Sekolah Vokasi
College of Vocational Studies

AID-E

AUTOMATED INSECT DETECTION AND ELIMINATION

CONTACT US : 0895-3252-82705 (MUHAMMAD ARSYI)

Daftar Isi

1. PENDAHULUAN

1.1 Deskripsi Alat

AID-E adalah robot pengendali hama Wereng Batang Coklat (WBC) pada tanaman padi. Alat ini berteknologi tinggi untuk membantu petani mengendalikan hama tanaman padi tanpa pestisida. Dengan menggunakan kecerdasan buatan, sensor, dan mikrokontroler, robot ini secara otomatis mendeteksi dan mengendalikan hama wereng tanpa merusak lingkungan. Dilengkapi dengan kamera dan LiDAR untuk pemantauan dan deteksi, serta menggunakan metode Support Vector Machine (SVM) untuk klasifikasi dengan akurasi 89%. Sistem IoT memungkinkan kontrol dan monitoring dari jarak jauh melalui internet. Dengan dimensi mekanik kompak dan bahan yang tahan lama, serta sistem penggerak roda rantai dan motor listrik, robot ini dapat bergerak secara otomatis. Diharapkan alat ini memberikan solusi efektif dalam pengendalian hama wereng pada tanaman padi tanpa merusak lingkungan.

1.2 Kelebihan Alat

Robot pengendali hama Wereng Batang Coklat (WBC) pada tanaman padi ini memiliki beberapa keunggulan yaitu sebagai berikut:

- Pertama, dapat mengendalikan hama tanaman padi tanpa pestisida untuk menjaga keseimbangan lingkungan pertanian.
- Kedua, menggunakan teknologi AI, sensor, dan mikrokontroler untuk mendeteksi dan mengendalikan hama secara otomatis, meningkatkan efisiensi pengendalian.
- Ketiga, dengan SVM mencapai akurasi 89%, klasifikasi hama wereng dilakukan dengan tingkat keberhasilan tinggi.
- Keempat, sistem IoT memungkinkan pengendalian jarak jauh melalui internet.
- Kelima, dengan dimensi mekanik yang disesuaikan dan penggerak roda rantai, alat ini fleksibel di berbagai kondisi lahan pertanian.

Diharapkan memberikan solusi efektif, efisien, dan ramah lingkungan dalam pengendalian hama tanaman padi.

1.3 Kekurangan Alat

Robot pengendali hama Wereng Batang Coklat (WBC) pada tanaman padi memiliki potensi kekurangan yang perlu diperhatikan yaitu sebagai berikut:

