

## BAB IV

### HASIL DAN PEMBAHASAN

Pada bab ini, penulis akan menjelaskan secara singkat mengenai tentang Kerja Praktek yang sudah dilaksanakan dan menjelaskan secara rinci hasil dari Capstone Projek yang menjadi tugas akhir program Bangkit 2021.

#### 4.1 Penjelasan Kerja Praktik

Kerja Praktek yang penulis lakukan merupakan bagian dari Capstone Project yang menjadi salah satu syarat kelulusan pada program Bangkit 2021. Pada Capstone tersebut penulis bekerja sama dengan mahasiswa yang berasal dari Perguruan Tinggi yang lain yang dibentuk menjadi satu tim untuk menyelesaikan Capstone Project tersebut. Adapun proyek yang penulis lakukan adalah merancang aplikasi berbasis Android untuk berguna untuk memonitoring pom bensin (SPBU).

Pada pembuatan aplikasi berbasis Android untuk monitoring SPBU terbagi menjadi tiga bagian, yaitu:

- a. Pada Pemrograman Android (Mobile Development) dilakukan dengan menerapkan Android Jet Pack dan integrasikan proyek Android Studio dengan Firestore. Aplikasi memiliki koneksi real-time menggunakan Firestore untuk menulis dan membaca data (dan pengguna dan pelanggan). Data pelanggan ditampilkan dalam bentuk daftar dan grafik (jumlah pelanggan dan konsumsi bensin).
- b. Pada jalur Cloud Computing dimanfaatkan dengan menerapkan pembelajaran mesin dan program Python menjadi bahasa komunikasi database menggunakan Compute Engine. Untuk database disini penulis menggunakan Firestore dalam mode asli yang dapat merekam data secara waktu nyata (real-time).
- c. Machine Learning, pada jalur pembelajaran ini dimanfaatkan untuk membuat model dari pelat nomor dan meteran digital menggunakan model jaringan syaraf tiruan dan input gambar akan diproses oleh pengenalan karakter optic (OCR) untuk string.

#### 4.2 Pengumpulan data

Pengumpulan data diperlukan untuk mempersiapkan data yang diperlukan proses Machine Learning dan testing pada pembuatan aplikasi monitoring.

Data yang diperlukan untuk pembuatan system monitoring SPBU ini memerlukan dua jenis dataset, yaitu:

- a. Plate number dataset (kumpulan data nomor pelat), yaitu data yang dipakai pada proyek ini diambil dari dataset nomor pelat Indonesia yang berasal dari Indramayu dan Cirebon.

Link: <https://github.com/SachaIZADI/Seven-Segment-OCR/tree/master/Datasets>

- b. Digital meter dataset (kumpulan data meter digital), pada dataset ini yang dipakai pada system ini diambil dari dokumentasi digital meter dalam bensin dispenser, yang menunjukkan jumlah liter dikonsumsi oleh setiap pelanggan.

Link: [https://www.kaggle.com/imamdigmi/indonesian-plate-number?select=plate number recognition datasets v1](https://www.kaggle.com/imamdigmi/indonesian-plate-number?select=plate%20number%20recognition%20datasets%20v1)

