

# PORTOFOLIO

**Muhammad Naufal Pratama**

[mhnpratama@gmail.com](mailto:mhnpratama@gmail.com)

<https://linkedin.com/in/mnpratamaa>

<https://github.com/mhnaufal>

<https://medium.com/@mprtmma>

<https://mhnaufal.vercel.app/>

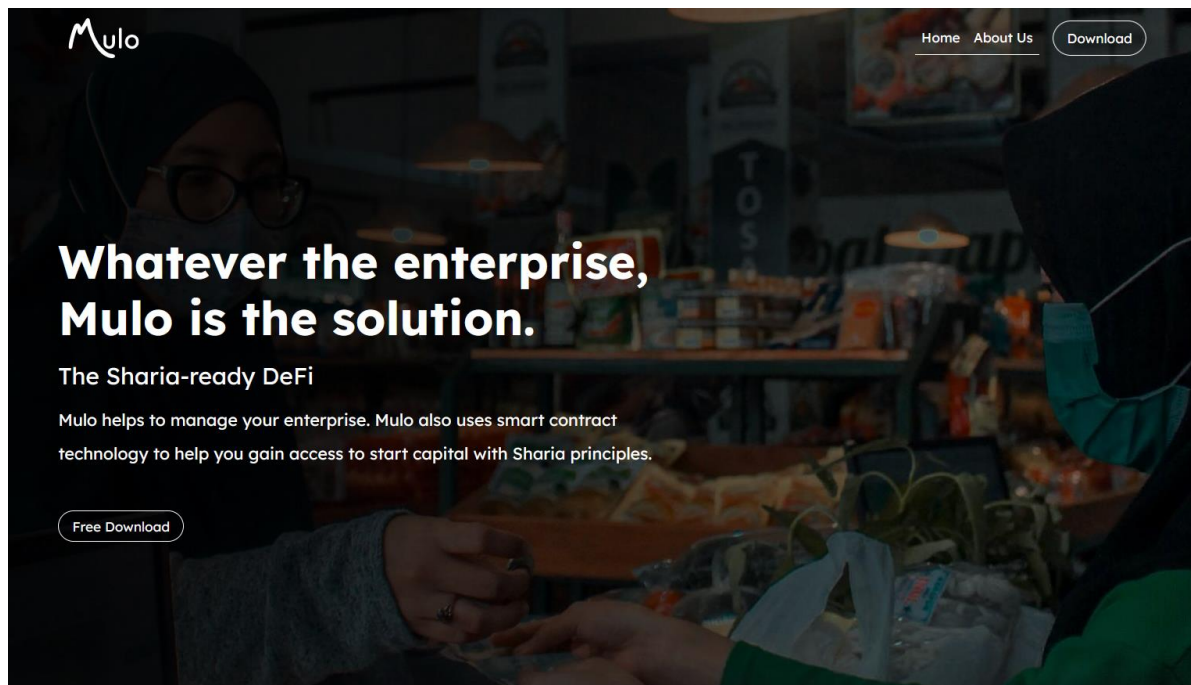
# Mulo

**About:** Mulo is a startup that develops financial management applications based in Wonosobo. I've been on this team for 1.5 years and was responsible for the backend system made using SailsJS and the android application made using Kotlin.

**Tech:** NodeJS, SailsJS, Kotlin, Git, MySQL, Insomnia

**Link:**

- <https://muloapp.id/en/>
- <https://www.instagram.com/mulo.app/?hl=en>
- <https://play.google.com/store/apps/details?id=app.mulo&hl=en&gl=US&pli=1>



