapat 1 model utama, yaitu :

a. Document

Model ini berfungsi sebagai container dari data dokumen yang ada di dalam sistem. Model ini memiliki 6 atribut yaitu id, user_id, kategori_id, nama, keterangan, status. Berikut ini adalah kode dari model dokumen

```
<?php
namespace App\Models;

use
Illuminate\Database\Eloquent\Factories\HasFactory;
;
use Illuminate\Database\Eloquent\Model;

class Document extends Model
{
    use HasFactory;
    protected $primaryKey = 'id';
    public $incrementing = false;
    protected $keyType = 'string';
    protected $table = 'documents';

    protected $fillable = [
        'id',
        'user_id',
        'kategori_id',
        'nama',
        'keterangan',
        'status'
    ];

    public static function generateDocId(){
        $latestModel = static::latest()->first();

        $lastNumber = $latestModel ?
intval(substr($latestModel->id, 4)) : 0;
        $nextNumber = $lastNumber + 1;

        return 'DOC0' . str_pad($nextNumber, 4,
'0', STR_PAD_LEFT);
    }

    public function category(){
        return $this->belongsTo(Category::class,
'kategori_id');
    }

    public function user(){
        return $this->belongsTo(User::class,
'user_id');
    }
}
```

3.2.2 View

View bertanggung jawab untuk menampilkan data dari Model kepada pengguna dan mengumpulkan input dari mereka. View tidak

memiliki pengetahuan tentang Model, namun hanya menampilkan informasi yang diberikan oleh Model. Dalam beberapa implementasi, View juga dapat memiliki beberapa logika tampilan sederhana. Pada aplikasi monitoring ini menggunakan view sebagai berikut :

a. View document

```
@extends('admin.layout.app')
@section('content')
    <div class="container-fluid my-3"
style="padding-left: 1%;">
        <a href="{{ route('admin.document.add')
}}" class="btn btn-primary">Tambah Data</a>
    </div>
    <br>

    <section class="content">
        <div class="container-fluid">
            <div class="row">
                <div class="col-12">
                    <!-- /.card -->

                    @if (session('success'))
                        <div class="alert alert-
success">{{ session('success') }}</div>
                    @endif
                    @if ($errors->any())
                        <div class="alert alert-danger"
role="alert">
                            {!! implode(", $errors-
>all('<li>:message</li>')) !!}
                        </div>
                    @endif

                    <div class="card">
                        <div class="card-header">
                            <h3 class="card-title">Data
Document</h3>
                        </div>
                        <!-- /.card-header -->
                        <div class="card-body">
                            <table id="example1"
class="table table-bordered table-striped">
                                <thead>
                                    <tr>
                                        <th>No</th>
                                        <th>ID</th>
                                        <th>Name</th>
                                        <th>Category</th>
                                        <th>User</th>
                                        <th>Status</th>
                                        <th>Keterangan</th>
                                        <th>Aksi</th>
                                    </tr>
                                </thead>
                                <tbody>
                                    @foreach ($document as
$index => $row)
                                        <tr>
```

```

    <td>{{ ++$index
  }}</td>
    <td>{{ $row->id
  }}</td>
    <td>{{ $row->nama
  }}</td>
    <td>{{ $row->category->nama }}</td>
    <td>{{ $row->user->name }}</td>
    <td>{{ $row->status
  }}</td>
    <td>{{ $row->keterangan }}</td>
    <td>
      <a href="{{
route('admin.document.edit', ['id' => $row->id])
}}"
        class="btn btn-
warning"><i class="fa fa-edit"></i></a>
      <a href="{{
route('admin.document.delete', ['id' => $row->id])
}}"
        onclick="return
confirm('Are you sure?'); return false;"
        class="btn btn-
danger"><i class="fa fa-trash"></i></a>
    </td>
  </tr>
@endforeach
</table>
</div>
<!-- /.card-body -->
</div>
<!-- /.card -->
</div>
<!-- /.col -->
</div>
<!-- /.row -->
</div>
<!-- /.container-fluid -->
</section>
@endsection

```

3.2.3 Controller

Controller bertindak sebagai perantara antara Model dan View. Controller menerima input dari pengguna melalui View, memproses input tersebut melalui perantara Model dan kemudian memperbarui View dengan informasi yang diperbarui dari Model. Controller memastikan bahwa Model dan View tetap terpisah satu sama lain.

a. DocumentController

```

<?php
namespace App\Http\Controllers\Admin;

```

```

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\Category;
use App\Models\Document;
use App\Models\User;

class DocumentController extends Controller
{
    public function index(){
        $document = Document::all();

        return view('admin.document.index',
compact(['document']));
    }

    public function add(){
        $category = Category::all();
        $user = User::where('role', 'user')->get();

        return view('admin.document.add',
compact(['category', 'user']));
    }

    public function create(Request $req){
        Document::create([
            'id' => Document::generateDocId(),
            'user_id' => $req->user,
            'kategori_id' => $req->category,
            'nama' => $req->nama,
            'keterangan' => $req->keterangan,
            'status' => $req->status,
        ]);

        return redirect()->route('admin.document.index');
    }

    public function edit($id){
        $category = Category::all();
        $user = User::where('role', 'user')->get();
        $document = Document::findOrFail($id);

        return view('admin.document.edit',
compact(['document', 'category', 'user']));
    }

    public function update(Request $req){
        $document = Document::findOrFail($req->id);

        $document->user_id = $req->user;
        $document->kategori_id = $req->category;
        $document->nama = $req->nama;
        $document->keterangan = $req->keterangan;
        $document->status = $req->status;

        $document->save();

        return redirect()->route('admin.document.index');
    }
}

