



# Deteksi Intensi Chatbot Berbahasa Indonesia dengan Menggunakan Metode Capsule Network

Fatharani, Khoirunnisa Putri Kania, Jonner Hutahaean, Sri Ratna Wulan\*

Teknik Komputer dan Informatika, Politeknik Negeri Bandung, Bandung

Jl. Gegerkalong Hilir, Ciwaruga, Kec. Parongpong, Kabupaten Bandung Barat, Jawa Barat, Indonesia

Email: <sup>1</sup>fatharani.tif418@polban.ac.id, <sup>2</sup>khoirunnisa.putri.tif418@polban.ac.id, <sup>3</sup>jonnerh@jtk.polban.ac.id,

<sup>4,\*</sup>sri.ratna@polban.ac.id

Email Penulis Korespondensi: sri.ratna@polban.ac.id

Submitted: 06/07/2022; Accepted: 31/07/2022; Published: 31/07/2022

**Abstrak**—Intent detection merupakan proses mengklasifikasi intent pelanggan berdasarkan tulisan atau lisan yang digunakan. Salah satu pemanfaatan intent detection adalah dalam chatbot suatu organisasi. Dengan intent detection, chatbot mampu memahami maksud pengguna. Namun, saat ini penggunaan intent detection belum diterapkan oleh kebanyakan perusahaan di Indonesia. Metode intent detection yang baik untuk chatbot adalah yang mampu mengklasifikasikan intensi pengguna dengan tepat dan cepat. Penelitian ini bertujuan untuk melakukan intent detection dari pesan-pesan yang berasal dari dataset chatbot berbahasa Indonesia yang diperoleh dari percakapan pelanggan PT. Kazeer dengan menggunakan metode Capsule Network (CapsNet). Dengan adanya penelitian ini diharapkan chatbot PT. Kazeer dapat merespon pelanggan dengan lebih sesuai. Eksperimen dilakukan terhadap dataset percakapan berisi pertanyaan seputar PT. Kazeer. Dataset tersebut terdiri dari dataset dengan 6 intensi dan dataset dengan 18 intensi. Hasil eksperimen dibandingkan dengan model intent detection yang telah dimiliki sebelumnya, yakni intent detection dengan menggunakan metode BERT. Hasil eksperimen menunjukkan bahwa metode Capsule Network memiliki waktu eksekusi yang lebih cepat dibandingkan dengan metode BERT. Namun, untuk kemampuan merespon dengan tepat, BERT masih lebih unggul daripada Capsule Network. Oleh karena itu, metode ini dapat dipertimbangkan untuk digunakan pada chatbot yang lebih mementingkan kecepatan eksekusi.

**Kata Kunci:** Capsule Network; CapsNet; Chatbot; Intent Detection; BERT; Klasifikasi Teks

**Abstract**—Intent detection is a process of classifying customer intention from given sentence or chatting. One of the uses of intent detection is in a chatbot. With intent detection, the chatbot can detect the customer intent. However, currently the use of intent detection has not been implemented by most companies in Indonesia. A good intent detection method for chatbots is one that is able to classify user intentions accurately and quickly. This study aims to perform intent detection of messages from the Indonesian language chatbot dataset obtained from customer conversations of PT. Kazeer using the Capsule Network (CapsNet) method. With this research, it is hoped that the chatbot of PT. Kazeer can respond to customers more appropriately. On this study we conducted experiments and analyze the use of Capsule Network (CapsNet) method in detecting the intent of PT. Kazeer customers conversation. The dataset of the experiments contains questions about PT. Kazeer. There are two types of datasets—a dataset with six intentions and a dataset with 18 intentions. We compare the result of this experiment with the BERT intent detection model we used previously. The experiment show that the execution time of CapsNet method is faster than that of BERT. However, BERT is still superior to the CapsNet for the ability to respond appropriately. The CapsNet method can be considered for use on chatbots that are more concerned with execution speed.

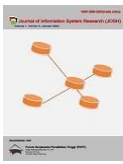
**Keywords:** Capsule Network; CapsNet; Chatbot; Intent Detection; BERT; Text Classification

## 1. PENDAHULUAN

*Chatbot* adalah program yang dibuat menggunakan teknologi Artificial Intelligence (AI) untuk meniru percakapan manusia. *Chatbot* memungkinkan mesin dan manusia berinteraksi menggunakan bahasa manusia. *Chatbot* dirancang untuk menjadi asisten virtual, sarana hiburan, membantu seseorang menyelesaikan tugas mulai dari menjawab pertanyaan, mendapatkan petunjuk arah mengemudi, menyalakan termostat di rumah pintar [1], hingga memainkan lagu favorit. *Chatbot* sangat bermanfaat pada beberapa aplikasi seperti aplikasi pendidikan, layanan informasi, bisnis dan perdagangan. [2] *Chatbot* populer dalam bisnis karena layanan *customer service* menjadi lebih murah [2]. Selain itu, *chatbot* juga bermanfaat untuk menghemat waktu karena dapat diakses di mana pun dan kapan pun [3]. Sehingga, *chatbot* yang dikembangkan harus memiliki kemampuan merespon dengan cepat [4].

