

**SMS SPAM DETECTION MENGGUNAKAN METODE LONG SHORT TERM
MEMORY**

SKRIPSI

**Oleh :
AINAFATUL NUR MUSLIKAH
NIM. 17650067**



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2021**

SMS SPAM DETECTION MENGGUNAKAN METODE *LONG SHORT TERM MEMORY*

SKRIPSI

**Diajukan kepada:
Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh:
AINAFATUL NUR MUSLIKAH
NIM. 17650067**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2021**

HALAMAN PERSETUJUAN

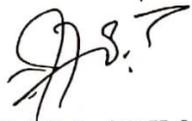
**SMS SPAM DETECTION MENGGUNAKAN METODE LONG SHORT
TERM MEMORY**

SKRIPSI

Oleh:
AINAFATUL NUR MUSLIKAH
NIM. 17650067

Telah Diperiksa dan Disetujui untuk Diuji
Tanggal : 14 Desember 2021

Dosen Pembimbing I



Khadijah Fahmi H.H, M.Kom
NIDT. 19900626 20160801 2 077

Dosen Pembimbing II



Dr. Cahyo Crysdiyan
NIP. 19740424 200901 1 008

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Dr. Fachrul Kurniawan, M.MT
NIP. 19771020 200912 1 001

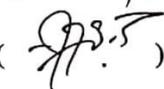
HALAMAN PENGESAHAN

**SMS SPAM DETECTION MENGGUNAKAN METODE LONG SHORT
TERM MEMORY**

SKRIPSI

Oleh:
AINAFATUL NUR MUSLIKAH
NIM. 17650067

Telah Dipertahankan di Depan Dewan Penguji
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Pada Tanggal: 22 Desember 2021

Penguji Utama	: Susunan Dewan Penguji Ajib Hanani, M.T NIDT. 19840731 20160801 1 076	
Ketua Penguji	: Johan Ericka Wahyu Prakasa, M.Kom NIP. 19831213 201903 1 004	()
Sekretaris Penguji	: Khadijah Fahmi H.H, M.Kom NIDT.19900626 20160801 2 077	()
Anggota Penguji	: Dr. Cahyo Crysdian NIP. 19740424 200901 1 008	()

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Dr. Fachrul Kurniawan, M.MT
NIP. 19771020 200912 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini :

Nama : Ainafatul Nur Muslikah
NIM : 17650067
Fakultas : Sains dan Teknologi
Jurusan : Teknik Informatika
Judul Skripsi : SMS *SPAM DETECTION* MENGGUNAKAN
METODE *LONG SHORT TERM MEMORY*

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar hasil karya saya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar Pustaka.

Apabila di kemudian hari terbukti atau dapat dibuktikan Skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 14 Desember 2021
Yang membuat pernyataan,



Ainafatul Nur Muslikah
NIM.17650067

HALAMAN MOTTO

إِنَّ مَعَ الْعُسْرِ يُسْرًا

“Sesungguhnya, bersama kesulitan ada kemudahan”

(Q.S. Al-Insyirah :6)

HALAMAN PERSEMBAHAN

الْعَالَمِينَ رَبِّ اللَّهُ الْحَمْد

Puji syukur kehadiran Allah SWT

Shalawat serta salam kepada Rasulullah SAW

Penulis persembahkan sebuah karya ini kepada :

Kedua orang tua penulis tercinta, Moh.Mulyono dan Likah Nurwanti yang selalu membimbing penulis, memberikan do'a, dukungan, serta motivasi yang tidak terhingga. Syukur Alhamdulillah dan ribuan ucapan terimakasih teruntuk mereka.

Adik penulis, Fauziah Nur Muslikah, Farizatul Nur Muslikah dan Muhammad Masdan Al-Fahrezi serta keluarga dirumah yang selalu menemani penulis sedari kecil, dan selalu memberikan dukungan dan do'a.

Dosen pembimbing penulis Khadijah Fahmi Hayati H, M.Kom dan Dr. Cahyo Crysdiyan yang telah dengan sabar membimbing jalannya penelitian skripsi ini dan selalu memberikan stimulus positif untuk tetap semangat menjalani setiap tahap ujian skripsi.

Seluruh dosen Teknik Informatika UIN Maulana Malik Ibrahim Malang, dan seluruh guru-guru penulis yang telah membimbing dan memberikan ilmunya yang sangat bermanfaat.

Dafa Wiratama, Rizky Hibatullah Rahman, Muhammad Ammarullah Ridho, dan Khamaida Safinah, sahabat yang mendukung satu sama lain dan selalu memberi masukan dalam pengerjaan penelitian ini.

Keluarga Teknik Informatika, terutama Unocore (Teknik Informatika Angkatan 2017) yang telah memberikan semangat dan doanya.

Penulis ucapkan *“jazakumullah khairan katsiiraa”*. Semoga ukhwah kita tetap terjaga dan selalu diridhoi Allah SWT. Aamiin Allahuma Aamiin

KATA PENGANTAR

Segala puji dan syukur kehadirat Allah SWT karena atas berkat Rahmat dan karunia-Nya, penulis diberikan kemudahan dalam menyelesaikan skripsi ini. Penyusunan skripsi ini bertujuan untuk memenuhi syarat kelulusan bagi mahasiswa Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang. Keberhasilan penulisan skripsi ini tidak lepas dari dorongan dan bimbingan dari berbagai pihak. Untuk itu dalam kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Prof. Dr. H. M. Zainuddin, MA selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim.
2. Dr. Sri Hariani, M.Si selalu dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim.
3. Dr. Fachrul Kurniawan, M.MT selaku Ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang yang senantiasa memberikan dorongan.
4. Khadijah Fahmi Hayati H, M.Kom selaku dosen pembimbing I yang telah bersedia meluangkan waktunya dalam membimbing dan memberi arah kepada penulis sehingga dapat menyelesaikan skripsi ini.
5. Dr. Cahyo Crysdiand selaku dosen pembimbing II yang juga bersedia meluangkan waktunya dalam membimbing dan memberi arah kepada penulis sehingga dapat menyelesaikan skripsi ini.

6. Bapak dan Ibu beserta keluarga yang telah memberikan dukungan baik moral maupun spiritual sehingga penulis diberi kemudahan dalam menyelesaikan skripsi ini.
7. Seluruh dosen dan staf Jurusan Teknik Informatika yang telah memberikan ilmu dan pengalaman yang berharga.
8. Dafa Wiratama, Rizky Hibatullah Rahman, Muhammad Ammarullah Ridho, dan Khamaida Safinah, sahabat yang mendukung satu sama lain dan selalu memberi masukan dalam pengerjaan penelitian ini.
9. Teman-teman seperjuangan UNOCORE dan teman-teman pengurus komunitas yang telah memberikan *support* dan pengalaman yang berharga.
10. Semua pihak yang telah membantu dalam menyelesaikan skripsi ini yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa masih banyak kekurangan dari laporan ini. Oleh karena itu, penulis memohon maaf atas segala kekurangan yang terjadi selama proses penyusunan skripsi ini. Semoga tugas akhir ini dapat memberikan kontribusi yang bermanfaat bagi penulis dan pembaca khususnya.

Malang, 14 Desember 2021

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERSETUJUAN	ii
HALAMAN PENGESAHAN	iii
PERNYATAAN KEASLIAN TULISAN	iv
HALAMAN MOTTO	v
HALAMAN PERSEMBAHAN	vi
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xii
DAFTAR TABEL	xiii
ABSTRAK	xiv
ABSTRACT	xv
المخلص.....	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Pernyataan Masalah.....	5
1.3 Tujuan Penelitian.....	5
1.4 Batasan Masalah.....	5
1.5 Manfaat Penelitian.....	6
BAB II TINJAUAN PUSTAKA.....	7
2.1 <i>Short Message Service (SMS)</i>	9
2.2 <i>Spam</i>	10
2.3 <i>Natural Language Processing (NLP)</i>	11
2.4 <i>Recurrent Neural Network (RNN)</i>	13
2.5 <i>Long Short Term Memory (LSTM)</i>	15
2.6 <i>Bidirectional LSTM</i>	19
2.7 Fungsi Aktivasi.....	21
2.8 <i>Binary Cross Entrophy</i>	23
BAB III METODOLOGI PENELITIAN	25
3.1 Desain Penelitian	25
3.2 Pengumpulan Data	27
3.3 Desain Sistem	28
3.4 <i>Preprocessing</i>	29
3.5 Pemodelan LSTM.....	32
3.5.1 Arsitektur Jaringan	32
3.5.2 Proses Training	44
3.5.3 Proses Validasi	50
BAB IV UJI COBA DAN PEMBAHASAN	52
4.1 Skenario Pengujian.....	52
4.2 Data Penelitian	54
4.3 Hasil Training.....	55
4.4 Evaluasi Model.....	56
4.5 Pembahasan	58

BAB V KESIMPULAN DAN SARAN	61
5.1 Kesimpulan.....	61
5.2 Saran.....	61
DAFTAR PUSTAKA	
LAMPIRAN	

DAFTAR GAMBAR

Gambar 2. 1 Arsitektur RNN (Wang et al., 2021)	13
Gambar 2. 2 <i>Flowchart Long Short Term Memory (LSTM)</i>	16
Gambar 2. 3 Arsitektur Bi-LSTM (Jang et al., 2020)	20
Gambar 2. 4 Distribusi Fungsi Sigmoid (P et al., 2016)	21
Gambar 2. 5 Fungsi Tanh (Feng & Lu, 2019)	22
Gambar 2. 6 Fungsi ReLU	23
Gambar 3. 1 Diagram Blok Prosedur Penelitian	25
Gambar 3. 2 Diagram Blok Perancangan Sistem	28
Gambar 3. 3 <i>Data Flow Training dan Validation</i>	29
Gambar 3. 4 Arsitektur Jaringan	32
Gambar 3. 5 <i>Word Embedding</i>	34
Gambar 3. 6 Inisialisasi <i>Input Backward LSTM</i>	40
Gambar 3. 7 Inisialisasi <i>Output Bi-LSTM</i>	41
Gambar 3. 8 <i>Flowchart Training</i>	44
Gambar 3. 9 Inisialisasi turunan <i>Binary Cross Entrophy</i>	46
Gambar 3. 10 Inisialisasi Turunan Sigmoid	46
Gambar 3. 11 Inisialisasi Turunan ReLU	46
Gambar 4. 1 Kurva Akurasi <i>Training</i>	56
Gambar 4. 2 Kurva Loss <i>Training</i>	56

DAFTAR TABEL

Tabel 3. 1 <i>Word Tokenization</i>	31
Tabel 3. 2 Hasil <i>Padding and Truncating Data</i>	32
Tabel 3. 3 Contoh Hasil <i>Word Embedding</i>	35
Tabel 3. 4 Tabel Contoh Bobot dan Bias	36
Tabel 4. 1 <i>Confusion Matrix</i>	53
Tabel 4. 2 Sampel Daftar SMS	54
Tabel 4. 3 Hasil <i>Confusion Matrix</i>	57

ABSTRAK

Muslikah, Ainafatul Nur. 2021. **SMS Spam Detection Menggunakan Metode Long Short Term Memory**. Skripsi. Jurusan Teknik Informatika, Fakultas Sains dan Teknologi. Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Khadijah Fahmi Hayati H, M.Kom, (II) Dr. Cahyo Crysdiyan

Kata Kunci : *Long Short Term Memory*, Klasifikasi Teks, Deteksi Spam

Kebutuhan informasi ialah kebutuhan fundamental pada setiap orang untuk melakukan komunikasi. Terdapat beberapa cara untuk melakukan penyaluran informasi, salah satunya ialah melewati pengiriman Short Message Service (SMS). Meningkatnya penggunaan layanan SMS banyak dimanfaatkan oleh pihak-pihak tertentu yang tidak bertanggung jawab. Mereka mengirimkan pesan-pesan yang tidak bermanfaat dan merugikan disebut SMS spam atau teks pesan spam. Teks pesan spam yang dikirimkan melalui sms banyak yang mengandung konten ilegal, sehingga menimbulkan rasa tidak nyaman bagi penerima pesan tersebut. Pesan spam yang banyak beredar di Indonesia adalah pesan spam yang didalamnya terdapat pengumuman promosi, pengumuman hadiah, dan diskon toko atau pesan yang tidak memiliki makna yang lainnya. Pada penelitian ini digunakan metode *Long Short Term Memory* (LSTM) untuk mengklasifikasikan sms spam dan non spam. Data yang digunakan merupakan data SMS berbahasa indonesia. Tujuan dari penelitian ini yaitu mengukur akurasi, presisi, *recall*, dan *f-measure* pada klasifikasi SMS spam. Hasil dari penelitian ini menunjukkan bahwa model LSTM yang telah dirancang menunjukkan bahwa model LSTM berjalan dengan baik dengan akurasi sebesar 96,09%, presisi sebesar 96,09%, *recall* sebesar 96,09% dan *f-measure* sebesar 96,09%.

ABSTRACT

Muslikah, Ainafatul Nur. 2021. **Detection of SMS spam using the Long Short Term Memory method.** Undergraduate Thesis. Informatics Engineering Department, Faculty of Science and Technology. Islamic State of Maulana Malik Ibrahim Malang. Supervisor: (I) Khadijah Fahmi Hayati H, M.Kom, (II) Dr. Cahyo Crysdiان.

Keywords : *Long Short Term Memory, Text Classification , Spam Detection*

The need for information is a basic need for everyone to communicate. One way to disseminate information is through the delivery of Short Message Service (SMS). Increased use of SMS services which are widely used by certain irresponsible parties. They send messages that are not useful and harmful called SMS spam or text spam messages. The text of spam messages sent via sms contains a lot of illegal content, causing discomfort to the recipient of the message. Spam messages that are widely circulated in Indonesia are spam messages in which there are announcements of announcements, prize announcements, and store discounts or messages that have no meaningful meaning. In this study, the Long Short Term Memory (LSTM) method is used to classify spam and non-spam sms. The data used is Indonesian language SMS data. The purpose of this study is to measure accuracy, precision, recall, and f-measure in the classification of SMS spam. The results of this study indicate that the LSTM model that has been designed shows that the LSTM model runs well with an accuracy of 96.09%, precision of 96.09%, recall of 96.09% and f-measure of 96.09%.

الملخص

النوعين من فئة مصالحة، 2021. القصيرة الرسالة لغير فيها المرغوب غير الرسالة عن الكشف الطريقة باستخدام *Long Short Term Memory*. البحث الجامعي. قسم الهندسة المعلوماتية. كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشريف الأول: (1) فهمي خادجة الماجستير هول هاي تي (2) كريد سديان الدوك تور.

الرؤية: *Long Short Term Memory*، المشدود ي البريد كشف، النص تصنيف

معلومات كيبوتوهان إلا كيبوتوهان الأساسية بادا ستياب أورانغ أنتوك ميلاكوكان كومونيكاسي. خدمة الرسائل القصيرة (SMS). يمكن أن يكون هناك الكثير من الأشياء التي يمكن أن يكون هناك الكثير من الأشياء. من المهم أن تكون قادرا على إرسال الرسائل القصيرة غير المرغوب فيها. إذا كنت ترغب في إرسال رسائل غير مرغوب فيها، فأنت بحاجة إلى إرسال رسائل نصية قصيرة. الرسائل غير المرغوب فيها بيسان يانغ بانياك بيريدار من اندونيسيا عدالة بيسان البريد المزعج يانغ ديدالامنيا تيردابات بينغومومان بروموسي، بينغومومان هاديا، وديسكون توكو أتاو بيسان يانغ تيداك ميميليكى ماكننا يانغ لينا. الذاكرة قصيرة المدى طويلة (LSTM) أوتوك مينغكلاسيكاسيكان سمز سبام وغير سبام. البيانات يانغ ديغوناكان ميروباكان البيانات سمز بريهاسا اندونيسيا. إذا كنت ترغب في الحصول على رسالة نصية قصيرة، فعليك أن تتذكر، وأن تقوم بقياس الرسائل القصيرة غير المرغوب فيها. هاسيل داري بينيليتيان إيني مينونجوكان باهوا نموذج لستم يانغ تيلاه ديرانكانغ مينونجوكان باهوا نموذج لستم بيرجالان دينجان بيك دينجان أكوراسي سيبيسار 96,09%، بريسيبي بريسيبي سيبيسار 96,09%، أذكر سيبيسار 96,09% دان و-قياس سيبيسار 96,09%.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kebutuhan informasi ialah kebutuhan fundamental pada setiap orang untuk melakukan komunikasi. Terdapat beberapa cara untuk melakukan penyaluran informasi, salah satunya ialah melewati pengiriman Short Message Service (SMS). SMS adalah data menggunakan tipe asynchronous message yang mana transmisi data SMS dilaksanakan dengan teknik protokol store and forward. Hal tersebut menandakan sebenarnya pengirim dan penerima SMS tidak memerlukan di dalam status berhubungan (connected/online) satu sama lain waktu akan saling bertukar pesan SMS. Pengiriman pesan SMS secara store and forward berarti pengirim pesan SMS mengirimkan pesan serta nomor telepon tujuan dan kemudian mengirimkannya (store) ke server SMS atau SMS-Center. SMS-Center (SMSC) berkewajiban dalam penanganan pesan SMS pada jaringan nirkabel (wireless) yang setelahnya berkewajiban dalam mengirimkan pesan tersebut (forward) ke nomor telepon yang dituju (Munitasri et al., 2018).