```

```

}

public function delete($id){
    Document::destroy($id);

    return redirect()-
>route('admin.document.index');
}
}

```

4. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan hasil dari seluruh kegiatan yang telah peneliti lakukan pada penelitian ini yaitu IMPLEMENTASI LARAVEL PADA APLIKASI MONITORING DOKUMEN BERBASIS *WEBSITE* DENGAN METODE EXTREM PROGRAMMING, kemudian melakukan perancangan dan implementasi sistem baru dapat ditarik kesimpulan berikut :

- a. Aplikasi ini dapat mendukung proses bisnis yang ada pada Perusahaan untuk melakukan monitoring dokumen.
- b. Aplikasi ini mampu menerapkan metode *extreme programming* sehingga menghasilkan aplikasi yang memenuhi kebutuhan pengguna dalam melacak dokumen.
- c. Aplikasi ini mampu membantu banyak bagian pada Perusahaan untuk memantau dan melacak dokumen.
- d. Aplikasi ini dapat menjadi alat manajemen yang baik dan efisien untuk pelacakan dan monitor dokumen.

4.2 Saran

Dari hasil perancangan dan penelitian yang dilakukan oleh penulis, terdapat pula beberapa kekurangan yang harus di kembangkan untuk ke depannya, yaitu berupa:

- a. Masih menggunakan aplikasi berbasis website dimana pengguna harus menggunakan browser untuk mengaksesnya.
- b. Belum ada fitur notifikasi email ketika terjadi perubahan status dokumen.

5. DAFTAR PUSTAKA

- [1] Ariyanti, L., Satria, M. N. D., & Alita, D. (2020). Sistem Informasi Akademik Dan Administrasi Dengan Metode Extreme Programming Pada Lembaga Kursus Dan Pelatihan. *Jurnal Teknologi Dan Sistem Informasi*, 1(1), 90-96.
- [2] Mujiono, M., Nalendra, A. K., Fauzi, D. H., & Karromah, N. (2023). IMPLEMENTASI IOT DALAM MONITORING SUHU DAN GAS AMONIA PADA KANDANG AYAM BERBASIS WEBSITE DENGAN FRAMEWORK LARAVEL. *Antivirus: Jurnal Ilmiah Teknik Informatika*, 17(1), 41-51.
- [3] Alpina, D., & Witriyono, H. (2022). Pemanfaatan Framework Laravel Dan Framework Bootstrap Pada Pembangunan Aplikasi Penjualan Hijab Berbasis Web. *Jurnal Media Infotama*, 18(1), 36-42.
- [4] Pulu, O. S., Purwanti, A. T. I., Fadly, M., & Bayu, A. (2022). PERANCANGAN DAN IMPLEMENTASI APLIKASI KEY PERFORMANCE INDICATOR (KPI) PERUSAHAAN BERBASIS WEBSITE DAN MOBILE. *Jurnal Darma Agung*, 30(1), 358-370.
- [5] Satya, J. B., Suhery, L., Sinlae, A. A. J., & Uliyatunisa, U. (2021). Pengembangan Sistem Pelayanan Publik Melalui Sistem Administrasi Kependudukan Menggunakan Metode Extreme Programming. *Jurnal Sistem Komputer dan Informatika (JSON)*, 3(2), 87-93.
- [6] Pertiwi, D. H. (2018). Metode extreme programming (xp) pada website sistem informasi franchise lkp palcomtech. *MIKROTIK: Jurnal Manajemen Informatika*, 8(1), 86-98.
- [7] Mulyadi, I. (2022, September). SM Sistem Monitoring Unit Kinerja Karyawan Melalui Telegram Bot API: Telegram Bot API. In *Seminar Nasional CORIS 2022*.
- [8] Fernandes, A. L. (2022). Perancangan Pengelolaan Arsip Data Berbasis Web Bidang Kemahasiswaan. *JR: Jurnal*

- Responsive Teknik Informatika, 6(02), 1-7.
- [9] Widyasari, Y. D. L. (2023). Sistem Monitoring Pengadaan Bahan Baku Menggunakan Metode Extreme Programming Pada Ayam Geprek Family. ABEC Indonesia, 207-214.
- [10] Hanif, M. R. (2023). Rancang Bangun Sistem Informasi Pembayaran Sekolah Berbasis Web Menggunakan Framework Laravel: Studi Kasus: SMK Walang Jaya (Doctoral dissertation, Sekolah Tinggi Teknologi Terpadu Nurul Fikri).
- [11] Hijriani, A. (2017). Aplikasi Informasi Dokter Spesialis di Bandar Lampung Berbasis Android dengan Menggunakan Teknologi Location Based Service. Jurnal Komputasi, 5(2).
- [12] Nurkholis, A., Susanto, E. R., & Wijaya, S. (2021). Penerapan Extreme Programming dalam Pengembangan Sistem Informasi Manajemen Pelayanan Publik. J-SAKTI (Jurnal Sains Komputer Dan Informatika), 5(1), 124-134.
- [13] Wahid, A. A. (2020). Analisis metode waterfall untuk pengembangan sistem informasi. J. Ilmu-ilmu Inform. dan Manaj. STMIK, no. November, 1-5.
- [14] Nur, H. (2019). Penggunaan Metode Waterfall Dalam Rancang Bangun Sistem Informasi Penjualan. Generation Journal, 3(1), 1-10.
- [15] Sasmito, G. W. (2017). Penerapan metode Waterfall pada desain sistem informasi geografis industri kabupaten Tegal. Jurnal Informatika: Jurnal Pengembangan IT, 2(1), 6-12.