Untuk membangun sebuah *chatbot* atau sistem percakapan, diperlukan model Natural Language Understanding (NLU) yang baik. NLU merupakan komponen paling penting agar mesin dapat memahami pesan pengguna. Namun, dalam pembangunan sistem percakapan, komponen ini yang paling sulit untuk dibangun [4] Agar mesin dapat memahami bahasa manusia, diterapkan metode *intent detection* terhadap teks yang dikirim pengguna. Dengan adanya *intent detection* ini, *chatbot* akan mengklasifikasikan teks yang dikirim pengguna berdasarkan intensi yang tersedia. Sehingga *chatbot* dapat menjawab pesan yang diketik pengguna berdasarkan klasifikasi intensinya [5]. Namun, beberapa perusahaan di Indonesia masih banyak menggunakan *chatbot* yang belum optimal dalam menerapkan *intent detection*. Hal ini dapat disebabkan oleh berbagai alasan. Diantaranya adalah *dataset* yang kurang baik dan metodenya yang kurang baik.

Penerapan *intent detection* dapat dilakukan dengan berbagai metode. Namun karena mendeteksi intensi dalam *chatbot* sulit, metode machine learning tradisional mulai ditinggalkan [6]. Metode tradisional yang dimaksud diantaranya adalah Naive Bayes, Adaboost, Support Vector Machine (SVM) dan Logistic Regression.



Metode tradisional ini tidak dapat memahami informasi dalam teks secara mendalam. Sehingga sulit untuk mengetahui intensi pengguna dengan akurat [5]. Metode yang saat ini sering digunakan diantaranya adalah Convolution Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM) Network, Gated Recurrent Unit (GRU), Attention Mechanism dan Capsule Network (CapsNets) [5][7]. Diantara metode tersebut, CapsNets memberikan hasil terbaik ketika dilatih menggunakan *dataset* teks berbahasa Inggris [5], [7]. CapsNet adalah sebuah model yang terdiri dari beberapa kapsul. Kapsul tersebut terbentuk dari kumpulan neuron yang akan memproses informasi yang diberikan. Sehingga dapat diketahui intensi dari sebuah teks berdasarkan perhitungan yang telah dilakukan.

Pada awalnya CapsNet diciptakan untuk mengatasi kelemahan dalam metode CNN yang kehilangan banyak informasi. CNN populer digunakan untuk mengolah objek gambar 2 dimensi. Oleh karena itu, CapsNet pun pada awalnya digunakan untuk mengolah objek gambar 2 dimensi. Namun penelitian [5], [7]–[10] menyatakan bahwa CapsNet dapat digunakan untuk mengolah *dataset* berupa teks. Bahkan ketika diberikan *dataset* multi label metode CapsNet masih memberikan hasil yang baik [5]. Selain itu, ada penelitian [8] yang memiliki tujuan untuk menyajikan ulasan tentang klasifikasi teks menggunakan CapsNet. Hasil dari penelitian ini adalah CapsNet telah mengungguli CNN dalam klasifikasi teks mencapai akurasi 87,17%. CapsNet dapat diterapkan dalam klasifikasi teks hierarkis agar hasilnya lebih baik. Sedangkan pada penelitian yang dilakukan oleh [9], dikatakan bahwa dengan menerapkan metode CapsNet pada metode transformer dapat meningkatkan nilai akurasi. Sehingga, dapat disimpulkan CapsNet memberikan hasil terbaik karena dapat memahami semantik yang dalam. Selain itu, CapsNet juga memastikan lokasi dari informasi semantik intensi tersebut [5]. Oleh karena itu, pada penelitian ini akan dilakukan eksperimen menggunakan metode CapsNet.

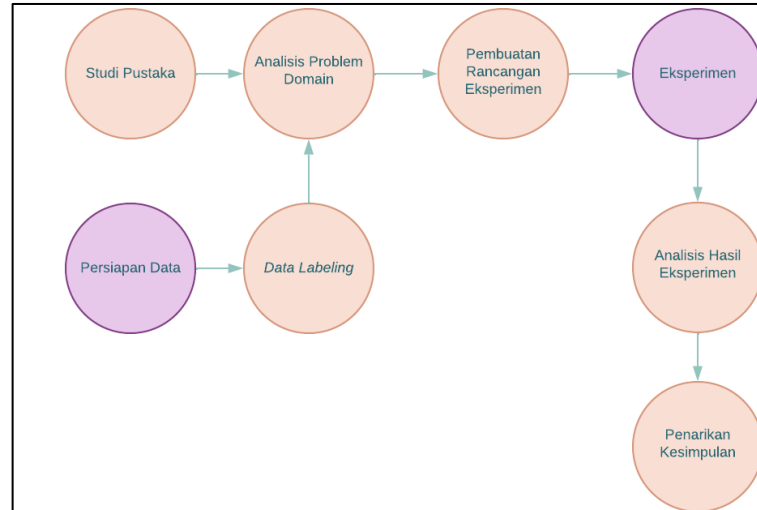
Eksperimen ini bertujuan untuk melakukan *intent detection* dari pesan-pesan yang berasal dari *dataset chatbot* berbahasa Indonesia. Eksperimen ini menggunakan *dataset* yang memiliki multi label dan berbahasa Indonesia. *Dataset* multi label merupakan *dataset* yang didalamnya terdapat kumpulan teks. Setiap teks dalam *dataset* multi label dapat memiliki lebih dari satu label atau intensi. Pada eksperimen ini terdapat dua macam *dataset*. *Dataset* pertama memiliki 6 intensi. *Dataset* kedua memiliki 18 intensi.