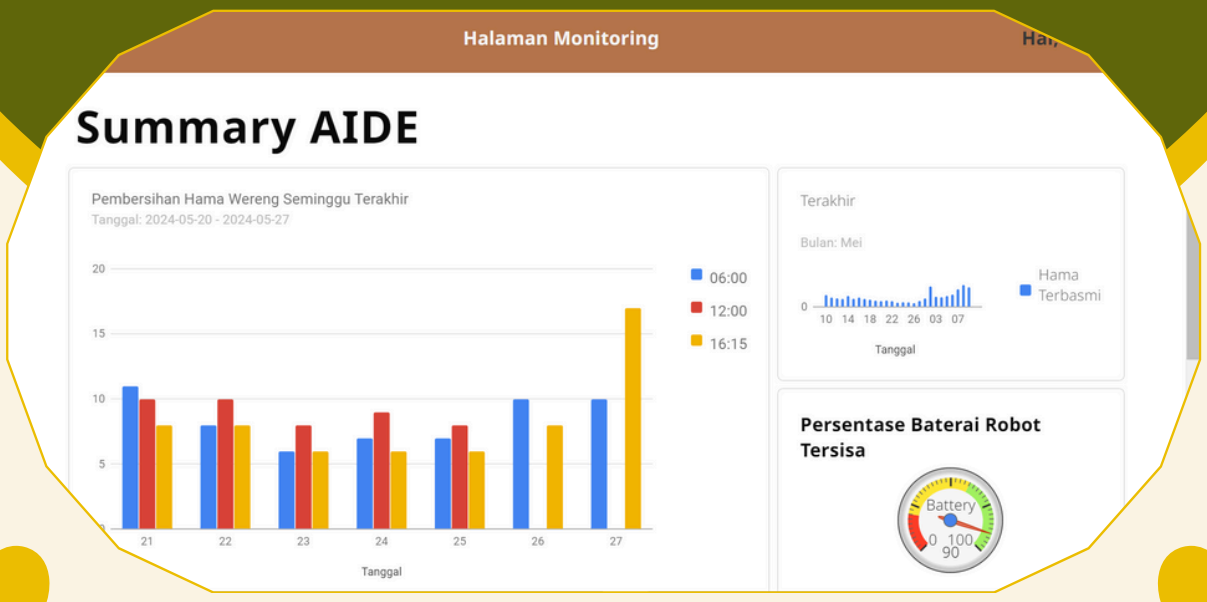
- Pertama, ketergantungan pada teknologi canggih dapat menyebabkan masalah jika terjadi kerusakan atau gangguan pada komponen.
- Kedua, penggunaan teknologi kompleks seperti SVM dan IoT dapat menimbulkan kesulitan teknis dalam pengoperasian dan pemeliharaan alat.
- Ketiga, biaya produksi yang tinggi mungkin menjadi kendala bagi petani dengan anggaran terbatas.
- Keempat, meskipun memiliki dimensi mekanik yang dapat disesuaikan, namun masih terdapat keterbatasan dalam penyesuaian dengan kondisi lahan pertanian yang beragam.
- Kelima, ketergantungan pada koneksi internet untuk sistem IoT dapat menjadi kendala jika terjadi gangguan atau ketidakstabilan jaringan.

Diperlukan perencanaan dan pengembangan yang matang untuk mengatasi kendala-kendala ini.

2. PENJELASAN

2.1 Penggunaan Alat

Untuk menggunakan alat ini, petani perlu memastikan semua komponen terpasang dengan benar dan baterai terisi penuh. Selanjutnya, sesuaikan jalur alat dengan lahan pertanian. Aktifkan sistem deteksi hama wereng dan biarkan alat bergerak otomatis di sepanjang jalur, menghisap hama yang terdeteksi. Gunakan sistem pemantauan IoT untuk kontrol jarak jauh. Lakukan pemeliharaan rutin untuk menjaga kinerja optimal. Dengan langkah-langkah ini, alat dapat digunakan secara efektif dalam mengendalikan hama tanaman padi secara otomatis dan ramah lingkungan, meningkatkan produktivitas pertanian.



2.2 Pengunaan Website

Website yang terhubung dengan alat ini merupakan platform yang memungkinkan pengguna, seperti petani, untuk memantau dan mengontrol alat secara efisien. Pengguna dapat login atau mendaftar untuk mengakses informasi tentang kondisi alat, aktivitas terkini, hasil deteksi hama, dan riwayat penggunaan. Fitur monitoring memungkinkan pengguna melihat data real-time, sementara fitur kontrol jarak jauh memungkinkan pengguna mengoperasikan alat dari kejauhan. Website juga memungkinkan analisis data untuk memahami kinerja alat dan efektivitas pengendalian hama. Ini memberikan kemudahan dalam pengelolaan pertanian modern dan optimalisasi penggunaan alat.

