

BAB II

PROSEDUR KERJA

2.1 Deskripsi Penugasan Kerja

Pada program MSIB (Magang Studi Independen Bersertifikat) AI Program Mastery di Orbitu Future Academy dengan skema pembelajarann yang mandiri. Difasilitasi *homeroom coach* dan *domain coach*, yang dimana *homeroom coach* bertugas menjadi mentor di kelas atau memantau secara berkala sedangkan *domain coach* yang menyampaikan beberapa pertemuan mengenai pembelajaran. Program ini terlaksana kurang lebih selama 5 bulan, mulai dari bulan Februari sampai dengan bulan Juli 2022. Program ini berlangsung setiap hari kerja (Senin sampai dengan Jumat) selama 8 jam per harinya, dengan rincian 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>

Pada Program AI Mastery mahasiswa atau peserta berkewajiban untuk melakukan pembelajaran mandiri serta mengerjakan tugas sesuai dengan *planning* yang telah disiapkan. Domain AI yang dipelajari diantaranya *Data Science*, *Computer Vision*, *Reinforcement Learning*, *Natural Language Processing* dan *Technical*. Setiap materi yang dipelajari terdapat quiz yang harus dikerjakan untuk memenuhi nilai yang dibutuhkan. Adapun beberapa deskripsi kewajiban peserta atau mahasiswa lainnya selama program ini berlangsung diantaranya :

- 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* hingga batas waktu tertentu.
- g. Mengerjakan *mini project* yang diberikan *homeroom* atau domain *coach* hingga batas waktu tertentu
- h. Mengikuti *post-test*.
- i. Mengerjakan Project Akhir secara berkelompok

Selama pengerjaan Project Akhir student diminta mengerjakan secara berkelompok. Pada pengerjaan Project Akhir student memiliki peran *modelling*, dengan deskripsi pekerjaan sebagai berikut:

- a. Setelah data sudah siap untuk dilakukan training, data tersebut dipakai untuk kemudian displit menjadi 3 bagian (*train data*, *test data* dan *validation data*).
- b. Membuat arsitektur CNN menggunakan arsitektur VGG 19 yang ditambahkan dengan *layer dropout*, *instance normalization*, *flatten* dan *Dense(4,activation='softmax')* yang terlihat pada gambar 2.1.

Layer (type)	Output Shape	Param #
input_3 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147808
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv4 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1190160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2380320
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2380320
block4_conv4 (Conv2D)	(None, 28, 28, 512)	2380320
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2380320
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2380320
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2380320
block5_conv4 (Conv2D)	(None, 14, 14, 512)	2380320
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
dropout_4 (Dropout)	(None, 7, 7, 512)	0
instance_normalization_3 (InstanceNormalization)	(None, 7, 7, 512)	1024
flatten_2 (Flatten)	(None, 25088)	0
dense_2 (Dense)	(None, 4)	100356

```

Total params: 20,125,764
Trainable params: 101,380
Non-trainable params: 20,024,384

```

Gambar 2.1 Arsitektur VGG 19 (yang telah disesuaikan).

- c. Model Arsitektur CNN tersebut kemudian digunakan untuk mentraining data dan ditentukan *tuning hyperparameters* nya seperti yang bisa dilihat pada gambar 2.2

```
[ ] optimizer = tf.keras.optimizers.Adam(learning_rate=1e-3)
model.compile(
    loss='categorical_crossentropy',
    optimizer=optimizer,
    metrics=['accuracy']
)

[ ] r = model.fit(
    training_set,
    validation_data=val_set,
    epochs=50,
    batch_size=32
)
```

Gambar 2.2 *Tuning Hyperparameters.*

- d. Kemudian dilakukan evaluasi model dan didapatkan hasil *accuracy* sebesar 99,37% seperti yang terlihat pada gambar 2.3.

```
[ ] loss, accuracy = model.evaluate(test_set)
40/40 [=====] - 8s 188ms/step - loss: 0.0422 - accuracy: 0.9937
```

Gambar 2.3 Evaluasi Model.

2.2 Teori Dasar Pendukung

2.2.1 AI Project Cycle

a. *Problem Scoping*

Diagnosis awal *Alzheimer disease* sangatlah penting. Mendeteksi penyakit *Alzheimer* sesegera mungkin dapat memberikan pengobatan yang lebih baik. Oleh karena itu, penulis memutuskan untuk mengangkat permasalahan ini. Dengan dibuatnya program ini, diharapkan dapat membantu dokter dalam *mendiagnosis* awal penyakit *Alzheimer*.