Penelitian ini berfokus pada hasil training model *intent detection*. Model dikatakan bagus dari score *accuracy*, *f1*, *recall*, dan presisi yang tinggi [11]. Selain itu kalimat yang diklasifikasikan oleh model harus memuat intensi yang sesuai. Model juga harus memenuhi kondisi *good fit* [12]. Selain dari kemampuannya melakukan pengklasifikasian dengan tepat, model juga harus memiliki kecepatan pemrosesan yang tinggi. Fokus utama dalam penelitian ini adalah meneliti model CapsNet untuk *intent detection*. Evaluasi model CapsNet ini dibandingkan dengan model BERT. Perbandingan dilakukan dengan metode BERT karena BERT telah menjadi salah satu *state-of-the-art* yang populer beberapa tahun belakangan ini. Sehingga metode BERT sudah diketahui dapat melakukan tugas *Natural Language Processing* seperti pengklasifikasian teks dengan baik [13]. Dengan adanya penelitian ini, diharapkan diketahui kinerja model CapsNet memiliki hasil yang sama baiknya atau lebih baik daripada BERT. Sehingga model CapsNet dapat berguna untuk keperluan teks klasifikasi pada *chatbot* dan lainnya.

## 2. METODOLOGI PENELITIAN

### 2.1 Tahapan Penelitian

Alur tahapan dari penelitian ini dapat dilihat pada Gambar 2. Secara umum, ada 3 bagian besar yang dilakukan: persiapan, eksperimen, dan penarikan kesimpulan. Pada bagian pertama, yakni persiapan, studi pustaka dilakukan terlebih dahulu. Pada tahapan studi pustaka akan dicari beberapa sumber literatur. Literatur yang dicari mengenai *intent detection* dan *metric* evaluasi model. Tahapan ini menghasilkan pengetahuan tentang proses kerja suatu model pendeteksi intensi. Selanjutnya dilakukan persiapan data. Pada tahapan ini data yang dibutuhkan untuk penelitian dikumpulkan dan diolah. Data tersebut dilabeli. Proses *data labeling* yang dilakukan adalah membuat *dataset* baru berjumlah 18 intensi. Tiap kalimat yang ada pada *dataset* akan diberi intensi secara manual. Jika telah selesai, maka dapat dilakukan analisis problem domain. Tahap analisis problem domain bertujuan mempelajari lebih dalam tentang metode yang digunakan pada penelitian ini. Selanjutnya, dilakukan perancangan eksperimen. Pada bagian ini rancangan eksperimen akan dipaparkan dengan jelas. Rancangan eksperimen meliputi tujuan, bahan, alat, perlengkapan, dan skenario eksperimen. Skenario eksperimen yang dibuat menyesuaikan dengan variable eksperimen yang akan diatur atau variabel bebas. Untuk lebih jelas terkait variabel penelitian, dapat dilihat pada subbab 2.3. Sedangkan skenario eksperimen dijelaskan lebih lanjut pada subbab 2.4. Pada bagian selanjutnya, eksperimen dilakukan. Pada tahap ini model CapsNet akan dibuat dan digunakan. Sebelum membuat model dilakukan *preprocessing data*. Lalu dilakukan *training* dengan metode CapsNet sehingga terbentuk model *intent detection* dengan metode CapsNet. Untuk lebih jelas tentang cara kerja CapsNet dapat dilihat pada subbab 2.2. Setelah model terbentuk, dilakukan pengujian terhadap model tersebut. Setelah itu, analisis hasil eksperimen dilakukan. Performa model akan dilihat dari akurasi, presisi, *recall*, dan *f1-score*. Pada bagian ketiga, penarikan kesimpulan dilakukan untuk mengetahui apakah rumusan masalah dari penelitian ini terjawab atau tidak.

**Gambar 1.** Tahapan penelitian

## 2.2 Capsule Network

Pengklasifikasian teks telah banyak menggunakan model *machine learning* salah satunya menggunakan CNN. Awalnya, CNN dibangun sebagai sistem *check-recognition* untuk membaca digit tulisan tangan. Namun, kemudian CNN berfungsi dengan baik untuk *computer vision* seperti deteksi, pelacakan dan pengenalan objek.

CNN adalah metode yang paling umum digunakan untuk klasifikasi objek dua dimensi. CNN pada awalnya dibuat untuk mengklasifikasi gambar. Namun, informasi seperti posisi dan arah dalam objek dibuang oleh CNN karena prosedur routing datanya. Saat melakukan pooling, CNN cenderung kehilangan informasi yang berguna. Informasi ini hilang saat melakukan tugas seperti segmentasi gambar dan deteksi objek. Pooling layer dapat kehilangan informasi berupa rotasi, lokasi, skala, dan atribut posisi objek yang berbeda. Dampak dari kehilangan informasi ini adalah proses deteksi dan segmentasi objek menjadi sangat sulit. Maka dari itu diusulkan metode baru yaitu Capsule Network atau CapsNet.

CapsNet adalah model yang terdiri dari kapsul. Sebuah kapsul terdiri dari sekelompok neuron. [14] Output dari CapsNet merepresentasikan properti yang berbeda dari sebuah entitas yang sama [15]. Awalnya, CapsNet diperkenalkan untuk pengenalan gambar. Namun, dari keberhasilan penggunaan CNN di *Natural Language Processing* (NLP) dapat disimpulkan bahwa *neural network* apa pun dapat menyelesaikan masalah di NLP.

