

Perancangan Aplikasi Pendataan Infrastruktur Dampak Bencana (Posdistre) untuk Kementerian Pekerjaan Umum dan Perumahan Rakyat

Darwin Gunawan^{#1}, Setia Budi^{*2}

[#]Program Studi SI Sistem Informasi, Universitas Kristen Maranatha
Jl. Surya Sumantri No. 65 Bandung, 40174

¹1873008@maranatha.ac.id

²setia.budi@it.maranatha.edu

Abstract — Posdistre stands for Post Disaster Structure Report. Posdistre is an android-based application with the implementation of Machine Learning and Google Cloud Platform. Posdistre was created to assist the Ministry of Public Works and Public Housing in collecting data on infrastructure affected by disaster. The main feature of this application is "Lapor" which is used to report the damaged infrastructure to the Ministry Leaders so a decision can be made quickly and resolve the problem for the post disaster area. The main feature is used with the implementation of Convolutional Neural Network which is used to predict images and help with the image classification so that it can be more effective and time efficient for the leaders to make quick decisions.

Keywords— Application, Disaster, Posdistre, Report.

I. PENDAHULUAN

Google Bangkit 2021 adalah program yang diinisiasi oleh Google bersama tiga unicorn di Indonesia (Tokopedia, Gojek, dan Traveloka) yang bertujuan untuk memperlengkapi mahasiswa dengan berbagai keterampilan yang dibutuhkan untuk dapat terjun dan berkiprah di industri. Pada penghujung program, terdapat suatu studi kasus atau *Capstone Project* dengan beberapa tema masalah yang dapat dipilih termasuk perkembangan infrastruktur.

Kementerian Pekerjaan Umum dan Perumahan Rakyat (PUPR) merupakan salah satu institusi yang sangat berjasa dalam perkembangan infrastruktur di Indonesia. Terdapat permasalahan yang dihadapi oleh Kementerian PUPR pendataan infrastruktur dampak bencana, sehingga dibuatkan aplikasi Posdistre yang berguna untuk membantu Kementerian PUPR dalam mendapatkan data agar pimpinan dapat menerima data lebih cepat dan tepat untuk pendukung pengambilan keputusan. Aplikasi berbasis Android ini dibangun dengan dukungan *Cloud Platform* dan *Machine Learning* serta mengimplementasikan TensorFlow Lite untuk *Image Classification*.

II. KAJIAN TEORI

A. Machine Learning

Machine Learning adalah ilmu dan seni pemrograman komputer untuk dapat belajar dari data [1]. *Machine Learning* adalah cabang *Artificial Intelligence* yang didasarkan pada gagasan bahwa sistem dapat belajar dari data, mengidentifikasi pola, dan membuat keputusan dengan intervensi manusia yang minimal. *Machine Learning* juga merupakan sebuah ilmu untuk membuat komputer dapat bertindak tanpa diprogram secara eksplisit [2]. Tujuan *Machine Learning* adalah memprogram computer untuk menggunakan contoh data atau pengalaman masa lalu untuk memecahkan masalah yang diberikan [3]. Dalam beberapa dekade ini, *Machine Learning* telah memberikan kita beberapa teknologi canggih seperti mobil *self-driving*, pengenalan ucapan, pencarian web efektif dan lainnya. *Machine Learning* sudah sangat meresap di hari ini sehingga rakyat pun akan menggunakan *Machine Learning* banyak kali tanpa menyadarinya. Dalam penggunaan *Machine Learning* pastinya akan terlibat pembuatan model, di mana model di *train* dengan *training data* kemudian ketika data lain dimasukkan maka *Machine Learning* akan melakukan prediksi. Secara umum *Machine Learning* dapat dibagi menjadi empat pembelajaran algoritma yaitu *Supervised Learning*, *Unsupervised Learning*, *Semi-Supervised*, dan *Reinforcement*. Perbedaan dari *Supervised* dan *Unsupervised Learning* terletak pada penggunaan *labeled datasets*. Secara

sederhananya, *Supervised Learning* menggunakan data input dan output berlabel, sedangkan *Unsupervised Learning* tidak menggunakan data *input* dan *output* berlabel [4].

B. Artificial Intelligence

Artificial Intelligence biasanya mengacu pada simulasi kecerdasan manusia dalam mesin yang diprogram untuk berpikir seperti manusia dan meniru tindakan manusia. Istilah ini juga dapat diterapkan pada mesin apa pun yang menunjukkan ciri-ciri yang terkait dengan pikiran manusia seperti pembelajaran dan pemecahan masalah [5]. *Artificial Intelligence* dibangun di atas elemen ilmu computer, matematika, dan statistic, serta psikologi dan ilmu lain yang mempelajari perilaku manusia [6]. *Artificial Intelligence* didasarkan pada prinsip bahwa kecerdasan manusia dapat didefinisikan sedemikian rupa sehingga mesin dapat dengan mudah menirunya dan menjalankan tugas, dari yang paling sederhana hingga yang lebih kompleks. Tujuan *Artificial Intelligence* termasuk meniru aktivitas kognitif manusia. Para peneliti dan pengembang di lapangan secara mengejutkan membuat langkah cepat dalam meniru kegiatan seperti pembelajaran, penalaran, dan persepsi, sejauh ini dapat didefinisikan secara konkret. Beberapa pengembang percaya bahwa *Artificial Intelligence* dapat dikembangkan melebihi kapasitas manusia untuk belajar atau menalar subjek apa pun. *Artificial Intelligence* bekerja pada penalaran induktif, mengolah dataset untuk prediksi hasil [7].