3. SOURCE CODE

3.1 Source Code Alat

#CODE UTAMA (main.py)

```
import os
import serial
import pickle
import time
import numpy as np
import os.path as path
# import ArducamDepthCamera as ac
from datetime import datetime as dt
from dotenv import load_dotenv
import logging
from module.connectDb import ConnectFirebase
from module.controller import navigation_robot
from module.navigation import AgentModel

# Add Log Info
logging.basicConfig(filename='output.log', level=logging.INFO)
logging.info("Mulai Program")

# Initialize camera
cam = ac.ArducamCamera()
if cam.open(ac.TOFConnect.CSI,0) != 0 :
    logging.info("initialization failed")
if cam.start(ac.TOFOutput.DEPTH) != 0 :
    logging.info("Failed to start camera")

# Initialize RL agent
if path.isfile('model.pkl'):
    with open('model.pkl','rb') as model:
        agent = model
else:
    agent = AgentModel()
```

```
# Initialize Coms Serial
while True:
    try:
        ser = serial.Serial('COM4',9600)
        break
    except Exception as e:
        logging.info(f"Gagal Terhubung Arduino Error: {str(e)}")

if __name__ == "__main__":
    try:
        load_dotenv() # Load Dependency on .env
        # Config firebase
        url_firebase = os.getenv('URL_FIREBASE', None)
        path_file_cred = os.getenv('CRED_PATH', None)
        app_fb = ConnectFirebase(path_file_cred, url_firebase)
        try:
            app_fb.get_connect() # Connect Firebase
            schedule = app_fb.get_schedule() # Get Schedule Operations
            hour_schedule = schedule.keys()
        except Exception as e:
            schedule = {'08:00': '15'}  # Default Schedule
            hour_schedule = schedule.keys()
            logging.info(f'Errors: {str(e)}')
        # Robot Running
        while True:
            # Initialize Current Time
            time_zone = dt.now()
            hour = time_zone.strftime('%H:%M')
            second = time_zone.strftime('%S')
            minute = time_zone('%M')
            # Check this time operations
            # Running Robot
            if hour in hour_schedule:
                duration = int(schedule[hour])
                navigation_robot(duration=duration, agentModel=agent, camera=cam)
            # Check 5 minute every time to update battery status
            if not int(minute)%5:
                response = ser.readline().decode().strip()
                if response:
                    try:
                        battery_value = int(float(response))
                        app_fb.update_battery(battery_value)
                        logging.info(app_fb.get_battery())
                    except Exception as e:
                        logging.info(f"Data tidak valid: {str(e)} | {type(response)}")
```

```
            # Update Schedule and Try Connect Firebase every hour
            if minute == '01':
                try:
                    # Connect to firebase
                    app_fb.get_connect()
                    # Update status robot connection log
                    app_fb.update_log()
                    # Get Schedule Operations
                    schedule = app_fb.get_schedule()
                    hour_schedule = schedule.keys()
                except Exception as e:
                    logging.info(f'Errors: {str(e)}')
    except Exception as e:
        logging.info(f'Errors {str(e)}')
```