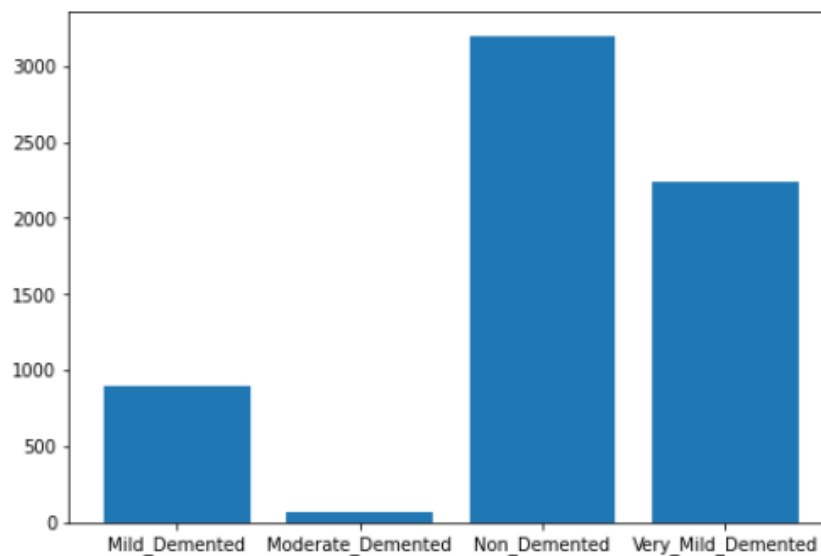
b. *Data Acquisition*

Sumber data diperoleh dari *Kaggle* yang merupakan salah satu *website* yang menyediakan banyak *public* dataset seperti dataset

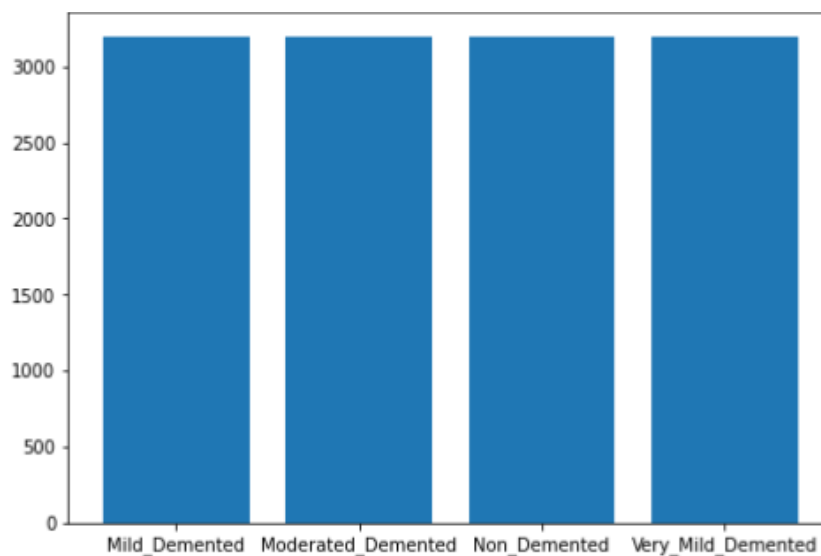
kesehatan, dataset *cancer*, gagal jantung, *corona*, data NLP, data *text*, dataset *music*, industri transportasi dll. Data tersebut adalah hasil scan MRI otak manusia yang berupa gambar.

c. Data Exploration

Sumber data yang didapat memiliki *class* yang sangat *imbalance* maka dari itu dilakukan data *augmentation* untuk membuat data menjadi *balance*. Ditunjukkan pada gambar 2.4, gambar 2.5.



Gambar 2.4 Data *Imbalance*



Gambar 2.5 Data *Balance (After Augmentation)*

d. Modelling

Algoritma yang digunakan adalah *Deep Learning* CNN yang menggunakan arsitektur VGG 19 yang ditambahkan dengan *layer dropout, instance normalization, flatten* dan *Dense (4,activation='softmax')*. Arsitektur VGG 19 merupakan arsitektur CNN yang bagus digunakan untuk *Image Classification* sesuai dengan proyek ini yaitu *Multiclass Classification*.

e. Evaluation

Metrics yang digunakan adalah *accuracy*. grafik *loss* dan *accuracy* dari hasil training dapat dilihat pada gambar 3.4, gambar 3.5. Pada grafik *loss* dan *accuracy* terlihat *train & val loss* dan *train & val accuracy* tidak terlalu mengalami *underfitting/overfitting* walaupun terlihat sedikit gap antara *train & val accuracy*.

f. Deployment

Proses *deployment* proyek ini menggunakan *Flask* yang hanya bisa *run* di *local computer*. Proses *deployment* bertujuan agar model bisa digunakan.