C. TensorFlow

TensorFlow adalah *platform open source end-to-end* untuk *Machine Learning*. Terdapat ekosistem tools, libraries, dan sumber daya komunitas yang komprehensif dan fleksibel yang memungkinkan peneliti mendorong state-of-the-art pada *Machine Learning* dan developers dengan mudah dapat membangun dan menerapkan aplikasi yang didukung *Machine Learning*. TensorFlow menawarkan beberapa tingkat abstraksi sehingga anda dapat memilih sesuai kebutuhan. *Build* dan *train* model menggunakan *high-level* Keras API, yang memudahkan memulai TensorFlow dan *Machine Learning* [8].

D. TensorFlow Lite

TensorFlow Lite adalah sebuah framework *Deep Learning* open source untuk inferensi di perangkat. Selain itu TensorFlow Lite juga memungkinkan *Machine Learning* di perangkat dengan membantu *developers* menjalankan model mereka di perangkat seluler atau *Internet of things* (IoT). Beberapa fitur yaitu untuk mengatasi lima kendala utama yaitu *latency*, *privacy*, *connectivity*, *size*, dan konsumsi daya. TensorFlow Lite memberi dukungan berbagai platform, mencakup Android dan iOS, Embedded Linux dan *microcontrollers*. TensorFlow Lite memberi dukungan bahasa yang beragam termasuk Java, Swift, Objective-C, C++, dan Python[8].

E. Convolutional Neural Network

Dalam *Deep Learning*, *Convolutional Neural Network* atau yang biasa disingkat menjadi CNN adalah jenis jaringan saraf multi-lapisan yang sangat istimewa. CNN Dirancang untuk mengenali pola visual langsung dari gambar dengan pemrosesan minimal [9]. CNN bisa digunakan untuk mendeteksi dan mengenali objek pada sebuah gambar. CNN juga disebut sebagai sebuah teknik yang terinspirasi dari cara mamalia-manusia menghasilkan persepsi visual. Penelitian oleh Hubel dan Wiesel [10], menunjukkan bahwa neuron dalam korteks visual pada otak mamalia tersusun menjadi sebuah topographical map, yang setiap levelnya fokus pada karakteristik tertentu. Dengan kata lain, sebuah gambar dipresentasikan ke dalam level fitur hierarki yang membentuk gambar tersebut secara keseluruhan.

F. Deep Learning

Deep Learning merupakan fungsi kecerdasan buatan atau biasa disebut *Artificial Intelligence* yang meniru cara kerja otak manusia dalam mengolah data dan menciptakan pola untuk digunakan dalam pengambilan keputusan. *Deep Learning* adalah bagian dari *Machine Learning* dalam *Artificial Intelligence* yang mampu belajar tanpa pengawasan dari data yang tidak terstruktur atau tidak berlabel. Juga dikenal sebagai pembelajaran *Deep Neural Learning* atau *Deep Neural Network*. *Deep Learning* adalah pembelajaran mesin yang memungkinkan komputer untuk belajar dari pengalaman dan memahami dunia dalam konsep hierarki [11].

G. Cloud Storage

Cloud Storage adalah media penyimpanan *file* berbasis online atau digital yang mengandalkan koneksi Internet untuk akses data. *Cloud Storage* memungkinkan penyimpanan tidak terbatas dan dapat diakses di seluruh dunia dan memiliki daya tahan yang tinggi. Beberapa fitur dari *Cloud Storage* yaitu *Cloud Storage* dapat konfigurasi data dengan *Object Lifecycle Management* untuk secara otomatis beralih ke kelas penyimpanan berbiaya lebih rendah saat memenuhi kriteria

yang ditentukan, seperti saat menyimpan versi data yang lebih baru. *Cloud Storage* memiliki daftar lokasi keranjang penyimpanan yang terus bertambah tempat, data dapat disimpan dengan beberapa opsi redundansi otomatis [12].