#CODE TRAINING MODEL (train.py)

```
import numpy as np
import time
import keyboard
import ArducamDepthCamera as ac
import serial
import pickle
import os.path as path
import logging

class AgentModel:
    def __init__(self, alpha=0.1, epsilon=0.1, road_condition=0, parameter=80, threshold=60,
    width_agent=57, parameter_road_condition=20):
        self.alpha = alpha
        self.epsilon = epsilon
        self.parameter = parameter  # Parameter untuk aturan jarak depan sensor
        self.threshold = threshold
        self.road_condition = road_condition  # 0 untuk kering, 1 untuk licin
        self.width_agent = width_agent
        self.parameter_road_condition = parameter_road_condition

    def choose_action(self, state):
        left, front, right = state
        # Jika jalan licin, gunakan ambang batas yang lebih tinggi
        if self.road_condition == 1:
            self.parameter = self.parameter + self.parameter_road_condition
```

```
        else:
            self.parameter = self.parameter
        # Parameter 0 adalah ambang batas untuk jarak tengah
        if front > self.parameter:
            action = 1
        # Parameter 1 adalah ambang batas lain untuk jarak tengah
        elif self.threshold < front <= self.parameter:
            if right > left:
                action = 2     # kanan
            else:
                action = 3     # kiri
        elif front <= self.threshold:
            action = 4       # mundur
        else:
            action = 0    # Mobil berhenti untuk menilai lingkungan
        return action

    def update_parameters(self, state, action):
        # Hitung gradien dari expected reward terhadap parameter
        grad = 0
        old_parameter = self.parameter
        self.parameter += self.epsilon * self.parameter
        reward_plus = self.get_expected_reward(state, action)
        self.parameter = old_parameter
        self.parameter -= self.epsilon * self.parameter
        reward_minus = self.get_expected_reward(state, action)
        grad = (reward_plus - reward_minus) / (2 * self.epsilon)
        self.parameter = old_parameter

        # Perbarui parameter menggunakan gradien
        self.parameter += self.alpha * grad

    def get_expected_reward(self, state, action):
        # Implementasikan fungsi ini untuk menghitung expected reward
        left, front, right = state
        if action == 0 or action == 4 or front < self.parameter:
            reward = 1
        elif action == 2 or action == 3:
            if abs(right - left) > (self.width_agent/2):
                reward = -1
        else:
            reward = 0
        return reward
```

```
    def get_com(self, action):
        # Mengirim sinyal ke Arduino berdasarkan action
        if action == 0:
            ser.write(b'berhenti')
            # pass
        elif action == 1:
            ser.write(b'maju')
            # pass
        elif action == 2:
            ser.write(b'kanan')
            # pass
        elif action == 3:
            ser.write(b'kiri')
            # pass
        elif action == 4:
            ser.write(b'mundur')
            # pass

def get_current_state(depth_buf):
    # Mengambil jarak di posisi tengah, kiri, dan kanan frame
    y = depth_buf.shape[0] // 2
    x_center = depth_buf.shape[1] // 2
    x_left = depth_buf.shape[1] // 4
    x_right = depth_buf.shape[1] * 3 // 4
    center_distance = depth_buf[y, x_center] * 100  # Mengubah jarak ke cm
    left_distance = depth_buf[y, x_left] * 100  # Mengubah jarak ke cm
    right_distance = depth_buf[y, x_right] * 100  # Mengubah jarak ke cm
    state = (left_distance, center_distance, right_distance)
    return state

# Initialize RL agent
if path.isfile('model.pkl'):
    with open('model.pkl','rb') as model:
        agent = model
else:
    agent = AgentModel()

# Add Log Info
logging.basicConfig(filename='output.log', level=logging.INFO)
logging.info("Mulai Program")
# Cek Kondisi jalanan
road_condition = 0 # Update dengan API
agent.road_condition = road_condition
```

```
# Membuat objek serial untuk komunikasi dengan Arduino
ser = serial.Serial('/dev/ttyACM0', 9600)
cam = ac.ArducamCamera()
# Initialize camera
if cam.open(ac.TOFConnect.CSI,0) != 0 :
    logging.info("initialization failed")
if cam.start(ac.TOFOutput.DEPTH) != 0 :
    logging.info("Failed to start camera")
# Main loop
start_time = time.monotonic()
while True:
    try:
        # Get current state
        frame = cam.requestFrame(200)
        if frame != None:
            depth_buf = frame.getDepthData()
            cam.releaseFrame(frame)
            state = get_current_state(depth_buf)
            # Choose action
            action = agent.choose_action(state)
            # Com serial
            agent.get_com(action)
            # Update Parameters
            agent.update_parameters(state, action)
        current_time = time.monotonic() # Check current time
        elapsed_time = current_time - start_time
        # Stop program when time out
        if elapsed_time >= 1:
            # Mengirim sinyal ke Arduino berdasarkan action
            if action == 0:
                status = 'berhenti'
            elif action == 1:
                status = 'maju'
            elif action == 2:
                status = 'kanan'
            elif action == 3:
                status = 'kiri'
            elif action == 4:
                status = 'mundur'
            logging.info(f"Status: {status} | Parameters: {agent.parameter}\nDistance: {state}")
            start_time = current_time
    except KeyboardInterrupt:
        break
with open('model.pkl','wb') as model:
    pickle.dump(agent,model)
```