Bertambahnya penggunaan layanan SMS banyak digunakan oleh beberapa pihak tertentu yang tidak bertanggung jawab. Mereka mengirimkan pesan yang tidak memiliki manfaat yang biasa dikatakan sebagai SMS spam atau teks pesan spam. Kalimat pesan spam kemudian dikirimkan menggunakan sms banyak yang didalamnya mayoritas adalah pesan ilegal, sehingga menimbulkan rasa tidak nyaman bagi penerima pesan. Banyaknya pesan spam yang tersebar di Indonesia adalah pesan spam yang didalamnya terdapat pengumuman promosi,

pengumuman hadiah, dan diskon toko atau pesan yang tidak memiliki makna yang lainnya. Contoh pengiriman pesan yang mengumukan hadiah yang mengatasnamakan PT MOBOINDOSAT pada tanggal 8 September 2020 oleh nomor 085823512145 dengan pesan “UNDIAN RESMI PT.MOBOINDOSAT TAHUN 2020 Selamat No.Anda Resmi Meraih Hadiah Ke-2 Rp.175jt Dgn PIN(25e477r) U/Info Hadian Klik di www.mobo-indsat227.blogspot.com”.

SMS spam mengalami peningkatan disetiap tahunnya. “Seperti yang terlihat dalam laporan kami dari tahun ke tahun, angka penipuan melalui telepon maupun SMS terus meningkat secara signifikan di Indonesia. Saat ini, ponsel memang sudah menjadi bagian tidak terpisahkan dalam kegiatan sehari-hari, namun oknum yang tidak bertanggung jawab justru menggunakan kesempatan ini untuk melakukan penipuan dengan mengelabui masyarakat umum. Karena itulah, di *Truecaller*, kami sangat bangga bisa membantu lebih dari 420.000 pengguna aktif harian di Indonesia untuk melindungi mereka dari panggilan scam,” kata Kim Fai Kok, *Director of Communications* Truecaller pada hari Kamis tanggal 5 bulan Desember tahun 2019. Sehingga dari pernyataan tersebut dapat disimpulkan bahwa banyak masyarakat yang telah dirugikan. Hal tersebut dikarenakan kebanyakan SMS Spam memuat penipuan, promosi dan intimidasi, sehingga masyarakat terusik dengan adanya SMS spam tersebut. Karena ketergangguan tersebut dapat memicu masyarakat merespon SMS spam tersebut dan menjadi korban dari SMS spam tersebut. Penipu akan meminta banyak hal kepada penerima SMS spam, dan penerima tersebut akan melakukan apa yang telah diminta penipu seperti mengirimkan pulsa atau mengirimkan uang ke rekening

penipu, sehingga dapat disimpulkan bahwa korban atau penerima SMS spam dirugikan secara finansial.

Dalam Kamus Besar Bahasa Indonesia (KBBI) penipuan berasal dari kata tipu yang berarti perbuatan atau perkataan yang tidak jujur (bohong, palsu, dan sebagainya) dengan maksud untuk menyesatkan, mengakali atau mencari untung sedangkan penipuan berarti proses, cara, perbuatan menipu, dan perkara menipu. Dalam pandangan agama, praktik penipuan merupakan hal yang harus dihindari oleh setiap manusia. Hal tersebut seperti yang tercantum pada Al-Qur'an dalam surat Asy-Syu'ara ayat 183 :

مُفْسِدِينَ الْأَرْضِ فِي تَعْتُوا وَلَا أَشْيَاءَهُمُ النَّاسَ تَبْخَسُوا وَلَا

“Dan janganlah kamu merugikan manusia pada hak-haknya dan janganlah kamu merajalela di muka bumi dengan membuat kerusakan” (Asy-Syu'ara ayat 183).

Berdasarkan tafsir dari Departemen Keagamaan Republik Indonesia (KEMENAG) pada Asy-Syu'ara ayat 183 Allah menerangkan terkait prinsip hubungan antar manusia menurut Islam adalah tidak boleh menzalimi dan tidak boleh dizalimi dengan cara apa pun dan dalam bidang apa pun. Zalim yang dimaksud boleh berupa penipuan atau kegiatan tidak terpuji lainnya. Maka dari itu, berlandaskan firman Allah tersebut, Allah melarang hambanya untuk mendzalimi atau memperlakukan manusia yang bukan pada hak-haknya.

Di Indonesia sendiri sudah memiliki atau diberlakukan dasar hukum terkait penipuan atau SMS palsu, yakni Pasal 28 ayat (1) UU Informasi dan Transaksi Elektronik (ITE), dijelaskan bahwa pelaku penipuan SMS diberikan

sanksi bahwa terancam hukuman penjara paling lama enam tahun atau denda Rp 1.000.000.000,- (Satu miliar rupiah).

Namun, pemberlakuan peraturan baik dari agama atau pun hukum pada penipuan SMS tidak mampu mengatasi permasalahan yang terjadi di masyarakat. Dalam mengatasi permasalahan tersebut maka dibutuhkan suatu sistem yang dapat mengklasifikasikan SMS yang berfungsi dalam mengkategorikan pesan yang terindikasi spam dan pesan yang tidak terindikasi spam. Sehingga sistem tersebut dapat membantu masyarakat dalam menyeleksi pesan dan meminimalisir jumlah korban penipuan SMS yang ada di Indonesia.

Spam Detection telah dimanfaatkan oleh sejumlah peneliti yang menggunakan metode *machine learning* seperti Naive Bayes (NB), Random Forest (RF), Support Vector Machine (SVM). Pada penelitian tersebut membutuhkan proses yang lama dan juga sulit untuk mengekstrak semua pesan dalam bentuk teks pendek. Dikarenakan permasalahan tersebut, peneliti akan menggunakan algoritma Long Short Term Memory. Algoritma Long Short Term Memory (LSTM) merupakan sebuah metode dari pengembangan *Recurrent Neural Network* dimana LSTM mampu menyimpan atau mengingat informasi pada sel memori untuk ruang yang sangat besar (Wildan et al., 2018).

Dalam penelitian ini mengembangkan model klasifikasi SMS Spam dengan menggunakan Teknik Natural Language Process (NLP) untuk pengembangan dan penyusunan model data berbasis LSTM. Dalam hal ini pengembangan model klasifikasi SMS spam berdasarkan Teknik *Deep Learning*

dikarenakan jenis SMS spam yang umum adalah data teks. NLP merupakan algoritma yang membuat komputer dapat memahami bahasa manusia.

Berdasarkan penjelasan diatas, peneliti memiliki ide untuk membuat sistem SMS *spam detection* menggunakan metode *Long Short-Term Memory* (LSTM) yang nantinya akan lebih mudah dimanfaatkan oleh masyarakat dalam mendeteksi SMS spam. Pembuatan sistem ini diharapkan dapat memudahkan masyarakat dalam mendeteksi SMS spam untuk meningkatkan kewaspadaan terhadap penipuan yang beredar di pesan teks.

1.2 Pernyataan Masalah

Berdasarkan latar belakang diatas, maka pernyataan masalah dalam penelitian ini adalah seberapa nilai akurasi, presisi, *recall*, dan *f-measure* metode *Long Short Term Memory* (LSTM) pada SMS *spam detection*.

1.3 Tujuan Penelitian

Tujuan masalah penelitian ini adalah untuk mengukur nilai akurasi, presisi, *recall*, dan *f-measure* metode *Long Short Term Memory* (LSTM) pada SMS *spam detection*.

1.4 Batasan Masalah

Batasan masalah penelitian ini adalah sebagai berikut :

1. Data SMS yang digunakan berbahasa Indonesia.
2. Data SMS Spam yang digunakan adalah data SMS yang berasal dari nomer yang tidak dikenal, berisi iklan, dan hadiah yang mana data tersebut bukan

dari *Official Store* dan pesan-pesan yang menimbulkan banyak kerugian kepada penerima pesan.

3. Metode yang digunakan merupakan LSTM dua arah atau Bi-LSTM.

1.5 Manfaat Penelitian

Dari penelitian ini diharapkan dapat memberikan manfaat bagi beberapa pihak, diantaranya adalah :

1. Manfaat bagi pengguna telepon selular terutama pengguna SMS, dengan adanya sistem ini diharapkan mampu membantu pengguna telepon selular dalam mengetahui dan menghindari SMS spam serta dapat meminimalisir jumlah korban penipuan SMS Spam di Indonesia.
2. Manfaat bagi penelitian selanjutnya, dengan adanya sistem ini diharapkan mampu menjadi referensi dalam menentukan metode dalam mendeteksi SMS Spam di penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA

Pada proses penelitian yang akan dilakukan diperlukan beberapa penelitian terdahulu sebagai referensi dan pemahaman bagi penulis untuk dapat meninjau dan mengembangkan penelitian ini. Maka dari itu, penulis telah mengumpulkan beberapa penelitian terdahulu terkait permasalahan yang diangkat pada penelitian ini.

Pada penelitian yang dilakukan oleh Indah Munitasari *et al* (2018), penelitian tersebut menggunakan klasifikasi Naïve Bayes yang akan membuat tambahan *Genetic Algorithm* pada proses pemilihan atribut yang akan digunakan pada proses klasifikasi menggunakan prosedur pemecahan Naïve Bayes. *Genetic Algorithm* dipergunakan sebagai atribut dari semua atribut pemilih yang diperoleh dari proses ekstraksi. Dilakukan perbandingan antara Naïve Bayes dan *Genetic Algorithm* memiliki hasil tingkat akurasi maksimal, Naïve Bayes memperoleh tingkat akurasi 89,39% sedangkan *Genetic Algorithm* memperoleh tingkat akurasi 89,73%. Dari hal tersebut dapat disimpulkan bahwa terdapat peningkatan akurasi sebesar 0,34% setelah penambahan *Genetic Algorithm*. Naïve Bayes dan *Genetic Algorithm* dapat diterapkan klasifikasi pesan singkat atau SMS, dikarenakan prosedur pemecahan Naïve Bayes ialah algoritma yang tidak mempertimbangkan korelasi antara atribut satu dengan yang lain (independensi).

Pada penelitian yang dilakukan oleh Gauri Jain *et al* (2018), penelitian tersebut menggunakan metode *Long Short Term Memory* (LSTM). Pada penelitian tersebut sebelum menggunakan LSTM untuk mengklasifikasi spam,

teks diubah menjadi vector kata semantic dengan bantuan word2vec, WordNet dan ConceptNet. Hasil Klasifikasi dibandingkan dengan beberapa metode yaitu *Support Vector Machine* (SVM), Naïve Bayes, *Artificial Neural Network* (ANN), *K-Nearest Network* (KNN) dan *Random Forest*. Data yang digunakan untuk perbandingan adalah dataset SMS spam dan dataset twitter. Dari penelitian tersebut dapat disimpulkan bahwa LSTM mampu mengungguli metode pembelajaran *machine learning* untuk mendeteksi spam.

Pada penelitian yang dilakukan Aini Suri Talita dan Aristiawan Wiguna (2019), penelitian tersebut menggunakan Teknik *deep learning* yaitu *Long Short Term Memory* (LSTM) yang diimplementasikan untuk mendeteksi ujaran kebencian atau *hate speech* yang berkaitan dengan Pemilihan Presiden (Pilpres) 2019. Dalam penelitian tersebut menggunakan data 950 kalimat dan diuji dengan 190 kalimat dari *dataset* yang diambil dari media social Facebook. Hasil dari penelitian tersebut memiliki nilai *recall* 0.7021, yang berarti bahwa dari keseluruhan data uji, 70,21% benar terdeteksi sebagai *hate speech* kemudian sisanya 29,79% salah dideteksi sebagai *hate speech*.

Pada penelitian yang dilakukan oleh Pumrapee Poomka *et al* (2019), penelitian tersebut membahas terkait deteksi spam SMS berbahasa inggris dengan menggunakan algoritma *Long Short Term Memory* dan *Gated Recurrent Unit*. Performa dari kedua algoritma tersebut dibandingkan dengan algoritma *machine learning* termasuk *Support Vector Machine* dan Naïve Bayes. Hasil dari penelitian tersebut menunjukkan bahwa model yang dibangun dengan menggunakan algoritma *Long Short Term Memory* (LSTM) memberikan akurasi keseluruhan

terbaik sebesar 98,18%. Pada penyaringan pesan spam yang akurat, LSTM dapat mendeteksi pesan spam dengan tingkat akurasi 90,96%, sedangkan persentase kesalahan dalam mengklasifikasikan spam sebesar 0,74%.

Pada penelitian yang dilakukan Hanane Elfaik dan El Habib Nfaoui (2020), penelitian tersebut membahas tentang *deep learning Bidirectional LSTM* dalam analisis sentimen berbasis pembelajaran untuk bahasa arab. Pada penelitian ini meneliti terkait kompleksitas bahasa arab dengan tidak adanya kata-kata sentimen eksplisit dalam sepotong implisit, dilakukan perbandingan pada data cleaning dengan menggunakan stemming dan tidak menggunakan stemming. Penelitian ini menggunakan dataset dari enam sumber yang berbeda, kemudian hasil dari penelitian ini dengan membandingkan Bi-LSTM dengan metode *traditional machine learning* seperti *random forest*, *support vector machine* dan metode *deep learning* yang lain yaitu *long short term memory*, *convolutional neural network*. Hasil dari penelitian ini menunjukkan bahwa Bi-LSTM berjalan lebih baik dengan dibandingkan dengan model *traditional machine learning* dan *deep learning* yang lain.

2.1 Short Message Service (SMS)

SMS adalah sebutan dari *Short Message Service*. SMS merupakan teknologi yang memungkinkan pengiriman dan penerimaan pesan antar ponsel. SMS pertama kali muncul di Eropa pada tahun 1992. SMS berevolusi dari standar *Global System for Mobile Communications*, spesifikasi jaringan telepon selular tersebut diterima secara internasional yang dibuat oleh Lembaga Standar

Telekomunikasi Eropa. Pada saat ini, Proyek Kemitraan Generasi ke-3 tetap mempertahankan standar SMS (Brown et al., 2007).

Pesan SMS ditangani melalui *Short Message Service Center* yang dikelola oleh operator nirkabel untuk perangkat akhir. SMSC dapat mengirim pesan ke perangkat akhir menggunakan muatan maksimum 140 oktet. Dari hal tersebut dapat mendefinisikan bahwa batas pesan SMS menjadi 160 karakter menggunakan pengkodean 7-bit (Brown et al., 2007).

Pesan teks mendukung bahasa internasional. Hal tersebut berfungsi dengan baik dengan semua bahasa yang didukung Unicode, termasuk Arab, Cina, Jepang dan Korea. Selain pesan teks, pesan SMS juga dapat membawa data biner. Pesan teks dapat digunakan untuk mengirim nada dering, gambar, logo operator, wallpaper, animasi, kartu nama dan konfigurasi WAP ke ponsel dengan pesan SMS (Brown et al., 2007).

Salah satu keunggulan utama SMS adalah didukung oleh semua ponsel GSM. Hampir semua paket langganan yang disediakan oleh operator nirkabel menyertakan layanan pesan SMS murah (Brown et al., 2007).

2.2 Spam

Menurut Kamus Besar Bahasa Indonesia (KBBI), spam adalah surat yang dikirim tanpa diminta melalui internet, biasanya berisi iklan. Pada SMS, spam dapat diartikan sebagai pesan yang dikirim tanpa diminta dan berskala massif secara terus menerus, biasanya berisi iklan. Namun, dalam kehidupan sehari – hari telah banyak ditemukan tujuan lain dari spam, yaitu :

- a. Promosi

Pada tujuan ini, pelaku spam memberikan penawaran berupa barang atau jasa yang dijadikan sebagai bentuk pemasaran dari sebuah perusahaan atau perseorangan yang menawarkan barang atau jasa.

b. Penipuan

Pada tujuan ini, pelaku spam mengirim pesan spam dengan tujuan untuk menipu penerima pesan. Modus penipuan yang dilakukan bermacam-macam, misalnya meminta uang, pencurian data, mengambil alih akun online (*phising*) dan lain sebagainya.

Modus penipuan yang sering terjadi pada saat ini adalah mengambil alih akun online (*phising*). Pada pesan spam tersebut biasanya disisipkan sebuah link yang menuju situs tertentu dimana si penerima spam tersebut diharuskan untuk login menggunakan akun email dan password. Pada saat korban menyetikkan email dan password, setiap ketikan yang korban telah direkam sehingga penipu telah mengetahui alamat email dan *password* korban.

Spam jenis penipuan adalah spam yang paling membahayakan dikarenakan spam tersebut lebih merugikan dari pada jenis-jenis spam lainnya. Jika korban memiliki akun Mobile Banking maka korban akan kehilangan data pribadi serta uang yang ada di rekening tersebut.

2.3 *Natural Language Processing (NLP)*

Natural language processing adalah sektor integral dari ilmu komputer tempat pembelajaran mesin dan linguistik komputasi digunakan secara luas. Bidang ini memiliki kaitan terkait pembuatan interaksi manusia dan komputer menjadi mudah tetapi efisien. Mesin mempelajari sintaks dan arti bahasa manusia,

memprosesnya, dan memberikan hasil pada pengguna. Area NLP melibatkan pembuatan sistem komputer untuk melakukan tugas-tugas yang berarti dengan bahasa yang alami dan dapat dimengerti manusia (A. Jain et al., 2019).

Alasan mengapa NLP sangat penting di masa depan ialah membantu peneliti dalam pembangunan model serta proses pengambilan bagian informasi sebagai *input* dan diimplementasikan terhadap suara atau teks atau keduanya dan diubah seperti algoritma di dalam komputer (A. Jain et al., 2019).