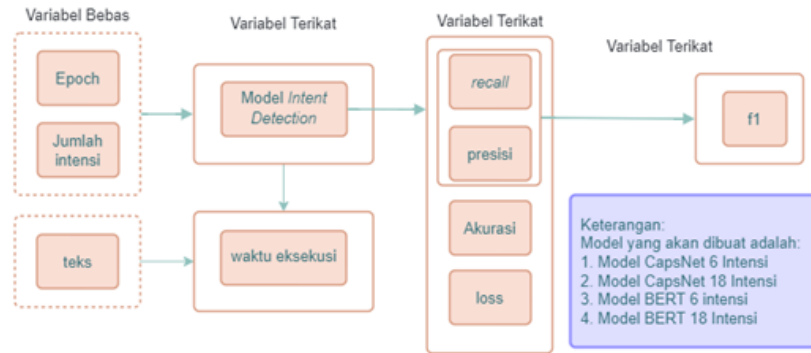
Arsitektur *capsule network* pada dasarnya terbagi menjadi 2, yaitu *encoder* dan *decoder*. Layer yang termasuk kedalam *encoder* yaitu *convolutional layer*, *primary layer*, dan *Convolutional Capsule Layer*. Layer pada *decoder* adalah *fully connected layer* [10]. Proses yang dilakukan pada *Convolutional Layer* adalah konvolusi. Konvolusi akan memperkecil jumlah vektor yang dimiliki. Pada *primary layer* ada pembentukan kapsul pada level bawah dan *squash function*. *Squash function* adalah proses perhitungan probabilitas tiap kata. Hasil *squash function* berada dari nilai 0 sampai 1. Semakin mendekati 1 probabilitas suatu kapsul menjadi *parent* dari kapsul di bawahnya semakin tinggi. Sebaliknya, jika mendekati 0 probabilitasnya menjadi lebih kecil. Selanjutnya ada *fully connected layer*. Layer ini berfungsi untuk *input* data yang merupakan gambar, untuk *text classification* layer ini dapat dihilangkan [10]

Pada *capsule network* terdapat proses *dynamic routing*. Tujuan dilakukannya *dynamic routing* adalah untuk menentukan induk suatu kata. Pada *Dynamic routing* dilakukan *mapping capsule* pada layer  $i$  terhadap layer  $i+1$ . Ide dasar dari *dynamic routing* adalah untuk membangun peta non-linier dengan cara yang berulang. Pemetaan ini memastikan bahwa *output* dari setiap kapsul akan dikirim ke induk yang sesuai di lapisan berikutnya. Untuk setiap calon induk atau label terdapat nilai *connection strength*. Dalam proses *dynamic routing* ini nilai *connection strength* akan dinaikan atau diturunkan. Ide *dynamic routing* ini dianggap lebih efektif daripada *max pooling*. Karena ketika *pooling* dilakukan, tidak semua data akan diambil. Jadi digunakan kapsul untuk bisa menyimpan informasi yang lebih detail dengan ukuran matriks yang lebih kecil. Setelah diketahui nilai *connection strength* terhadap setiap induk, induk yang memiliki nilai *connection strength* melebihi nilai *threshold* yang ditentukan, yaitu 0.5 akan dianggap sebagai label dari teks yang diberikan [10].

## 2.3 Variabel Eksperimen

Variabel eksperimen dapat dilihat pada Gambar 2. Terdapat dua jenis variabel yang akan menjadi fokus pengamatan pada penelitian ini. Variabel tersebut yaitu variabel bebas (*independen*) dan variabel terikat (*dependen*). Variabel bebas adalah variabel yang dapat dikendalikan nilainya dan akan mempengaruhi nilai dari variabel terikat [16]. Pada penelitian ini yang termasuk sebagai variabel bebas adalah jumlah intensi dan *epoch*. Variabel terikat adalah variabel yang nilainya dipengaruhi oleh nilai variabel bebas [16]. Pada penelitian ini yang merupakan variabel terikat adalah akurasi, f1, recall, dan presisi. Nilai *epoch* dan jumlah intensi mempengaruhi

model *intent detection*. Hasil dari pembelajaran model mempengaruhi nilai akurasi, recall, akurasi, dan presisi. Nilai presisi dan recall memengaruhi nilai f1. Hasil prediksi dari proses proses tersebut memengaruhi jenis intensi yang dihasilkan oleh suatu kalimat. Selain itu, pada penelitian ini juga akan dilihat kondisi perbandingan nilai *loss* hasil pelatihan model tersebut. Nilai *loss* didapatkan dari hasil prediksi model. Selain itu, terdapat beberapa teks yang akan diujikan sehingga teks termasuk sebagai variabel bebas. Teks yang diujikan pada sebuah model ini akan mempengaruhi waktu eksekusi.



**Gambar 2.** Variabel penelitian

Untuk parameter penelitian, akan mengikuti penelitian [10]. Nilai parameter tidak akan berubah, sehingga nilai parameter tidak termasuk variabel.