H. Cloud SQL

Cloud SQL merupakan layanan basis data relasional dari *Google Cloud Platform* yang menyediakan berbagai fitur untuk memudahkan kita dalam membuat, mengatur dan mengelola data pada *Cloud*. *Cloud SQL* mendukung sepenuhnya untuk MySQL, PostgreSQL, dan SQL Server. Beberapa fitur utama dari *Cloud SQL* yaitu *Cloud SQL* secara otomatis memastikan database yang digunakan reliable, aman, dan scalable sehingga bisnis yang dijalankan akan terus berjalan tanpa gangguan. Akses *Cloud SQL* terhubung dengan mudah dari *App Engine*, *Compute Engine*, *Google Kubernetes Engine* dan workstation. Membuka kemungkinan analitik dengan menggunakan *BigQuery* untuk langsung *query database Cloud SQL*. *Database Migration Service* memudahkan migrasi database ke *Cloud SQL* dengan downtime yang minimal [13].

I. Cloud Function

Cloud Function adalah *Scalable pay-as-you-go functions as a service (FaaS)* untuk menjalankan kode backend pengguna tanpa menggunakan server. Beberapa fitur utama dari *Cloud Function* yaitu *Cloud Function* hanya menagih pengguna untuk waktu eksekusi fungsi, *Cloud Function* menggunakan kerangka *function as a services* untuk menjalankan fungsi di berbagai lingkungan dan mencegah penguncian. Lingkungan yang didukung mencakup *Cloud Function*, lingkungan *developer local*, *local*, *Cloud Run*, dan *Knative-based serverless* lainnya [14].

J. Database

Database adalah kumpulan data yang dikelola dan disusun sedemikian rupa dengan berdasarkan ketentuan yang saling berhubungan sehingga mudah dalam pengelolaannya. Melalui pengelolaannya tersebut dapat mempermudah pengguna untuk mencari informasi, menyimpan informasi dan membuang informasi [15].

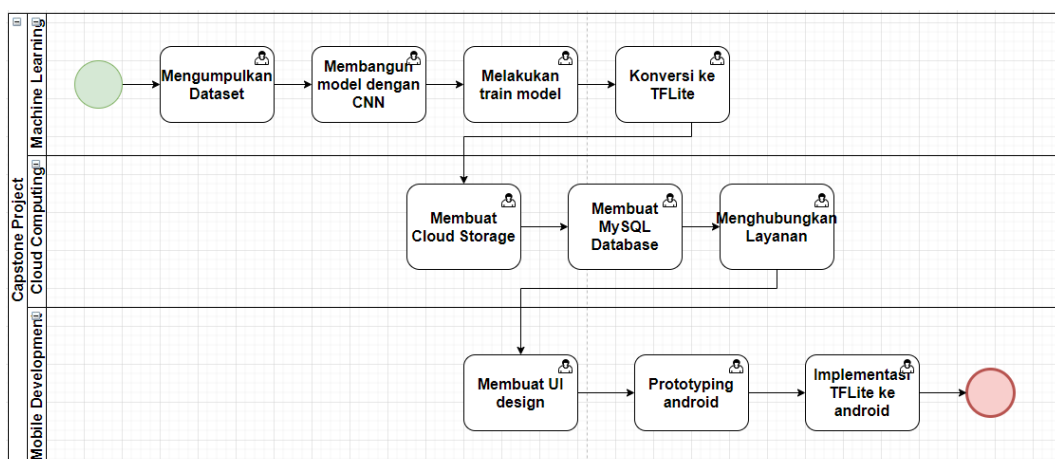
III. ANALISIS DAN RANCANGAN SISTEM

A. Analisis Aplikasi Posdistre

Posdistre adalah aplikasi berbasis Android yang akan digunakan oleh pengguna umum untuk mempermudah Kementerian PUPR dalam melakukan pengambilan data infrastruktur setelah bencana. Aplikasi ini memiliki *Image Classification* yang digunakan untuk mempermudah klasifikasi data yang akan disimpan.

B. Proses Bisnis

Proses bisnis adalah gabungan dari aktivitas yang memiliki peran untuk mencapai tujuan yang tersusun sesuai dengan alur aktivitas. Proses bisnis berguna untuk memberikan gambaran mengenai proses berjalannya sebuah bisnis. Berikut merupakan proses bisnis yang dilakukan oleh penulis dan timnya untuk membuat aplikasi Posdistre.

