

## Aplikasi Android Pengklasifikasi Semantik Teks Menggunakan Tensorflow Lite Pada Ringkasan Karya Ilmiah

Adam Permana<sup>1\*</sup>, Khairi Budayawan<sup>2</sup>

<sup>1</sup>Prodi Pendidikan Teknik Informatika Fakultas Teknik Universitas Negeri Padang

<sup>2</sup>Jurusan Teknik Elektronika Universitas Negeri Padang

Jl. Prof. Hamka Kampus UNP Air Tawar Padang

\*Corresponding author e-mail: adampermananine@gmail.com

### ABSTRAK

Perkembangan *neural network* di era industri 4.0 semestinya dapat membantu berbagai bidang pekerjaan, salah satunya literatur karya ilmiah. Permasalahannya adalah karya tulis ilmiah masih menggunakan penyortiran tema/semantik secara manual. Tujuan dari studi ini adalah untuk membangun aplikasi klasifikasi semantik teks yang memungkinkan pengguna melakukan penyortiran berdasarkan tema/semantik menggunakan model *neural network* yang disematkan pada ponsel cerdas. Pengembangan aplikasi menggunakan metode *waterfall* yang didalamnya terdapat analisis dan perancangan sistem. Aplikasi mengimplementasi fitur *text recognition* dari Firebase ML-Kit. Pengembangan model menggunakan metode atau pendekatan umum siklus *machine learning* yang terdiri dari identifikasi data, persiapan data, pemilihan algoritma, pelatihan model, evaluasi model dan deploy model. Model dibangun menggunakan data abstrak karya ilmiah dari Perpustakaan Teknik Elektronika Universitas Negeri Padang. Dari total data didapatkan 84 data latih dan 21 data uji menggunakan perbandingan 80:20 persen untuk uji validasi. Model *neural network* menggunakan spesifikasi AverageWordVec yang disediakan oleh Tensorflow Lite Model Maker dengan 3 output klasifikasi. Uji validasi model mencapai 0.7619 nilai akurasi dengan 0.7782 nilai loss. Model di eksekusi menggunakan *interpreter* Tensorflow Lite yang tertanam pada aplikasi. Hasil aplikasi memenuhi keseluruhan analisis kebutuhan fungsional sistem.

**Kata kunci** : Tensorflow Lite, ML-Kit, Neural Network, Semantik Teks

### ABSTRACT

*The development of neural network in the era of industry 4.0 should be able to help various work fields, one of which is scientific literature. The problem is that scientific papers still use manual sorting of themes/semantics. The purpose of this study is to build semantic text classification apps that allowed user to sort based on theme / semantics with a smartphone-embedded neural network model. Application development uses the waterfall method which contain system analysis and system design. The app implements the text recognition feature from Firebase ML-Kit. Model development uses a general method or approach called the machine learning cycle which consist of data identification, data preparation, algorithm selection, training, model evaluation and model deploy. The model was built using abstract data of scientific papers from the Electronics Engineering Library, Universitas Negeri Padang. From the total data, 84 training data and 21 test data were obtained using 80:20 percent comparison for the validation test. The neural network model uses the AverageWordVec specification provided by Tensorflow Lite Model Maker with 3 classification outputs. The validation test reached 0.7619 accuracy values with 0.7782 loss values. Models are executed using the embedded Tensorflow Lite interpreter in the application. The application results fulfill the overall system functional requirements analysis.*

**Keywords:** Tensorflow Lite, ML-Kit, Neural Network, Text Semantic

## I. PENDAHULUAN

*Machine learning* adalah satu dari sekian faktor pendukung revolusi industri 4.0 yang saat ini kita jalani [1]. *Machine learning* sendiri adalah bentuk kecerdasan buatan yang memungkinkan sistem belajar dari data tanpa perlu diprogram secara eksplisit [2]. Salah satu model dari *machine learning* adalah *neural network*.

Perkembangan *neural network* saat ini telah memungkinkan perangkat lunak untuk melakukan pekerjaan layaknya indra dengan otak manusia seperti pengenalan suara, pengenalan wajah, deteksi objek, pengenalan teks, klasifikasi teks dan lain-lain. Pengenalan teks dan klasifikasi teks ini semestinya mampu membantu pekerjaan manusia di berbagai bidang yang berkaitan dengan dokumen seperti administrasi perkantoran, psikologi tulisan, evaluasi esai, dan terutama yang menjadi fokus pada studi ini yakni literatur karya tulis ilmiah.