Dengan demikian input dapat berupa ucapan, teks atau gambar dimana output dari sistem NLP dapat diproses Speech serta Teks Tertulis. Berbagai algoritma yang dikembangkan untuk meningkatkan efisiensi pemrosesan bahasa dalam bentuk teks yang akan kita bahas di sini adalah: (A. Jain et al., 2019)

- *Long short term memory*
- *Sequence 2 Sequence model*
- *Named Entity Recognition model*
- *User preference graph model*
- *Word Embedding model*
- Ekstaksi kalimat berbasis fitur menggunakan aturan inferensi fuzzy
- Algoritma berbasis template menggunakan peringkasan teks otomatis

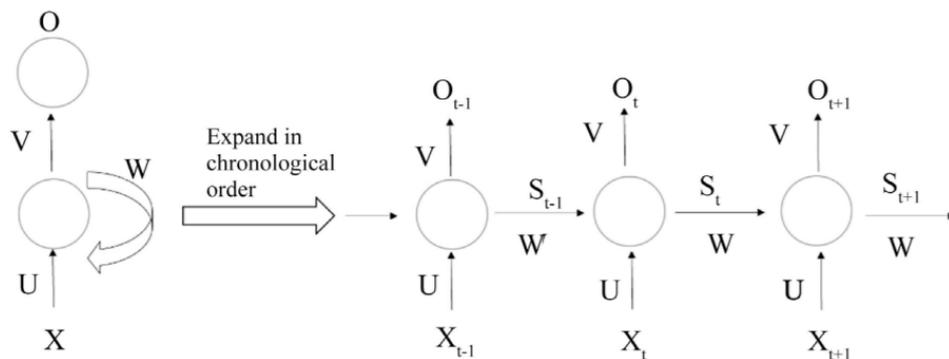
Demikian pula bahasa dapat diproses meskipun masukan dalam bentuk ucapan. Untuk itu berbagai algoritma dikembangkan dan yang terbaik dari semuanya adalah: (A. Jain et al., 2019)

- *Word Recognition*
- *Acoustic Modelling*

- *Connectionist temporal classification*
- *Neural machine translation*
- *Google neural machine translation*

2.4 Recurrent Neural Network (RNN)

Traditional Neural Network tidak dapat memprediksi informasi dari node berikutnya berdasarkan informasi dari node sebelumnya, dan kata-kata dalam setiap kalimat tidak independen di dalam teks (Wang et al., 2021). Kelebihan dari RNN adalah RNN dapat mengirimkan informasi sebelumnya ke output saat ini dari suatu urutan terkait dengan output sebelumnya (Wang et al., 2021). Hal tersebut dapat didefinisikan sebagai data sekuensial yaitu data yang diproses sesuai urutan, dan suatu data memiliki hubungan satu sama lain (Faadilah, 2020). Berikut adalah arsitektur sederhana dari Recurrent Neural Network :



Gambar 2. 1 Arsitektur RNN (Wang et al., 2021)

Pada Gambar 2.1, sisi kanan gambar merupakan perluasan dari arsitektur pada sisi kiri, dengan informasi sebelumnya pada setiap waktu (t), yang dikendalikan oleh bobot (W) (Wang et al., 2021). Pada Gambar 2.1 dapat dilihat bahwasanya jaringan mempunyai *input layer* berupa x , *hidden layer* s dan output

layer berupa y . Input jaringan dalam waktu yaitu x_t , output dalam waktu yaitu y_t dan lapisan tersembunyi yaitu s_t . Input dalam bentuk vektor dilambangkan x_t dibuat dengan menggabungkan vektor w yang merupakan perwakilan kata saat ini, dan *output layer* konteks ketika $t - 1$ (Faadilah, 2020).

Tujuan dari proses training RNN adalah menghasilkan nilai loss paling minimum. Pada arsitektur RNN bergantung pada antrian waktu sebelumnya, sehingga untuk mencapai tujuan dari proses training dapat menggunakan algoritma *Backpropagation Thorough Time* (BPTT). Setelah itu dalam pengkalkulasian gradien ketika langkah waktu t , dilakukan perhitungan *derivative* pada langkah $t - 1$, $t - 2$, $t - 3$ sampai mencapai $t = 1$. Sehingga jaringan saraf tiruan dapat mengenali informasi dari beberapa proses di setiap waktu pada *hidden layer* ketika BPTT mempelajarinya.

Berdasarkan proses yang telah dijelaskan sebelumnya RNN memiliki permasalahan yaitu *short-term memory* yang mana di dalam urutan yang lebih panjang, RNN kesulitan dalam menjalankan dari urutan waktu awal ke urutan waktu berikutnya (Faadilah, 2020). BPTT yang dilakukan algoritma RNN memiliki kekurangan yaitu *vanishing gradient decent*. *Vanishing gradient decent* yaitu kondisi dimana pembaruan gradien dari urutan waktu awal ke urutan waktu akhir semakin mengecil, sehingga arsitektur atau model tidak berjalan dengan baik. Pada *Recurrent Neural Network*, lapisan yang memiliki *update* nilai gradien kecil akan menghentikan tahapan learning (Faadilah, 2020). Dikarenakan layer tersebut menghentikan tahapan training maka RNN melupakan informasi di

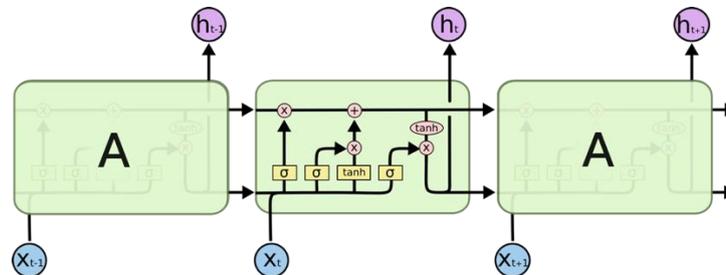
sequence yang lebih panjang, maka dari itu RNN memiliki masalah short-term memory (Faadilah, 2020).

2.5 *Long Short Term Memory (LSTM)*

LSTM merupakan varian dari *Recurrent Neural Network* (RNN). Secara teori RNN berjalan dengan baik, tetapi memiliki kesulitan dalam mengingat memori jangka Panjang, yaitu karena jarak antara *layer* saat ini dan *layer* sebelumnya meningkat sehingga membuat memori menjadi lemah. Hal tersebut dikarenakan kesalahan atau eror dari *backpropagation* yang mengharuskan jaringan menghitung gradien untuk bobot yang diperbaharui sehingga membuat RNN menghadapi masalah gradien menghilang atau meledak (gradien menjadi terlalu kecil atau terlalu besar). Dalam mengatasi hal tersebut, LSTM digunakan sebagai pengganti RNN. Jaringan LSTM dapat mengingat informasi untuk periode waktu yang lebih lama atau untuk siklus yang lebih lama tanpa masalah gradien yang hilang dan meledak (G. Jain et al., 2019).

LSTM meningkatkan algoritma RNN dengan memecahkan masalah gradien yang hilang dengan menambahkan status sel untuk mengingat atau melupakan data. Status sel mengandung struktur yang disebut *cell gates* atau gerbang sel. Gerbang sel terdiri dari empat bagian yaitu *input gate*, *forget gate*, *memory-cell state gate* dan *output gate*. *Input gate* merupakan *gate* yang digunakan untuk mengontrol input data yang layak disimpan atau tidak. *Forget gate* digunakan untuk mengontrol keadaan tersembunyi sebelumnya yang akan disimpan di *cell-memory* dari keadaan tersembunyi saat ini. *Memory-cell state gate* digunakan untuk memperbaharui data berdasarkan informasi dari *input gate*

dan *forget gate*. *Output gate* digunakan untuk menghitung *output data* dari jaringan berdasarkan keadaan *memory-cell* (Poomka et al., 2019).



Gambar 2. 2 Flowchart Long Short Term Memory (LSTM)

Pada gambar diatas menyebutkan perihal bagaimana alur kerja dari *memory cells* pada setiap neurons LSTM bekerja. Seperti yang sudah dijelaskan sebelumnya, bahwa terdapat empat tahapan yang menggunakan *activation function* dalam setiap *input* pada *neurons* yang disebut *gates units* yaitu *forget gates, input gates, cell gates, dan output gates*.

a. *Forget Gates*

Forget Gates akan mengolah informasi disetiap data *input* dan menentukan data mana saja yang akan disimpan atau dibuang pada *memory cells*. Fungsi aktivasi yang dipergunakan pada *forget gates* yaitu fungsi aktivasi sigmoid dimana hasil keluarannya adalah antara 0 dan 1. Jika keluarannya adalah 1 maka semua data akan disimpan serta kebalikannya jika keluarannya merupakan 0 maka data akan dibuang (Wildan et al., 2018). Berikut adalah rumus yang digunakan :

$$f_t = \sigma(W_{xf} \cdot x_t + W_{hf} \cdot h_{t-1} + b_f) \quad (2.1)$$

Keterangan :

f_t = Forget gate

σ = Fungsi aktivasi sigmoid

W_{xf} = Bobot *input* pada *forget gate*

x_t = Input ke-t

W_{hf} = Bobot *hidden state* pada *forget gate*

h_{t-1} = *Output hidden state* sebelumnya

b_f = Bias pada *forget gate*

b. *Input Gates*

Input Gates memiliki dua *gates* yang akan dilakukan perhitungan, *gates* pertama akan memutuskan nilai yang akan diperbaharui menggunakan fungsi aktivasi sigmoid kemudian fungsi aktivasi tanh akan membuat vektor nilai baru yang kemudian tersimpan pada *memory cell* (Wildan et al., 2018). Berikut adalah persamaan yang digunakan :

$$i_t = \sigma(W_{xi} \cdot x_t + W_{hi} \cdot h_{t-1} + b_i) \quad (2.2)$$

$$\check{c}_t = \tanh(W_{xc} \cdot x_t + W_{hc} \cdot h_{t-1} + b_c) \quad (2.3)$$

Keterangan :

i_t = *Input gate*

\check{c}_t = Cell Aktivasi

σ = Fungsi aktivasi sigmoid

\tanh = Fungsi aktivasi tanh

x_t = Input ke-t

h_{t-1} = *Output hidden state* sebelumnya

W_{xi} = Bobot *input* pada *Input gate*

W_{xc} = Bobot *input* pada cell aktivasi

W_{hi} = Bobot *hidden state* pada *Input gate*

W_{hi} = Bobot *hidden state* pada cell aktivasi

b_i = Bias pada *Input gate*

b_c = Bias pada *cell aktivasi*

c. *Cell Gates*

Cell Gates akan mengganti nilai yang ada pada *memory cell* sebelumnya menggunakan nilai *memory cell* yang baru dimana nilai tersebut didapat dari penggabungan nilai yang ada pada *forget gate* dan *input gate* (Wildan et al., 2018). Berikut adalah rumus yang digunakan :

$$c_t = f_t * c_{t-1} + i_t * \check{c}_t \quad (2.4)$$

Keterangan :

c_t = *cell gates*

f_t = *forget gate*

c_{t-1} = *output cell gate* sebelumnya

i_t = *input gate*

\check{c}_t = cell aktivasi

d. *Output Gates*

Output gates memiliki dua gates yang akan dilakukan tahapan didalamnya, yang pertama akan diputuskannya nilai pada bagian *memory cell* dimana menggunakan fungsi aktivasi sigmoid. Kemudian akan ditempatkan nilai pada *memory cell* menggunakan fungsi aktivasi tanh. Kedua gates tersebut kemudian dikalikan sehingga dapat menghasilkan nilai yang akan dikeluarkan (Wildan et al., 2018). Berikut adalah rumus yang digunakan:

$$o_t = \sigma(W_{x_o} \cdot x_t + W_{h_o} \cdot h_{t-1} + b_o) \quad (2.5)$$

$$h_t = o_t * \tanh(c_t) \quad (2.6)$$

Keterangan :

o_t = *Output Gate*

c_t = *cell gates*

σ = Fungsi aktivasi sigmoid

\tanh = Fungsi aktivasi tanh

W_{x_o} = Bobot *input* pada *Output Gate*

x_t = Input ke-t

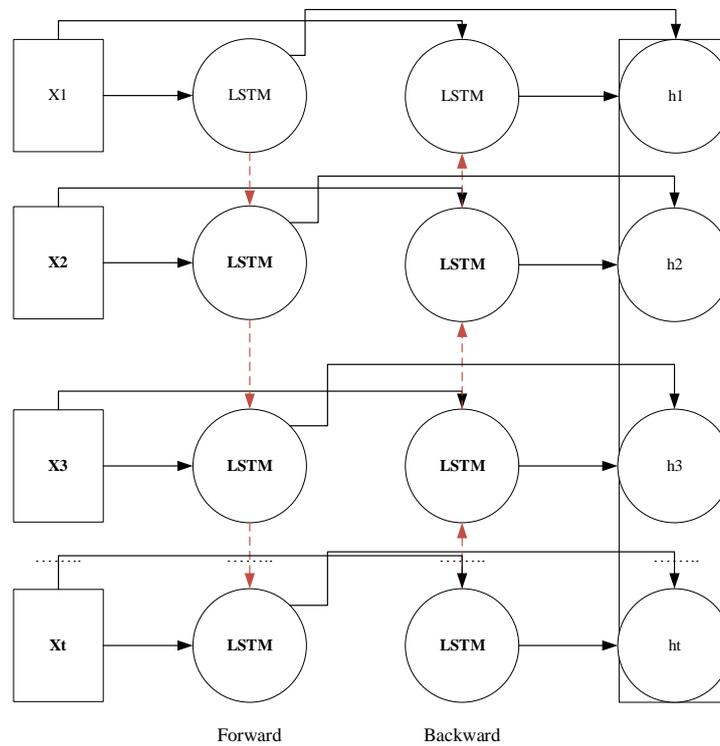
W_{h_o} = Bobot *hidden state* pada *Output Gate*

h_{t-1} = *Output hidden state* sebelumnya

b_o = Bias pada *Output Gate*

2.6 *Bidirectional LSTM*

LSTM dua arah (BiLSTM) berisi dua LSTM independen, yang beroperasi di kedua arah, forward atau maju dan backward atau mundur dimana Bi-LSTM dapat menggabungkan informasi masa lalu dan masa depan sehingga dua arah maju dan mundur tersebut digunakan untuk menangkap ketergantungan dari dalam dua konteks (Jang et al., 2020). Struktur jaringan Bi-LSTM sebagai berikut:



Gambar 2. 3 Arsitektur Bi-LSTM (Jang et al., 2020)

Secara khusus, pada setiap langkah waktu t , LSTM maju menghitung keadaan tersembunyi fh_t berdasarkan keadaan tersembunyi fh_{t-1} sebelumnya dan vektor input x_t , sedangkan LSTM mundur menghitung keadaan tersembunyi bh_t berdasarkan keadaan tersembunyi yang berlawanan bh_{t-1} dan vektor masukan x_t . Akhirnya, vektor dari kedua arah digabung sebagai status tersembunyi terakhir dari model BiLSTM. Dua parameter jaringan saraf LSTM di jaringan BiLSTM tidak bergantung satu sama lain dan berbagi embedding kata yang sama dari kalimat tersebut. Output akhir, h_t , dari model BiLSTM pada langkah t adalah sebagai berikut persamaan: (C. W. Chen et al., 2020)

$$h_t = [fh_t, bh_t] \quad (2.7)$$

Keterangan :

h_t = output hidden state

fh_t = Output forward lstm

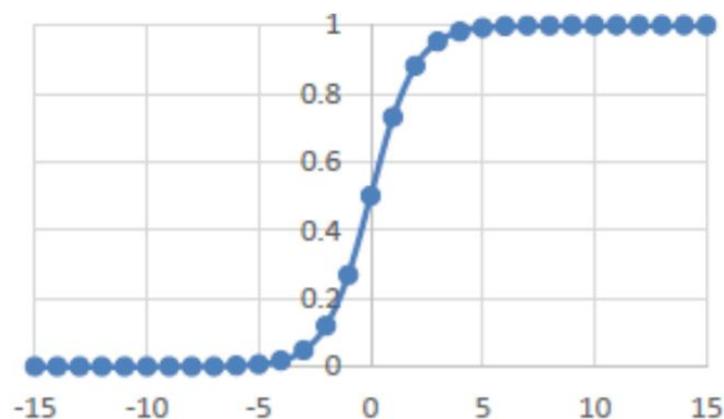
bh_t = Output backward lstm

2.7 Fungsi Aktivasi

Fungsi aktivasi merupakan modifikasi non linier yang digunakan untuk setiap neuron pada hidden layer guna mendapatkan output dalam bentuk yang telah dinormalisasi atau dalam rentang atau skala tertentu. Berikut merupakan fungsi aktivasi pada penelitian ini:

1. Sigmoid

Fungsi aktivasi sigmoid mentransformasikan range nilai dari input x menjadi antara 0 dan 1 dengan bentuk distribusi fungsi (P et al., 2016). Berikut adalah gambar dari distribusi fungsi sigmoid :



Gambar 2. 4 Distribusi Fungsi Sigmoid (P et al., 2016)

Dari Gambar 2.4 menunjukkan bahwa fungsi sigmoid memiliki bentuk perhitungan seperti dibawah ini :

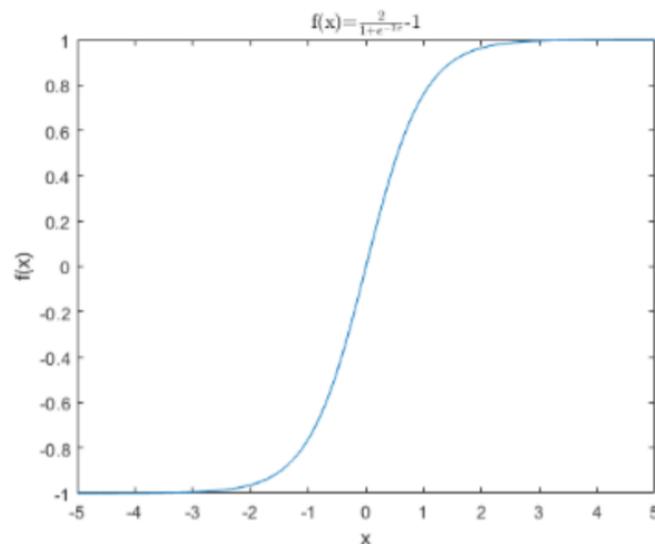
$$\sigma(x) = \frac{1}{(1+e^{-x})} \quad (2.8)$$

Dimana e merupakan sebuah konstanta atau ketetapan yang bernilai 2.71828. Jika input bernilai negatif sehingga hasil yang dikeluarkan adalah 0, sedangkan jika input bernilai positif maka hasil yang dikeluarkan 1. Fungsi aktivasi sigmoid mempunyai kelemahan pada range nilai output tidak terpusat pada angka nol (P et al., 2016).