# MULO | Pencatatan Kas Keuangan

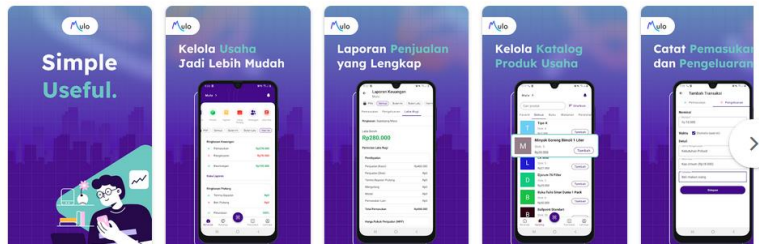


Mulo Solution

100+  
Download Semua Umur

Tambahkan ke wishlist

Aplikasi ini tidak tersedia untuk semua perangkat Anda

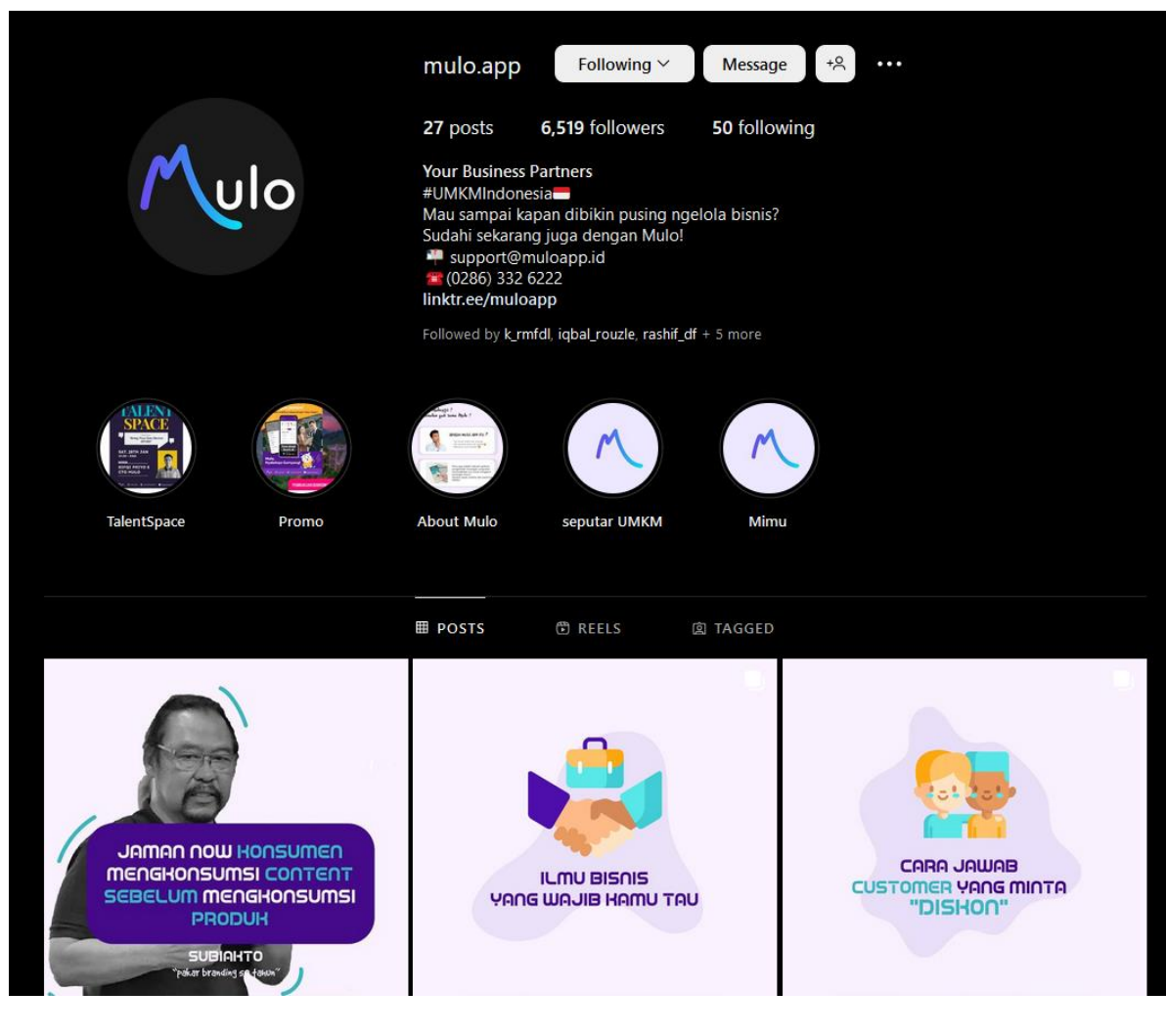


Kontak developer

Situs  
<https://muloapp.id>

Email  
[support@muloapp.id](mailto:support@muloapp.id)

Alamat  
Jalan Diponegoro Km. 2, Bugangan, Wonosobo,  
Jawa Tengah 56319



## Personal Protective Equipment (PPE) Detection

**About:** PPE Detection is one of three final projects during my internship at PT Nodeflux Teknologi Indonesia. I worked with the other three members in this project to develop a PPE detection. The PPE detection runs on the web platform, and the purpose of this software is to detect whether there are workers who do not use safety equipment such as helmets and vests. The detection is done using deep learning models. My role for this project is Software Engineer, who manages the software architecture and codebase.

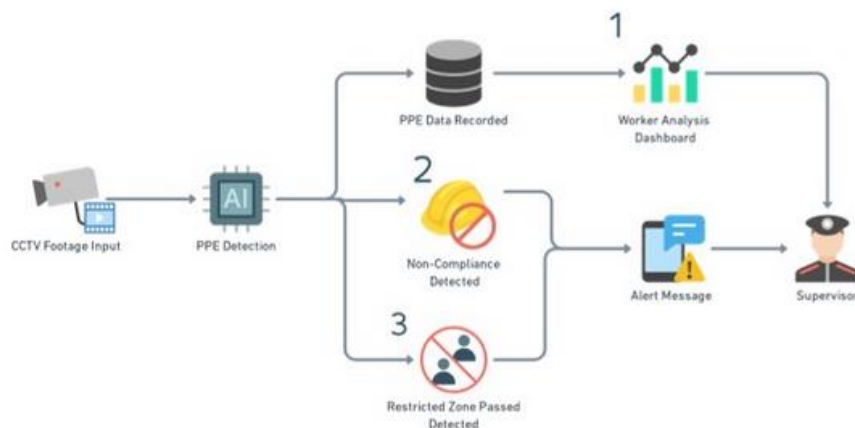
**Tech:** Python, Streamlit, Tensorflow, Git

**Link:**

- [https://www.linkedin.com/posts/christianale\\_personal-protective-equipment-prd-visionaire-activity-7015619743797694464-b901?utm\\_source=share&utm\\_medium=member\\_desktop](https://www.linkedin.com/posts/christianale_personal-protective-equipment-prd-visionaire-activity-7015619743797694464-b901?utm_source=share&utm_medium=member_desktop)

## Product Use Case

“As a user, I want to **know whether each worker always uses their PPE properly or not** so that I can take actions that ensure their safety“. From this user story, there are three use cases where this knowledge can be used.