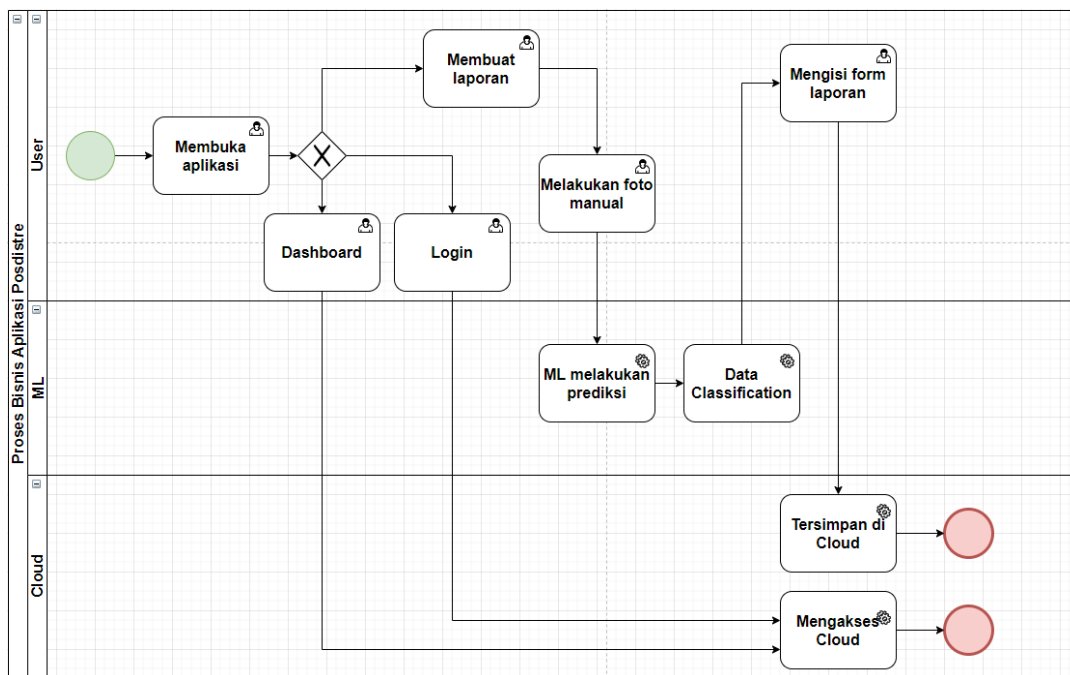


Gambar 1. Proses Bisnis Capstone Project

Berikut adalah langkah-langkah yang diambil oleh penulis dan timnya dalam pembuatan aplikasi Posdistre :

1. Mengumpulkan dataset secara manual dan menyimpannya pada Google Drive
2. Membangun sebuah model menggunakan *Convolutional Neural Network*
3. Melakukan train pada model dan menghasilkan *SavedModel*
4. Melakukan konversi ke TensorFlow Lite
5. Membuat *Cloud Storage*
6. Membuat MySQL *Database* dengan Cloud SQL
7. Menghubungkan layanan dengan *Cloud Function*
8. Membuat *User Interface* design
9. Prototyping
10. Implementasi TensorFlow Lite ke Android.

Adapun proses bisnis untuk memberikan gambaran bagaimana user menggunakan aplikasi Posdistre.



Gambar 2. Proses Bisnis Aplikasi Posdistre

Ketika Pengguna atau user berada di lokasi setelah bencana dan ingin melapor infrastruktur yang rusak pengguna dapat memulai dengan membuka aplikasi kemudian pengguna dapat melakukan *view dashboard* yaitu untuk mengakses ke *Cloud* berdasarkan apa yang telah dilapor oleh pengguna, atau pengguna dapat *login* dan dapat mengakses semua laporan, atau pengguna dapat membuat laporan dengan melakukan foto secara manual dengan perangkat seluler kemudian aplikasi akan melakukan prediksi *image classification* yang kemudian setelah prediksi sukses maka akan dilakukan berdasarkan klasifikasi data sehingga setelah pengguna mengisi form laporan, data akan tersusun sesuai kategori jenis, saat ini jenis yang ada adalah bangunan, jalanan, jembatan, mobil, dan motor.

C. Rancangan Dataset

Dataset adalah data yang akan digunakan atau diolah, penulis menggunakan *dataset* yang dikumpul secara manual oleh penulis dan tim bagian *Machine Learning*, dataset yang digunakan oleh penulis adalah dalam bentuk gambar dan diambil 5 classes yaitu Bangunan, Jalan, Jembatan, Mobil, dan Motor. Setiap classes memiliki 400 gambar untuk train dan 100 gambar untuk *validation*. Data gambar yang diambil oleh penulis bersumber dari Open Image Dataset V6 dan Kaggle.

Open Image Dataset V6 adalah kumpulan data yang mulai dari 9 juta gambar yang dianotasi dengan *image label, object bounding boxes, object segmentation masks, visual relationships*, dan narasi lokal. Sedangkan Kaggle adalah sebuah

komunitas belajar online yang dimana anggota komunitasnya bisa berkompetisi, saling berbagi ilmu dan juga terdapat lebih dari 1000 dataset.

IV. IMPLEMENTASI

A. Implementasi Machine Learning

Pada pembuatan aplikasi Posdistre, penulis dan tim penulis di bagian *Machine Learning* membuat sebuah model dengan susunan *Convolutional Neural Network* sebagai berikut :

Model: "sequential_1"		
Layer (type)	Output Shape	Param #
conv2d_2 (Conv2D)	(None, 298, 298, 16)	448
max_pooling2d_2 (MaxPooling2)	(None, 99, 99, 16)	0
conv2d_3 (Conv2D)	(None, 97, 97, 32)	4640
max_pooling2d_3 (MaxPooling2)	(None, 32, 32, 32)	0
flatten_1 (Flatten)	(None, 32768)	0
dense_3 (Dense)	(None, 32)	1048608
dropout_1 (Dropout)	(None, 32)	0
dense_4 (Dense)	(None, 64)	2112
dropout_2 (Dropout)	(None, 64)	0
dense_5 (Dense)	(None, 128)	8320
dense_6 (Dense)	(None, 5)	645