2. Tanh

Fungsi Tanh juga dikenal sebagai Tangent Hyperbolic Function, yang dapat diperoleh dari fungsi sigmoid (Feng & Lu, 2019). Tanh adalah sebuah fungsi dengan pusat nol yang rentangnya terletak diantara -1 sampai 1, sehingga output dari fungsi tanh dapat didapatkan dengan persamaan berikut :

$$\tanh(x_i) = \frac{\sinh x_i}{\cosh x_i} = \frac{e^{x_i} - e^{-x_i}}{e^{x_i} + e^{-x_i}} = \frac{2}{1+e^{-2x_i}} - 1 \quad (2.9)$$



Gambar 2. 5 Fungsi Tanh (Feng & Lu, 2019)

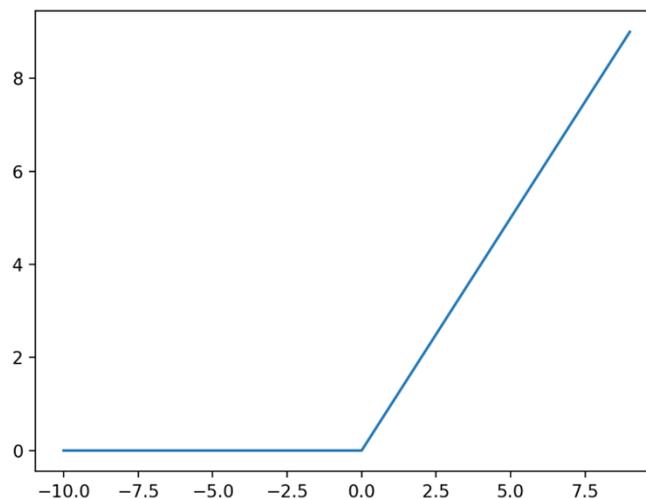
Fungsi tanh merupakan sebuah fungsi yang lebih digemari dibandingkan dengan fungsi sigmoid karena memberikan kinerja pelatihan yang lebih baik

untuk neural network multi layer (Nwankpa et al., 2018). Namun pada fungsi tanh tidak dapat menyelesaikan masalah vanishing gradient yang mana masalah tersebut juga dialami pada fungsi sigmoid (Nwankpa et al., 2018).

3. ReLU

ReLU atau Rectified Linear Unit adalah lapisan aktivasi yang mengambil nilai maksimal yang dapat di definisikan sebagai berikut :

$$f(x) = \max(0, x) \quad (2.9)$$



Gambar 2. 6 Fungsi ReLU

Selama proses training, beberapa neuron mati atau tidak aktif secara efektif sehingga mereka berhenti mengeluarkan apapun selain 0. Fungsi dari fungsi aktivasi ReLU adalah untuk mengembalikan nilai 0 jika inputnya negatif dan jika input positif maka ia mengembalikan nilai tersebut.

2.8 *Binary Cross Entrophy*

Pada penelitian ini menggunakan *loss function binary cross entropy* dimana loss standar yang dipergunakan pada masalah klasifikasi biner

mempunyai nilai sasaran berada pada set $\{0,1\}$ (Janocha & Czarnecki, 2016).

Berikut adalah persamaan rumus dari binary cross entropy :

$$CE = \begin{cases} -\log(p), & \text{if } y = 1 \\ -\log(1-p) & \text{sebaliknya} \end{cases} \quad (2.10)$$

Keterangan :

CE = *cross entropy*

p = probabilitas / *output layer*

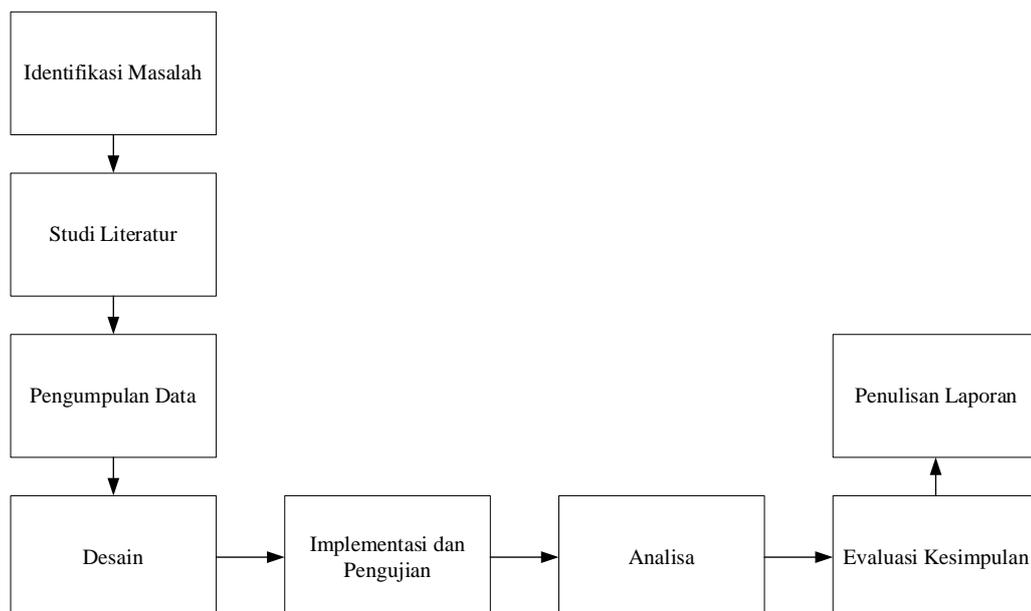
y = aktual

BAB III

METODOLOGI PENELITIAN

3.1 Desain Penelitian

Penelitian ini menggunakan pendekatan kuantitatif. Menurut Sugiyono (2014), metode penelitian kuantitatif dapat diartikan sebagai metode penelitian yang berlandaskan pada filsafat *positivisme*, digunakan untuk meneliti pada populasi atau sampel tertentu, teknik pengambilan sampel pada umumnya dilakukan secara acak, pengumpulan data menggunakan instrumen penelitian, analisis data bersifat kuantitatif atau statistik dengan tujuan untuk menguji hipotesis yang telah ditetapkan. Kegiatan penelitian ini bertujuan memudahkan masyarakat dalam mendeteksi SMS spam untuk meningkatkan kewaspadaan terhadap penipuan yang beredar di pesan teks. Berikut adalah gambaran prosedur penelitian dari penelitian:



Gambar 3. 1 Diagram Blok Prosedur Penelitian

Penjelasan dari Gambar 3.1 adalah :

1. Identifikasi Masalah

Tahap awal sebelum membangun sistem adalah melakukan identifikasi masalah yaitu tahap dimana menemukan masalah berdasarkan latar belakang penelitian.

2. Studi Literatur

Studi literatur yaitu mempelajari dan memahami teori-teori yang akan digunakan serta penelitian yang berhubungan dengan sistem yang akan digunakan di dalamnya. Studi literatur bisa didapatkan dari paper nasional maupun internasional, artikel dan buku yang terkait.

3. Pengumpulan Data

Melakukan pengumpulan data yang selanjutnya akan dikelola pada pembangunan sistem.

4. Desain

Melakukan pembuatan desain sistem yang sesuai dengan tujuan penelitian.

5. Implementasi dan Pengujian

Melakukan perubahan desain sistem menjadi kode program dan melakukan pengujian terhadap sistem.

6. Analisa

Melakukan Analisa terhadap sistem yang telah diuji.

7. Evaluasi dan Kesimpulan

Jika hasil dan pengujian telah sesuai, maka dilakukan tahap evaluasi dan kesimpulan dari penelitian sehingga dapat diketahui manfaat dan fungsi dari sistem serta perbaikan yang dapat dilakukan di masa mendatang.

8. Penulisan Laporan

Hal terakhir yang dilakukan adalah penulisan laporan mengenai sistem yang telah dibuat.

3.2 Pengumpulan Data

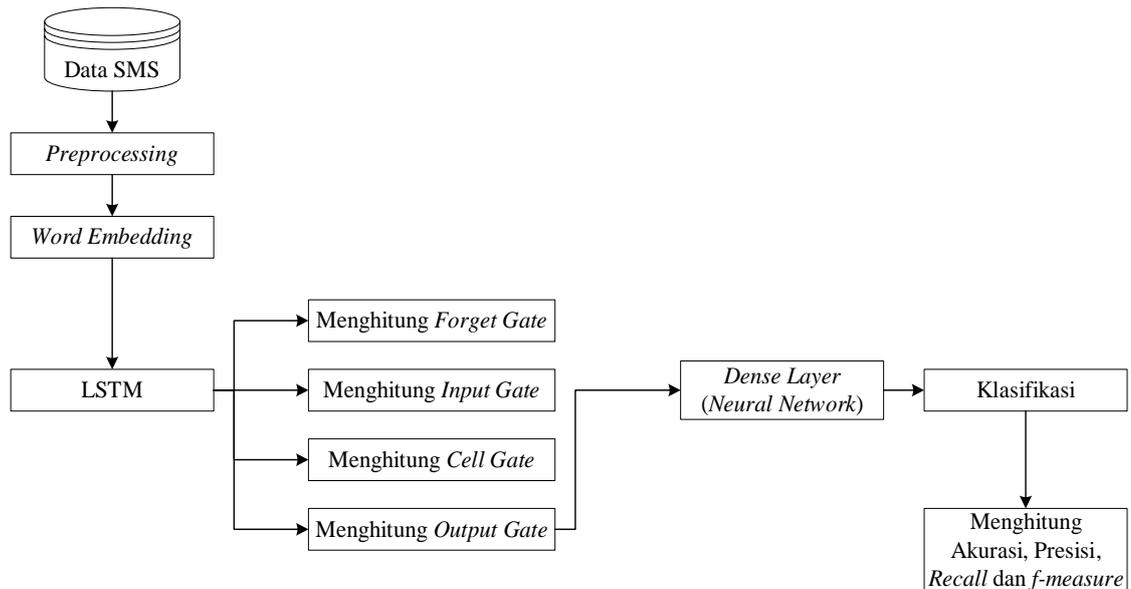
Sumber data yang digunakan dalam penelitian ini adalah data primer dan data sekunder. Data primer merupakan sumber data yang langsung memberikan data kepada pengumpul data sedangkan data sekunder merupakan sumber data tidak langsung memberikan data kepada pengumpul data (Sugiyono, 2017).

Pada data primer, data diperoleh dari pengguna operator atau pengguna SMS. Sumber data tersebut didapat dari aplikasi SMS Collector yang disediakan oleh peneliti sehingga pengguna sms hanya menginstall dan memilih SMS yang boleh dikirimkan kepada peneliti, sehingga peneliti hanya mendapatkan SMS yang diizinkan oleh responden. Pengumpulan data primer pada penelitian ini telah dilakukan pada tanggal 26 Desember 2020 sampai dengan tanggal 28 Desember 2020. Pengumpulan data tersebut diikuti oleh 62 Responden serta mendapatkan 41.397 data. Kemudian pada data sekunder, data diperoleh dari *repository* Institut Pertanian Bogor berjumlah 1.142 pesan.

Sebelum memasuki tahapan *preprocessing*, data tersebut digabungkan dan dilakukan pengecekan terhadap data duplikat yang kemudian data yang duplikat dihapus dan disisakan satu SMS saja. Kemudian data-data tersebut dipisahkan untuk data *spam* dan *non spam*. Data yang digunakan pada penelitian ini sebanyak 6514 dengan pembagian data sebanyak 3161 data *SMS spam* dan 3353 data *SMS non spam*. Kemudian data-data tersebut disimpan pada file dengan format csv.

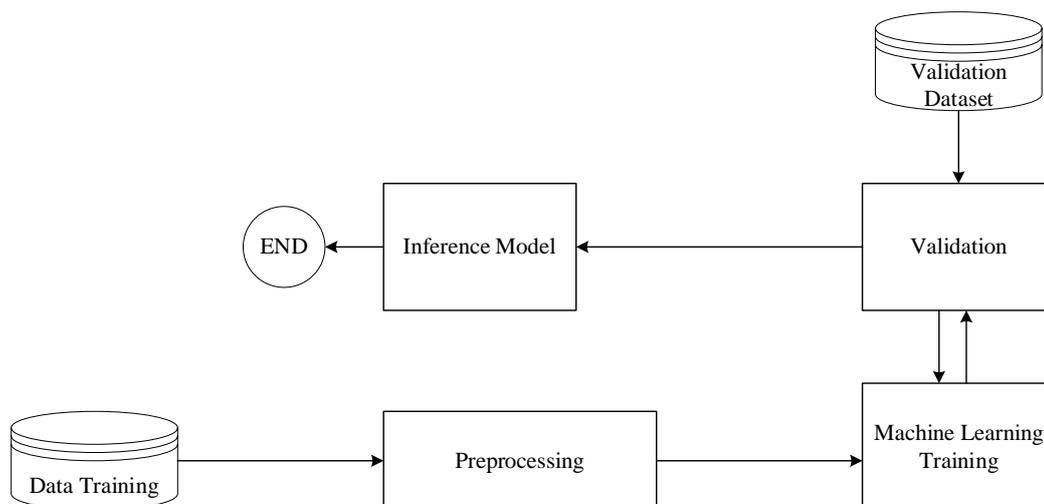
3.3 Desain Sistem

Desain sistem pada sistem *spam detection* menggunakan metode *Long Short Term Memory (LSTM)* seperti tercantum pada Gambar 3.2 dan Gambar 3.3, sebagai berikut :



Gambar 3. 2 Diagram Blok Perancangan Sistem

Pada Gambar 3.2 diketahui bahwa input berupa data sms yang akan diproses pada beberapa tahap yaitu *preprocessing* yang kemudian akan dibentuk model dengan menggunakan metode *Long Short Term Memory*. Dari model tersebut akan dilakukan evaluasi dengan menghitung akurasi,presisi, *recall* dan *f-measure*.



Gambar 3. 3 Data Flow Training dan Validation

Berdasarkan Gambar 3.3, dataset terbagi menjadi 80% data training dan 20% data validation. Pada Gambar 3.3 menjelaskan terkait alur data atau data flow pada proses *training* dimana data *training* atau data latih berisi data sms yang telah dikumpulkan oleh peneliti yang kemudian data tersebut masuk pada tahap *preprocessing* dimana data training digunakan untuk pembuatan *dictionary* yang kemudian *dictionary* tersebut digunakan pada proses *preprocessing*, kemudian hasil dari *preprocessing* masuk kedalam proses *machine learning* dimana pada *machine learning* menghasilkan sebuah prediksi, kemudian hasil prediksi masuk pada tahap *validation* dimana pada tahap *validation* dilakukan pengujian dari hasil prediksi machine learning dengan *dataset validation*.

3.4 *Preprocessing*

Pada penelitian yang dilakukan Poomka *et al* (2019), *preprocessing* pada metode LSTM dilakukan sebagai berikut :

1. Melakukan standarisasi yaitu mengubah semua kata menjadi *lower case* atau huruf kecil (*case folding*) dan *punctuation removal* atau menghapus tanda baca. Case Folding bertujuan untuk menyamakan semua jenis karakter yang ada pada SMS sehingga dapat memudahkan dalam proses penghapusan karakter atau istilah-istilah eksklusif yang tidak diinginkan pada penelitian ini. Tanda baca yang dimaksud merupakan tanda baca seperti ? ! , / = + - \ > < ; “ () { } [] . : | dan lainnya. Menghapus tanda baca dilakukan karena tanda baca tidak dihiraukan selama proses training sehingga dengan melakukan penghapusan tanda baca proses training akan menjadi lebih sederhana.

Tabel 3. 2 Standarisasi Kata

Sebelum Standarisasi	Setelah Standarisasi
ASS, KAMI MENAWARKAN PINJAMAN ONLINE DI SELURUH WILAYAH INDONESIA MINIMAL RP. 5JT-500JT INFO CHAT Whatsapp 082316715651	ass kami menawarkan pinjaman online di seluruh wilayah indonesia rp 5jt 500jt info chat whatsapp 082316715651

2. Memisahkan kalimat menjadi kata berdasarkan karakter dalam bentuk array satu dimensi.

Tabel 3. 3 Pemisahan Kata Menjadi Array

Sebelum Pemisahan Kata	Setelah Pemisahan Kata
ass kami menawarkan pinjaman online di seluruh wilayah indonesia rp 5jt 500jt info chat whatsapp 082316715651	[“ass”, “kami”, “menawarkan”, “pinjaman”, “online”, “di”, “seluruh”, “wilayah”, “indonesia”, “rp”, “5jt”, “500jt”, “info”, “chat”, “whatsapp”, “082316715651”]

3. Proses selanjutnya adalah *Word Tokenization* atau tokenisasi kata merupakan sebuah proses yang mengubah kata di dalam suatu kalimat menjadi nilai indeks yang berbentuk angka. Pada proses ini, nilai indeks berasal dari *vocabulary* yang telah dibuat dimana penomoran indeks berdasarkan kata yang sering muncul pada dataset untuk membuat tokenisasi kata. Tokenizer mengubah kata menjadi indeks dan mengatur indeks menjadi 0 untuk kata-kata yang tidak dikenal.

Tabel 3. 1 *Word Tokenization*

Sebelum diberi nilai indeks	Setelah diberikan nilai indeks
[“ass”, “kami”, “menawarkan”, “pinjaman”, “online”, “di”, “seluruh”, “wilayah”, “indonesia”, “rp”, “5jt”, “500jt”, “info”, “chat”, “whatsapp”, “082316715651”]	[290 1895 82 4565 79 3 347 747 208 283 29 163 428 7 28 149 1352]

4. Setelah kata di tokenisasi kemudian berlanjut pada proses *padding and truncating data* atau mengisi dan memotong data. Hal tersebut dilakukan karena pada dataset, setiap pesan teks memiliki jumlah kata yang berbeda-beda sehingga panjang vector harus sama. Maka dari itu dilakukan padding data atau mengisi data jika panjang pesan teks kurang dari panjang maksimal, sehingga agar panjang vektor sama, vektor diisi dengan angka 0 untuk mengisi urutan kata yang kosong. *Truncating data* atau pemotongan data dilakukan jika panjang pesan melebihi panjang maksimal, sehingga agar panjang vektor sama dilakukan pemotongan data diawal atau diakhir agar panjangnya sama dengan panjang maksimal. Pada penelitian ini menggunakan panjang vektor dengan jumlah 64. Struktur data yang terbentuk pada proses ini merupakan masukan yang digunakan di *layer word embedding*.