 # CODE MODULE FIREBASE (connectDb.py)

```
import firebase_admin
from firebase_admin import db
from datetime import datetime as dt

class ConnectFirebase:
    def __init__(self, path_cred, url_firebase):
        self.path_cred = path_cred
        self.url_firebase = url_firebase

    def get_connect(self):
        try:
            cred_obj = firebase_admin.credentials.Certificate(self.path_cred)
            firebase_admin.initialize_app(cred_obj, {'databaseURL': self.url_firebase})
            return f"Berhasil terhubung dengan Firebase!"
        except Exception as e:
            print(f"Gagal terhubung dengan Firebase. Error: {str(e)}")

    def get_battery(self):
        ref = db.reference('/battery')
        return ref.get()

    def update_battery(self,battery):
        ref = db.reference('/battery')
        try:
            ref.set(battery)
            return f"Berhasil update status battery {self.get_battery()}"
        except Exception as e:
            print(f'Gagal update status battery. Error: {str(e)}')

    def get_schedule(self):
        ref = db.reference('/operations').get()
        data = dict()
        for key, value in ref.items():
            data[value['started']] = value['duration']
        return data

    def add_pest(sef, hour, pest):
        time = dt.now()
        year, month, day = time.strftime('%Y'), time.strftime('%m'), time.strftime('%d')
        try:
            ref = db.reference(f'/pests/{year}/{month}/')
            ref.child(day).child(hour).set(pest)
            return "Berhasil menambahkan data hama"
```

```
        except Exception as e:
        print(f"Gagal menambahkan jumlah hama. Error {str(e)}")

def update_log(self):
    log = dt.now().strftime(f'%Y-%m-%d %H:%M')
    ref = db.reference(f'/log')
    try:
        ref.set(log)
        return f"Berhasil update status log {log}"
    except Exception as e:
        print(f"Gagal update status log. Error {str(e)}")
```