Penyusunan atau penyortiran karya tulis ilmiah di perpustakaan umumnya masih dilakukan secara manual. Yang dimaksud dengan penyortiran manual disini adalah menyortir tema atau semantik pada abstrak/ringkasan dengan ‘mata kepala’ dan pengetahuan manusia (pustakawan). Penyortiran manual memiliki kekurangan yakni rentan terhadap aus dan membutuhkan kemampuan menganalisis. Hal ini secara potensial dipermudah oleh teknologi *neural network*.

Perusahaan jasa layanan internet Google di bulan Februari tahun 2017 merilis *software stack* pengembangan *neural network* dengan nama Tensorflow. Ditahun yang sama tepatnya pada bulan Mei, versi khususnya dirilis dengan nama Tensorflow Lite. Tensorflow Lite sendiri adalah versi ringan dari Tensorflow yang mampu dieksekusi lebih cepat pada perangkat seluler bahkan pada perangkat tertanam seperti mikrokontroler [3]. Tensorflow Lite juga menyediakan Model Maker, yakni *software stack* untuk mengubah model Tensorflow menjadi format FlatBuffer agar dapat dieksekusi oleh Tensorflow Lite Interpreter.

Tensorflow Lite Model Maker menyediakan sample berupa spesifikasi model yang diberi nama AverageWordVec. Spesifikasi model ini memiliki arsitektur yang bertujuan untuk mendapatkan semantik dari korpus tulisan dengan cara melakukan rerata vektor (*averaging vector*). Tersedianya spesifikasi model ini memungkinkan pengembang menyesuaikan konfigurasi spesifikasi dengan data dan *output* yang ia kehendaki.

Google melalui layanan Firebase-nya merilis *software development kit* yang diberi nama ML-Kit. Layanan Firebase ML-Kit memungkinkan pengembang perangkat lunak membangun fitur-fitur

*machine learning* yang salah satunya adalah pengenalan teks (*Text Recognition*). Firebase sendiri adalah platform berbagai layanan untuk mempermudah pengembang perangkat lunak membangun aplikasi *mobile* berkualitas tinggi dengan cepat dan efisien [4]. Selain ML-Kit, layanan Firebase lainnya adalah Realtime Database, Cloud Firestore, Hosting, Cloud Function dan lain-lain yang tentunya sangat bermanfaat dalam pengembangan aplikasi *mobile*.

Studi ini bertujuan untuk membangun prototipe aplikasi dengan menggabungkan fitur Tensorflow Lite dan Firebase ML-Kit sehingga menjadi aplikasi pengklasifikasi semantik teks. Aplikasi *mobile* dibangun menggunakan IDE Android Studio dengan bahasa pemrograman Kotlin dan Java. Sedangkan model *neural network* dibangun menggunakan Jupyter Notebook dari layanan Google Colab menggunakan bahasa pemrograman Python.

## II. METODE PERANCANGAN SISTEM

Metode perancangan sistem terdiri dari 2 bagian, yakni perancangan aplikasi dan perancangan model *neural network*. Perancangan aplikasi menggunakan pendekatan *waterfall* yakni sebuah *software development life cycle* yang sistematis dan bersifat tetap[5]. Ini dikarenakan pendekatan ini fokus pada langkah-langkah yang jelas dan tujuan akhir terdefinisi di awal. Dalam pendekatan *waterfall* terdapat dokumentasi berupa analisis dan perancangan sistem. Sedangkan perancangan model menggunakan pendekatan umum siklus *machine learning* (*the machine learning cycle*) karena mengaplikasikan algoritma *machine learning* adalah proses iteratif [2].

### Analisis Sistem

Analisis sistem adalah kegiatan untuk melihat sistem yang sudah berjalan, melihat bagian mana yang bagus dan kemudian mendokumentasikan kebutuhan yang akan dipenuhi dalam sistem yang baru [6]. Dalam studi ini analisis sistem terdiri dari analisis kebutuhan, analisis kemananan, analisis dokumen *input-output* dan analisis prosedur.

### Analisis Kebutuhan Sistem

Analisis kebutuhan sistem terdiri dari analisis kebutuhan fungsional dan analisis kebutuhan perangkat. Kedua analisis tersebut disajikan dalam tabel berikut:

Tabel 1. Analisis kebutuhan fungsional.