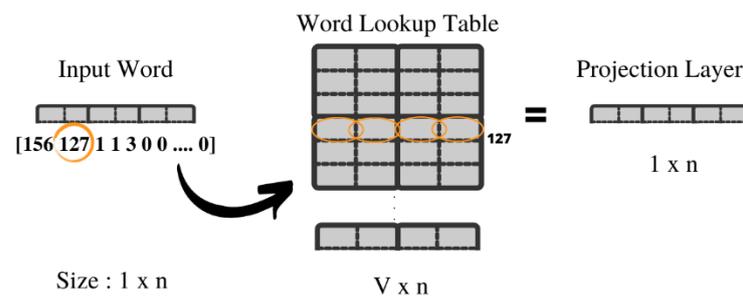
Model LSTM yang digunakan pada penelitian ini adalah LSTM Bi-Directional sesuai dengan ilustrasi Gambar 3.4. Pada layer pertama merupakan sebuah input yang kemudian terdapat proses *embedding* yang dilanjutkan oleh LSTM Bi-Directional pada layer kedua dan ketiga, yang kemudian pada layer keempat dan kelima merupakan *dense layer* dengan aktivasi ReLU dilanjutkan pada layer terakhir dengan menggunakan fungsi aktivasi sigmoid.

3.5.1.1 Word Embedding

Berdasarkan Gambar 3.4 Arsitektur Jaringan, pada layer pertama yang digunakan sebagai input. Layer pertama pada arsitektur jaringan adalah embedding yaitu teknik vektorisasi teks, dan teknik ini mengubah kata-kata dalam kosakata menjadi vektor bilangan real kontinu (Augustyniak et al., 2019) . Embedding pada neural network berguna dikarenakan dapat mengurangi dimensi variabel mutlak dan merepresentasikan kategori dengan merubah suatu jarak tanpa merubah makna dari jarak tersebut. Tujuan dari layer pertama ini adalah untuk mempelajari pemetaan yang menyematkan setiap kata dalam kosa kata diskrit ke ruang vector berdimensi lebih rendah (Rao & Spasojevic, 2016).

Menurut Poomka et al , preprocessing data dilakukan pada layer input dimana sebelum mendapatkan output dari word embedding harus melewati proses yang telah disebutkan dan jelaskan pada sub bab *preprocessing*. Teknik ini digunakan untuk mengubah urutan kata yang sudah di preprocessing menjadi representasi vector yang dapat disebut dengan embedding space yang berisi lebih banyak dimensi dari pada dimensi kata normal yang akan digunakan untuk melatih algoritma LSTM (Poomka et al., 2019). Setelah dilakukan *padding* dan

truncating kata, peneliti menggunakan *word embedding* untuk membuat lebih banyak dimensi pada data berurutan dengan mengatur *embedding size* 64. Proses *word embedding* dapat dilihat pada gambar dibawah ini :



Gambar 3. 5 *Word Embedding*

Pada Gambatr 3.6 menunjukkan proses *word embedding* dari satu kata atau satu token. Input kata direpresentasikan oleh hasil dari *padding* dan *truncating* pada sebuah kalimat, V adalah jumlah dari *vocabulary* dan n adalah jumlah kata pada input atau panjang vektor input.

Sebelum dilakukan lookup, data dilakukan normalisasi *standard score* dengan rumus seperti dibawah ini :

$$z = \frac{x - \mu}{\sigma} \quad (3.1)$$

Keterangan :

x = input

μ = mean

σ = standar deviasi

Kemudian setelah dinormalisasi setiap data yang telah dinormalisasi dikalikan dengan nilai bobot sejumlah output dim dengan nilai bobot lookup

konstant atau tetap, sehingga pada lookup tabel memiliki bentuk jumlah vocab x output_dim.

Pada Gambar 3.6 terdapat *projection layer* yang berarti *output layer* pada setiap kata. Pada Gambar 3.6 terlihat bahwa *projection layer* adalah *output* dari *vocabulary* yang memiliki token 127, sehingga output dari token 127 dapat direpresentasikan oleh 1xn. Output dari layer ini memiliki shape (64,64) dimana setiap kata direpresentasikan dengan panjang vektor 64 dan pada kalimat memiliki panjang vektor 64, sehingga output pada layer ini adalah (batch_size, input_length, output_dim). Berikut merupakan contoh dari hasil *word embedding* dengan *shape* (1,64,64):

Tabel 3. 3 Contoh *Output Embedding Layer*

Contoh Output Embedding Layer
[[[-0.00718591 -0.03526856 -0.02290094 ... -0.03731718 -0.04376619 -0.03987709]
[0.03615633 0.01801633 0.006908 ... 0.0210815 0.02665273 -0.04083342]
[-0.01264602 0.03180597 0.03683798 ... 0.02611795 -0.00011413 -0.04658569]
...
[0.02824035 0.03974258 -0.0112664 ... -0.04357586 0.04489925 -0.00616703]
[-0.01650316 -0.0120414 0.00780848 ... 0.00887295 0.02469072 -0.03522788]
[-0.02298703 0.00369114 0.04687988 ... -0.04709866 -0.03633734 0.00925423]]

3.5.1.2 Long Short Term Memory (LSTM)

Pada penelitian ini menggunakan *Bidirectional LSTM*. *Bidirectional Long Short Term Memory* atau biLSTM adalah model pemrosesan berurutan yang terdiri dari dua LSTM : satu mengambil input dalam arah maju (*forward direction*), dan yang lainnya di arah belakang (*backward direction*). Bi-LSTM

meningkatkan jumlah informasi yang tersedia untuk jaringan, meningkatkan konteks yang tersedia pada algoritma (misalnya, mengetahui kata apa yang segera mengikuti dan mendahului kata dalam kalimat) (Cornegruta et al., 2016).

Pada penelitian ini dapat dilihat pada gambar 3.4 terkait arsitektur, dalam mendapatkan output terdapat beberapa layer setelah layer word embedding yaitu :

Layer kedua merupakan bi-LSTM layer dengan menggunakan 32 unit neuron. Pada layer ini, input yang digunakan berupa hasil dari layer sebelumnya yaitu layer word embedding. Pada bab sebelumnya telah dijelaskan terkait LSTM dan Bi-LSTM.

Pada LSTM terdapat empat gerbang yaitu : forget gates, input gates, cell gates, dan output gates. Sebelum menghitung empat gerbang tersebut, berikut adalah parameter yang akan digunakan untuk perhitungan pada empat gerbang tersebut :

Tabel 3. 4 Tabel Contoh Bobot dan Bias

W_{xf}	W_{hf}	b_f	W_{xi}	W_{hi}	b_i	W_{xc}	W_{hc}	b_c	W_{xo}	W_{ho}	b_o
0.7	0.15	0.2	0.95	0.8	0.65	0.45	0.15	0.2	0.6	0.25	0.1
0.45			0.8			0.25			0.4		

Diketahui $x_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, $x_2 = \begin{bmatrix} 0.5 \\ 3 \end{bmatrix}$ dimana nilai input tersebut adalah hasil dari proses word embedding atau layer pertama. Matriks W_h dan b memiliki ukuran 1×1 dimana pada ukuran W_h didapat dari hidden neuron dikalikan jumlah neuron dan pada b memiliki ukuran 1 dikali hidden neuron sedangkan matriks W_x merupakan bobot dari x sehingga memiliki ukuran yang sama dengan x . Selain parameter diatas diatas juga dibutuhkan nilai h_{t-1} dan C_{t-1} . Pada perhitungan ini, $h_{t-1} = [0]$ dan $C_{t-1} = [0]$ dikarenakan belum ada proses dan

nilai sebelumnya atau perhitungan ini merupakan $t = 1$. Setelah parameter yang dibutuhkan terpenuhi, berikut merupakan contoh perhitungan metode LSTM :

1. Forget Gates

Berdasarkan persamaan 2.1 dan parameter yang telah diketahui, forget gates dapat dihitung seperti dibawah ini :

$$f_1 = \sigma(W_{xf} \cdot x_t + W_{hf} \cdot h_{t-1} + b_f)$$

$$f_1 = \sigma([0.7 \ 0.45] \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix} + [0.1] \cdot [0] + [0.15])$$

$$f_1 = \sigma([1.6] + [0.15])$$

$$f_1 = \sigma([1.75])$$

$$f_1 = \frac{1}{(1 + e^{-1.75})}$$

$$f_1 = [0.85195]$$

2. Input Gates

Berdasarkan persamaan 2.2 dan parameter yang telah diketahui, input gates dapat dihitung seperti dibawah ini :

$$i_1 = \sigma(W_{xi} \cdot x_t + W_{hi} \cdot h_{t-1} + b_i)$$

$$i_1 = \sigma([0.95 \ 0.8] \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix} + [0.8] \cdot [0] + [0.65])$$

$$i_1 = \sigma([2.55] + [0.65])$$

$$i_1 = \sigma([3.2])$$

$$i_1 = \frac{1}{(1 + e^{-3.2})}$$

$$i_1 = [0.96083]$$

Kemudian menghitung fungsi aktivasi tanh dimana hasil dari perhitungan fungsi aktivasi tanh akan digunakan untuk perhitungan cell gates.

$$\check{c}_1 = \tanh(W_{xc} \cdot x_c + W_{hc} \cdot h_{t-1} + b_c)$$

$$\check{c}_1 = \tanh([0.45 \ 0.25] \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix} + [0.15] \cdot [0] + [0.2])$$

$$\check{c}_1 = \tanh([0.95] + [0.2])$$

$$\check{c}_1 = \tanh([1.15])$$

$$\check{c}_1 = \frac{2}{(1+e^{-2 \cdot 0.21})} - 1$$

$$\check{c}_1 = [0.81775]$$

3. Cell Gates

Berdasarkan persamaan 2.4 dan parameter yang telah diketahui, cell gates dapat dihitung seperti dibawah ini :

$$c_1 = f_t * c_{t-1} + i_t * \check{c}_t$$

$$c_1 = [0.85195] * [0] + [0.96083] * [0.81775]$$

$$c_1 = [0.78572]$$

4. Output Gates

Berdasarkan persamaan 2.5 dan parameter yang telah diketahui, output gates dapat dihitung seperti dibawah ini :

$$o_1 = \sigma(W_{xo} \cdot x_t + W_{ho} \cdot h_{t-1} + b_o)$$

$$o_1 = \sigma([0.6 \ 0.4] \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix} + [0.25] \cdot [0] + [0.1])$$

$$o_1 = \sigma([1.4] + [0.1])$$

$$o_1 = \sigma([1.5])$$

$$o_1 = \frac{1}{(1 + e^{-0.12})}$$

$$o_1 = [0.81757]$$

Kemudian menghitung nilai pada *memory cell* menggunakan fungsi aktivasi tanh dan dikalikan dengan hasil o_t sehingga menghasilkan nilai yang akan dikeluarkan.

$$h_1 = o_t * \tanh(c_t)$$

$$h_1 = [0.81757] * \tanh([0.78572])$$

$$h_1 = [0.53631]$$

Pada perhitungan diatas di dapatkan output $h_t = [0.53631]$ dan $c_t = [0.78572]$, kedua output tersebut akan digunakan untuk time step berikutnya sampai dengan time step maksimal. Berikut merupakan contoh perhitungan untuk $t=2$:

$$f_2 = \sigma([0.7 \ 0.45] \cdot \begin{bmatrix} 0.5 \\ 3 \end{bmatrix} + [0.1] \cdot [0.53631] + [0.15]) = [0.87030]$$

$$i_2 = \sigma([0.95 \ 0.8] \cdot \begin{bmatrix} 0.5 \\ 3 \end{bmatrix} + [0.8] \cdot [0.53631] + [0.65]) = [0.98118]$$

$$\check{c}_{t_2} = \tanh([0.45 \ 0.25] \cdot \begin{bmatrix} 0.5 \\ 3 \end{bmatrix} + [0.15] \cdot [0.53631] + [0.2]) = [0.84980]$$

$$c_2 = [0.87030] * [0.78572] + [0.84980] * [0.98118] = [1.5176]$$

$$o_2 = \sigma([0.6 \ 0.4] \cdot \begin{bmatrix} 0.5 \\ 3 \end{bmatrix} + [0.25] \cdot [0.53631] + [0.1]) = [0.84993]$$

$$h_2 = [0.84993] * \tanh([1.5176]) = [0.77197]$$

Contoh perhitungan diatas merupakan contoh perhitungan LSTM menggunakan proses *forward* dimana proses input dimulai pada x pertama, pada proses *backward* digunakan persamaan yang sama dimulai dengan input x_t atau

input terakhir pada t . Pada pemrograman, input pada proses *backward* dapat diinisialisasikan seperti dibawah ini :

```
self.input_reserve = np.flip(x, axis=1)
```

Gambar 3. 6 Inisialisasi Input Backward LSTM

Pada Gambar 3.5 terdapat *syntax* flip yang berarti x pada proses *forward* dibalik urutannya sebagai input *backward*. Contoh perhitungan pada proses *backward* seperti dibawah ini :

$$f_1 = \sigma([0.7 \ 0.45] \cdot \begin{bmatrix} 0.5 \\ 3 \end{bmatrix} + [0.1] \cdot [0] + [0.15]) = [0.86412]$$

$$i_1 = \sigma([0.95 \ 0.8] \cdot \begin{bmatrix} 0.5 \\ 3 \end{bmatrix} + [0.8] \cdot [0] + [0.65]) = [0.97527]$$

$$\check{c}_{t_1} = \tanh([0.45 \ 0.25] \cdot \begin{bmatrix} 0.5 \\ 3 \end{bmatrix} + [0.15] \cdot [0] + [0.2]) = [0.82586]$$

$$c_1 = [0.87030] * [0] + [0.84980] * [0.98118] = [0.83380]$$

$$o_1 = \sigma([0.6 \ 0.4] \cdot \begin{bmatrix} 0.5 \\ 3 \end{bmatrix} + [0.25] \cdot [0] + [0.1]) = [0.83201]$$

$$h_1 = [0.84993] * \tanh([1.5176]) = [0.56786]$$

Maka output dari proses *backward* t_1 adalah $h_1 = [0.56786]$ dan $c_1 = [0.83380]$ dan kedua nilai tersebut digunakan untuk time step berikutnya sampai dengan time step maksimal. Berikut merupakan contoh perhitungan untuk $t=2$:

$$f_2 = \sigma([0.7 \ 0.45] \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix} + [0.1] \cdot [0.56786] + [0.15]) = [0.85897]$$

$$i_2 = \sigma([0.95 \ 0.8] \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix} + [0.8] \cdot [0.56786] + [0.65]) = [0.97477]$$

$$\check{c}_{t_2} = \tanh([0.45 \ 0.25] \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix} + [0.15] \cdot [0.56786] + [0.2]) = [2.86686]$$

$$c_2 = [0.87030] * [0.83380] + [0.84980] * [0.98118] = [3.51075]$$

$$o_2 = \sigma([0.6 \ 0.4] \cdot \begin{bmatrix} 1 \\ 2 \end{bmatrix}) + [0.25] \cdot [0.56786] + [0.1] = [0.83780]$$

$$h_2 = [0.84993] * \tanh([1.5176]) = [0.83630]$$

Pada persamaan diatas dapat disimpulkan bahwa perbedaan *forward direction* dan *backward direction* terletak pada nilai x atau input dimana x_1 pada *backward direction* merupakan x terakhir pada *forward direction*. Berdasarkan persamaan 2.7 terkait output dari *Bidirectional LSTM* dapat diinisialisasikan seperti dibawah ini :

```
np.concatenate((forward, backward), axis=-1)
```

Gambar 3. 7 Inisialisasi *Output Bi-LSTM*

Pada Gambar 3.6 terdapat *syntax concatenate* yang berarti bahwa output dari Bi-LSTM merupakan gabungan dari *output* proses *forward* dan *backward*. Pada contoh perhitungan *forward direction* dan *backward direction* dapat disimpulkan bahwa output dari Bi-LSTM adalah

$$h_1 = [fh_1, bh_1] = [0.53631 \ 0.56786]$$

$$h_2 = [fh_{t2}, bh_2] = [0.77197 \ 0.83630]$$

Pada layer ketiga juga merupakan layer Bi-LSTM dengan jumlah neuron berbeda yaitu 16. Output dari layer kedua menjadi input pada layer ketiga, dimana dilakukan perhitungan atau proses yang sama seperti layer sebelumnya.

3.5.1.3 ReLU

Pada Gambar 3.4 terkait arsitektur jaringan terdapat layer dense pada layer keempat. Layer dense yang dimaksud pada arsitektur tersebut adalah layer neural network dengan penambahan fungsi aktivasi ReLU. Fungsi aktivasi pada layer ini

digunakan untuk mengembalikan nilai 0 jika inputnya negatif dan jika input positif maka ia mengembalikan nilai tersebut.

Diketahui hidden $i = 1$, $x_i = [-0.524 \ 0.850 \ 0.345 \ -0.053]$, $W_i = [0.146 \ 0.238 \ 0.651 \ 0.421]$, $b_i=[0.15]$. Berikut merupakan perhitungan dari layer dense dengan aktivasi ReLU :

$$i = \left(\sum_{i=1}^n (W_{ij}x_i + b_{ij}) \right) \quad (3.3)$$

$$i = [0.146 \ 0.238 \ 0.651 \ 0.421] * [-0.524 \ 0.850 \ 0.345 \ -0.053] + [0.15]$$

$$i = [-0.378 \ 0.199 \ 0.224 \ -0.022] + [0.15]$$

$$i = [-0.228 \ 0.349 \ 0.374 \ 0.128]$$

Kemudian nilai x dimasukkan pada persamaan 2.9.

$$f(-0.228) = \max(0, -0.228) = 0$$

$$f(0.349) = \max(0, 0.349) = 0.349$$

$$f(0.374) = \max(0, 0.374) = 0.374$$

$$f(0.128) = \max(0, 0.128) = 0.128$$

Dari hasil diatas dapat disimpulkan output dari layer keempat adalah **[0 0.349 0.374 0.128]** .