#### 4.2.1 Neural Network (Model Jaringan syaraf tiruan)

Metode jaringan saraf yang digunakan oleh penulis karena input nya dalam bentuk gambar, dan kemudian gambar tersebut akan diproses oleh optic pengenalan karakter sehingga karakter yang terkandung dalam gambar dapat diambil dalam bentuk huruf dan angka.

```
# Neural Network Model
def OCRModel():
    image=keras.layers.Input((32, 784, 1))
    conv1=keras.layers.Conv2D(16, (3, 3), activation='relu', padding='same')(image)
    mp1=keras.layers.MaxPooling2D((2, 2), padding='same')(conv1)
    conv2=keras.layers.Conv2D(32, (3, 3), activation='relu', padding='same')(mp1)
    mp2=keras.layers.MaxPooling2D((2, 2), padding='same')(conv2)
    conv3=keras.layers.Conv2D(64, (3, 3), activation='relu', padding='same')(mp2)
    mp3=keras.layers.MaxPooling2D((2, 2), padding='same')(conv3)
    conv4=keras.layers.Conv2D(128, (3, 3), activation='relu', padding='same')(mp3)
    mp4=keras.layers.MaxPooling2D((2, 1), padding='same')(conv4)
    conv5=keras.layers.Conv2D(256, (3, 3), activation='relu', padding='same')(mp4)
    mp5=keras.layers.MaxPooling2D((2, 1), padding='same')(conv5)
    conv6=keras.layers.Conv2D(256, (3, 3), activation='relu', padding='same')(mp5)
    mp6=keras.layers.MaxPooling2D((1, 3), padding='same')(conv6)
    bn=keras.layers.BatchNormalization()(conv6)
    sq=keras.backend.squeeze(bn, axis=1)

    rn1=keras.layers.Bidirectional(keras.layers.LSTM(256, return_sequences=True))(sq)
    rn2=keras.layers.Bidirectional(keras.layers.LSTM(256, return_sequences=True))(rn1)

    exd=keras.backend.expand_dims(rn2, axis=2)
    mapping=keras.layers.Conv2D(len(symbols), (2, 2), activation='relu', padding='same')(exd)
    mapping=keras.backend.squeeze(mapping, axis=2)
    mapping = tf.keras.layers.Softmax()(mapping)

    # bn = keras.layers.BatchNormalization()(conv3)
    model=keras.Model(image,mapping)
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

    return model
```

Gambar 4.1 Hasil tingkat akurasi

##### a. Tingkat akurasi Digital Meter

```
300/300 [=====] - 17s 55ms/sample - loss: 0.0213 - accuracy: 0.9877
Epoch 90/100
300/300 [=====] - 17s 56ms/sample - loss: 0.0202 - accuracy: 0.9890
Epoch 91/100
300/300 [=====] - 17s 55ms/sample - loss: 0.0191 - accuracy: 0.9872
Epoch 92/100
300/300 [=====] - 17s 57ms/sample - loss: 0.0171 - accuracy: 0.9897
Epoch 93/100
300/300 [=====] - 17s 56ms/sample - loss: 0.0159 - accuracy: 0.9903
Epoch 94/100
300/300 [=====] - 17s 56ms/sample - loss: 0.0137 - accuracy: 0.9917
Epoch 95/100
300/300 [=====] - 17s 56ms/sample - loss: 0.0115 - accuracy: 0.9932
Epoch 96/100
300/300 [=====] - 17s 56ms/sample - loss: 0.0107 - accuracy: 0.9930
Epoch 97/100
300/300 [=====] - 16s 55ms/sample - loss: 0.0112 - accuracy: 0.9902
Epoch 98/100
300/300 [=====] - 17s 56ms/sample - loss: 0.0100 - accuracy: 0.9926
Epoch 99/100
300/300 [=====] - 17s 56ms/sample - loss: 0.0095 - accuracy: 0.9907
Epoch 100/100
300/300 [=====] - 17s 56ms/sample - loss: 0.0082 - accuracy: 0.9937
[7]: <tensorflow.python.keras.callbacks.History at 0x7fec846600b8>
```