### 2.4 Skenario Eksperimen

Skenario eksperimen yang akan dilakukan akan mengikuti variabel yang telah dituliskan pada sub-bab sebelumnya. Salah satu proses yang akan dilakukan dalam eksperimen adalah melakukan *training dataset*. *Dataset* tidak dilatih sekali saja, seberapa banyaknya *training* yang dilakukan oleh *dataset* disebut *epoch*. Jadi, *epoch* adalah ketika seluruh *dataset* sudah melalui proses *training* pada *Neural Network* sampai dikembalikan ke awal untuk sekali putaran. Jumlah *epoch* tidak dapat ditentukan. Jumlah *epoch* berbeda untuk *dataset* yang berbeda. Jadi, dapat disimpulkan bahwa *epoch* tergantung dengan *dataset* yang dimiliki. [17]

Terdapat empat skenario eksperimen. Empat eksperimen ini dikelompokkan menjadi eksperimen A dan eksperimen B. Eksperimen A akan dilakukan menggunakan metode BERT. Eksperimen A.1 menggunakan metode BERT dan dilakukan menggunakan *dataset* dengan enam intensi. Sedangkan eksperimen A.2 menggunakan metode BERT dan dilakukan menggunakan *dataset* dengan delapan belas intensi. Eksperimen B akan dilakukan menggunakan metode CapsNet. Eksperimen B.1 menggunakan metode CapsNet dan dilakukan menggunakan *dataset* dengan enam intensi. Sedangkan eksperimen B.2 menggunakan metode CapsNet dan dilakukan menggunakan *dataset* dengan delapan belas intensi.

## 3. HASIL DAN PEMBAHASAN

### 3.1 Eksperimen Capsule Network

Agar lebih jelas memahami tahap-tahap tersebut, bagian ini akan digambarkan ilustrasi alur proses pengklasifikasian teks. Tahapan dalam eksperimen tersebut adalah sebagai berikut:

**a. Menyiapkan Persyaratan**

Persyaratan yang dimaksud adalah modul yang perlu diunduh dan dipasang terlebih dahulu. Persyaratan yang dibutuhkan untuk membuat model CapsNet adalah tensorflow versi 1.15, keras versi 2.3.0, h5py versi 2.10.0, Sastrawi. Selain itu, diaktifkan juga cuda agar dapat melakukan komputasi secara paralel. Penggunaan cuda membutuhkan GPU. Namun, GPU ini hanya digunakan pada proses training BERT.

**b. Load Data**

*Load data* merupakan sebuah proses memuat data agar dapat diolah di google colab. Data tersebut telah dibagi-bagi ke dalam data *training* dan *testing*.

**c. Preprocessing**

Setelah *dataset* dimuat dalam *dataframe*, dilakukan *preprocessing* terhadap *dataset* tersebut. *Preprocessing* ini terdiri dari *case folding*, *remove punctuation*, *stopword removal*, tokenisasi, *word indexing*, dan *padding*.

**d. Embedding**

*Embedding* adalah proses mengubah kata dari bahasa manusia menjadi bahasa yang dimengerti mesin. Pada proses *embedding* kata yang sama akan memiliki representasi nilai yang sama.

**e. Load pre-trained model BERT**

BERT sudah memiliki model *pre-train*. Model *pre-train* adalah model yang sudah terlatih. Entah itu dilatih oleh orang lain, atau dilakukan secara mandiri terlebih dahulu. Model *pre-train* dilatih menggunakan data yang

besar. Hal ini dapat bermanfaat pada pekerjaan deep learning karena untuk beberapa hal tidak perlu dilakukan training dari awal [18]. Model *pre-trained* yang digunakan adalah *cahya/bert-base-indonesian-522M*.

**f. Training**

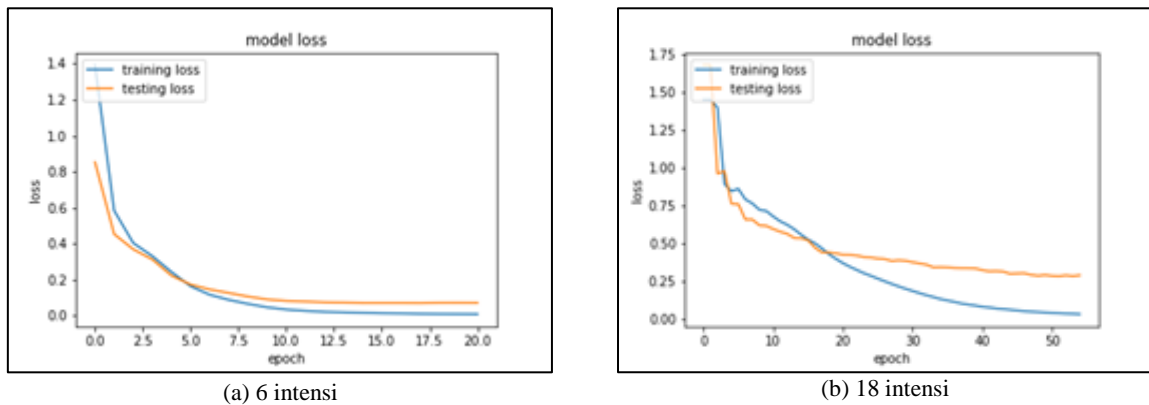
Setelah dilakukan *embedding*, data akan di *train*. *Training* akan dilakukan dalam beberapa *epoch*. Jumlah *epoch* ini ditentukan oleh fungsi *early stop* yang disediakan oleh *library* Keras. Jika dalam tiga *epoch* berturut-turut tidak ada penurunan nilai *testing loss* maka proses *training* dihentikan.

