

BAB II PROSEDUR KERJA

2.1 Deskripsi Penugasan Kerja

Pelaksanaan kegiatan ini dilaksanakan selama kurang lebih 5 bulan yang dimulai dari tanggal 21 Februari 2022 hingga 22 Juli 2022. Kegiatan MBKM yang dilakukan adalah MSIB atau Magang Studi Independen Bersertifikat yang dilakukan dengan mitra Orbit Future Academy dan berfokus pada pembelajaran *Artificial Intelligence* (AI). Mahasiswa yang mengikuti program di Orbit Future Academy ini harus mengikuti kegiatan dari mulai *pre-test* untuk menguji kemampuan dasar mahasiswanya hingga pengerjaan Project Akhir untuk menerapkan hasil dari pembelajaran dengan deskripsi perkerjaan sebagai berikut:

- a. Mengikuti pre-test.
- b. Mengikuti kelas sesi pagi pada pukul 08.00 hingga 11.30 WIB.
- c. Mengikuti kelas sesi siang pada pukul 13.00 hingga 16.30 WIB.
- d. Mengulang materi yang telah disampaikan di kelas sesi pagi dan siang, setelah kelas sesi siang, selama 1 jam (*self-study*).
- e. Mengerjakan latihan individu atau kelompok yang diberikan oleh homeroom atau domain coach saat kelas berlangsung.
- f. Mengerjakan tugas yang diberikan homeroom atau domain coach hinggabatas waktu tertentu.
- g. Mengerjakan *mini project* yang diberikan homeroom atau domain coachhingga batas waktu tertentu
- h. Mengikuti post-test.
- i. Pengerjaan Project Akhir

Pelaksanaan kegiatan pembelajaran tersebut dipantau langsung oleh *homeroom coach* sebagai pembimbing ataupun mentor di setiap kelas dan *domain coach* yang memiliki tugas untuk menyampaikan materi setiap domain utama antara lain *Data Science, Computer Vision, Reinforcement Learning, Natural Language Processing dan Technical*. Kegiatan ini berlangsung setiap

hari Senin sampai dengan Jum'at yang memiliki 2 sesi yaitu sesi pagi dan sesi siang selama 8 jam perharinya, dengan jadwal pembelajaran sebagai berikut:

Tabel 2.1 Agenda Kelas

Pukul (WIB)	Durasi (jam)	Aktivitas
08.00 s.d. 11.30	3.5	Kelas Sesi Pagi
13.00 s.d. 16.30	3.5	Kelas Sesi Siang
16.30 s.d. 17.30	1	<i>Self-Study</i>

2.2 Teori Dasar Pendukung

AI Project Cycle merupakan proses dalam pembuatan proyek AI secara utuh. Terdapat beberapa tahap dalam pencancangannya dan dijabarkan sebagai berikut:

- *Problem Scooping*

Sebelum memulai proyek yang akan dibuat, pada tahap pertama penulis menentukan permasalahan yang berkaitan dengan penerapan teknologi *Artificial Intelligence (AI)* yang telah dipelajari di Orbit Future Academy. Setelah pendiskusian, permasalahan yang didapatkan oleh penulis yaitu mengenai pengidentifikasian jenis – jenis batuan. Penyusunan kerak bumi dan kumpulan mineral – mineral yang mengkristal atau biasa disebut dengan batuan ini memiliki banyak macam yang harus diidentifikasi untuk bisa mengetahui nama, klasifikasi jenis dan juga karakteristik setiap jenis batuan tersebut. Dengan menerapkan teknologi *Artificial Intelligence (AI)* pengguna bisa mengidentifikasi gambar dari jenis batuan yang beragam tersebut dengan cepat dan mudah karena menggunakan aplikasi berbasis *website* yang bisa diakses oleh semua orang. Tetapi, perangkat yang digunakan harus terhubung dengan internet. Penulis berharap dengan adanya aplikasi berbasis *website* ini dapat membantu semua orang yang ingin mempelajari ataupun mengidentifikasi gambar jenis batuan yang dengan cepat dan mudah. Cara menggunakan aplikasi berbasis *website* ini juga sangat mudah, pengguna hanya meng-*input*-kan gambar jenis batuan yang dimiliki selanjutnya sistem akan mengidentifikasi dan

mengkalsifikasikan informasi tersebut kedalam beberapa jenis batuan yang ada didalam *dataset* sehingga menghasilkan output berupa gambar hasil klasifikasi disertai nama dan juga karakteristiknya. Hal tersebut sangat memudahkan pengguna untuk proses pembelajaran.