Gambar 4.2 Tingkat akurasi digital meter

- Epoch            100
- Loss             : 0.0082
- Accuracy        : 0.9937

b. Tingkat akurasi data pelat nomor

```

Epoch 90/100
167/167 [=====] - 9s 55ms/sample - loss: 0.0608 - accuracy: 0.9825
Epoch 91/100
167/167 [=====] - 9s 56ms/sample - loss: 0.0596 - accuracy: 0.9831
Epoch 92/100
167/167 [=====] - 9s 56ms/sample - loss: 0.0557 - accuracy: 0.9854
Epoch 93/100
167/167 [=====] - 9s 56ms/sample - loss: 0.0564 - accuracy: 0.9844
Epoch 94/100
167/167 [=====] - 9s 55ms/sample - loss: 0.0544 - accuracy: 0.9861
Epoch 95/100
167/167 [=====] - 9s 54ms/sample - loss: 0.0524 - accuracy: 0.9850
Epoch 96/100
167/167 [=====] - 10s 57ms/sample - loss: 0.0519 - accuracy: 0.9852
Epoch 97/100
167/167 [=====] - 9s 55ms/sample - loss: 0.0509 - accuracy: 0.9861
Epoch 98/100
167/167 [=====] - 9s 53ms/sample - loss: 0.0501 - accuracy: 0.9864
Epoch 99/100
167/167 [=====] - 10s 57ms/sample - loss: 0.0495 - accuracy: 0.9868
Epoch 100/100
167/167 [=====] - 9s 54ms/sample - loss: 0.0494 - accuracy: 0.9864

```

Gambar 4.3 Tingkat akurasi data pelat motor

- Epoch            100
- Loss             : 0.0494
- Accuracy        : 0.9864

### 4.3 Tahap Cloud Computing

a. Menggunakan Compute Engine untuk menyebarkan Machine Learning dan menyimpan data.

- Pembelajaran mesin (Machine Learning) dan database dibentuk menggunakan python yang dapat berjalan secara virtual pada linux.
- Karena masih prototipe, program Machine Learning masih dilakukan secara manual.

b. Menggunakan Firestore sebagai Database

- Data akun login dan output machine learning disimpan di Firestore.
- Otentikasi Firestore adalah menggunakan Firebase Admin SDK.json file.

Pembelajaran mesin (Machine Learning) dan database komunikasi program menggunakan Bahasa pemrograman Python.

```

c0080788@gas-station: ~ - Google Chrome
ssh.cloud.google.com/projects/gas-station-monitoring-v1/zones/us-central1-a/instances/gas-station?authuser=1&hl=en_US&projectN...
:5D:CB:BE:3A:29:22:99:48:57:E0:35:9D:55:12:54:15:0F:2C
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1044-gcp x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Sun Jun  6 16:22:25 UTC 2021

System load:  0.34          Processes:      114
Usage of /:   36.1% of 19.21GB    Users logged in:  0
Memory usage: 5%           IP address for ens4: 10.128.0.5
Swap usage:   0%

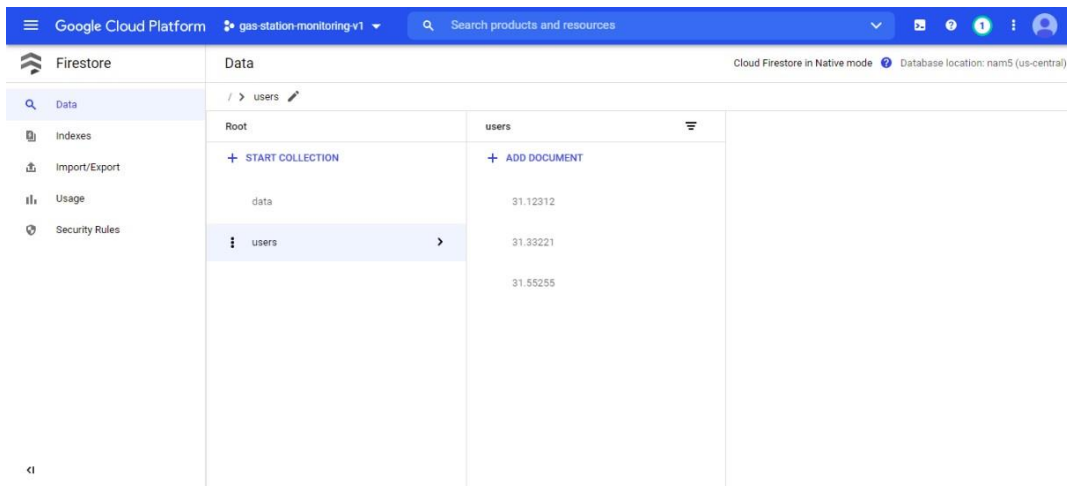
 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.
   https://ubuntu.com/blog/microk8s-memory-optimisation

Last login: Fri Jun  4 02:41:00 2021 from 35.235.241.16
c0080788@gas-station:~$