**g. Evaluasi dan Prediksi**

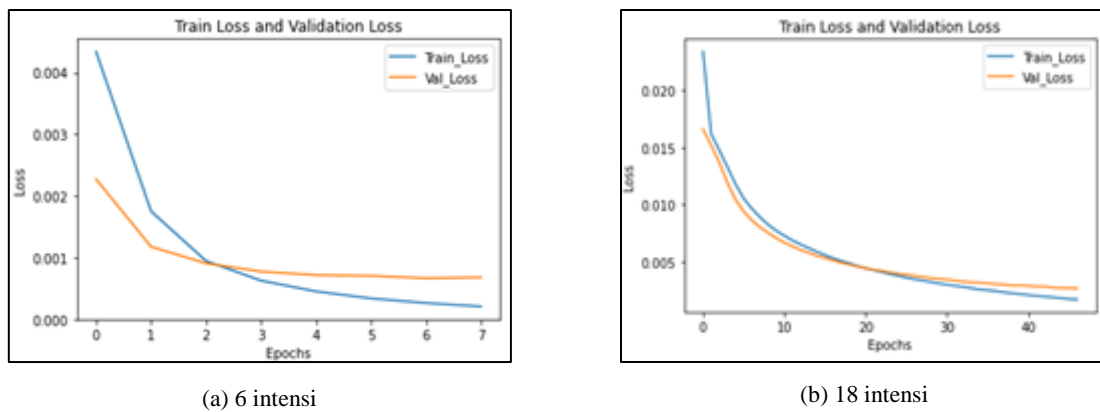
Untuk memastikan performa dari *capsule network* dilakukan perbandingan dengan metode *intent detection* lainnya yaitu BERT. Hal-hal yang akan dibandingkan adalah nilai akurasi, presisi, *recall* dan *f1*. Masing-masing metode dibandingkan dengan kondisi jumlah intensi yang sama. Selain itu akan dilihat juga grafik perbandingan *loss* untuk data *training* dan data validasi dari seluruh skenario. Model yang telah dibuat juga akan diuji untuk memprediksi enam belas kata. 16 kata yang digunakan telah mewakili seluruh intensi yang terdapat di *dataset*.

**3.2 Grafik Perbandingan Loss**

Hasil eksperimen pertama adalah perbandingan grafik *loss*. *X-axis* menandakan jumlah *epoch* yang dilakukan ketika *training*. Sedangkan *y-axis* menandakan nilai *loss* pada setiap *epoch*. Berdasarkan grafik, kedua skenario CapsNet menandakan kondisi *good fitting* karena kedua garisnya berhimpit.



**Gambar 3.** Grafik perbandingan loss CapsNet



**Gambar 4.** Grafik perbandingan loss BERT

Dilihat dari Gambar 3 dan Gambar 4, seluruh grafik menunjukkan jarak nilai *training loss* dan *validation loss* yang dekat. Hasil ini dapat diperoleh karena menggunakan fungsi *early stop* ketika *training*. Hal ini dapat dilihat dari kerapatan antar garis *loss* dan jumlah *epoch*. Kerapatan yang dilakukan untuk setiap skenario itu berbeda. *Training* akan dihentikan ketika *testing loss* atau *validation loss* tidak mengalami penurunan dalam tiga *epoch*. Jika dilihat dari masing-masing grafik pada Gambar 4 dan Gambar 5, nilai *loss* pada beberapa *epoch* pertama mengalami penurunan drastis. Sedangkan nilai *loss* dalam beberapa *epoch* terakhir hanya mengalami sedikit penurunan. Bahkan di tiga *epoch* terakhir tidak ada penurunan. Kondisi model saat proses *training* dihentikan berada pada kondisi dengan nilai *loss* yang cukup stabil. Sehingga kondisi ini dinyatakan *good fitting*. Sehingga semua skenario seharusnya dapat mempelajari *dataset* dengan baik. Namun berdasarkan hasil pengujian teks ternyata masih didapatkan beberapa kesalahan dalam melakukan klasifikasi. Hal ini akan dijelaskan lebih lanjut pada bab berikutnya.



### 3.3 Analisis Nilai Confusion Matrix

Nilai *loss* memperhatikan jarak antara hasil prediksi dan nilai sebenarnya. Sedangkan nilai *confusion matrix* akan memperhatikan jumlah banyaknya data yang tepat prediksi dan salah prediksi. Nilai akurasi, presisi, recall dan f1 didapatkan dari sebuah *confusion matrix*. Pada *confusion matrix* terdapat *true positive*, *true negative*, *false positive*, *false negative*. *True positive* terjadi jika label yang terdapat dalam kalimat berhasil dideteksi sebagai label kalimat tersebut. *True negative* terjadi jika label yang tidak terdapat dalam kalimat tidak dideteksi sebagai label kalimat. *False positive* terjadi jika label yang dideteksi model sebagai label kalimat tersebut bukan label sebenarnya. *False negative* terjadi jika model tidak mendeteksi suatu label sebagai label kalimat tertentu. Untuk lebih jelas, dapat dilihat pada Tabel 1.

Tabel 1. Nilai hasil evaluasi

Komponen	Capsule Network		BERT	
	6 intensi	18 intensi	6 intensi	18 intensi
Akurasi	0.7071	0.7445	0.0653	0.066
Presisi	0.9948	0.9812	0.1168	0.0823
Recall	0.9973	0.9812	0.0433	0.0226
F1	0.9960	0.9811	0.0556	0.0245
loss	0.007	0.0291	0.0059	0.0027

Nilai akurasi pada model CapsNet berkisar pada 0.7. Hal ini menandakan bahwa hanya ada sekitar 70% label yang merupakan hasil prediksi yang benar. Nilai akurasi CapsNet 18 intensi lebih tinggi daripada CapsNet 6 intensi. Hal ini menandakan ada lebih banyak label yang tepat prediksi pada model CapsNet 18 intensi. Sedangkan BERT mendapatkan hasil yang baik yaitu 0.9792 untuk model enam intensi. Untuk model 18 intensinya, BERT memiliki nilai sebesar 0.9666. Hal ini menandakan bahwa pada model BERT 6 intensi lebih banyak label yang tepat prediksi.