Total params: 1,064,773		
Trainable params: 1,064,773		
Non-trainable params: 0		

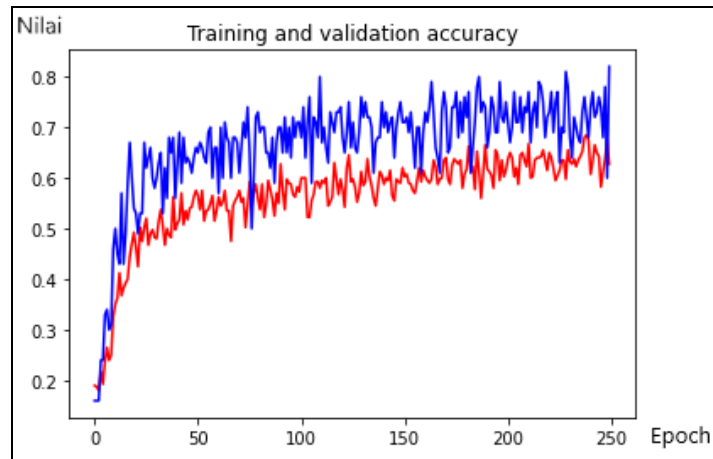
Gambar 3. Model Sequential

Dari gambar 3 dapat dilihat bahwa *Convolutional Layer* yang digunakan berjumlah dua, *Max Pooling Layer* berjumlah dua, *Flatten Layer* berjumlah satu, *Dense Layer* berjumlah empat, dan *Dropout Layer* berjumlah dua. Model tersebut menghasilkan total parameter sejumlah 1.064.773, *trainable* parameter berjumlah 1.064.773 dan *Non-trainable* parameter berjumlah nol. Kemudian model di compile oleh tim penulis dan dilakukan *train* dengan *epochs* berjumlah 250. Berikut adalah hasil dari *epoch* terakhir :

Epoch 250/250 8/8 [=====] - 10s 1s/step - loss: 0.9290 - accuracy: 0.6275 - val_loss: 0.5544 - val_accuracy: 0.8200
--

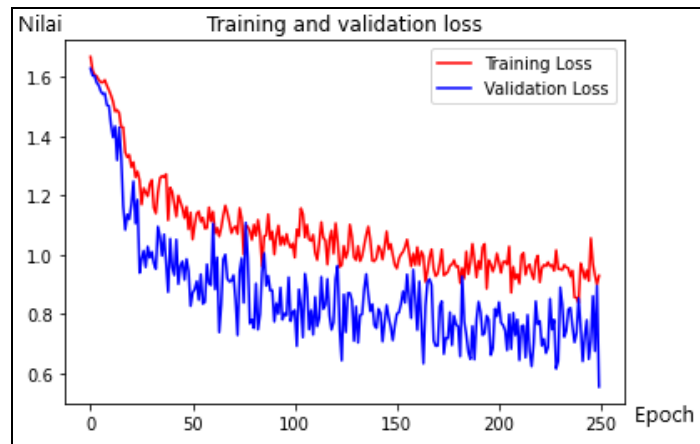
Gambar 4. Epoch Terakhir

Dapat dilihat dari gambar diatas bahwa hasil yang didapatkan oleh tim *Machine Learning* penulis pada *epoch* terakhir adalah *training loss* sebesar 0.929, *training accuracy* sebesar 0.6275, *validation loss* sebesar 0.5544 dan *validation accuracy* sebesar 0.82 dengan *steps per epoch* adalah 8. Kemudian juga setelah *train* model, tim penulis juga membuat grafik garis untuk melihat nilai pada *training accuracy* dan *validation accuracy*.



Gambar 5. Grafik garis *training* dan *validation accuracy*

Dari grafik garis di atas menunjukkan bahwa *training accuracy* dan *validation accuracy* terus meningkat. Pada *epoch* pertama, *training accuracy* sebesar 0.19 dan *validation accuracy* sebesar 0.16. Pada *epoch* terakhir, *training accuracy* sebesar 0.6275 dan *validation accuracy* sebesar 0.82. Selain grafik garis *training accuracy* dan *validation accuracy*, juga terdapat grafik garis dari *training loss* dan *validation loss*.