No	Fungsional	Deskripsi Fungsional
1	Aplikasi mampu menangkap gambar ( <i>Camera Capture</i> )	Aplikasi menggunakan sumberdaya Android untuk mengaktifkan kamera dan menangkap gambar tersebut untuk diolah.
2	Aplikasi mampu membaca pola teks pada gambar	Menggunakan fitur <i>Text Recogniton</i> dari ML Kit, aplikasi mampu mengubah setiap teks yang ada pada gambar hasil tangkapan menjadi teks <i>string</i> .
3	Aplikasi mampu menyimpan teks yang ada menjadi file teks (.txt)	Untuk mempermudah proses <i>data collecting</i> saat membangun model <i>neural network</i> , aplikasi menyediakan tiga tombol untuk menyimpan teks <i>string</i> kedalam folder kategori dalam format <i>file</i> teks (.txt)
4	Aplikasi mampu menampilkan hasil prediksi	Hasil prediksi menampilkan kategori beserta nilai persentasi yang dengan rentang 0 s/d 1. Prediksi dilakukan melalui inferensi model Tensforflow Lite yang terpasang pada aplikasi
5	Aplikasi mampu menyimpan hasil prediksi	Hasil prediksi yang berupa teks terbaca ( <i>recognized text</i> ) dan kategori/class terprediksi dapat disimpan ke sebuah penyimpanan <i>non-volatile</i> seperti memori <i>internal smartphone</i> atau <i>database</i>

Tabel 2. Analisis kebutuhan perangkat.

No	Perangkat	Deskripsi Kebutuhan
1	Kebutuhan perangkat keras ( <i>Hardware</i> )	Pengembangan : Personal Computer dengan spesifikasi minimum processor Intel Core i5(~2.3 Ghz) dan RAM 4 GB (Diperlukan Smartphone tambahan untuk proses <i>debugging</i> ).  Penggunaan : Ponsel atau tablet dengan camera minimal 8 Megapixel.
2	Kebutuhan perangkat lunak ( <i>Software</i> )	Pengembangan : IDE Android Studio, Google Colabs(Diakses via web), Java, Kotlin dan Android SDK.  Penggunaan : Sistem operasi Android Lollipop (SDK 21).
3	Kebutuhan sumber daya manusia ( <i>Brainware</i> )	Pengembangan : Pengguna dengan pemahaman pengembangan perangkat lunak Android dan konsep neural network modern.  Penggunaan :

Tidak memerlukan spesifikasi. Didesain untuk umum, sehingga aplikasi dapat dioperasikan oleh siapapun.

## Analisis Keamanan Sistem

Analisis keamanan sistem adalah bagian yang perlu diperhatikan pengguna saat menggunakan aplikasi ini. Bagian ini merupakan saran-saran keamanan dalam menggunakan :

Tabel 2. Analisis keamanan sistem.

No	Saran	Pertimbangan
1	Seluruh korpus (badan tulisan) tertangkap kamera.	Setiap kata yang tidak berhasil terbaca oleh kamera akan mempengaruhi semantik bahkan proses training model. Penting untuk memastikan keseluruhan korpus tertangkap kamera.
2	Tulisan yang ditangkap berupa hasil cetak komputer.	Model yang ada pada Firebase TextRecognition dilatih melalui <i>font-font</i> umum yang digunakan dan pula tulisan tangan yang cukup signifikan dibedakan oleh komputer. Tulisan tangan yang sangat tidak signifikan akan dibaca berbeda oleh Firebase TextRecognition. Contohnya antara "0" dan "o", oleh tulisan tangan yang tak memperhatikan spasi antara kata, huruf "o" bisa saja terbaca "0" oleh sistem.
3	Posisi kamera berada lurus diatas kertas/objek tangkapan kamera	Perspektif (sudut pandang kamera) dalam memproses gambar objek tangkapan mempengaruhi tinggi/panjang karakter yang dibaca.

## Analisis Dokumen I/O

Pada rancangan aplikasi pengklasifikasi semantik teks pengguna melakukan input data tangkapan kamera. Output yang dihasilkan aplikasi berupa data kategori klasifikasi beserta nilai akurasi (*confident value*)-nya.