3.5.1.4 Sigmoid

Pada Gambar 3.4 terkait arsitektur jaringan terdapat *layer dense* pada layer kelima. *Layer dense* yang dimaksud pada arsitektur tersebut adalah *layer neural network* dengan penambahan fungsi aktivasi Sigmoid yang mana pada Gambar 3.4 menunjukkan bahwa *layer* kelima ini adalah *layer output*. Fungsi aktivasi pada *layer* ini digunakan untuk menghasilkan output atau prediksi dari model yang telah dibangun. Pada penelitian ini bertujuan untuk mengetahui sms spam

atau non spam, dikarenakan fungsi sigmoid berada diantara 0 sampai 1 maka fungsi aktivasi sigmoid adalah pilihan yang tepat untuk klasifikasi biner.

Diketahui *input* pada *layer* kelima merupakan *output* dari *layer* keempat, *hidden layer* pada *layer* ini adalah 1, dan $b=[0.1]$. Berikut perhitungan untuk mendapatkan nilai *output* :

$$y = \sigma \left(\sum_{i=1}^n (W_{ij}x_i + b_j) \right) \quad (3.4)$$

$$y = \sigma ([0.104] + [0.1])$$

$$y = \sigma ([0.204])$$

$$y = \frac{1}{(1 + e^{-0.204})}$$

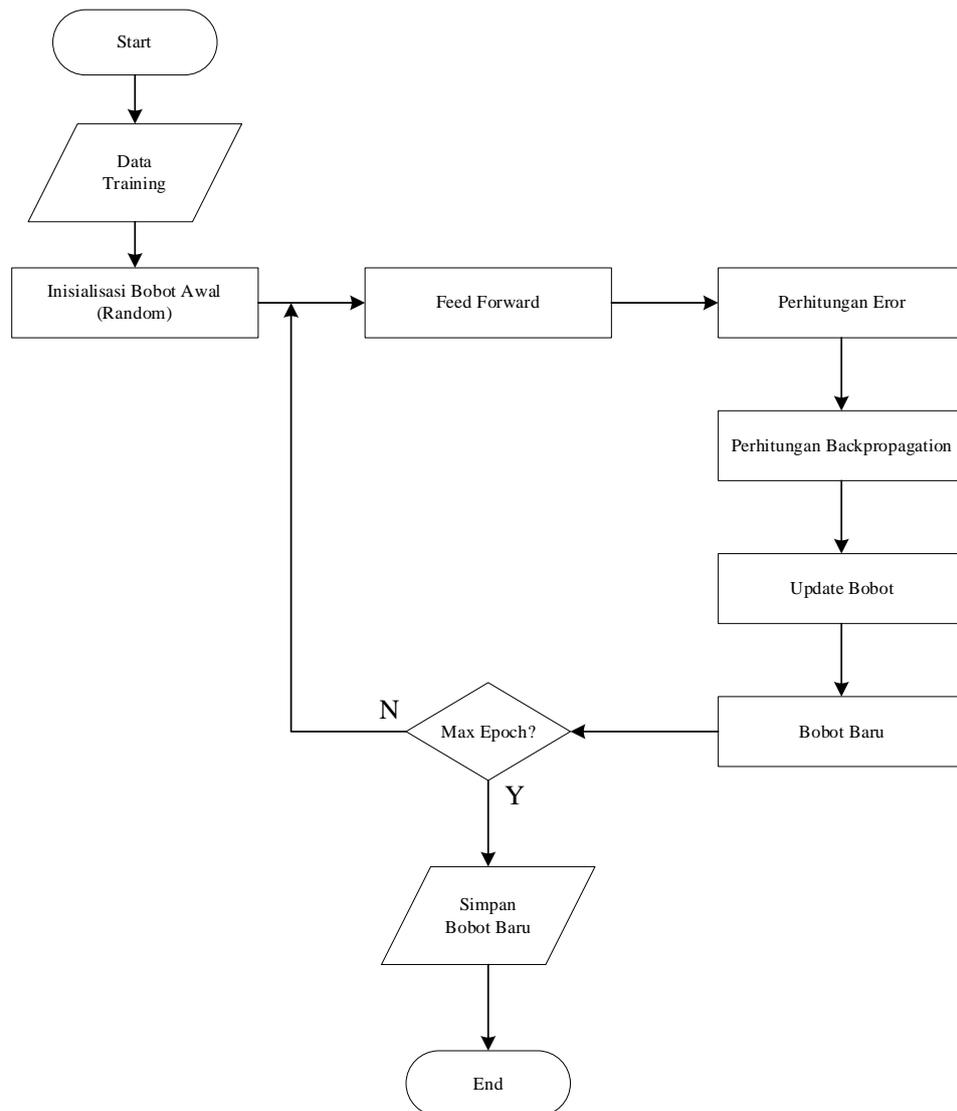
$$y = [0.55]$$

Pada *layer* ini dikatakan 0 atau *non spam* jika nilai *output* pada *layer* *output* dibawah 0.5 dan jika dikatakan 1 atau *spam* maka nilai *output* pada *layer* *output* adalah diatas 0.5.

Pada penelitian ini merupakan *binary classification*, maka nilai atau hasil dari perhitungan diatas dalam menghitung nilai loss menggunakan loss fungsi *binary cross entropy* dan dengan *learning rate* sebesar 0.01. *Batch size* yang digunakan adalah 128 dan *epoch* berjumlah 16 dan untuk mengetahui nilai akurasi terhadap proses training menggunakan fungsi '*accuracy*' yang digunakan untuk menghitung tingkat akurasi training pada model dan *confusion matrix* pada data validasi untuk mengukur akurasi, presisi, *recall* dan *f-measure*.

3.5.2 Proses Training

Berdasarkan arsitektur jaringan pada Gambar 3.3, maka alur proses training pada penelitian ini sebagai berikut :



Gambar 3. 8 *Flowchart Training*

Pada Gambar 3.7 merupakan alur dari proses training model pada penelitian ini, dimana nilai bobot diinisialisasi random kemudian dilanjutkan dengan proses *feed forward* yang mana telah dijelaskan pada sub bab 3.5.1 Pada

penelitian ini menggunakan maksimum epoch sebanyak 16 yang mana pada proses training akan mencari nilai optimal untuk bobot.

Berdasarkan penjelasan pada sub bab sebelumnya, perhitungan error pada penelitian ini menggunakan *binary cross entropy* loss standar yang dipergunakan pada masalah klasifikasi biner mempunyai nilai sasaran berada pada set $\{0,1\}$ (Janocha & Czarnecki, 2016).

$$CE = \begin{cases} -\log(p), & \text{if } y = 1 \\ -\log(1-p) & \text{sebaliknya} \end{cases}$$

Dimana p merupakan probabilitas atau *output layer* dan y merupakan nilai aktual atau *groundtruth*. Pada persamaan diatas berarti jika *groundtruth* memiliki nilai 1 maka persamaan yang digunakan adalah $-\log(p)$ dan jika jika *groundtruth* memiliki nilai 0 maka persamaan yang digunakan adalah $-\log(1-p)$.

Kemudian setelah menghitung nilai eror dilakukan perhitungan *backpropagation* yang mana pada proses ini dimaksudkan untuk melakukan perhitungan balik dari output neuron agar memiliki nilai bobot yang sesuai pada jaringan. Berdasarkan penjelasan tersebut dapat disimpulkan bahwa tujuan dari perhitungan *backpropagation* adalah untuk memodifikasi bobot dalam melatih jaringan neural untuk memetakan input ke output dengan benar serta untuk mempelajari bobot dalam keterkaitannya pada jaringan berlapis lapis agar memiliki nilai eror minimum (Cynthia & Ismanto, 2017).

Pada proses *backpropagation*, hal pertama yang dilakukan adalah dengan menghitung turunan dari *loss*, dimana turunan pada *binary cross entropy* dapat diinisialisasikan sebagai berikut :

```
np.where(y_true == 1, -1 / y_pred, 1 / (1 - y_pred))
```

Gambar 3. 9 Inisialisasi turunan *Binary Cross Entropy*

Kemudian mencari turunan dari fungsi aktivasi yang digunakan dimana pada penelitian ini, pada layer output menggunakan aktivasi sigmoid dan aktivasi ReLU pada layer keempat. Berikut inisialisasi turunan dari aktivasi yang digunakan:

```
class Sigmoid(Activation):
    def forward_activation(self, x):
        return 1 / (1 + np.exp(-x))

    def backward_activation(self, x):
        a = self.forward_activation(x)
        return a * (1 - a)
```

Gambar 3. 10 Inisialisasi Turunan Sigmoid

```
class ReLU(Activation):
    def forward_activation(self, x):
        # dimana x lebih dari 0 nilai tetap x dan kurang dari 0 nilai 0
        return np.where(x > 0, x, 0)

    def backward_activation(self, x):
        # dimana x lebih dari 0 nilai 1 dan kurang dari 0 nilai 0
        return np.where(x > 0, 1, 0)
```

Gambar 3. 11 Inisialisasi Turunan ReLU

Pada *dense layer* perhitungan *backpropagation* yang digunakan menggunakan teknik *chain rule* yaitu dengan persamaan berikut:

$$\frac{\delta error}{\delta W} = \frac{\delta error}{\delta prediksi} * \frac{\delta prediksi}{\delta W} \quad (3.5)$$

Keterangan :

$\delta error$ = turunan eror

δW = turunan bobot

$\delta prediksi$ = turunan prediksi atau output sistem

Kemudian pada *dense layer*, perhitungan pembaruan bobot dengan menggunakan *learning rate* sebesar 0.01 dapat direpresentasikan pada persamaan berikut:

$$W^{new} = W^{old} - \lambda * \left(\frac{\delta error}{\delta W} \right) \quad (3.6)$$

Keterangan :

W^{new} = bobot baru

W^{old} = bobot lama

λ = *learning rate*

$\frac{\delta error}{\delta W}$ = perhitungan gradien/eror pada eror terhadap bobot

Pada metode LSTM, algoritma *backpropagation* yang digunakan adalah *Backpropagation Through Time* (BPTT) dimana BPTT merupakan sebuah algoritma *backpropagation* pada jaringan *Recurrent Neural Network* (RNN) (K. Chen et al., 2015). Pada LSTM, *backpropagation* dilakukan dengan menghitung turunan pada setiap *gate* nya dan dapat direpresentasikan pada persamaan berikut :

1. Menghitung nilai turunan pada *hidden state* dan *output gate*.

$$\delta h_t = \Delta_t + \Delta h_t \quad (3.7)$$

$$\delta o_t = \delta h_t * \tanh(c_t) * o_t * (1 - o_t) \quad (3.8)$$

Keterangan:

δh_t = turunan *hidden state*

Δ_t = error ke-t

Δh_t = error hidden state ke-t

δo_t = turunan *output state*

tanh = fungsi aktivasi tanh

o_t = *output gate*

2. Menghitung nilai turunan pada *cell state*.

$$\delta c_t = \delta h_t * o_t * (1 - \tanh^2(c_t)) + \delta c_{t+1} * f_{t+1} \quad (3.9)$$

Keterangan:

δc_t = turunan *cell state*

δh_t = turunan *hidden state*

o_t = *output gate* ke-t

\tanh = fungsi aktivasi \tanh

c_t = *cell state* ke-t

δc_{t+1} = turunan *cell state* selanjutnya

f_{t+1} = *forget gate* selanjutnya

3. Menghitung nilai turunan *input gate*, dimana pada bab sebelumnya telah dijelaskan bahwa *input gate* memiliki dua aktivasi, *cell* aktivasi dengan aktivasi \tanh dan *input gate* dengan aktivasi sigmoid.

$$\delta \check{c}_t = \delta c_t * i_t * (1 - \check{c}_t^2) \quad (3.10)$$

$$\delta i_t = \delta c_t * \check{c}_t * i_t * (1 - i_t) \quad (3.11)$$

Keterangan:

$\delta \check{c}_t$ = turunan *cell* aktivasi

δc_t = turunan *cell state*

i_t = *input gate*

δi_t = turunan *input gate*

\check{c}_t = *cell* aktivasi ke-t

4. Menghitung nilai turunan *forget gate*.

$$\delta f_t = \delta c_t * c_{t-1} * f_t * (1 - f_t) \quad (3.12)$$

Keterangan:

δf_t = turunan *forget gate*

δc_t = turunan *cell state*

c_{t-1} = *cell state* sebelumnya

f_t = *forget gate*

5. Menghitung turunan *input*.

$$\delta x_t = w_x^T * \delta gates_t \quad (3.13)$$

Keterangan:

δx_t = turunan *input*

w_x^T = hasil transpose matriks bobot *input*

$\delta gates$ = turunan *output* disetiap *gates*

6. Menghitung error *hidden state* sebelum.

$$\Delta h_{t-1} = w_h^T * \delta gates_t \quad (3.14)$$

Keterangan :

Δh_{t-1} = error *hidden state* sebelum

w_h^T = hasil transpose matriks bobot *hidden state*

$\delta gates$ = turunan *output* disetiap *gates*

Perhitungan *backpropagation* pada LSTM telah selesai, kemudian dilakukan perhitungan berapa banyak bobot yang akan berubah dengan persamaan berikut :

$$\delta W_x = \sum_{t=0}^T \delta gates * x_t \quad (3.15)$$

$$\delta W_h = \sum_{t=0}^T \delta gates * h_t \quad (3.16)$$

$$\delta b = \sum_{t=0}^T \delta gates_{t+1} \quad (3.17)$$

Keterangan :

δW_x = turunan bobot x

δW_h = turunan bobot *hidden state*

δb = turunan bias

$\delta gates$ = turunan *output* disetiap gates

x_t = input

h_t = *hidden state*

b = bias

Kemudian diperlukannya untuk melakukan pembaharuan bobot dengan *learning rate* sebesar 0.01, dengan persamaan seperti dibawah ini:

$$W^{new} = W^{old} - \lambda * \delta W^{old} \quad (3.18)$$

Keterangan :

W^{new} = bobot baru

W^{old} = bobot lama

λ = *learning rate*

δW^{old} = turunan bobot lama

3.5.3 Proses Validasi

Proses validasi adalah suatu proses dengan tujuan untuk menguji hasil training dari model yang telah dibuat. Setelah mendapatkan nilai bobot yang optimal, nilai tersebut digunakan pada proses validasi. Proses validasi ini dilakukan dengan menggunakan data validasi sebagai input disertai dengan menggunakan nilai bobot dan bias optimal yang telah di dapat dari proses *training*. Tahapan proses validasi hampir sama dengan proses *training*, yang

membedakan adalah pada proses validasi tidak terdapat proses inisialisasi bobot dan bias. Dibawah ini merupakan langkah-langkah pada proses validasi :

1. Melakukan inisialisasi bobot dan bias yang didapatkan dari proses *training*.
2. Melakukan perhitungan LSTM unit 1 dilanjutkan LSTM unit 2 seperti yang telah dijelaskan pada sub bab 3.5.1.2.
3. Melakukan perhitungan dense layer dengan fungsi aktivasi ReLU seperti yang telah dijelaskan pada sub bab 3.5.1.3.
4. Melakukan perhitungan probabilitas output pada *output layer* menggunakan perhitungan *neural network* dengan aktivasi sigmoid seperti yang telah dijelaskan pada sub bab 3.5.1.4.

BAB IV

UJI COBA DAN PEMBAHASAN

Pada bab ini akan dibahas tentang implementasi dari langkah-langkah pada skenario uji coba yang disiapkan oleh peneliti. Kemudian akan ditunjukkan hasil dan pembahasan uji coba serta integrasi penelitian ini dengan islam.

4.1 Skenario Pengujian

Pada tahap ini dilakukan skenario uji coba bagaimana metode *Long Short Term Memory* (LSTM) mengolah data validasi yang telah disiapkan dari hasil proses metode *Long Short Term Memory* (LSTM), data validasi juga diperoleh dari data yang telah dikumpulkan yang sama dengan latih namun data validasi tidak dimasukkan ke dalam proses pelatihan.

Tujuan dari penelitian ini adalah mengukur tingkat akurasi, presisi, *recall*, dan *f-measure* metode *Long Short Term Memory* (LSTM) pada *SMS spam detection*. Dalam mengetahui kinerjanya maka diperlukan suatu metode perhitungan kinerja yaitu dengan menggunakan metode *confusion matrix*.

Confusion Matrix yaitu mempresentasikan hasil evaluasi model dengan menggunakan tabel matriks, jika dataset terdiri dari dua kelas, maka kelas pertama dianggap positif, dan kelas dua dianggap negative. Evaluasi dari *confusion matrix* tersebut menghasilkan nilai akurasi, presisi, *recall*, dan *f-measure* (Putra & Wibowo, 2020).

Tabel 4. 1 *Confusion Matrix*

Nilai Prediksi	Nilai Sebenarnya		
		<i>TRUE</i>	<i>FALSE</i>
	<i>TRUE</i>	(TP)	(FP)
<i>FALSE</i>	(FN)	(TN)	

Berikut adalah persamaan model *Confusion Matrix* berdasarkan tabel 3.4 :

$$1. \text{ Rumus akurasi: } \frac{TP+TN}{TP+ TN+ FP+ FN} \quad (3.1)$$

$$2. \text{ Rumus presisi : } \frac{TP}{TP+FP} \quad (3.2)$$

$$3. \text{ Rumus } recall : \frac{TP}{TP+FN} \quad (3.3)$$

$$4. \text{ Rumus } f\text{-measure : } 2 \cdot \frac{\text{presisi} \cdot \text{recall}}{\text{presisi}+\text{recall}} \quad (3.4)$$

Keterangan:

TP = Jika sistem memprediksi sama dengan *ground truth* dimana *output* dari sistem sama dengan *groundtruth*.