### 1. Worker Analysis

Automatically gather and provide information about each worker's compliance score and events that happened over time.

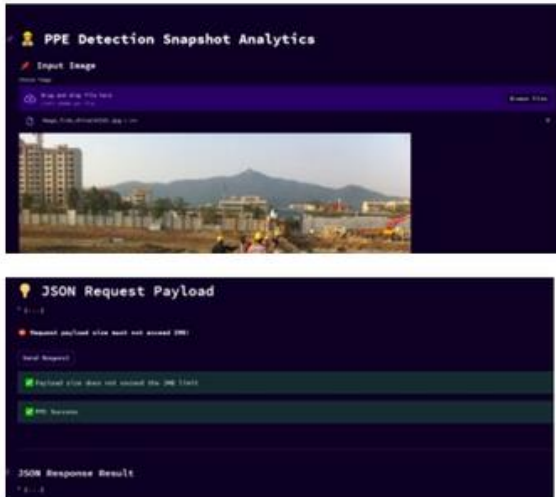
### 2. PPE Non-Compliance Alert

Automatically sends an alert message to the supervisor when non-compliant workers are detected.

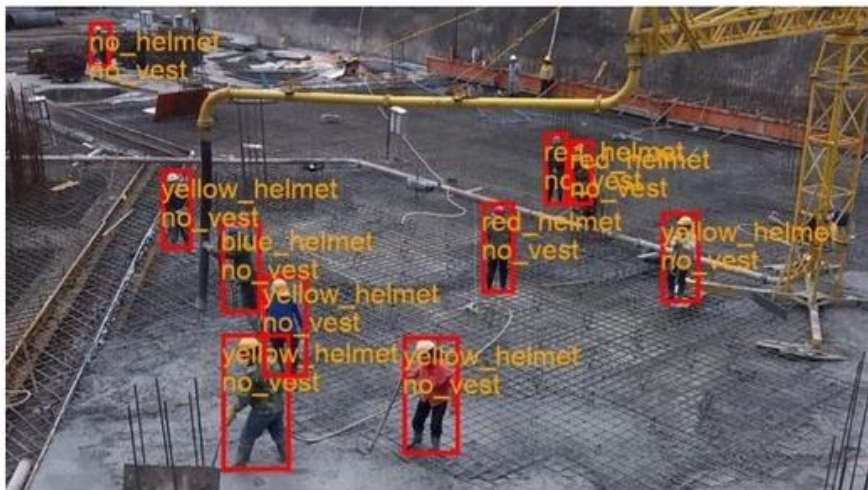
### 3. Restricted Zone Alert

Automatically sends an alert message to the supervisor when the restricted zone is passed by unauthorized workers.

## User Interface



## Example of Output Image

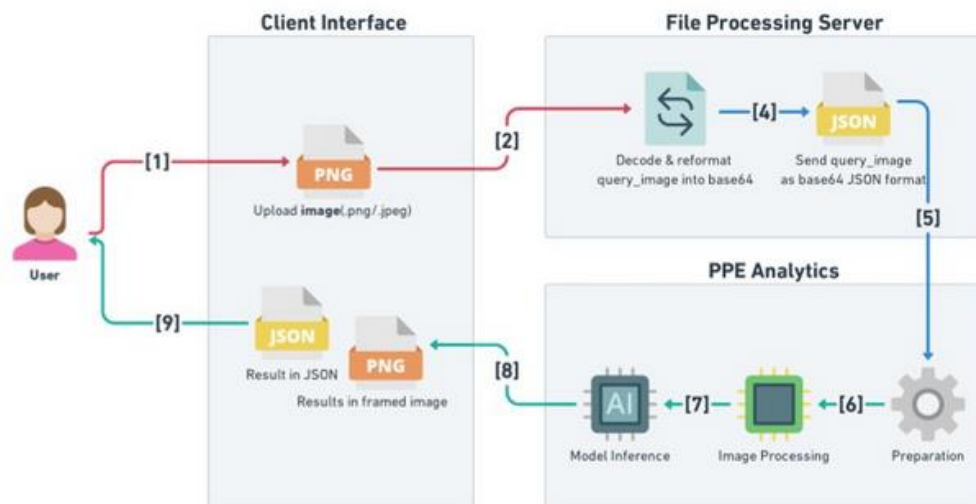


**PT. NODEFLUX TEKNOLOGI INDONESIA**  
Jl. Kemang Timur No.24  
Jakarta Selatan 12730  
P : 021 - 22718184

## Software

The software will be a website where the user can:

1. Upload the image from the user
2. Process the file as API input, send the API request, and receive the API response
3. View the result in JSON and image with the bounding boxes





## Software Defect Prediction

**About:** Software defect prediction is an Undip research focused on detecting defects in software using machine learning models and deep learning. We worked as a team of 5 people, and here I contributed. I worked as the programmer who did the code and helped others write the final paper.