```

Gambar 4.4 Compute Engine Visualization ( Menghitung Visualisasi Mesin)

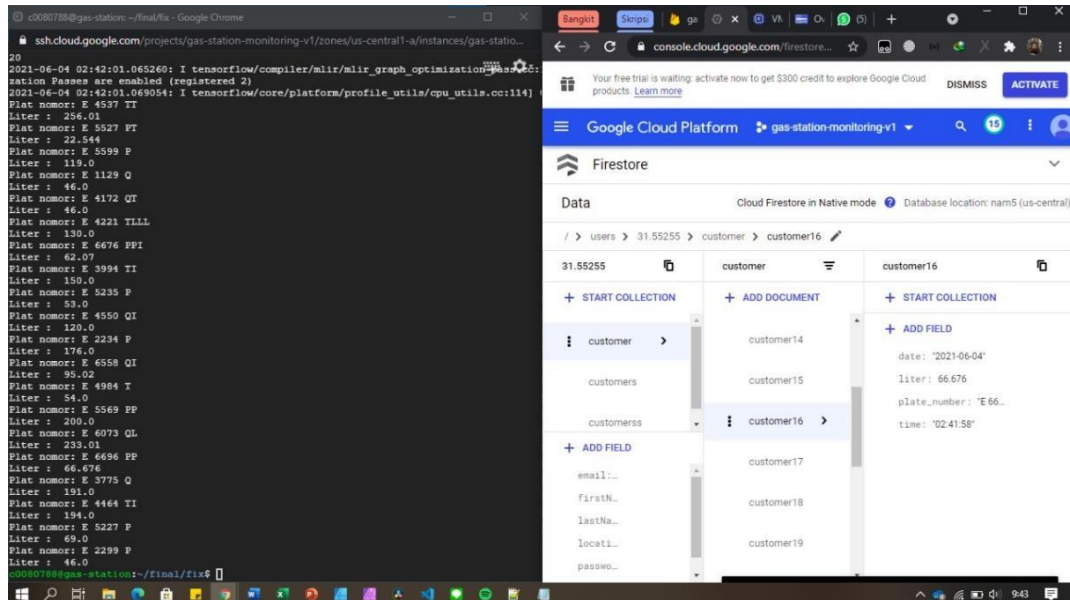
- Ubuntu 18.04
- Python 3.95



Gambar 4.5 Firestore Visualization

- Firestore Native mode
- Architecture
  - Users
  - Spbucode
  - Customer
  - Customer\_number
    - Date

- Liter
- Plate\_number
- Time



Gambar 4.6 Result Firestore

#### 4.4 Mobile Development Andorid (Pemrograman Android)

- a. Menggunakan Android Studio untuk membangun aplikasi
  - Menggunakan perangkat lunak asli untuk mengembangkan aplikasi Android.
  - Menggunakan Bahasa pemrograman Kotlin beberapa Android Jet Pack (Lifecycle, ViewModel, Data Binding, LiveData).
- b. Mengintegrasikan proyek Android Studio dengan Firestore
  - Daftarkan proyek Android dengan menggunakan Firestore Console
  - Menginstal dependensi dari Firebase dan Firestore pada proyek Android Studio
  - Untuk saat ini, aturan untuk menulis dan membaca data di Firestore tersedia untuk semua pengguna.
- c. Menulis/Membaca data ke/dari Firestore
  - Untuk menulis dan membaca data di Firestore, dibutuhkan alamat koneksi dan dokumen  
Format alamat : collection/document/collection/document/...
- d. Menulis data ke Firestore
  - Menulis data digunakan untuk mendaftarkan akun pengguna baru ke Firestore.  
Aalamat : users/<spbuicode>