3. SOURCE CODE

3.2 Source Code Web

#CODE ROUTE WEBSITE (web.php)

```
<?php

use App\Http\Controllers\ProfileController;
use App\Http\Controllers\MonitorController;
use App\Http\Controllers\AdminController;
use App\Http\Controllers\AboutController;
use App\Http\Controllers\FeatureController;
use App\Http\Controllers\DocumentationController;
use App\Http\Controllers\TeamController;
use App\Models\About;
use App\Models\Feature;
use App\Models\Documentation;
use App\Models\Team;
use Illuminate\Support\Facades\Route;

// Index Home
Route::get('/', function () {
    $abouts = About::get();
    $features = Feature::get();
    $documentations = Documentation::get();
    $teams = Team::get();

    return view('welcome',[
        'abouts' => $abouts,
        'features' => $features,
        'documentations' => $documentations,
        'teams' => $teams,
    ]);
})->name('home');
```

```
// Profile Seting
Route::middleware('auth')->group(function () {
    Route::get('/profile', [ProfileController::class, 'edit'])->name('profile.edit');
    Route::patch('/profile', [ProfileController::class, 'update'])->name('profile.update');
    Route::delete('/profile', [ProfileController::class, 'destroy'])->name('profile.destroy');
});

// Auth Resource
require __DIR__.'/auth.php';

// Monitor Resource
Route::prefix('dashboard')->middleware('auth')->group(function () {
    Route::get('/', [MonitorController::class, 'index'])->name('dashboard.index');
    Route::post('/store', [MonitorController::class, 'store'])->name('dashboard.store')-
>middleware('admin');
    Route::post('/update', [MonitorController::class, 'update'])->name('dashboard.update')-
>middleware('admin');
    Route::post('/destroy', [MonitorController::class, 'destroy'])->name('dashboard.destroy')-
>middleware('admin');
});

// Admin Controller
Route::prefix('admin')->middleware('admin')->group(function(){
    Route::get('/', [AdminController::class, 'index'])->name('admin.index');
    Route::post('/update/{id}', [AdminController::class, 'update'])->name('admin.update')-
>middleware('root');
    // About
    Route::resource('about', AboutController::class)->except([
        'index','create', 'show'
    ]);
    // Feature
    Route::resource('feature', FeatureController::class)->except([
        'index','create', 'show'
    ]);
    // Documentation
    Route::resource('documentation', DocumentationController::class)->except([
        'index','create', 'show'
    ]);
    // Team
    Route::resource('team', TeamController::class)->except([
        'index','create', 'show'
    ]);
});
```

```
@extends('layouts.guest')
@section('title', "Landing Page")

@section('content')
<!-- Navbar -->
<header class="container-md-fluid pt-md-3 mb-md-3 px-md-5 bg-nav fixed-top">
    <nav class="navbar navbar-expand-md px-4 py-3" id="navbar">
        <div class="container-fluid">
            <a class="navbar-brand" href="{{ route('home') }}">
                <img src="{{ url('dev/image/logo.png') }}" alt="Logo AIDE">
                AIDE
            </a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <ul class="navbar-nav ms-md-auto mt-md-0 mt-2 text-center text-md-start">
                    <li class="nav-item">
                        <a class="nav-link" aria-current="page" href="#about">Tentang Kami</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="#goals">Fitur</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="#docum">Dokumentasi</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="#tim">Tim</a>
                    </li>
                    @guest
                    <li class="nav-item ms-md-4">
                        <a class="btn btn-primary" href="{{ route('login') }}">Mulai</a>
                    </li>
                    @endguest
                    @auth
                    <li class="nav-item dropdown">
                        <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-
toggle="dropdown" aria-expanded="false">
                            Hai, {{ Str::limit(Auth::user()->name, 5, '...') }}
                        </a>
                        <ul class="dropdown-menu bg-primary p-2">
                            <li><a class="dropdown-item fs-5" href="{{ route('dashboard.index')
}}">Dashboard</a></li>
```

```
                        <li><a class="dropdown-item fs-5" href="{{ route('profile.edit')
}}">Profil</a></li>
                        <li>
                            <hr class="dropdown-divider">
                        </li>
                        @if(Auth::user()->role == 'admin' or Auth::user()->role == 'root')
                        <li><a class="dropdown-item fs-5" href="{{ route('admin.index')
}}">Admin</a></li>
                        @endif
                        <button type="button" class="dropdown-item btn btn-danger fs-5
text-center" data-bs-toggle="modal" data-bs-target="#modalLogout">Keluar</button>
                    </ul>
                </li>
                @endauth
            </ul>
        </div>
    </div>
</nav>
</header>

@auth
<!-- Modal Logout -->
<div class="modal fade" id="modalLogout" tabindex="-1" aria-
labelledby="modalLogoutLabel" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered modal-dialog-scrollable">
        <form method="POST" action="{{ route('logout') }}" class="modal-content">
            @csrf
            <div class="modal-header">
                <h1 class="modal-title fs-5" id="modalLogoutLabel">Kembali Kehalaman
Awal</h1>
                <button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
            </div>

            <div class="modal-body">
                <span>Apakah anda yakin untuk keluar dari halaman monitoring sebagai
Admin?</span>
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary w-auto" data-bs-
dismiss="modal">Batal</button>
                <button type="submit" class="btn btn-danger w-auto">Keluar</button>
            </div>