**Tech:** Python, Tensorflow, Git

**Link:**

- <https://github.com/mhnaufal/Software-Defect-Finale>
- <https://github.com/mhnaufal/Software-Defect-Finale/blob/main/reports/PAPER-Software%20Defect%20Prediction.pdf>



# Software Defect Prediction

Machine learning model for 'Software Defect Prediction' using deep learning and based on Python & Tensorflow

## Introduction

- Datasets

Datasets for this project taken from [PROMISE public dataset](#)

- Models

CNN, RNN, LSTM, Random Forest, and more

- Results

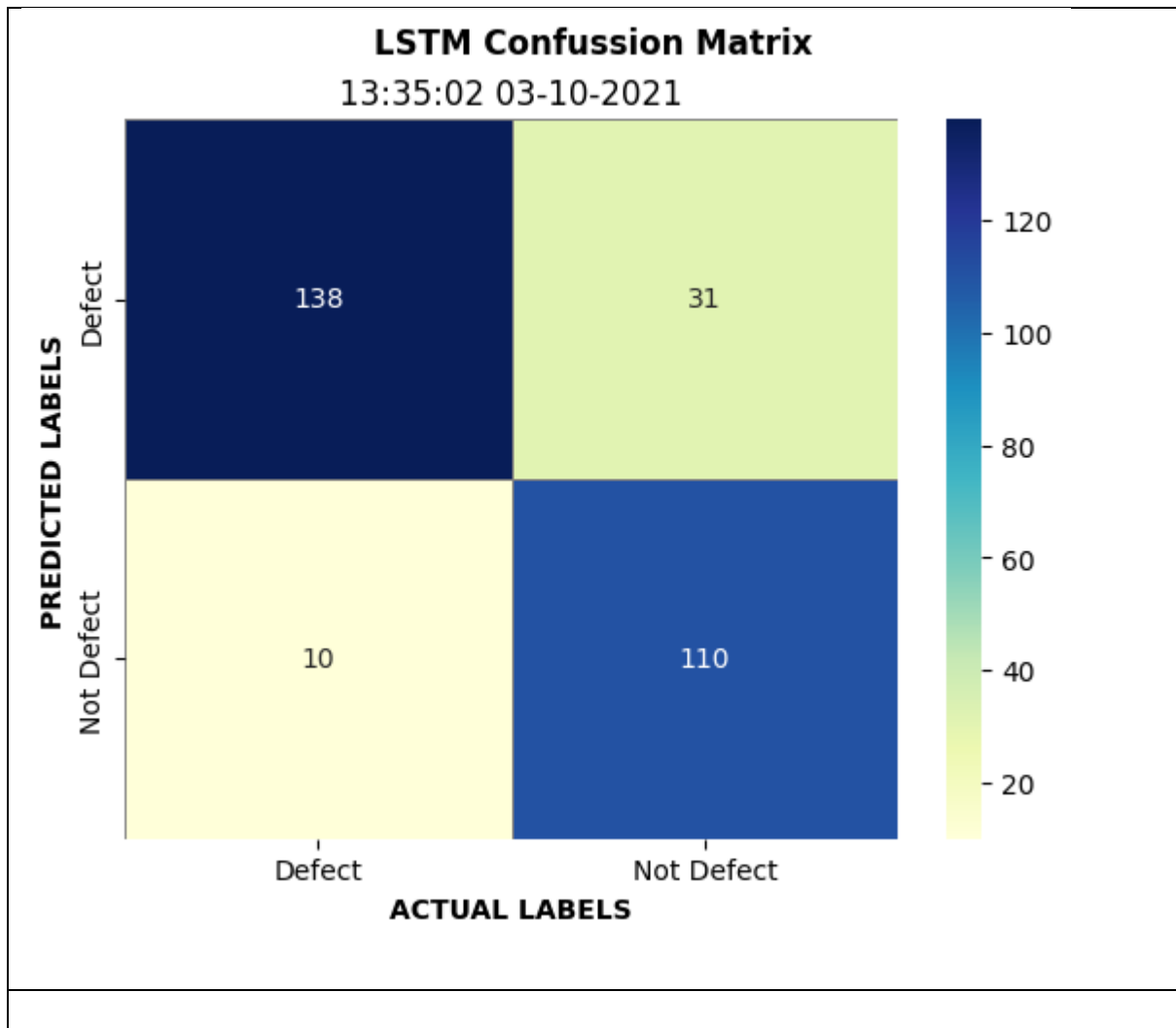
[Results directory](#)

[Paper](#)

*Results from running model(s) shown in reports folder*

## Project Structure

```
| datasets
|  └─ processed
|     └─ big_data1.csv
|     └─ big_data2.csv
|     └─ pc4.csv
|     └─ pc3.csv
|     └─ ...
|  └─ raw
|     └─ ...
|
| references
|  └─ README.md
|
| reports
|  └─ figures
|     └─ confusion matrix
|     └─ random_forest.png
```



# Software Defect Prediction Using Deep Learning

Achmad Iqbal Al Faizin<sup>1, a)</sup>, Edy Suharto<sup>1, b)</sup>, Aris Puji Widodo<sup>1, c)</sup>, Hendinur Faizal<sup>1, d)</sup>, Muhammad Naufal Pratama<sup>1, e)</sup>, Raihan Mufadhal<sup>1, f)</sup>, and Rafli Azra Virendra Azhari<sup>1, g)</sup>

## Author Affiliations

<sup>1</sup>*Department of Computer Science/Informatics, Diponegoro University, Semarang, Indonesia*