## Analisis Prosedur

Analisis prosedur dilakukan untuk menetapkan proses apa saja yang akan dilakukan sistem. Dalam proses pelakasnaanya dilakukan sesuai dengan prosedur yang telah ditetapkan, yaitu:

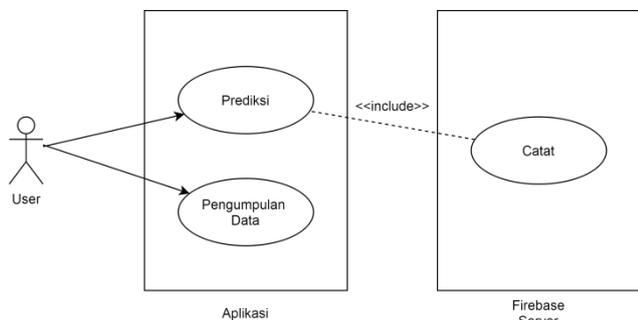
1. User membuka aplikasi
2. User menangkap gambar dengan kamera
3. User mendapatkan informasi hasil prediksi dari aplikasi.

## Perancangan Sistem

Rancangan sistem secara umum dilakukan dengan maksud memberikan gambaran umum tentang sistem yang akan dibuat. Salah satu cara mendokumentasikan rancangan adalah dengan menggunakan pemodelan. Karena pemodelan menspesifikasikan kebutuhan untuk menentukan penilaian dan kebenarannya, kelengkapannya serta konsistensinya [7]. Dalam studi ini pemodelan menggunakan UML (*Unified Modelling Language*). UML sendiri merupakan penyampaian model sistem yang mendukung paradigma *object oriented* [8]. UML juga dapat berfungsi sebagai cetak biru karena bersifat detail dan lengkap [8]. Mempertimbangkan beberapa diagram sangat sederhana untuk dimodelkan dengan UML, maka diagram yang digunakan adalah Use Case Diagram, Activity Diagram, Class Diagram, Object Diagram dan Component Diagram.

### Use Case Diagram

Use Case Diagram merepresentasikan aspek dinamis dari sistem. Secara spesifik, diagram ini digunakan untuk mendapatkan persyaratan (requirement) sistem termasuk pengaruh internal dan eksternal. Diagram ini juga berfungsi untuk menggambarkan interaksi antar sistem serta kebutuhan fungsional dari sistem itu sendiri [9].



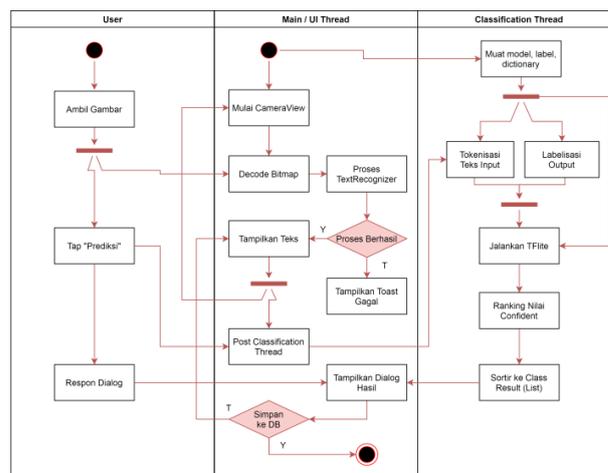
Gambar 1. Use case diagram aplikasi klasifikasi semantik teks

Requirement yang harus dimiliki pengguna adalah fitur prediksi dan pengumpulan data. Use case hanya terdiri dari 1 user yang dalam pengoperasian fitur prediksi memerlukan sistem eksternal berupa layanan Firebase untuk melakukan pencatatan hasil prediksi.

### Activity Diagram

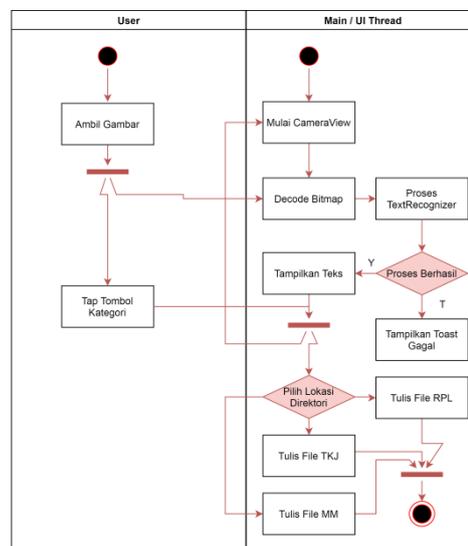
Activity Diagram pada dasarnya adalah sebuah flowchart untuk merepresentasikan aliran dari satu activity ke activity lainnya. Dalam pengembangan aplikasi ini terdapat 2 activity yang benar-benar memiliki fungsi dan telah dijabarkan pada Use Case Diagram. Activity ini adalah MainActivity sebagai implementasi dari fitur

prediksi, dan DataCollectingActivity sebagai implementasi fitur pengumpulan data.