Pada *dataset* multilabel, suatu kalimat dapat memiliki lebih dari satu intensi. Misalnya suatu kata memiliki tiga intensi, namun model memprediksi terdapat empat intensi. Dari keempat intensi tersebut, satu diantaranya benar sedangkan sisanya salah. Untuk mengetahui seberapa banyak label yang benar dari hasil prediksi, digunakan presisi. Pada model CapsNet memiliki presisi yang cukup tinggi, nilainya mendekati 1. Semakin besar nilai presisi maka semakin sedikit nilai *false positive*. Jadi hanya ada sedikit label yang diprediksi ada pada kalimat namun sebenarnya tidak ada. Perbedaan nilai presisi model CapsNet 6 intensi dan model CapsNet 18 intensi tidak begitu jauh. Namun, model CapsNet 6 intensi memiliki nilai lebih tinggi, berarti lebih sedikit nilai *false positive*. Nilai presisi pada BERT mendekati 1. Nilai BERT enam intensi sedikit lebih tinggi daripada BERT delapan belas intensi. Sehingga pada BERT enam intensi kemungkinan ada lebih sedikit kasus *false positive*.

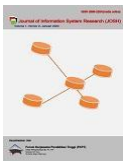
*Recall* digunakan untuk mencari tahu ada berapa banyak label yang tidak terdapat dalam hasil prediksi. Model CapsNet memiliki nilai *recall* yang cukup tinggi. Model CapsNet dengan enam intensi memiliki nilai *recall* yang sedikit lebih tinggi. Namun, CapsNet 18 intensi maupun enam intensi, memiliki nilai *recall* yang bagus, yaitu mendekati satu. Hal ini berarti nilai *false negative* pada model ini sangat rendah. Berarti hanya ada sedikit label yang tidak terdapat dalam hasil prediksi. Model BERT dengan enam intensi memiliki nilai *recall* yang sedikit lebih tinggi. Model BERT 18 intensi juga memiliki nilai *recall* yang bagus, yaitu mendekati satu.

Untuk membandingkan rata-rata nilai precision dan *recall* digunakan metric F1. Berdasarkan presisi dan *recall* model dengan baik CapsNet enam intensi maupun 18 intensi, didapatkan nilai F1 yang baik, yaitu mendekati 1. Berarti model memiliki jumlah kesalahan prediksi yang kecil karena *false negative* dan *false positif*nya kecil. Pada BERT enam intensi maupun 18 intensi, didapatkan nilai F1 yang baik, yaitu mendekati 1. Berarti model memiliki jumlah kesalahan prediksi yang kecil juga.

Pada subbab ini disimpulkan bahwa secara keseluruhan dapat dinyatakan model dengan enam intensi memberikan hasil yang lebih baik. Namun perbedaan tersebut tidak terlalu signifikan. Sehingga model yang dilatih dengan *dataset* delapan belas intensi juga memiliki hasil yang baik. Terakhir, model BERT memiliki nilai kinerja yang lebih tinggi daripada CapsNet. Oleh karena itu, BERT dinyatakan lebih baik daripada CapsNet.

### 3.4 Analisis Waktu Eksekusi

Pengujian ini dilakukan untuk mengetahui kecepatan model untuk melakukan klasifikasi. Pengujian dilakukan terhadap enam belas teks yang sudah disiapkan dan dicatat dalam satuan detik. Hasil rata-rata pengujian kecepatan pengklasifikasian ini tercatat dalam Tabel 2. Hasil eksperimen menunjukkan metode CapsNet memiliki waktu eksekusi yang lebih cepat dari BERT. Hal ini dapat terjadi karena metode BERT memiliki arsitektur yang lebih rumit daripada CapsNet. Sedangkan kecepatan antara metode yang sama dengan jumlah intensi yang berbeda tidak tetap. Namun, pada model CapsNet jika dilihat dari rata-rata waktu pemrosesan, model enam intensi memiliki waktu eksekusi lebih singkat pada. Sedangkan pada model BERT, model dengan 18 intensi memiliki waktu eksekusi yang lebih singkat. Namun perbedaan waktu eksekusi model BERT 6 intensi dan BERT 18 intensi tidak begitu jauh.



Tabel 2. Perbandingan waktu eksekusi

Waktu eksekusi (dalam detik)			
BERT 6 Intensi	CapsNet 6 Intensi	BERT 18 Intensi	CapsNet 18 Intensi
0,00506	0,01660	0,00664	0,01645