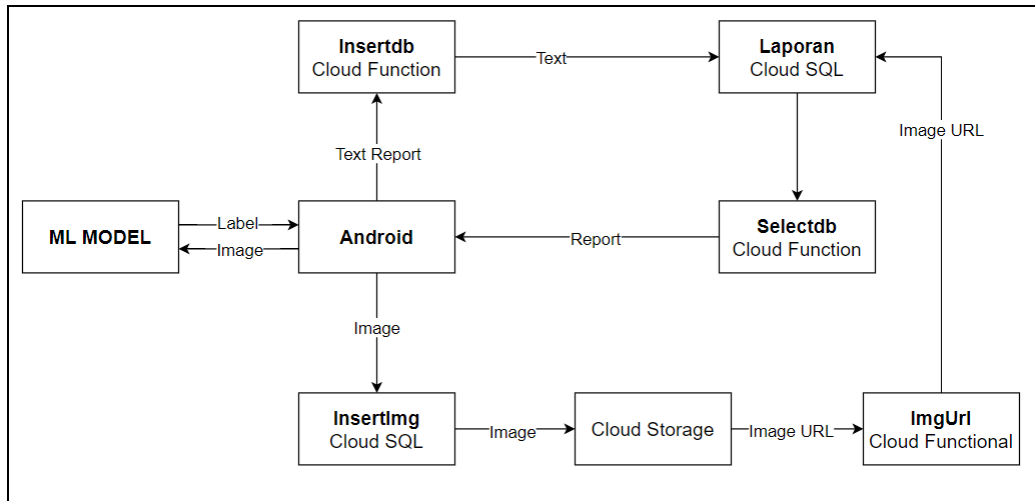


Gambar 6. Grafik garis *training* and *validation loss*

Dari grafik garis di atas menunjukkan bahwa *training loss* dan *validation loss* terus menurun. Pada *epoch* pertama, *training loss* sebesar 1.6671 dan *validation loss* sebesar 1.6274. Pada *epoch* terakhir, *training loss* sebesar 0.929 dan *validation loss* sebesar 0.5544.

B. Implementasi Cloud

Pada implementasi *Cloud*, tim penulis bagian *Cloud Computing* menggunakan tiga services dari *Google Cloud Platform* (GCP) yaitu *Cloud Storage*, *Cloud SQL* dan *Cloud Functions*. *Cloud Storage* digunakan untuk menyimpan data gambar yang ketika pengguna mengambil gambar menggunakan perangkat seluler, *Cloud SQL* digunakan untuk menyimpan semua laporan yang dibuat pengguna dan *Cloud Function* digunakan untuk menghubungkan services lainnya agar dapat terhubung dengan aplikasi Posdistre.



Gambar 7. Infrastruktur Cloud

Gambar diatas akan menjelaskan bagaimana cara kerja aplikasi dengan *Cloud*. Pertama, Aplikasi akan mengirimkan laporan dari aplikasi ke *Cloud* dengan mengirimkan *HTTP Post Request* ke *Cloud Storage* dan *Cloud SQL* dengan menggunakan *Cloud Function* melalui format *JSON*. Kedua, *Cloud Function* akan melakukan mengubah *HTTP Post Request* menjadi *query* dan kode untuk dapat dimasukkan ke *Cloud SQL* dan untuk gambar akan dimasukkan ke *Cloud Storage*. Pada akhirnya, aplikasi akan mendapatkan kembali laporannya dengan mengirimkan *Get Request* ke *Cloud Function* dan *Cloud Function* akan mengambil laporan dari *Cloud SQL* dan *Cloud Storage* menjadi sebuah format *JSON* untuk dikirim ke aplikasi *Posdistre*.

C. Implementasi Android

Pada implementasi *Android*, Tim penulis bagian *Mobile Development* membuat sebuah desain tampilan antarmuka berbasis *Android* menggunakan *Figma*. *Figma* adalah alat desain digital dan alat *prototyping* yang dapat digunakan untuk desain *UI* dan *UX*. Berikut adalah tampilan aplikasi *Posdistre* :

1) Tampilan *Splash Screen*



Gambar 8. Tampilan *Splash Screen*

Gambar 8 adalah tampilan ketika pengguna membuka aplikasi, maka akan muncul *splash screen* tersebut sebelum masuk ke tampilan berikutnya.

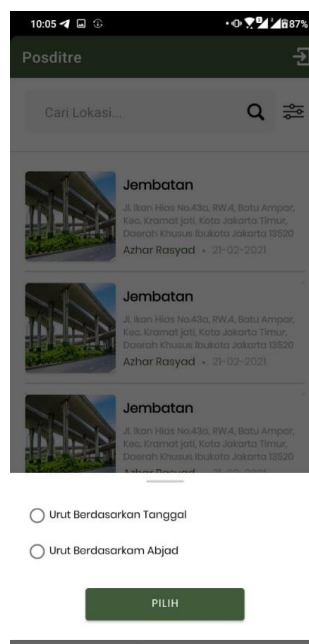
2) Tampilan Dashboard



Gambar 9. Tampilan Dashboard

Ketika pengguna berhasil masuk ke aplikasi maka pengguna akan melihat tampilan *Dashboard*, pada gambar 9 pengguna bisa melihat data laporan yang sebelumnya dilapor oleh pengguna, kemudian pengguna juga bisa menggunakan fitur sederhana yaitu *search* dan *filter* untuk memudahkan pengguna mencari data laporan.