## Author Emails

<sup>a)</sup> [iqbalrouzle@students.undip.ac.id](mailto:iqbalrouzle@students.undip.ac.id)

<sup>b)</sup> [edys@lecturer.undip.ac.id](mailto:edys@lecturer.undip.ac.id)

<sup>c)</sup> [arispu@gmail.com](mailto:arispu@gmail.com)

<sup>d)</sup> [hendimurfaizal@students.undip.ac.id](mailto:hendimurfaizal@students.undip.ac.id)

<sup>e)</sup> [muhhammadnaufalp@students.undip.ac.id](mailto:muhhammadnaufalp@students.undip.ac.id)

<sup>f)</sup> [rmfdl@students.undip.ac.id](mailto:rmfdl@students.undip.ac.id)

<sup>g)</sup> [rafliazra@students.undip.ac.id](mailto:rafliazra@students.undip.ac.id)

**Abstract.** In the process of software development, a defect is often present but hard to detect if the software in development is complex. This could be a problem for customers or the end user. To increase the reliability of the software a plethora of methods can be used to detect these defects, one of them is using deep learning. The use of deep learning can help us to extract features automatically using a neural network instead of extracting them manually. The dataset used for this paper is taken from the public PROMISE repository. The result of this research will be gauged against preceding methods on its effectiveness.

## KEYWORDS

Software defects, machine learning, deep learning.

## INTRODUCTION

Software defect is a condition where a software product is not up to the standards of softwares or when the software is not working as expected of the end user. In other words, a defect is an error in the coding or logic that causes the program to not function properly or have outcomes that are wrong or unexpected.

Software defect is also defined as an error happening to the software because of a faulty code, documentation, or to the design that causes a failure in the software's performance. By making predictions on software defects, the expected outcome is to decrease the chances of a software to experience failure mostly during production so that it will be easier when testing occurs.

To detect the defects in the softwares, we are using two approaches, first using machine learning and secondly using deep learning algorithms. Using these methods, we will compare which model from the two approaches that will hold the better outcome. The dataset that will be used in this research are from the PROMISE public dataset