Gambar 1. Activity diagram fitur prediksi

Pada aplikasi Android, thread yang dapat disaksikan langsung oleh pengguna disebut main thread atau UI thread. Sedangkan thread selain dari main thread disebut dengan workerthread. Classification Thread adalah workerthread yang dijalankan ketika konstruktor dari class TextClassificationClient di eksekusi.

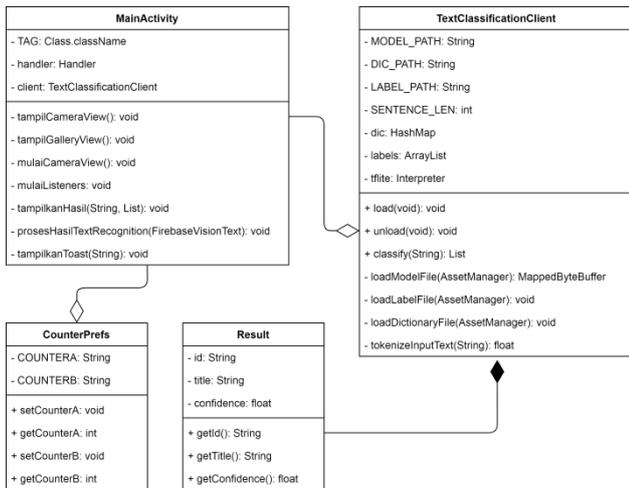


Gambar 3. Pemodelan fitur pengumpulan data dalam activity diagram

### Class Diagram

Class diagram merepresentasikan setiap class yang ada pada aplikasi. Class sendiri adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek [10]. Setiap class terdiri dari atribut dan method. MainActivity sebagai class yang membentuk objek

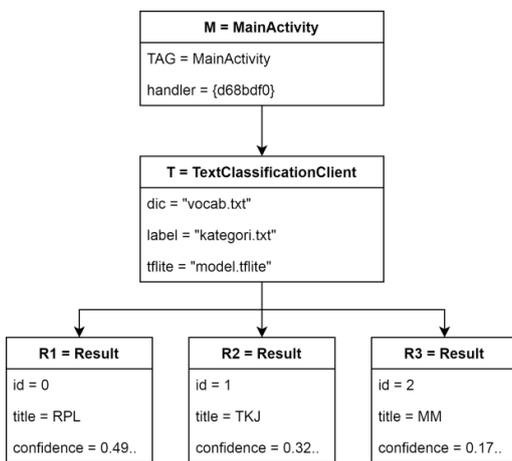
dari aktivitas utama memiliki hubungan agregasi terhadap TextClassificationClient dan CounterPrefs.



Gambar 1. Class diagram aplikasi klasifikasi semantik teks

### Object Diagram

Object Diagram merepresentasikan objek atau instance yang dibentuk dari class yang ada. Setiap objek memiliki nilai untuk masing-masing atributnya. Object MainActivity sebagai thread utama akan memanggil handler sebagai pengendali TextClassificationClient (TCC). Objek TCC ini membentuk objek dari Result sebanyak kategori yang ditentukan oleh atribut “label”. Masing-masing objek result di tumpuk dalam struktur data List.

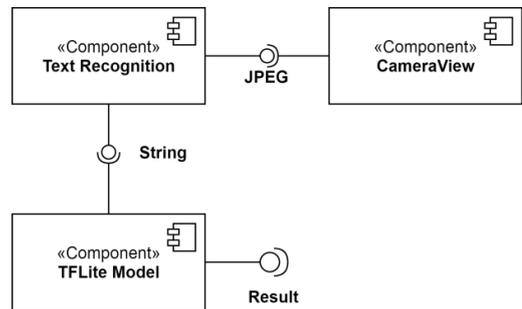


Gambar 1. Objek diagram saat proses prediksi.

### Component Diagram

Component Diagram merepresentasikan sekumpulan komponen dan hubungannya. Pada rancangan aplikasi pengklasifikasi semantik ini terdapat 3 komponen yang diimplementasikan. Cameraview terdiri dari beberapa class yang mengatur driver camera pada perangkat android,

class tersebut diantaranya adalah Audio, CameraListener dan CameraUtils. Text Recognition terdiri dari beberapa class yakni FirebaseVision, FirebaseVisionImage, dan FirebaseVisionText.