3) Tampilan Filter



Gambar 10. Tampilan Filter

Gambar 10 adalah tampilan yang akan muncul ketika pengguna ingin menggunakan fitur *filter*, pada fitur *filter* ini pengguna dapat mengurutkan data berdasarkan tanggal atau abjad.

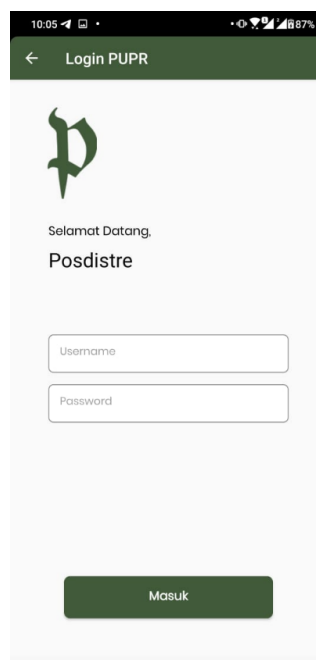
4) Tampilan Detail Laporan



Gambar 11. Tampilan detail laporan

Pada gambar 11, pengguna dapat melihat kembali detail laporan yang sebelumnya dilapor oleh pengguna. Pada gambar 11 terdapat gambar, keterangan gambar, deskripsi dan alamat gambar diambil.

5) Tampilan Login



Gambar 12. Tampilan Login

Pada gambar 12, pengguna yang memiliki akun dapat melakukan *login* ke dalam aplikasi untuk bisa melihat semua data laporan yang ada di *Cloud*. Fitur *login* ini dibuat untuk anggota PUPR jika nantinya ingin melihat semua data lapor yang terdapat di *Cloud*.

6) Tampilan Form Laporan



Gambar 13. Tampilan Form Laporan

Pada tampilan Form Laporan, pengguna dapat membuat laporan dengan mengambil gambar melalui perangkat seluler pengguna, kemudian pengguna dapat mengisi form hingga lengkap untuk dapat membuat laporan. Untuk alamat lengkap dan tanggal akan terisi otomatis berdasarkan posisi dan tanggal pada perangkat seluler tersebut.

V. PENGUJIAN

A. Rencana Pengujian

Sebelum aplikasi Posdistre ini dikatakan layak, diperlukan suatu pengujian untuk memastikan dan menunjukkan bahwa sistem yang dikembangkan telah sesuai dengan tujuan pengembang serta kebutuhan pengguna. Oleh karena itu, penulis akan membuat serangkaian rencana pengujian dengan metode *black box testing*, dimana pengujian terhadap aplikasi Posdistre ini dilakukan dengan *test case* untuk setiap pengujian validasi. Metode testing ini nantinya akan menghasilkan keluaran yang diharapkan dari sistem terhadap *test case* yang dijalankan dan akan menghasil kesimpulan kelayakan aplikasi. Berikut adalah rencana pengujian aplikasi Posdistre:

Tabel I
Rencana Pengujian Aplikasi Posdistre

Item Uji	Butir Uji	Jenis Pengujian
Login	Validasi Inputan	Black Box
	Kesesuaian Proses	
Lapor	Validasi Inputan	Black Box
	Kesesuaian Proses	
Filter	Validasi Inputan	Black Box
	Kesesuaian Proses	

B. Pengujian Black Box

Hasil Pengujian *black box* dibuat untuk memperlihatkan serta membuktikan bahwa aplikasi yang dikembangkan sudah sesuai dengan tujuan pembuatan. Rencana pengujian *black box* dengan *test case* adalah sebagai berikut:

1) Pengujian Login

Tabel II
Pengujian Login

No	Data Uji	Input	Harapan	Hasil
1	Username, Password	admin, admin	Cek data login	Valid
2	Username, Password	admin, null	Tampil pesan, Password harus diisi	Valid
3	Username, Password	null, admin	Tampil pesan, Username harus diisi	Valid

2) Pengujian Laporan

Tabel III
Pengujian Laporan

No	Data Uji	Input	Harapan	Hasil
1	Gambar, Keterangan, Nama Lengkap, Deskripsi, Alamat Lengkap, Tanggal	Format Gambar, Otomatis setelah ambil gambar, Darwin Gunawan, Jembatan rusak setelah gempa, Alamat otomatis diisi, Tanggal otomatis diisi	Data laporan disimpan	Valid
2	Gambar, Keterangan, Nama Lengkap, Deskripsi, Alamat Lengkap, Tanggal	Format Gambar, Otomatis setelah ambil gambar, Darwin Gunawan, Jembatan rusak setelah gempa, Alamat otomatis diisi, Tanggal otomatis diisi	Tampil pesan, Nama Lengkap harus diisi	Valid
3	Gambar, Keterangan, Nama Lengkap, Deskripsi, Alamat Lengkap, Tanggal	Format Gambar, Otomatis setelah ambil gambar, Darwin Gunawan, Jembatan rusak setelah gempa, Alamat otomatis diisi, Tanggal otomatis diisi	Tampil pesan, Deskripsi harus diisi	Valid