2.2.2 Profil Tim dan Deskripsi Pembagian Tugas

Tabel 2.2 Profil Tim dan Tupoksi

Nama	NIM	Kelas	Tupoksi
Dhany Umar	1910314032	Snowdrops	<i>Modelling</i>
Olivia Sherly Indah Ulyly	19101002	Snowdrops	<i>Deployment</i>
Zein Alwildan	1051930	Snowdrops	<i>Data Preparation & Data Exploration</i>
Kevin Geovany	1806147956	Snowdrops	<i>Data Preparation & Data Exploration</i>

2.2.3 Deskripsi Aplikasi

a. Nama dan Fungsi Aplikasi

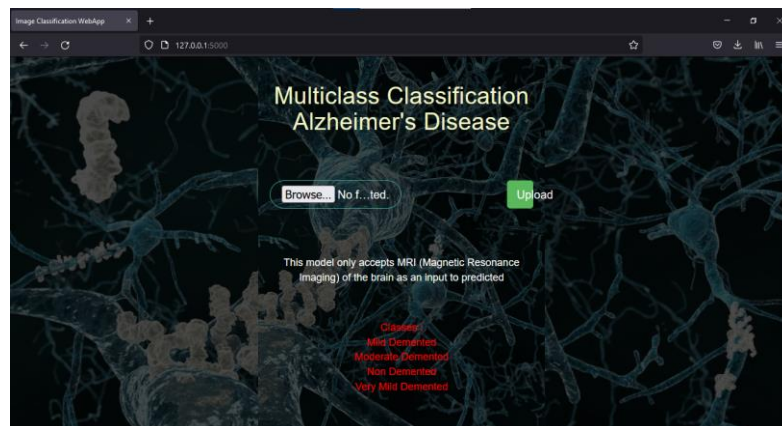
Multiclass Classification Alzheimer's Disease(MCAD) ini adalah aplikasi yang bisa memprediksi *Alzheimer's Disease*. Target user dari aplikasi ini sendiri adalah dokter, aplikasi ini bertujuan untuk memudahkan dokter melakukan *early diagnosis* pada penyakit *Alzheimer*, dengan cara dokter melakukan scan MRI pada otak pasien kemudian hasil dari scan MRI tersebut yang berupa gambar akan di prediksi oleh aplikasi ini, yang mana nanti akan mengeluarkan hasil prediksi beserta probabilitasnya.

b. Jenis Aplikasi dan *Specific Requirement*

Aplikasi ini berbasis *web*, jadi selama laptop ataupun mobile phone user bisa mengakses *website* dan terhubung ke internet maka dapat mengakses aplikasi ini dan tidak ada *requirement* khusus dari aplikasi ini.

c. *User Interface*

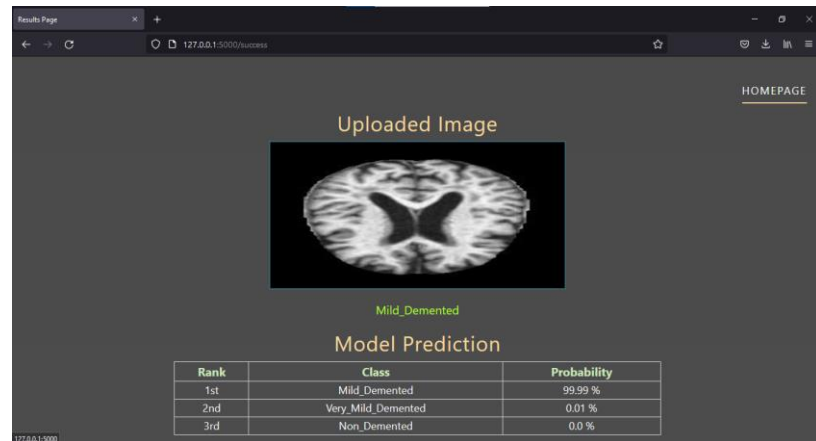
Berikut ini adalah *User Interface* dari aplikasi ini :



Gambar 2.6 Tampilan Awal *User Interface* Aplikasi

Seperti yang terlihat pada gambar 2.6 user interface dari aplikasi ini terlihat *standard* dan sangat sederhana, yang mana ada dua tombol pada *user interface*. Tombol *Browse* yang berfungsi untuk mencari gambar yang ingin diprediksi, kemudian ada tombol *upload* yang

berfungsi untuk mengupload gambar yang sudah dipilih pada tombol *browse*, kemudian gambar tersebut akan diproses oleh model untuk dilakukan prediksi dan masuk ke tampilan *user interface* hasil yang terlihat pada gambar 2.7.



Gambar 2.7 Tampilan Hasil *User Interface* Aplikasi

Pada tampilan hasil aplikasi terlihat gambar yang diupload, hasil prediksi serta probabilitasnya. Seperti terlihat pada gambar 8.4, pada tampilan akhir ada tombol *Homepage* yang berfungsi untuk kembali ke tampilan awal aplikasi.

d. Keterangan Lainnya

Adapun kelebihan dari aplikasi ini yaitu:

1. Mudah digunakan.
2. Dapat membantu dokter dalam melakukan *early diagnosis* pada *Alzheimer's Disease*.

Aplikasi ini juga memiliki kekurangan yaitu:

1. Aplikasi ini hanya dideploy di *local computer*.
2. UI dari aplikasi ini masih sangat standar dan memiliki fitur yang terbatas.

Dilihat dari kekurangan yang ada, aplikasi ini masih perlu pengembangan lebih lanjut. Pengembangan yang bisa dilakukan

yaitu melakukan proses *deployment* yang lebih advance baik web deployment maupun mobile deployment serta membuat UI yang yang lebih menarik dan menambahkan beberapa fitur yang bermanfaat.