Gambar 1. Componen diagram pada aplikasi klasifikasi semantik teks.

### Perancangan Model Neural Network

Perancangan membangun model neural network pada penelitian ini memedomani *The Machine Learning Cycle*. Siklus ini bersifat kontinu karena mengoperasikan algoritma machine learning adalah proses berulang [2]. Siklus ini terdiri dari identifikasi data, persiapan data, pemilihan algoritma, training/pelatihan, evaluasi dan model deploy.

#### Identifikasi Data

Data pada tahap ini di identifikasi secara manual. Adapun ketentuan yang digunakan dalam menyeleksi abstrak skripsi sebagai data latih adalah sebagai berikut:

- Tidak ada redundansi judul abstrak
- Judul skripsi yang bertema analisis kepuasan ditanggihkan
- Judul skripsi yang bertema pendidikan di tanggihkan kecuali memiliki tema mata pelajaran yang berkorelasi dengan 3 tema klasifikasi yakni multimedia, komputer jaringan dan rekayasa perangkat lunak.

#### Persiapan Data

Pada tahap ini data diambil dari aplikasi yang menyimpan teks di smartphone menggunakan fitur pengumpulan data. Data kemudian di posisikan kedalam folder training dan test dengan perbandingan validasi 80:20 persen. Total file terkumpul saat proses identifikasi data adalah 154 file, setelah dilakukan persiapan data maka informasi file menjadi seperti tabel berikut:

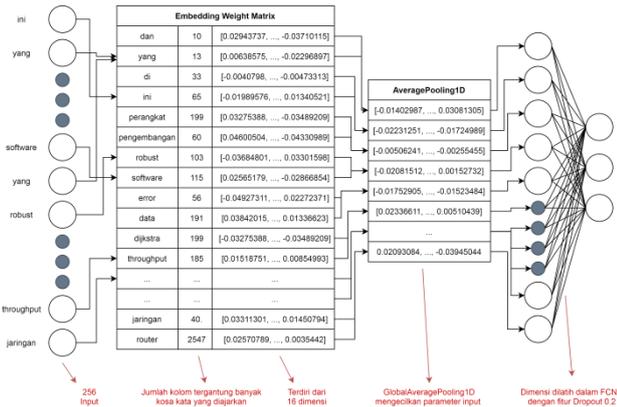
Tabel 2. Informasi file di tahap persiapan data.

No	Informasi	Jumlah
1	Total file	105 file / 8 folder
2	Total file	84 file / 3 folder

training		
3	Total file testing	21 file / 3 folder

### Pemilihan Algoritma

Khusus pengembangan aplikasi ini peneliti memilih arsitektur / spesifikasi model AverageWordVec yang merupakan model deep neural network untuk klasifikasi semantik. Adapun loss function yang digunakan adalah categorical cross entropy dengan optimizer-nya ialah RMSprop.



Gambar 1. Ilustrasi arsitektur AverageWordVec.

### III. HASIL DAN PEMBAHASAN

Hasil model machine learning akan dibahas sesuai tahap lanjutan dari The Machine Learning Cycle yakni training data, evaluasi dan model deploy. Sedangkan hasil aplikasi dibahas dalam hasil tampilan activity dan pengujian fungsional.

#### Training / Pelatihan Model

Pada tahap ini pengembang melatih model sebanyak 100 epoch dan mencapai nilai akurasi seperti berikut:

```
Epoch 95/100
2/2 [=====] - 0s 22ms/step - loss: 0.5513 - accuracy: 0.9167
Epoch 96/100
2/2 [=====] - 0s 25ms/step - loss: 0.5669 - accuracy: 0.8854
Epoch 97/100
2/2 [=====] - 0s 20ms/step - loss: 0.6152 - accuracy: 0.8229
Epoch 98/100
2/2 [=====] - 0s 21ms/step - loss: 0.5649 - accuracy: 0.8542
Epoch 99/100
2/2 [=====] - 0s 23ms/step - loss: 0.5357 - accuracy: 0.9479
Epoch 100/100
2/2 [=====] - 0s 19ms/step - loss: 0.5381 - accuracy: 0.9167
```

Gambar 1. Hasil tangkap layar proses training data.