        </form>
    </div>
```

```
</div>
@endauth

<!-- Main Content -->
<main data-bs-spy="scroll" data-bs-target="#navbar" data-bs-root-margin="0px 0px
-40%" data-bs-smooth-scroll="true" tabindex="0">
    <div class="banner">
        <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 1440 320">
            <path fill="#3b291d" fill-opacity="1"
d="M0,0L48,10.7C96,21,192,43,288,74.7C384,107,480,149,576,176C672,203,768,213,864,202.7
C960,192,1056,160,1152,144C1248,128,1344,128,1392,128L1440,128L1440,320L1392,320C1344,3
20,1248,320,1152,320C1056,320,960,320,864,320C768,320,672,320,576,320C480,320,384,3
20,288,320C192,320,96,320,48,320L0,320Z"></path>
        </svg>
        <div class="container">
            <h1 class="text-center">
                Mulai Revolusi Pertanian Indonesia dengan Langkah Kecil yang Ramah
Lingkungan
            </h1>
        </div>
    </div>

    @isset($abouts)
    <section id="about" class="about pb-5 mb-4">
        <h1 class="text-center">Tentang Kami</h1>
        <div class="container-fluid mt-2 px-md-5">
            <div class="row justify-content-center align-items-center">
                @foreach ($abouts as $about)
                <div class="col-lg-5 col-md-6 p-3">
                    <img src="{{ Storage::url($about->path) }}" alt="About" class="img-thumbnail
rounded-4">
                </div>
                <div class="col-lg-7 col-md-6 p-3">{!! Str::markdown($about->description) !!}
</div>
                @endforeach
            </div>
        </div>
    </section>
    @endisset
    @isset($features)
    <section id="goals" class="goals pb-5">
        <h1 class="text-center">Fitur</h1>
        <div class="container mt-2">
            <div class="row justify-content-center align-items-stretch">
```

```
                @foreach ($features as $feature)
                <div class="col-lg-4 col-md-8 p-3">
                    <div class="card text-white bg-primary rounded-4 w-100 h-100">
                        <img class="card-img-top rounded-4" src="{{ Storage::url($feature->path)
}}" alt="{{ $feature->name }}">
                        <div class="card-body">
                            <h3 class="card-title text-center">{{ $feature->name }}</h3>
                            <p class="card-text">{!! Str::markdown($feature->description) !!}</p>
                        </div>
                    </div>
                </div>
                @endforeach
            </div>
        </div>
    </section>
    @endisset

    @isset($documentations)
    <section id="docum" class="docum pb-5">
        <h1 class="text-center">Dokumentasi</h1>
        <div class="container pt-4">
            <div class="row justify-content-center align-items-start">
                <!-- Nav Docum -->
                <div class="col-md-4 col-lg-3 p-2">
                    <div class="nav flex-column me-3" id="docum-tab" role="tablist" aria-
orientation="vertical">
                        @foreach ($documentations as $i => $docum)
                        <button @class(['nav-link', 'active'=> $i == 0]) id="docum-{{ $docum->id
}}-tab" data-bs-toggle="pill" data-bs-target="#docum-{{ $docum->id }}" type="button"
role="tab" aria-controls="docum-{{ $docum->id }}" aria-selected="true">{!! Str::title($docum-
>name) !!}</button>
                        @endforeach
                    </div>
                </div>

                <!-- Content Docum -->
                <div class="col-md-8 col-lg-9 p-2">
                    <div class="tab-content" id="docum-tabContent">
                        @foreach ($documentations as $i => $docum)
                        <div @class(['tab-pane fade', 'show active'=> $i == 0]) id="docum-{{
$docum->id }}" role="tabpanel" aria-labelledby="docum-{{ $docum->id }}-tab"
tabindex="0">
                            <figure class="figure p-lg-4 p-md-3 p-2">
                                <div class="figure-img img-fluid rounded" style="background-
image: url('{{ Storage::url($docum->path) }}');"></div>
```

```
                                <figcaption class="figure-caption">
                                {!! Str::markdown($docum->description) !!}
                            </figcaption>
                        </figure>
                    </div>
                    @endforeach
                </div>
            </div>
        </div>
    </section>
    @endisset
    @isset($teams)
    <section id="tim" class="tim pb-5">
        <h1 class="text-center">Anggota Tim</h1>
        <div class="container-fluid px-md-5 pt-4">
            <div class="row justify-content-around align-items-start">
                @foreach ($teams as $team)
                <div class="col-md-4 col-lg-3 col-6 p-md-4 p-2">
                    <div class="card">
                        <div class="card-body">
                            <div class="img-thumbnail" style="background-image: url({{
Storage::url($team->path) }});"></div>
                            <h6 class="card-subtitle mt-4 text-body-secondary">{{
Str::title($team->division) }}</h6>
                            <h5 class="card-title">{{ Str::title($team->name) }}</h5>
                            <h6 class="card-subtitle mb-2 text-body-secondary">{{
Str::upper($team->nim) }}</h6>
                            <a href="{{ $team->linkedin }}" target="_blank" class="card-link">
                                <i class="bi bi-linkedin"></i>
                            </a>
                        </div>
                    </div>
                </div>
                @endforeach
            </div>
        </div>
    </section>
    @endisset
</main>
@endsection
@section('add-script')
<!-- Navbar JS -->
<script src="{{ url('dev/script/index.js') }}"></script>
@endsection
```

Sekolah Vokasi
College of Vocational Studies

AID-E

AUTOMATED INSECT DETECTION AND ELIMINATION

CONTACT US : 0895-3252-82705 (MUHAMMAD ARSYI)