TN = Jika *output* yang tidak diprediksi sistem dan juga bukan *ground truth*.

FP = Jika *output* yang diprediksi oleh sistem ternyata bukan *ground truth*.

FN = Jika *output* yang tidak diprediksi oleh sistem ternyata adalah *ground truth*.

Nilai akurasi adalah persentase jumlah data yang diklasifikasikan dengan benar oleh suatu algoritma. Nilai presisi digunakan untuk membandingkan *True Positive* dengan keseluruhan hasil positif. Nilai *recall* adalah proporsi kasus

dengan hasil diagnosa positif. Sedangkan nilai *f-measure* adalah hasil perhitungan evaluasi dengan menggabungkan nilai presisi dan *recall*.

4.2 Data Penelitian

Pada penelitian ini, dataset dibagi menjadi dua bagian yaitu 80% untuk data training dan 20% data validasi. Hal tersebut dilakukan untuk membuktikan bahwa hasil dari training model telah berhasil. Sumber data didapatkan dari 62 responden dan dari *repository* Institut Pertanian Bogor seperti yang telah dijelaskan pada bab sebelumnya. Berikut adalah sampel dari dataset yang telah diberi label dan akan digunakan pada proses training.

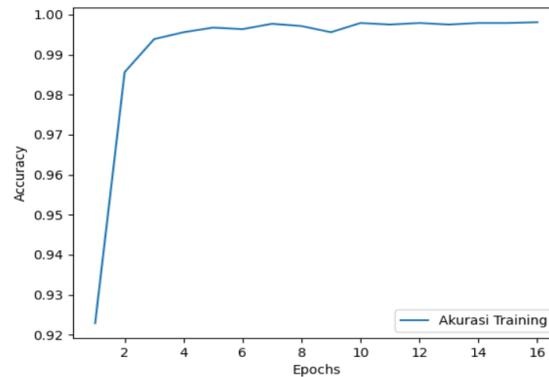
Tabel 4. 2 Sampel Daftar SMS

No.	SMS	Label
1	Pelanggan yth, Paket Combo Sakti akan berakhir pada tanggal 08/10/2020 jam 23:59 WIB.	0
2	[Info] bagi teman2 yg mengumpulkan data asisten lab/matakuliah bukti berupa surat keterangannya sudah bisa diambil ke pak andri	0
3	Selamat, Paket Combo Sakti 15 GB, 300 Mnt Tsel, 400 SMS Tsel dan Voucher Rise of Nowlin Rp50000 /30 hari Rp 75000 telah aktif, berlaku s/d tgl 08/10/2020 pkl. 23:59 WIB. Cek status/berhenti berlangganan melalui My Telkomsel Apps atau hub *363#. Info : 188.	0
4	Selamat, Paket Combo Terbaik Untukmu 17 GB, 300 Mnt Tsel, 400 SMS Tsel dan Voucher Shellfire Rp 50000 /30 hari Rp 80000 telah aktif, berlaku s/d tgl 05/03/2020 pkl. 23:59 WIB. Cek status/berhenti berlangganan melalui My Telkomsel Apps atau hub *363#. Info : 188.	0
5	Ada yang 400 rb yang kamar mandinya sama2 kalau yang ada kamar mandi di dalam kalau yang kecil 600 dan yang agak besar 700 rb	0

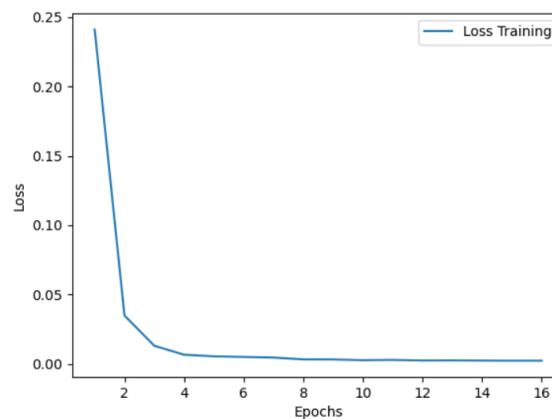
6	Mohon Maaf Kami Hanya Menginfokan Bahwa KSP Utama Karya Sudah Melanyani Pinjaman Dana Online Dengan Bunga 1%/TAHUN U/ INFO Klik Link https://wa.me/6281944488247	1
7	ANDA BUTUH DANA CEPAT CAIR??? Solusinya KSP SEJAHTERA BERSAMA menawarkan pinjaman Dgn bunga 0,6% mulai 10jt S/D 1M, chat WA:085348530057	1
8	BUTUH DANA BUAT: *Dagang *Kuliah *Usaha *Kebutuhan sehari2 dll, Kami solusinya Jika Minat Chat Wa/no; 085351978854	1
9	Yth Bpk/Ibu. BNI menyatakan Rekening anda terpilih sebagai pemenang hadiah Rp. 35jt. kode cek anda 03299757 info klik www.bankbni2016.blogspot.com	1
10	Ingin Tahu Angka Jitu! Yg Keluar Putaran Pada Hari ini 20/01/2013 Hubungi : 082- 335555-454. mbah TOGEL semarang "Jamin 100% Tembus" Raja SHIO (Buka 24 Jam)	1

4.3 Hasil Training

Data yang digunakan pada pelatihan model berjumlah 6514 yang mana data tersebut dibagi untuk data training sebanyak 5211 dan 1303 untuk data validasi. Pada pelatihan model dilakukan beberapa tahap seperti yang telah dijelaskan pada bab sebelumnya sesuai dengan arsitektur pada Gambar 3.4. Berikut adalah hasil training dari model yang telah dibentuk sesuai dengan Gambar 3.4 :



Gambar 4. 1 Kurva Akurasi Training



Gambar 4. 2 Kurva Loss Training

Pada gambar diatas menunjukkan bahwa loss paling rendah pada epoch ke-16 dengan nilai sebesar 0,0024 dan akurasi dari training adalah 0.9985 atau 99,85%. Kemudian setelah dilakukan proses pelatihan, maka model akan dievaluasi dengan data validasi dengan menggunakan *confusion matrix*.

4.4 Evaluasi Model

Berdasarkan skenario uji coba yang telah dijelaskan pada sub bab 4.1 maka pada tahap ini akan dilakukan pengujian pada model yang sudah melewati proses pelatihan. Prediksi yang akan dihasilkan oleh sistem akan dibandingkan dengan *ground truth*. Jika output dari sistem memprediksi benar terhadap satu

kelas, kemudian TN akan memiliki nilai 1 dikarenakan terdapat 1 kelas yang bukan *ground truth* dan diprediksi benar oleh sistem. Jika sistem menghasilkan prediksi yang salah, maka kolom TP akan bernilai 0 karena tidak ada prediksi yang sesuai dengan *groundtruth*. FP memiliki nilai 1 karena terdapat satu kelas yang seharusnya diprediksi benar oleh sistem sebagai kelas pada pesan sedangkan FN akan bernilai 1 karena terdapat satu kelas yang seharusnya diprediksi salah oleh sistem. Hasil uji coba secara lengkap dapat dilihat pada lampiran 1.

Kemudian dari hasil tersebut maka didapatkan total nilai *True Positive* (TP), *True Negative*(TN), *False Positive* (FP) dan *False Negative* (FN), seperti pada tabel berikut :

Tabel 4. 3 Hasil *Confusion Matrix*

Nilai Prediksi	Nilai Sebenarnya		
		TRUE	FALSE
	TRUE	1252	51
FALSE	51	1252	

Dari Tabel 4.4, dapat diukur hasil evaluasi dari sistem spam detection dengan menggunakan metode *Long Short Term Memory* (LSTM) sebagai berikut:

$$Accuracy = \frac{1252 + 1252}{1252 + 1252 + 51 + 51} \times 100\% = 96,09\% \quad (4.1)$$

$$Precision = \frac{1252}{1252 + 51} \times 100\% = 96,09\% \quad (4.2)$$

$$Recall = \frac{1252}{1252 + 51} \times 100\% = 96,09\% \quad (4.3)$$

$$F - measure = 2 \cdot \frac{(0,9609 \times 0,9609)}{0,9609 + 0,9609} \times 100\% = 96,09\% \quad (4.4)$$

Berdasarkan pengertian terkait perhitungan yang dilakukan, penelitian ini menggunakan dua klasifikasi yaitu *spam* dan *non spam* dimana dengan perhitungan *confusion matrix* pada penelitian ini menghasilkan nilai *accuracy*, *precision*, *recall*, *f-measure* yang sama yaitu 96,09%. Hal tersebut terjadi dikarenakan keseimbangan dari jumlah nilai *tp*, *tn*, *fp*, *fn* seperti yang telah terlampir pada lampiran 1.

4.5 Pembahasan

Berdasarkan hasil uji coba yang telah dilakukan dengan menggunakan data validasi sebanyak 1.303 kalimat, model LSTM yang telah dirancang dalam penelitian ini menunjukkan bahwa model LSTM berjalan dengan baik. Berdasarkan Tabel 4.4 menunjukkan bahwa model LSTM dapat membedakan SMS spam dan non spam dengan baik dan menghasilkan nilai parameter akurasi sebesar 96,09%. Nilai parameter lainnya juga cenderung baik dengan nilai parameter presisi sebesar 96,09%, *recall* sebesar 96,09% dan *f-measure* sebesar 96,09%.

Pada percobaan yang telah dilakukan terdapat suatu kesalahan atau *error* pada pembangunan model. Pada pembangunan model deep learning membutuhkan memori yang lebih pada proses training dan validasi dimana pada penelitian ini menggunakan *dictionary* yang didalamnya terdapat maksimal *vocabulary* dan data berurutan yang panjang yang keduanya direpresentasikan dalam bentuk vektor. Sehingga, pada penelitian ini tidak dapat melakukan *hyperparameter tuning* dengan maksimal.

Sms *spam detection* ini dapat membantu masyarakat dan meningkatkan kehati-hatian dalam menerima serta mempercayai pesan teks dari nomor yang tidak dikenal. Di dalam Al-Qur'an terdapat banyak ayat terkait konsep tolong menolong. Allah SWT berfirman dalam surah Al – Maidah ayat 2 :

..... الْعِقَابِ شَدِيدُ اللَّهِ إِنَّ اللَّهَ وَاتَّقُوا اللَّهَ وَالْعُدْوَانَ الْإِثْمِ عَلَى تَعَاوُنًا وَلَا

“..... *Dan tolong-menolonglah kamu dalam (mengerjakan) kebajikan dan takwa, dan jangan tolong-menolong dalam berbuat dosa atau pelanggaran. Dan bertakwalah kamu kepada Allah, sesungguhnya Allah amat berat siksa-Nya.*” (Al – Maidah ayat 2)

Menurut Tafsir Jalalain, (Bertolong-tolonglah kamu dalam kebaikan) dalam mengerjakan yang dititahkan (dan ketakwaan) dengan meninggalkan apa-apa yang dilarang (dan janganlah kamu bertolong-tolongan) pada ta`aawanu dibuang salah satu di antara dua ta pada asalnya (dalam berbuat dosa) atau maksiat (dan pelanggaran) artinya melampaui batas-batas ajaran Allah. (Dan bertakwalah kamu kepada Allah) takutlah kamu kepada azab siksa-Nya dengan menaati-Nya (sesungguhnya Allah amat berat siksa-Nya) bagi orang yang menentang-Nya (Al-Mahalli & As Suyuti, 2017).

Selain itu sms spam yang telah teridentifikasi dengan baik dapat dengan mudah dipahami oleh penerima sms, sehingga penerima sms lebih mengetahui dan mengerti substansi dari sebuah pesan teks yang masuk, di dalam Al – Qur'an juga diberikan sebuah peringatan terkait kehati-hatian dalam menerima berita. Berita yang dimaksud dapat dikatakan sebagai pesan teks. Allah SWT berfirman dalam surah Al – Hujurat ayat 6 :

بِجَهَالَةٍ قَوْمًا تُصِيبُوا أَنْ فَتَبَيَّنُوا بِنَبِيٍّ فَاسِقٍ حَاءَكُمْ إِنْ آمَنُوا الَّذِينَ أُيُّهَا يَا

نَادِمِينَ فَعَلْتُمْ مَا عَلَيَّ فَتُصِيبُحُوا

“Hai orang-orang yang beriman, jika datang kepadamu orang fasik membawa suatu berita, maka periksalah dengan teliti agar kamu tidak menimpakan suatu musibah kepada suatu kaum tanpa mengetahui keadaannya yang menyebabkan kamu menyesal atas perbuatanmu itu.” (Al – Hujurat ayat 6).

Menurut Tafsir Jalalain, (Hai orang-orang yang beriman! Jika datang kepada kalian orang fasik membawa suatu berita) (maka periksalah oleh kalian) kebenaran beritanya itu, apakah ia benar atau berdusta. Menurut suatu qiraat dibaca Fatatsabbatuu berasal dari lafal Ats-Tsabaat, artinya telitilah terlebih dahulu kebenarannya (agar kalian tidak menimpakan musibah kepada suatu kaum) menjadi Maf’ul dari lafal Fatabayyanuu, yakni dikhawatirkan hal tersebut akan menimpa musibah kepada suatu kaum (tanpa mengetahui keadaannya) menjadi Hal atau kata keterangan keadaan dari Fa’il, yakni tanpa sepengetahuannya (yang menyebabkan kalian) membuat kalian (atas perbuatan kalian itu) yakni berbuat kekeliruan terhadap kaum tersebut (menyesal) selanjutnya Rasulullah mengutus Khalid kepada mereka sesudah mereka kembali ke negerinya. Ternyata Khalid tiada menjumpai mereka melainkan hanya ketaatan dan kebaikan belaka, lalu ia menceritakan hal tersebut kepada Nabi (Al-Mahalli & As Suyuti, 2017).

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pada penelitian ini menggunakan data sebanyak 6.514 dengan pembagian data sebanyak 3.161 data *SMS spam* dan 3.353 data *SMS non spam*. Data tersebut dibagi menjadi 80% untuk data *training* dan 20% untuk data validasi. Pada proses *training* menghasilkan akurasi sebesar 99,85%. Berdasarkan hasil *training* tersebut dilakukan proses validasi dengan hasil akurasi pada pengujian sebesar 96,09% dan nilai parameter lainnya juga cenderung baik dengan nilai parameter presisi sebesar 96,09%, *recall* sebesar 96,09% dan *f-measure* sebesar 96,09%. Hal tersebut menunjukkan bahwa model LSTM yang telah dirancang pada penelitian ini berjalan dengan baik, sehingga sistem dapat membedakan *SMS spam* dan *non spam* dengan baik.

5.2 Saran

Berdasarkan hasil percobaan pada penelitian ini diharapkan bagi penelitian selanjutnya dapat meningkatkan hasil klasifikasi yang lebih akurat dan lebih baik. Maka dari itu, penulis memiliki saran untuk pengembangan penelitian di masa yang akan datang, yaitu sebagai berikut :

1. Penambahan jumlah dataset, dikarenakan jumlah dataset sangat berpengaruh pada hasil akurasi dan minimal *error*.
2. Melakukan metode yang berbeda di dalam proses *word embedding* seperti TF-IDF, Word2Vec dan metode vektorisasi teks yang lain.

3. Melakukan pembangunan model LSTM dengan arsitektur dan *hyperparameter* yang berbeda.
4. Menggunakan teknik *continuous machine learning* dimana *user* dapat melakukan *update* pada proses *training*.
5. Menambahkan sistem bot untuk melaporkan pesan *spam* yang tidak terdeteksi *spam* oleh sistem.