- *Data Acquisition*

Data yang digunakan untuk sumber informasi untuk menjalankan model pada projek aplikasi berbasis *website* ini berasal dari data primer dan data sekunder. Data tersebut diambil dari sumber data yang sering digunakan yaitu kaggle. Kaggle sendiri ini merupakan situs untuk berbagi ide, mendapatkan inspirasi, bersaing dengan *data scientist* lain, mempelajari informasi baru dan trik *coding*, serta melihat berbagai contoh aplikasi *data science* di dunia nyata. Didalam kaggle tersebut terdapat kumpulan data atau *dataset* yang bisa dicari berdasarkan nama untuk *dataset* tertentu dan dapat digunakan dalam model *data science*. Penulis menggunakan *dataset* dengan format csv untuk mengembangkan model yang digunakan dalam membuat aplikasi tersebut. *Dataset* yang ingin digunakan bisa di-*download* untuk dimasukkan ke dalam model. Penulis menggunakan *Rock Classification Dataset* yang dilihat dalam *data explorer*-nya, *dataset* ini memiliki 168.96 MiB, dengan tiga klasifikasi *directory* yaitu dengan nama *igneous*, *metamorphic* dan *sedimentary*.

- *Data Exploration*

Seperti yang sudah dijelaskan, pengambilan *dataset* untuk tahap *modelling* bersumber dari kaggle, walaupun banyak *dataset* yang bisa digunakan penulis tetap harus menyesuaikan *dataset* tersebut dengan tujuan dan kegunaan pembuatan aplikasi. Untuk model ini, penulis memerlukan penambahan *directory* atau klasifikasi batuan yang digunakan selain *igneous*, *metamorphic* dan *sedimentary* yaitu *labradorite* dan *pyrite*. Penulis menambahkan citra lebih banyak dan menghapus beberapa citra yang memiliki kualitas yang kurang baik, setelah itu penulis juga melakukan *labelling* ulang agar memudahkan

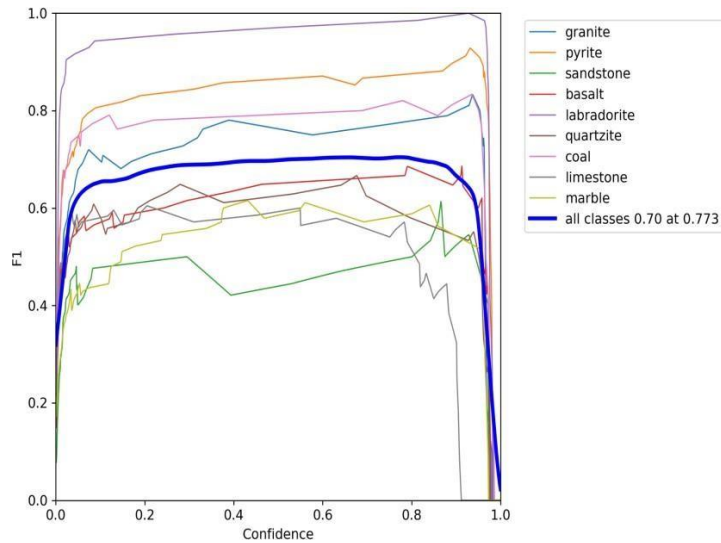
model agar sesuai dengan *output* yang ingin dihasilkan selain itu *labelling* juga mempermudah model untuk mengakses dataset tersebut.

- *Modelling*

Tahap *modelling* dalam membuat projek ini menggunakan algoritma YOLO. Sistem *You Only Look Once (YOLO)* ini mendeteksi dengan menggunakan *repurpose classifier* atau *localizer* untuk melakukan deteksi tersebut. Dalam kasus ini sebuah model diterapkan pada sebuah citra, yang bersumber dari gambar berformat jpg. YOLO merupakan perkembangan dari algoritma deteksi objek *Convolutional Neural Networks*. YOLO hanya menggunakan lapisan konvolusi dan lapisan *pooling*. Untuk lapisan konvolusi terakhir disesuaikan dengan jumlah kelas dan jumlah kotak prediksi yang diinginkan.