#### 4. KESIMPULAN

*Intent detection* adalah salah satu bagian Natural Language Processing (NLP). *Intent detection* adalah tugas mengambil masukan dan mengklasifikasikannya berdasarkan apa yang ingin dicapai pengguna. Penelitian ini menggunakan metode BERT dan Capsule Network (CapsNet) untuk mendapatkan *intent detection* dari *chatbot* berbahasa Indonesia. Skenario eksperimen yang dilakukan adalah membuat model CapsNet dengan *dataset* berisi pertanyaan. *Dataset* yang ada memiliki 6 dan 18 intensi. Model yang dihasilkan akan dilihat nilai akurasi, presisi, *recall*, *f1*, dan *loss* nya. Selain *metric* tersebut, dilakukan juga perhitungan waktu untuk menghasilkan *output* ketika program berjalan. *Output* yang dimaksud adalah intensi dari kalimat yang menjadi *input*. Hasil dari eksperimen menggunakan CapsNet dan BERT dibandingkan. Berdasarkan hasil eksperimen dapat disimpulkan bahwa penggunaan CapsNet berhasil memberikan nilai akurasi 70%. Namun, nilai *loss* yang dihasilkan kecil. Jadi, dapat dikatakan bahwa *capsule network* dapat mendeteksi intensi dengan baik. Metode CapsNet juga menghasilkan nilai presisi, *f1*, dan *recall* yang baik yaitu sekitar 90%. Grafik model yang dihasilkan pun sudah menunjukkan kondisi *good fit*. Hasil model CapsNet 6 intensi sudah cukup sebaik BERT. Namun model CapsNet mengalami penurunan nilai akurasi untuk 18 intensi. Hasil analisis menunjukkan CapsNet lebih cepat memberikan *output* daripada BERT. Karena BERT memiliki arsitektur yang lebih rumit. Jika metode ini digunakan dalam *chatbot*, *chabot* mampu merespon pengguna. Namun, respon yang diberikan masih banyak yang tidak sesuai. Untuk *chatbot* yang bersifat *task based* model BERT masih lebih bagus digunakan daripada model CapsNet. Namun, jika dilihat dari segi waktu, CapsNet lebih unggul.

#### REFERENCES

- [1] E. Adamopoulou and L. Moussiades, “An Overview of Chatbot Technology,” in *IFIP Advances in Information and Communication Technology*, 2020, vol. 584 IFIP, pp. 373–383. doi: 10.1007/978-3-030-49186-4\_31.
- [2] R. B. R., R. N., and S. S., “Chatbot for university related FAQs,” Sep. 2017.
- [3] P. N. P., P. D. R., P. D. A., and P. R. R., “AI and Web-Based Human-Like Interactive University Chatbot (UNIBOT),” Jun. 2019.
- [4] Y. Wang, L. Tang, and T. He, “Attention-based CNN-BLSTM networks for joint intent detection and slot filling,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, vol. 11221 LNAI, pp. 250–261. doi: 10.1007/978-3-030-01716-3\_21.
- [5] J. Liu, Y. Li, and M. Lin, “Review of Intent Detection Methods in the Human-Machine Dialogue System,” in *Journal of Physics: Conference Series*, Jul. 2019, vol. 1267, no. 1. doi: 10.1088/1742-6596/1267/1/012059.
- [6] S. M. Y. H., A. R. M., and E. S. R., *Chatbot, Comparison Of Multinomial Naive Bayes Algorithm And Logistic Regression For Intent Classification In Chatbot*. Proceedings of the 2018 International Conference on Applied Engineering (ICAIE): Batam, Indonesia, October 3-4, 2018, 2018.
- [7] W. Zhao, J. Ye, M. Yang, Z. Lei, S. Zhang, and Z. Zhao, “Investigating Capsule Networks with Dynamic Routing for Text Classification,” Mar. 2018, [Online]. Available: <http://arxiv.org/abs/1804.00538>
- [8] K. R. and A. Y., “Study on Text Classification using Capsule Networks,” 2019.
- [9] B. Chen, Z. Xu, X. Wang, L. Xu, and W. Zhang, “Capsule Network-Based Text Sentiment Classification,” in *IFAC-PapersOnLine*, 2020, vol. 53, no. 5, pp. 698–703. doi: 10.1016/j.ifacol.2021.04.160.
- [10] A. Khikmatullaev, J. Lehmann, K. Singh, C. Gmbh, and Z. Research, “Capsule Neural Networks for Text Classification Boosting Named Entity Recognition (NER) Systems View project SANSA-Stack View project”, doi: 10.13140/RG.2.2.12359.24488.
- [11] H. N. B., “Confusion Matrix, Accuracy, Precision, Recall, F1 Score,” Dec. 11, 2019.
- [12] J. Brownlee, “How to Diagnose Overfitting and Underfitting of LSTM Models,” Sep. 01, 2017.
- [13] S. González-Carvajal and E. C. Garrido-Merchán, “Comparing BERT against traditional machine learning text classification,” May 2020, [Online]. Available: <http://arxiv.org/abs/2005.13012>
- [14] S. Sabour, N. Frosst, and G. E. Hinton, “Dynamic Routing Between Capsules.”
- [15] J. Tao, X. Zhang, X. Luo, Y. Wang, C. Song, and Y. Sun, “Adaptive Capsule Network,” *Computer Vision and Image Understanding*, vol. 218, p. 103405, Apr. 2022, doi: 10.1016/J.CVIU.2022.103405.
- [16] Muh. Fitriah and Luthfiyah, *Metodologi penelitian: penelitian kualitatif, tindakan kelas & studi kasus*, 1st ed., vol. 1. Sukabumi: CV Jejak, 2017.
- [17] I. Digmi, “Memahami Epoch Batch Size Dan Iteration - JournalToday,” Jan. 25, 2018. <https://imam.digmi.id/post/memahami-epoch-batch-size-dan-iteration/> (accessed Jul. 04, 2022).
- [18] X. P. Qiu, T. X. Sun, Y. G. Xu, Y. F. Shao, N. Dai, and X. J. Huang, “Pre-trained models for natural language processing: A survey,” *Science China Technological Sciences*, vol. 63, no. 10. Springer Verlag, pp. 1872–1897, Oct. 01, 2020. doi: 10.1007/s11431-020-1647-3.