3) Pengujian Filter

Tabel IV
Pengujian Filter

No	Data Uji	Input	Harapan	Hasil
1	Filter	Tanggal	Tampil pesan, Urut berdasarkan Tanggal dipilih	Valid
2	Filter	Abjad	Tampil pesan, Urut berdasarkan Abjad dipilih	Valid

C. Kesimpulan Pengujian

Berdasarkan pengujian black box dengan test case pada *Login*, *Laporan* dan *Filter* dapat ditarik kesimpulan bahwa validasi dan kesesuaian proses sistem berjalan dengan baik sehingga sistem dinyatakan layak. Pengujian ini membantu memastikan bahwa sistem telah berjalan dengan baik.

VI. KESIMPULAN

A. Simpulan

Kesimpulan dari karya tulis yang berjudul “Pembuatan Aplikasi Pendataan Infrastruktur Dampak Bencana (Posdistre) untuk Kementerian Pekerjaan Umum dan Perumahan Rakyat” ini adalah sebagai berikut :

- 1) Aplikasi ini bertujuan untuk membantu Kementerian PUPR untuk mendapatkan data infrastruktur yang terdampak setelah bencana.
- 2) Membuka kesempatan bagi masyarakat untuk bisa berpartisipasi dalam pengumpulan data infrastruktur rusak akibat bencana untuk membantu mempercepat proses kerja Kementerian PUPR.
- 3) Mempermudah Kementerian PUPR untuk mendapatkan data secara cepat, efisien dan mudah untuk mengambil keputusan.

B. Saran

Dari karya tulis ini penulis menyadari bahwa masih banyak hal yang dapat dikembangkan pada sistem ini untuk menjadi aplikasi yang lebih baik kedepannya yaitu sebagai berikut :

- 1) Perancangan Aplikasi Posdistre membutuhkan tampilan khusus untuk pengguna Kementerian PUPR yang akan login ke aplikasi ini.
- 2) Aplikasi masih belum berjalan dengan baik sehingga masih dibutuhkan perbaikan pada bagian yang tidak sesuai harapan.
- 3) Perancangan aplikasi Posdistre untuk bagian *Image Classification* masih membutuhkan fitur yaitu saat gambar yang diambil tidak terdeteksi oleh model sehingga gambar yang diambil dapat disimpan ke *Cloud* dan dilakukan training ulang untuk model tersebut sehingga kedepannya bisa mendeteksi gambar yang lebih banyak.
- 4) Hasil akurasi yang didapatkan dari train model masih terbilang belum cukup meyakinkan sehingga akan lebih baik jika akurasi bisa dinaikkan.

DAFTAR PUSTAKA

- [1] A. Geron, *Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow*, O'Reilly Media; 2nd Edition, 2019.
- [2] O. Theobald, *Machine Learning For Absolute Beginners*, Independently published, 2018.
- [3] E. Alpaydin, “Introduction to Machine Learning, fourth edition (Adaptive Computation and Machine Learning series),” The MIT Press, 2020.
- [4] A. Burkov, *The Hundred-Page Machine Learning Book*, Andriy Burkov, 2019.
- [5] S. Russell, *Artificial Intelligence: A Modern Approach*, Pearson; 4th edition, 2021.
- [6] A. Gulli, “Deep Learning with TensorFlow 2 and Keras: Regression, ConvNets, GANs, RNNs, NLP, and more with TensorFlow 2 and the Keras API,” Packt Publishing, 2019.
- [7] E. J. Larson, “The Myth of Artificial Intelligence: Why Computers Can’t Think the Way We Do,” Belknap Press, 2021.
- [8] L. Moroney, *AI and Machine Learning for Coders: A Programmer’s Guide to Artificial Intelligence*, O'Reilly Media; 1st Edition, 2020.
- [9] M. Sewak, *Practical Convolutional Neural Networks: Implement advanced deep learning models using Python*, Packt Publishing, 2018.
- [10] D. H. Hubel dan T. N. Wiesel, “Binocular interaction and functional architecture in the cat's visual cortex,” *The Journal of physiology*, vol. I, pp. 106-154, 1962.
- [11] I. Goodfellow, Y. Bengio dan A. Courville, *Deep Learning*, The MIT Press, 2006.
- [12] J. Bernstein, *Cloud Storage Made Easy: Securely Backup and Share Your Files*, James Bernstein, 2018.
- [13] N. Sabharwal, *Hands on Google Cloud SQL and Cloud Spanner; Deployment, Administration and Use Cases with Python*, Apress, 2019.
- [14] K. Chowman, *Hands-on Serverless Computing: Build, run and orchestrate serverless applications Using AWS Lambda , Microsoft Azure Functions, and Google Cloud Function*, Packt Publishing, 2018.
- [15] A. Petrov, “Database Internals: A Deep Dive Into How Distributed Data Systems Work,” O'Reilly Media, 2019.