- *Evaluation*

Pada bagian evaluasi itu menggunakan *metrics mAP*. *mAP* (rata-rata Presisi Rata-rata) adalah metrik evaluasi yang digunakan dalam model deteksi objek seperti YOLO. Perhitungan *mAP* membutuhkan *IOU*, *Precision*, *Recall*, *Precision Recall Curve*, dan *AP*. *IOU* adalah model deteksi objek memprediksi kotak pembatas dan kategori objek dalam sebuah gambar. *Intersection Over Union (IOU)* digunakan untuk menentukan apakah kotak pembatas diprediksi dengan benar. Presisi adalah kemampuan model untuk mengidentifikasi hanya objek yang relevan. *Recall* adalah kemampuan model untuk menemukan semua kotak pembatas kebenaran dasar. *Kurva Precision Recall* adalah *plot Precision* pada sumbu vertikal dan *Recall* pada sumbu horizontal. Berikut adalah gambar grafik dari *F-1 Score*:



Gambar 3.1 *F-1 Score*

- *Deployment*

Pada bagian *deployment* ini memiliki tujuan yaitu menyampaikan aplikasi yang sudah dibuat, dimana aplikasi ini merupakan aplikasi yang digunakan untuk mengidentifikasi dari jenis batuan yang diharapkan dapat mempermudah pengguna untuk mempelajari jenis batuan dengan cepat dan mudah. Aplikasi berbasis *website* ini hanya perlu memasukkan gambar dengan format jpg dan *submit* gambar tersebut. Maka kemudian prediksi dan akan muncul pada tampilan web. Kemudian pada bagian proses *deployment* ini dilakukan dengan menggunakan 2 program yang digunakan untuk menghubungkan antarmodel dan tampilan web yang akan digunakan untuk menampilkan hasil dari prediksi. Dimana pada *deployment* ini menggunakan program yaitu *app.py* dan *html* untuk menyambungkan dengan model yang sudah dibuat. *App.py* yang dibuat menggunakan *Flask*. *Flask* ini merupakan web *framework* dari bahasa *python* yang menyediakan, *libraries* dan kumpulan kode yang bisa digunakan untuk membangun *website*, tanpa perlu melakukan semuanya dari nol. Untuk menggunakan *Flask* terlebih dahulu

harus meng-*import* *Flask* dan membuat objek dari kelas *Flask*. Setelah itu penulis membuat *routing*-nya dengan *route('/')*. *Route('/')* adalah dekorator yang digunakan untuk mendefinisikan URL apa yang memanggil fungsi dibawahnya. Lalu berikut ini adalah tampilan dari *app.py* dan *html*:

```
app4.py 3.0 • app2.py 3.0 batu-classification-4.html U batu-classification.html U
app4.py > ...
1 import argparse
2 import io
3 import os
4 from PIL import Image
5 from detect import run
6 from flask import Flask, render_template, request
7
8
9 app = Flask(__name__)
10
11 @app.route("/", methods=["GET", "POST"])
12 def predict():
13     print(request.method)
14     if request.method == 'GET':
15         print("ini method GET")
16         return render_template("batu-classification-4.html")
17
18     elif request.method == 'POST':
19         print("ini method POST")
20         x=request.files["image"]
21         print(x.save("upload/hasil.jpg"))
22
23         print("test")
24
25         run(weights='best.pt', source='upload/hasil.jpg', exist_ok=True)
26
27     return render_template("batu-classification-4.html")
28
29 if __name__ == '__main__':
30     app.run(port=1111, debug=True)
31
```

Gambar 3.2 Tampilan *App.py*

```
app4.py 3.0 • app2.py 3.0 batu-classification-4.html U X batu-classification.html U
templates > batu-classification-4.html > html > head > meta
1
2 <!DOCTYPE html>
3 <html lang="en">
4 <head>
5     <meta charset="UTF-8" />
6     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
7     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
8     <link
9         href="https://unpkg.com/tailwindcss@2/dist/tailwind.min.css"
10        rel="stylesheet"
11    />
12 <title>Identifikasi Batuan</title>
13 </head>
14 <body>
15     <section
16         class="
17             flex
18             justify-center
19             items-center
20             w-screen
21             h-screen
22             bg-indigo-500
23         "
24     >
25         <main
26             class="
27                 flex flex-col
28                 justify-center
29                 items-center
30                 p-16
31                 bg-white
32                 rounded-lg

```

Gambar 8.3 Tampilan html