#### Evaluasi Model

Pada tahap ini didapatkan hasil validasi menggunakan data testing dimana loss = 0.7782 dan akurasi = 0.7619.

```
model.summary()
Model: "sequential"
Layer (type)                Output Shape                Param #
-----
embedding (Embedding)       (None, 256, 16)            40768
global_average_pooling1d (G1) (None, 16)                  0
dense (Dense)                (None, 16)                  272
dropout (Dropout)            (None, 16)                  0
dense_1 (Dense)              (None, 3)                   51
-----
Total params: 41,091
Trainable params: 41,091
Non-trainable params: 0

loss, acc = model.evaluate(test_data)
1/1 [=====] - 0s 131ms/step - loss: 0.7782 - accuracy: 0.7619
```

Gambar 1. Hasil tangkap layar evaluasi dan summary model.

Jika diperhatikan nilai akurasi dan loss menggunakan data uji (test data) lebih rendah dari data latih (training data) yang dilakukan. Boleh dikatakan ini karena data masih sedikit, sehingga pola yang didapat dari data latih belum mencapai akurasi optimal dalam memprediksi data uji.

#### Model Deploy

Pada tahap ini model di export dengan format flat buffer Tensorflow Lite yakni (.tflite). Dalam Tensorflow Lite versi 0.2.2, spesifikasi model AverageWordVec secara default melakukan ekspor vocab file dan label menjadi metadata model.

#### Activity Menu Utama

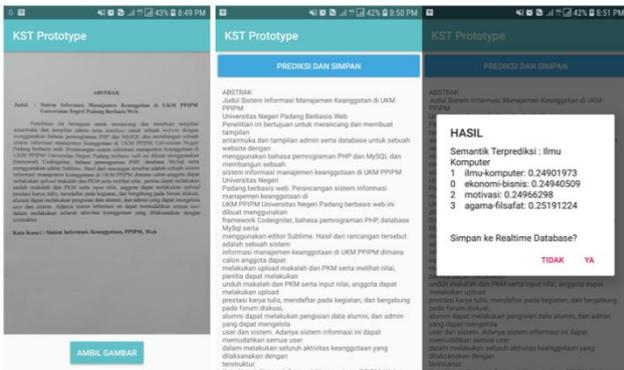
Bertindak sebagai launcher dan tidak memerlukan splashscreen dikarenakan aset-aset telah di muat secara multithreading. Sesuai namanya activity ini menjadi menu utama pada aplikasi.



Gambar 1. Menu utama aplikasi.

### Activity Fitur Prediksi

Activity Fitur Prediksi atau MainActivity memiliki 2 layout yang melakukan pergantian ditengah proses. Layout pertama adalah overlay camera, Sedangkan layout kedua adalah overlay result.

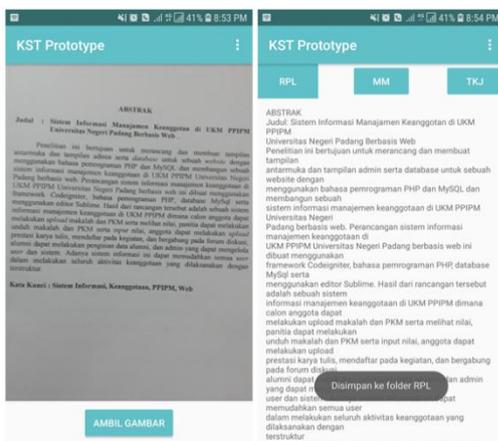


Gambar 1. Fitur prediksi

Overlay camera adalah layout yang dipanggil saat pertama kali activity berjalan. Camera overlay memiliki komponen UI CameraView untuk menangkap gambar. Sedangkan overlay result akan dimuat setelah proses pengenalan teks selesai dengan menampilkan teks tersebut pada komponen Textview sekaligus menyediakan tombol tunggal untuk mengeksekusi prediksi. Tombol tunggal ini juga menampilkan dialog untuk menyimpan data hasil prediksi tersebut ke Firebase Realtime Database.

### Activity Fitur Pengumpulan Data

Tata letak pada activity ini memiliki tidak jauh berbeda dengan Fitur Prediksi. Dikarenakan terdapat proses pengenalan teks, activity ini juga memiliki dua layout yakni camera overlay dan result overlay.



Gambar 1. Fitur pengumpulan data

Hanya saja pada result overlay, layout ini menampilkan tombol-tombol untuk menyimpan teks kedalam file teks (.txt). Saat ditekan, tombol terpilih akan menjadi datar dan sistem akan menampilkan komponen Toast untuk memberitahu pengguna dimana file tersebut tersimpan.