e. Membaca data dari Firestore

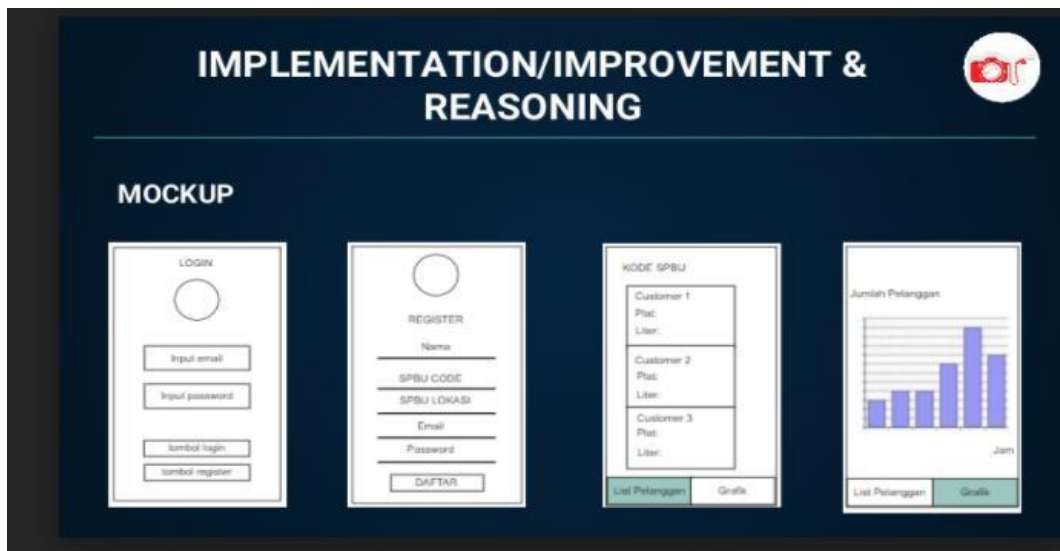
- Untuk memanggil semua data akun pengguna melakukan proses pemeriksaan selama pendaftaran (sehingga tidak terdapat data ganda) dan login (untuk mencocokkan data input dengan database).

Alamat : users/<alldocument>

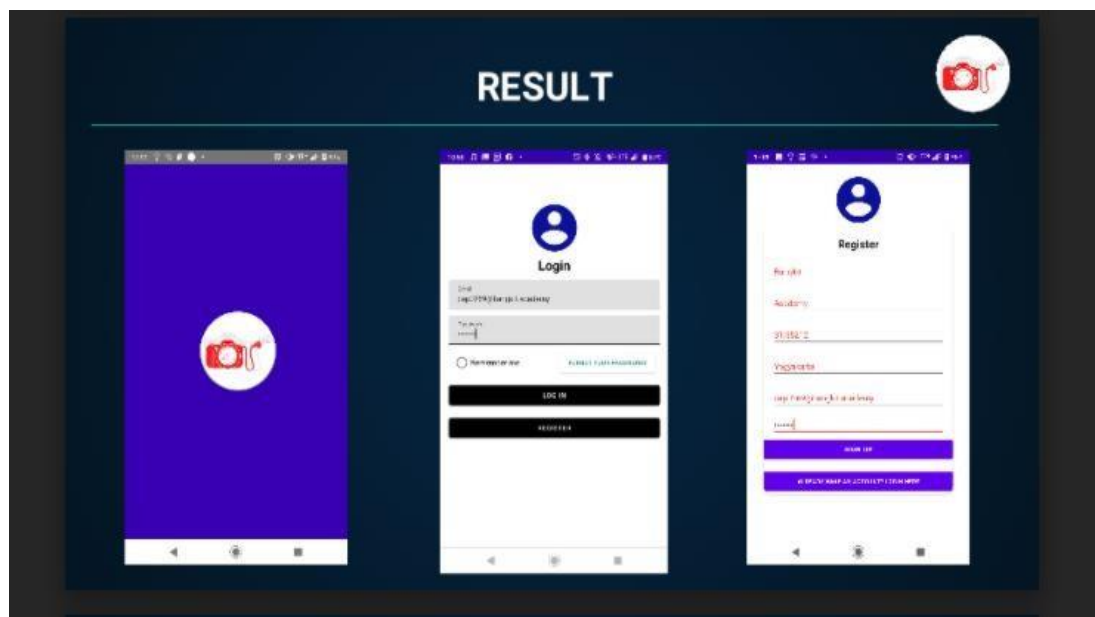
- Untuk memuat semua data pelanggan sesuai dengan spbucode alamat untuk menampilkan daftar data dan grafik.