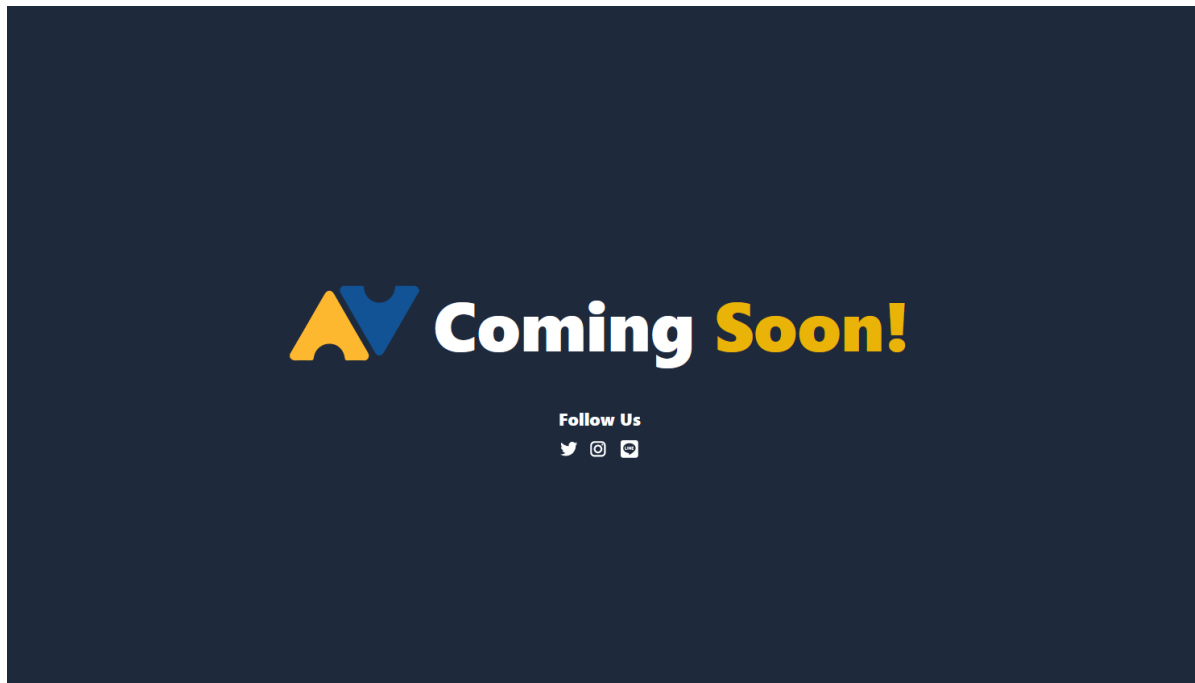
## Anforcom 2021

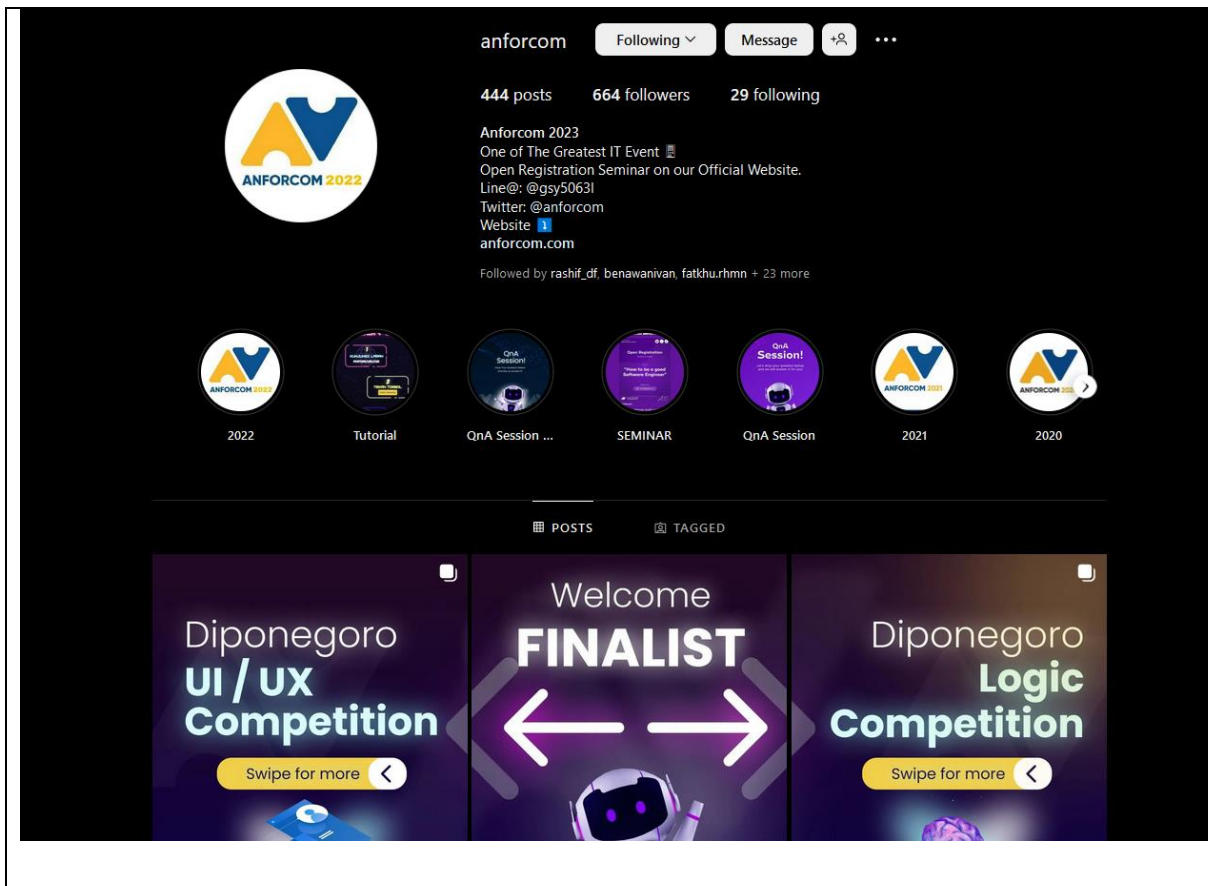
**About:** Anforcom is an IT event and competition held annually by Informatics students of Universitas Diponegoro. In this team project, with the other 5 members, I was responsible for the backend system of Anforcom. Using Laravel, MySQL, Git, and hosting providers, we deliver the website to the participants so that they can get more information about Anforcom and register for the events and competitions.

**Tech:** Laravel, PHP, MySQL, Git

**Link:**

- <https://www.anforcom.com/>





## Miscellaneous

**About:** A Covid-19 API made using the hardest programming language syntax to read, C++

**Tech:** C++, Docker, CMake

**Link:**

- <https://github.com/mhnaufal/kopid-api>

### Kopit 19

COVID 19 API built using C++++ because it's C++ + Crow + COVID

#### About

☰ README.md

#### Getting Started

Below are some of the instructions on how to use this API or head to this [Docs](#)

#### Prerequisites

Make sure you fulfill these requirements

- 🐼 C++ (11/14) yes I use this language dude 😭
- 🦉 Crow it helps me to survive the C++ world
- 🐳 Docker (optional) thanks you are here, dude 🙏 ("of course")