### Hasil Pengujian Aplikasi

Hasil pengujian menjawab analisis kebutuhan fungsional sistem.

Tabel 3. Hasil pengujian aplikasi.

No	Fungsional	Status
1	Aplikasi mampu menangkap gambar (Camera Capture)	Berfungsi
2	Aplikasi mampu membaca pola teks pada gambar	Berfungsi
3	Aplikasi mampu menyimpan teks yang ada menjadi file teks (.txt)	Berfungsi
4	Aplikasi mampu menampilkan hasil prediksi	Berfungsi
5	Aplikasi mampu menyimpan hasil prediksi	Berfungsi

### IV. KESIMPULAN

Perancangan dan pembuatan Aplikasi Pengklasifikasi Semantik Teks Menggunakan Tensorflow Lite Pada Ringkasan Karya Ilmiah ini menggunakan metode pengembangan perangkat lunak prototyping, maka didapatkan kesimpulan dalam poin-poin berikut :

1. Teknologi Firebase ML Kit dengan Tensorflow Lite dapat diandalkan dalam membangun aplikasi pengklasifikasi semantik teks. Namun model neural network yang didesain khusus memiliki kekurangan yakni data training yang terlalu sedikit. Sehingga nilai akurasi belum optimal.
2. Hasil pengujian prototype sesuai dengan analisis fungsional secara keseluruhan berhasil direalisasikan dari rancangan.

### V. SARAN

Adapun saran-saran dalam merancang aplikasi pengklasifikasi semantik teks kedepannya adalah :

1. Karena data yang dibutuhkan untuk membangun model *deep neural network* sangat banyak, disarankan untuk menggunakan data abstrak berupa *softcopy* dari *database* jurusan.
2. Untuk pengembangan model berikutnya, ada baiknya membangun spesifikasi model baru dengan fitur *drop-out* lebih kecil.
3. Pertimbangkan model dengan *input sequence* lebih besar dari 256 input, apabila beberapa abstrak/ringkasan karya tulis ilmiah yang digunakan sebagai data memiliki ketentuan maksimal diatas 256 kata.

## UCAPAN TERIMA KASIH

Terimakasih sebesar-besarnya saya ucapkan pada pihak jurusan Elektronika Universitas Negeri Padang yang memberikan saya kesempatan untuk melakukan studi di perpustakaan khususnya Ketua Jurusan. Terimakasih pula saya ucapkan kepada Pembimbing tugas akhir ini yang telah meluangkan waktu, tenaga dan pikiran demi terselesaikanya studi ini.

## DAFTAR PUSTAKA

- [1] N. Syam & A.Sharma, "Waiting for a sales renaissance in the fourth industrial revolution: Machine learning and artificial intelligence in sales research and practice". *Industrial Marketing Management*, vol. 69, pp. 135–146, 2018.
- [2] J. Hurwitz & Kirsch. D, *Machine Learning for Dummies. IBM Limited Edition*, John Wiley & Sons Inc, Hoboken, NJ, 2018.
- [3] O.Alsing, "Mobile Object Detection using TensorFlow Lite and Transfer Learning", *Degree Project, KTH Royal Institute of Technology*, Sweden, 2018.
- [4] A. Pradesh, "Language Identification Using ML Kit". *International Journal of Engineering Technology Research & Management*, vol. 4, no.13, pp. 106–107, 2020.
- [6] M. Shalahuddin & Rosa. A.S, *Media Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*, Modula, Bandung, 2011.
- [7] RS. Pressman, *Rekayasa Perangkat Lunak – Buku Satu Pendekatan Praktis Edisi 7*, Andi, Yogyakarta, 2012.
- [5] Sasmito, G. W. (2017). Penerapan Metode Waterfall Pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal. *Jurnal Informatika: Jurnal Pengembangan IT*, 2(1), 6-12.
- [8] Hidayanti, Luthfi Nur. "Pengembangan Sistem Informasi Akademik User Friendly Untuk Sma Muhammadiyah Kota Tegal (Siata)." *Media ElektriKa* 11.2 (2019): 65-74.
- [9] Novaliendry, Dony, and Viony DwiYana Puteri. "E-Retail Percetakan Anambaleh Desain Menggunakan Framework Laravel." *Jurnal Teknologi Informasi dan Pendidikan* 13.1 (2020): 10-19.
- [10] Dharwiyanti, S., & Wahono, R. S. (2003). Pengantar Unified Modeling Language (UML). *Ilmu Komputer*, 1-13.