Alamat : users/<spbucode>/customer/<alldocument>

#### 4.5 MOCKUP



Gambar 4.7.a Result Mockup



Gambar 4.7.b Result Mockup



Gambar 4.7.c Result Mockup

#### 4.6 Langkah install aplikasi

Alur untuk menjalankan semua program pada proyek SPBU Montirong.

- a. Aplikasi Android (Langkah 1 - 4, Langkah 12 - 15)
  - b. Pembelajaran Mesin (Langkah 5 - 11)
- Instal aplikasi android
  - Daftarkan akun SPBU untuk mendapatkan kunci akun yang akan digunakan dalam uji coba penggunaan program pembelajaran mesin.
  - Simpan kunci akun. Kunci akun berupa kode SPBU yang didaftarkan pada saat pembuatan akun diterapkan.
  - Simpan alamat email dan kata sandi Anda untuk masuk ke aplikasi nanti, untuk memantau dari Firestore.
  - Simpan gambar yang ingin Anda uji di tempat yang tetap dan tidak berubah. Gambar dapat menggunakan contoh gambar yang disediakan.
  - Simpan model 'plat nomor' dan model 'liter bensin' di tempat yang tetap dan tidak berubah.
  - Dalam menjalankan program machine learning, siapkan dulu alamat tempat menyimpan file foto yang ingin dicoba.
  - Ubah alamat foto pada program 'load image', dengan alamat yang telah disiapkan.
  - Ganti alamat model pada program 'model beban', dengan alamat model 'plat nomor' pada model1 dan alamat model 'liter bensin' pada model2.
  - Ubah nama string data 'documents' pada variabel 'users\_ref', dengan key akun yang sudah Anda miliki sesuai langkah sebelumnya.
  - Jalankan program .py, maka setiap hasil pembacaan machine learning akan disimpan di Firestore.

- Buka aplikasi dan login menggunakan email dan password yang digunakan saat pendaftaran.
- Aplikasi akan secara otomatis mengambil data dari Firestore sesuai dengan alamat 'dokumen' yang telah dibuat.
- Aplikasi dapat memonitor hasil pembacaan plat nomor dan bensin.
- Data ditampilkan pada dashboard aplikasi dalam bentuk item daftar dan grafik.

#### 4.7 Timeline

TAHAP KEGIATAN	BULAN KE:																							
	1				2				3				4				5				6			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Machine Learning																								
Pengumpulan dataset																								
Anotasi Dataset																								
Perbaikan Model dan Program																								
Android																								
Peningkatan Fitur Grafik																								
Peningkatan Fitur Umum																								
Peningkatan UI Aplikasi																								
Penambahan Security Pengiriman Data																								
Cloud Computin																								
Managing Deployment Apk in Cloud																								
Penambahan Security Cloud																								
Monitoring and Logging Cloud																								

Gambar 4.8 Timeline



## 4.8 Budgeting

### Budgeting and Needed Resource for One Gasoline Dispenser in SPBU

No.	Resource	Amount	Price @Unit	Budget
1	Camera	2	Rp. 1.500.000,00	Rp. 3.000.000,00
2	Raspberry-Pi 4	1	Rp. 1.200.000,00	Rp. 1.200.000,00
3	Cloud Storage (Minimum)	6 month	Rp. 500.000,00	Rp. 3.000.000,00
4	Human Resource	6	Rp. 5.000.000,00	Rp. 30.000.000,00
5	Internet (Wi-Fi)	6 month	Rp. 400.000,00	Rp. 2.400.000,00
6	Hardware installation	1	Rp.1.400.000,00	Rp. 1.400.000,00
<b>Total</b>				Rp. 41.000.000,00

### Needed Resource without Budget

No.	Resource
1	Channel in SPBU

Gambar 4.9 Budgeting