#### Installation

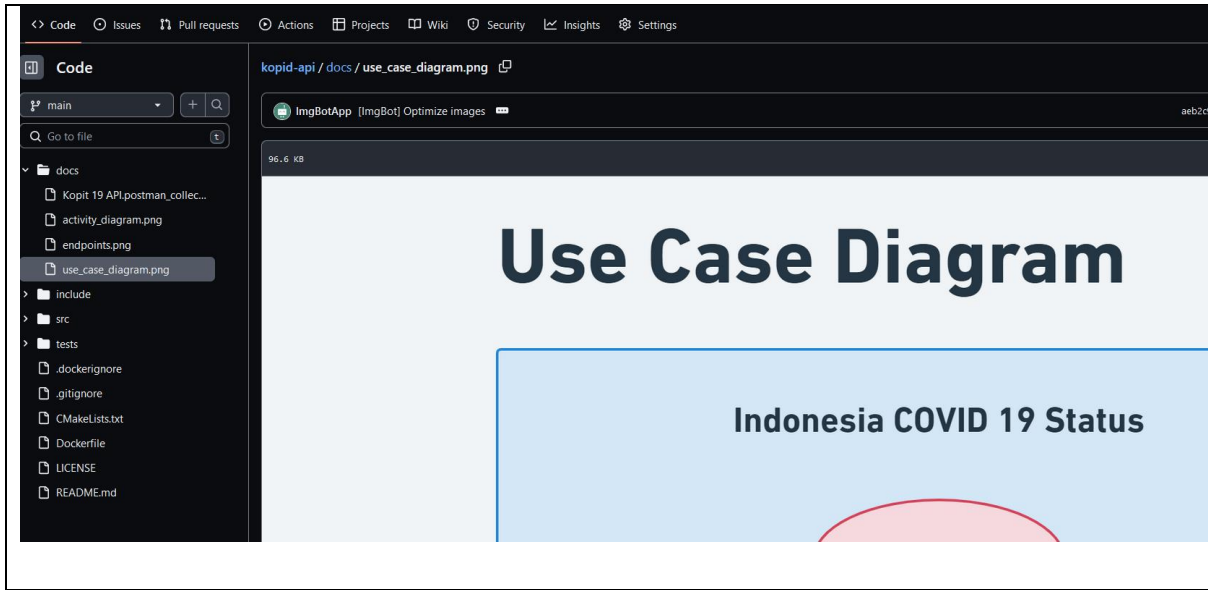
Follow these steps to install this API

##### Manual (Recommended)

1. Clone and go into the directory

```
git clone https://github.com/mhnaufal/kopid-api.git
cd kopid-api
```

2. Configure CMake





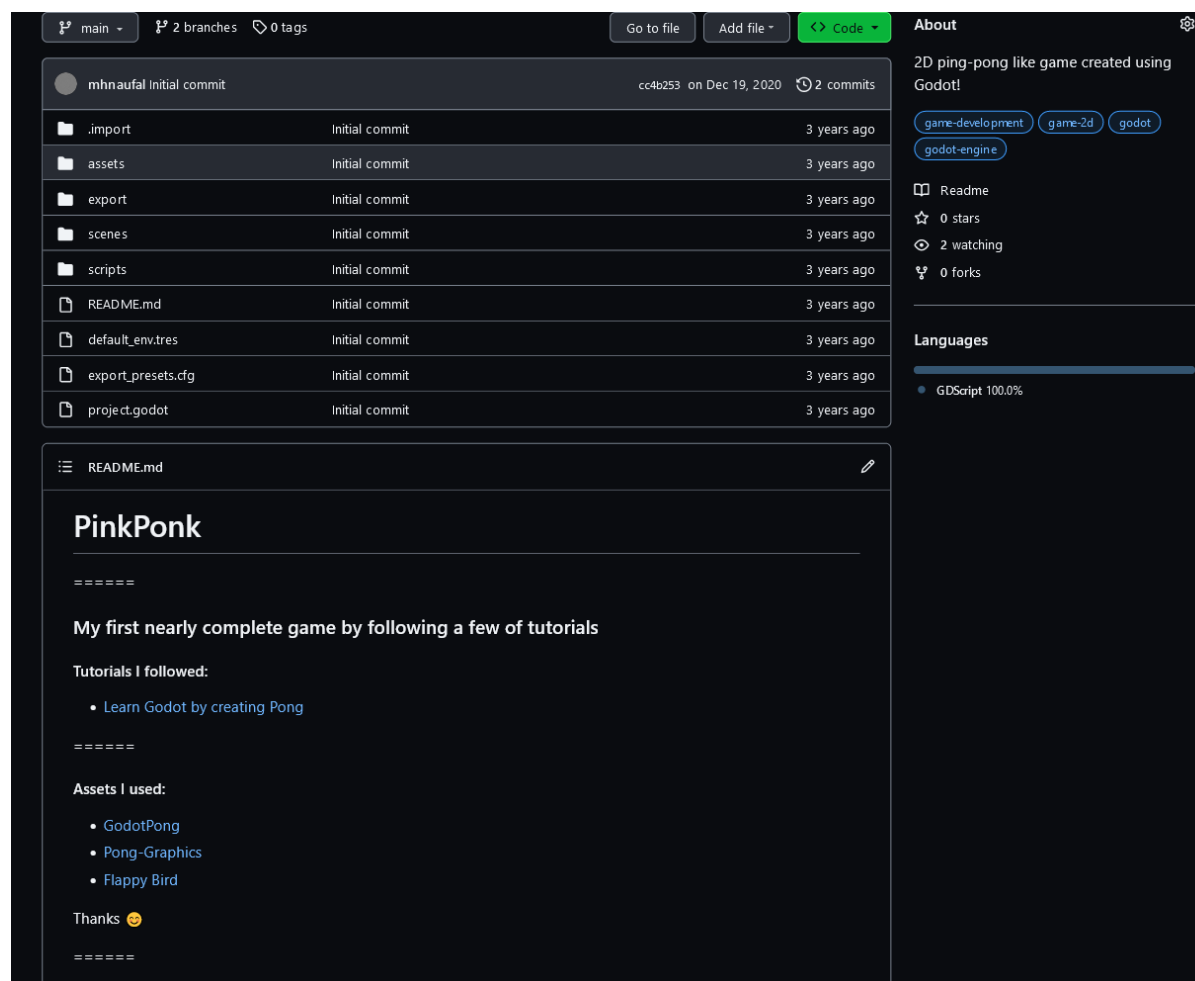
## Game Related

**About:** Some of game related projects I made. Although it's not much, I learn much from them.

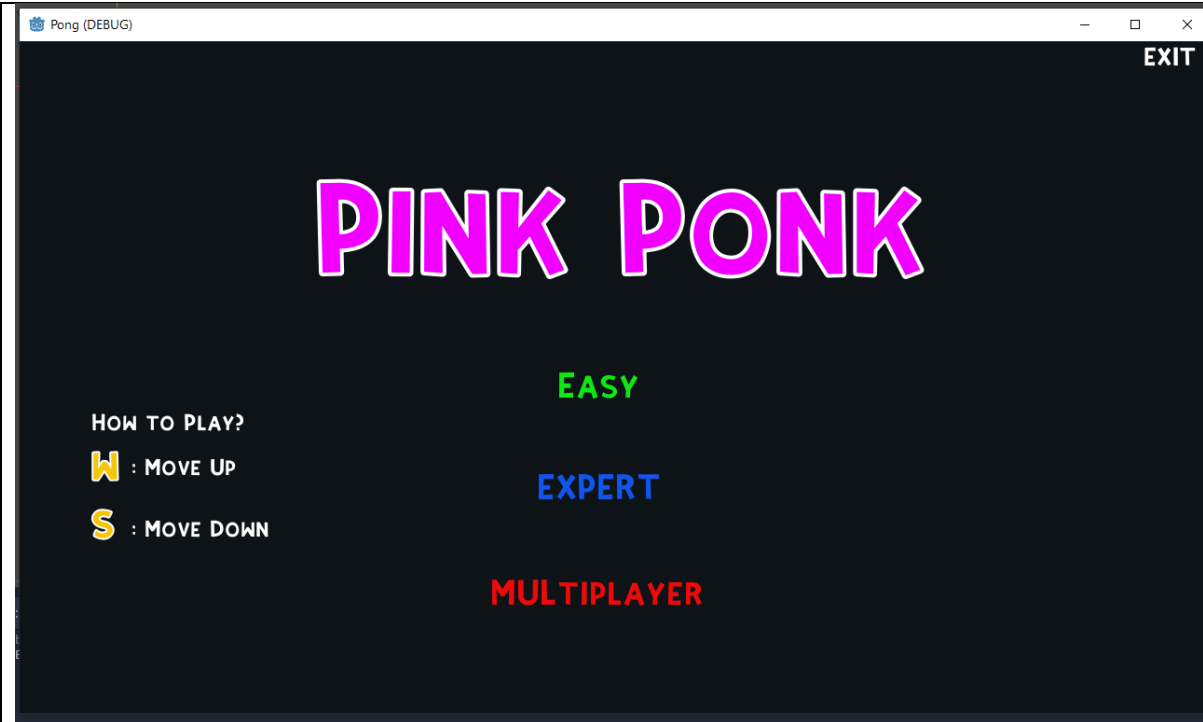
**Tech:** Godot, C,

**Link:**

- <https://github.com/mhnaufal/PinkPonk>
- <https://github.com/mhnaufal/nothing-to-c>




The screenshot shows a GitHub repository page for 'mhnaufal/PinkPonk'. The repository is on the 'main' branch, has 2 branches, and 0 tags. The commit history shows an initial commit by mhnaufal on Dec 19, 2020, with 2 commits. The file list includes folders for .import, assets, export, scenes, and scripts, along with files like README.md, default\_env.tres, export\_presets.cfg, and project.godot. The README.md file is open, showing the title 'PinkPonk' and a description: 'My first nearly complete game by following a few of tutorials'. It lists tutorials followed (Learn Godot by creating Pong) and assets used (GodotPong, Pong-Graphics, Flappy Bird). The repository is described as a '2D ping-pong like game created using Godot!' and has 0 stars, 2 watchers, and 0 forks. The language is GDScript at 100.0%.



mhnaufal update tictactoe to always play 19e738f 1 hour ago 32 commits

.vscode	init toe-tac-tic game	last week
compiler	update compiler README	last year
hangman	init hangman game	2 hours ago
not-redis	update readme	12 hours ago
shell	shell	last year
toe-tac-tic	update tictactoe to always play	1 hour ago
.gitignore	update some functions	last year
Makefile	init hangman game	2 hours ago
README.md	init hangman game	2 hours ago
main.c	main	last year

README.md 

## Me, Trying To Remember What C Feels Like

Not all of the projects have been done, only some of them (or even only one of them), but I will continue to working on the others. Stay tuned 😊

- [not-redis](#)
- [Toe-Tac-Tic](#)
- [Hangman](#)

```
  | 0 |  
---|---|---  
  |   |  
---|---|---  
  |   |
```

ENEMY: Enter row number (1-3): 2  
ENEMY: Enter col number (1-3): 2

```
  | 0 |  
---|---|---  
  | X |  
---|---|---  
  |   |
```

PLAYER: Enter row number (1-3): 1  
PLAYER: Enter column number (1-3): 1

```
 0 | 0 |  
---|---|---  
  | X |  
---|---|---  
  |   |
```

ENEMY: Enter row number (1-3): 1  
ENEMY: Enter col number (1-3): 3

```
 0 | 0 | X  
---|---|---  
  | X |  
---|---|---  
  |   |
```

PLAYER: Enter row number (1-3):