DAFTAR PUSTAKA

- Al-Mahalli, I. J., & As Suyuti, I. J. (2017). *TERJEMAHAN TAFSIR JALALAIN BERIKUT ASBABU NUZUL JILID 2*. Sinar Baru Algensindo Bandung.
- Augustyniak, Ł., Kajdanowicz, T., & Kazienko, P. (2019). Aspect Detection Using Word and Char Embeddings With (Bi)Lstm and Crf. *ArXiv*. <https://doi.org/10.1109/AIKE.2019.00016>
- Brown, J., Shipman, B., & Vetter, R. (2007). SMS: The short message service. *Computer*, 40(12), 106–110. <https://doi.org/10.1109/MC.2007.440>
- Chen, C. W., Tseng, S. P., Kuan, T. W., & Wang, J. F. (2020). Outpatient text classification using attention-based bidirectional LSTM for robot-assisted servicing in hospital. *Information (Switzerland)*, 11(2). <https://doi.org/10.3390/info11020106>
- Chen, K., Yan, Z. J., & Huo, Q. (2015). A context-sensitive-chunk BPTT approach to training deep LSTM/BLSTM recurrent neural networks for offline handwriting recognition. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, 2015-November*, 411–415. <https://doi.org/10.1109/ICDAR.2015.7333794>
- Cornegruta, S., Bakewell, R., Withey, S., & Montana, G. (2016). *Modelling Radiological Language with Bidirectional Long Short-Term Memory Networks*. 17–27. <https://doi.org/10.18653/v1/w16-6103>
- Cynthia, E. P., & Ismanto, E. (2017). Jaringan Saraf Tiruan Algoritma Backpropagation dalam Memprediksi Ketersediaan Komoditi Pangan Provinsi Riau. *Jurnal Teknologi Dan Sistem Informasi Univrab*, 2(2), 196–209.
- Faadilah, A. (2020). *Analisis Sentimen Pada Ulasan Aplikasi Tokopedia di Gogle Playstore Menggunakan Metode Long Short Term Memory*.
- Feng, J., & Lu, S. (2019). Performance Analysis of Various Activation Functions in Artificial Neural Networks. *Journal of Physics: Conference Series*, 1237(2). <https://doi.org/10.1088/1742-6596/1237/2/022030>
- Jain, A., Kulkarni, G., & Shah, V. (2019). Natural Language Processing. *International Journal of Computer Sciences and Engineering, January 2018*. <https://doi.org/10.26438/ijcse/v6i1.161167>
- Jain, G., Sharma, M., & Agarwal, B. (2019). Optimizing semantic LSTM for spam detection. *International Journal of Information Technology (Singapore)*, 11(2), 239–250. <https://doi.org/10.1007/s41870-018-0157-5>
- Jang, B., Kim, M., Harerimana, G., Kang, S. U., & Kim, J. W. (2020). Bi-LSTM model to increase accuracy in text classification: Combining word2vec CNN and attention mechanism. *Applied Sciences (Switzerland)*, 10(17). <https://doi.org/10.3390/app10175841>

- Janocha, K., & Czarnecki, W. M. (2016). *On Loss Functions for Deep Neural Networks in Classification*. 25(December), 49–59. <https://doi.org/10.4467/20838476SI.16.004.6185>
- Munitasri, I., Santosa, S., & Supriyanto, C. (2018). Klasifikasi Pesan Sms Menggunakan Algoritma Naive Bayes Dengan Seleksi Fitur Genetic Algorithm. *Jurnal Teknologi Informasi*, 14(1). <http://research>.
- Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). *Activation Functions: Comparison of trends in Practice and Research for Deep Learning*. 1–20. <http://arxiv.org/abs/1811.03378>
- P, I. W. S. E., Wijaya, A. Y., & Soelaiman, R. (2016). *Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) pada Caltech 101*. 5(1).
- Poomka, P., Pongsena, W., Kerdprasop, N., & Kerdprasop, K. (2019). SMS Spam Detection Based on Long Short-Term Memory and Gated Recurrent Unit. *International Journal of Future Computer and Communication*, 8(1), 11–15. <https://doi.org/10.18178/ijfcc.2019.8.1.532>
- Putra, D., & Wibowo, A. (2020). *Prediksi Keputusan Minat Penjurusan Siswa SMA Yadika 5 Menggunakan Algoritma Naïve Bayes*. 2, 84–92.
- Rao, A., & Spasojevic, N. (2016). *Actionable and Political Text Classification using Word Embeddings and LSTM*. <http://arxiv.org/abs/1607.02501>
- Sugiyono. (2017). *Metode Penelitian Pendidikan Pendekatan Kuantitatif, Kualitatif dan R&D*. AlfaBeta.
- Wang, Q., Li, W., & Jin, Z. (2021). Review of Text Classification in Deep Learning. *OALib*, 08(03), 1–8. <https://doi.org/10.4236/oalib.1107175>
- Wildan, M., Aldi, P., & Aditsania, A. (2018). Analisis dan Implementasi Long Short Term Memory Neural Network untuk Prediksi Harga Bitcoin. *E-Proceeding of Engineering*, 5(2), 3548–3555.

LAMPIRAN

Lampiran 1

Hasil Uji Coba

Data	Ground Truth	Output	TP	TN	FP	FN
1	1	1	1	1	0	0
2	1	1	1	1	0	0
3	1	1	1	1	0	0
4	1	1	1	1	0	0
5	1	1	1	1	0	0
6	1	1	1	1	0	0
7	1	1	1	1	0	0
8	1	1	1	1	0	0
9	1	1	1	1	0	0
10	1	1	1	1	0	0
11	1	1	1	1	0	0
12	1	1	1	1	0	0
13	1	1	1	1	0	0
14	1	1	1	1	0	0
15	1	1	1	1	0	0
16	1	1	1	1	0	0
17	1	1	1	1	0	0
18	1	1	1	1	0	0
19	1	1	1	1	0	0
20	1	1	1	1	0	0
21	1	1	1	1	0	0
22	1	1	1	1	0	0
23	1	1	1	1	0	0
24	1	1	1	1	0	0
25	1	1	1	1	0	0
26	1	1	1	1	0	0
27	1	1	1	1	0	0
28	1	1	1	1	0	0
29	1	1	1	1	0	0
30	1	1	1	1	0	0
31	1	1	1	1	0	0
32	1	1	1	1	0	0
33	1	1	1	1	0	0

34	1	1	1	1	0	0
35	1	1	1	1	0	0
36	1	1	1	1	0	0
37	1	1	1	1	0	0
38	1	1	1	1	0	0
39	1	1	1	1	0	0
40	1	1	1	1	0	0
41	1	1	1	1	0	0
42	1	1	1	1	0	0
43	1	1	1	1	0	0
44	1	1	1	1	0	0
45	1	1	1	1	0	0
46	1	1	1	1	0	0
47	1	1	1	1	0	0
48	1	1	1	1	0	0
49	1	1	1	1	0	0
50	1	1	1	1	0	0
51	1	1	1	1	0	0
52	1	1	1	1	0	0
53	1	1	1	1	0	0
54	1	1	1	1	0	0
55	1	1	1	1	0	0
56	1	1	1	1	0	0
57	1	1	1	1	0	0
58	1	1	1	1	0	0
59	1	1	1	1	0	0
60	1	1	1	1	0	0
61	1	1	1	1	0	0
62	1	1	1	1	0	0
63	1	1	1	1	0	0
64	1	1	1	1	0	0
65	1	1	1	1	0	0
66	1	1	1	1	0	0
67	1	1	1	1	0	0
68	1	1	1	1	0	0
69	1	1	1	1	0	0
70	1	1	1	1	0	0
71	1	1	1	1	0	0
72	1	1	1	1	0	0

73	1	1	1	1	0	0
74	1	1	1	1	0	0
75	1	1	1	1	0	0
76	1	1	1	1	0	0
77	1	1	1	1	0	0
78	1	1	1	1	0	0
79	1	1	1	1	0	0
80	1	1	1	1	0	0
81	1	1	1	1	0	0
82	1	1	1	1	0	0
83	1	1	1	1	0	0
84	1	1	1	1	0	0
85	1	1	1	1	0	0
86	1	1	1	1	0	0
87	1	1	1	1	0	0
88	1	1	1	1	0	0
89	1	1	1	1	0	0
90	1	1	1	1	0	0
91	1	1	1	1	0	0
92	1	1	1	1	0	0
93	1	1	1	1	0	0
94	1	1	1	1	0	0
95	1	1	1	1	0	0
96	1	1	1	1	0	0
97	1	1	1	1	0	0
98	1	1	1	1	0	0
99	1	1	1	1	0	0
100	1	1	1	1	0	0
101	1	1	1	1	0	0
102	1	1	1	1	0	0
103	1	1	1	1	0	0
104	1	1	1	1	0	0
105	1	1	1	1	0	0
106	1	1	1	1	0	0
107	1	1	1	1	0	0
108	1	1	1	1	0	0
109	1	1	1	1	0	0
110	1	1	1	1	0	0
111	1	1	1	1	0	0

112	1	1	1	1	0	0
113	1	1	1	1	0	0
114	1	1	1	1	0	0
115	1	1	1	1	0	0
116	1	1	1	1	0	0
117	1	1	1	1	0	0
118	1	1	1	1	0	0
119	1	1	1	1	0	0
120	1	1	1	1	0	0
121	1	1	1	1	0	0
122	1	1	1	1	0	0
123	1	1	1	1	0	0
124	1	1	1	1	0	0
125	1	1	1	1	0	0
126	1	1	1	1	0	0
127	1	1	1	1	0	0
128	1	1	1	1	0	0
129	1	1	1	1	0	0
130	1	1	1	1	0	0
131	1	1	1	1	0	0
132	1	1	1	1	0	0
133	1	1	1	1	0	0
134	1	1	1	1	0	0
135	1	1	1	1	0	0
136	1	1	1	1	0	0
137	1	1	1	1	0	0
138	1	1	1	1	0	0
139	1	1	1	1	0	0
140	1	1	1	1	0	0
141	1	1	1	1	0	0
142	1	1	1	1	0	0
143	1	1	1	1	0	0
144	1	1	1	1	0	0
145	1	1	1	1	0	0
146	1	1	1	1	0	0
147	1	1	1	1	0	0
148	1	1	1	1	0	0
149	1	1	1	1	0	0
150	1	1	1	1	0	0

151	1	1	1	1	0	0
152	1	1	1	1	0	0
153	1	1	1	1	0	0
154	1	1	1	1	0	0
155	1	1	1	1	0	0
156	1	1	1	1	0	0
157	1	1	1	1	0	0
158	1	1	1	1	0	0
159	1	1	1	1	0	0
160	1	1	1	1	0	0
161	1	1	1	1	0	0
162	1	1	1	1	0	0
163	1	1	1	1	0	0
164	1	1	1	1	0	0
165	1	1	1	1	0	0
166	1	1	1	1	0	0
167	1	1	1	1	0	0
168	1	1	1	1	0	0
169	1	1	1	1	0	0
170	1	1	1	1	0	0
171	1	1	1	1	0	0
172	1	1	1	1	0	0
173	1	1	1	1	0	0
174	1	1	1	1	0	0
175	1	1	1	1	0	0
176	1	1	1	1	0	0
177	1	1	1	1	0	0
178	1	1	1	1	0	0
179	1	1	1	1	0	0
180	1	1	1	1	0	0
181	1	1	1	1	0	0
182	1	1	1	1	0	0
183	1	1	1	1	0	0
184	1	1	1	1	0	0
185	1	1	1	1	0	0
186	1	1	1	1	0	0
187	1	1	1	1	0	0
188	1	1	1	1	0	0
189	1	1	1	1	0	0

190	1	1	1	1	0	0
191	1	1	1	1	0	0
192	1	1	1	1	0	0
193	1	1	1	1	0	0
194	1	1	1	1	0	0
195	1	1	1	1	0	0
196	1	1	1	1	0	0
197	1	1	1	1	0	0
198	1	1	1	1	0	0
199	1	1	1	1	0	0
200	1	1	1	1	0	0
201	1	1	1	1	0	0
202	1	1	1	1	0	0
203	1	1	1	1	0	0
204	1	1	1	1	0	0
205	1	1	1	1	0	0
206	1	1	1	1	0	0
207	1	1	1	1	0	0
208	1	1	1	1	0	0
209	1	1	1	1	0	0
210	1	1	1	1	0	0
211	1	1	1	1	0	0
212	1	1	1	1	0	0
213	1	1	1	1	0	0
214	1	1	1	1	0	0
215	1	1	1	1	0	0
216	1	1	1	1	0	0
217	1	1	1	1	0	0
218	1	1	1	1	0	0
219	1	1	1	1	0	0
220	1	1	1	1	0	0
221	1	1	1	1	0	0
222	1	1	1	1	0	0
223	1	1	1	1	0	0
224	1	1	1	1	0	0
225	1	1	1	1	0	0
226	1	1	1	1	0	0
227	1	1	1	1	0	0
228	1	1	1	1	0	0

229	1	1	1	1	0	0
230	1	1	1	1	0	0
231	1	1	1	1	0	0
232	1	1	1	1	0	0
233	1	1	1	1	0	0
234	1	1	1	1	0	0
235	1	1	1	1	0	0
236	1	1	1	1	0	0
237	1	1	1	1	0	0
238	1	1	1	1	0	0
239	1	1	1	1	0	0
240	1	1	1	1	0	0
241	1	1	1	1	0	0
242	1	1	1	1	0	0
243	1	1	1	1	0	0
244	1	1	1	1	0	0
245	1	1	1	1	0	0
246	1	1	1	1	0	0
247	1	1	1	1	0	0
248	1	1	1	1	0	0
249	1	1	1	1	0	0
250	1	1	1	1	0	0
251	1	1	1	1	0	0
252	1	1	1	1	0	0
253	1	1	1	1	0	0
254	1	1	1	1	0	0
255	1	1	1	1	0	0
256	1	1	1	1	0	0
257	1	1	1	1	0	0
258	1	1	1	1	0	0
259	1	1	1	1	0	0
260	1	1	1	1	0	0
261	1	1	1	1	0	0
262	1	1	1	1	0	0
263	1	1	1	1	0	0
264	1	1	1	1	0	0
265	1	1	1	1	0	0
266	1	1	1	1	0	0
267	1	1	1	1	0	0

268	1	1	1	1	0	0
269	1	1	1	1	0	0
270	1	1	1	1	0	0
271	1	1	1	1	0	0
272	1	1	1	1	0	0
273	1	1	1	1	0	0
274	1	1	1	1	0	0
275	1	1	1	1	0	0
276	1	1	1	1	0	0
277	1	1	1	1	0	0
278	1	1	1	1	0	0
279	1	1	1	1	0	0
280	1	1	1	1	0	0
281	1	1	1	1	0	0
282	1	1	1	1	0	0
283	1	1	1	1	0	0
284	1	1	1	1	0	0
285	1	1	1	1	0	0
286	1	1	1	1	0	0
287	1	1	1	1	0	0
288	1	1	1	1	0	0
289	1	1	1	1	0	0
290	1	1	1	1	0	0
291	1	1	1	1	0	0
292	1	1	1	1	0	0
293	1	1	1	1	0	0
294	1	1	1	1	0	0
295	1	1	1	1	0	0
296	1	1	1	1	0	0
297	1	1	1	1	0	0
298	1	1	1	1	0	0
299	1	1	1	1	0	0
300	1	1	1	1	0	0
301	1	1	1	1	0	0
302	1	1	1	1	0	0
303	1	1	1	1	0	0
304	1	1	1	1	0	0
305	1	1	1	1	0	0
306	1	1	1	1	0	0

307	1	1	1	1	0	0
308	1	1	1	1	0	0
309	1	1	1	1	0	0
310	1	1	1	1	0	0
311	1	1	1	1	0	0
312	1	1	1	1	0	0
313	1	1	1	1	0	0
314	1	1	1	1	0	0
315	1	1	1	1	0	0
316	1	1	1	1	0	0
317	1	1	1	1	0	0
318	1	1	1	1	0	0
319	1	1	1	1	0	0
320	1	1	1	1	0	0
321	1	1	1	1	0	0
322	1	1	1	1	0	0
323	1	1	1	1	0	0
324	1	1	1	1	0	0
325	1	1	1	1	0	0
326	1	1	1	1	0	0
327	1	1	1	1	0	0
328	1	1	1	1	0	0
329	1	1	1	1	0	0
330	1	1	1	1	0	0
331	1	1	1	1	0	0
332	1	1	1	1	0	0
333	1	1	1	1	0	0
334	1	1	1	1	0	0
335	1	1	1	1	0	0
336	1	1	1	1	0	0
337	1	1	1	1	0	0
338	1	1	1	1	0	0
339	1	1	1	1	0	0
340	1	1	1	1	0	0
341	1	1	1	1	0	0
342	1	1	1	1	0	0
343	1	1	1	1	0	0
344	1	1	1	1	0	0
345	1	1	1	1	0	0

346	1	1	1	1	0	0
347	1	1	1	1	0	0
348	1	1	1	1	0	0
349	1	1	1	1	0	0
350	1	1	1	1	0	0
351	1	1	1	1	0	0
352	1	1	1	1	0	0
353	1	1	1	1	0	0
354	1	1	1	1	0	0
355	1	1	1	1	0	0
356	1	1	1	1	0	0
357	1	1	1	1	0	0
358	1	1	1	1	0	0
359	1	1	1	1	0	0
360	1	1	1	1	0	0
361	1	1	1	1	0	0
362	1	1	1	1	0	0
363	1	1	1	1	0	0
364	1	1	1	1	0	0
365	1	1	1	1	0	0
366	1	1	1	1	0	0
367	1	1	1	1	0	0
368	1	1	1	1	0	0
369	1	1	1	1	0	0
370	1	1	1	1	0	0
371	1	1	1	1	0	0
372	1	1	1	1	0	0
373	1	1	1	1	0	0
374	1	1	1	1	0	0
375	1	1	1	1	0	0
376	1	1	1	1	0	0
377	1	1	1	1	0	0
378	1	1	1	1	0	0
379	1	1	1	1	0	0
380	1	1	1	1	0	0
381	1	1	1	1	0	0
382	1	1	1	1	0	0
383	1	1	1	1	0	0
384	1	1	1	1	0	0

385	1	1	1	1	0	0
386	1	1	1	1	0	0
387	1	1	1	1	0	0
388	1	1	1	1	0	0
389	1	1	1	1	0	0
390	1	1	1	1	0	0
391	1	1	1	1	0	0
392	1	1	1	1	0	0
393	1	1	1	1	0	0
394	1	1	1	1	0	0
395	1	1	1	1	0	0
396	1	1	1	1	0	0
397	1	1	1	1	0	0
398	1	1	1	1	0	0
399	1	1	1	1	0	0
400	1	1	1	1	0	0
401	1	1	1	1	0	0
402	1	1	1	1	0	0
403	1	1	1	1	0	0
404	1	1	1	1	0	0
405	1	1	1	1	0	0
406	1	1	1	1	0	0
407	1	1	1	1	0	0
408	1	1	1	1	0	0
409	1	1	1	1	0	0
410	1	1	1	1	0	0
411	1	1	1	1	0	0
412	1	1	1	1	0	0
413	1	1	1	1	0	0
414	1	1	1	1	0	0
415	1	1	1	1	0	0
416	1	1	1	1	0	0
417	1	1	1	1	0	0
418	1	1	1	1	0	0
419	1	1	1	1	0	0
420	1	1	1	1	0	0
421	1	1	1	1	0	0
422	1	1	1	1	0	0
423	1	1	1	1	0	0

424	1	1	1	1	0	0
425	1	1	1	1	0	0
426	1	1	1	1	0	0
427	1	1	1	1	0	0
428	1	1	1	1	0	0
429	1	1	1	1	0	0
430	1	1	1	1	0	0
431	1	1	1	1	0	0
432	1	1	1	1	0	0
433	1	1	1	1	0	0
434	1	1	1	1	0	0
435	1	1	1	1	0	0
436	1	1	1	1	0	0
437	1	1	1	1	0	0
438	1	1	1	1	0	0
439	1	1	1	1	0	0
440	1	1	1	1	0	0
441	1	1	1	1	0	0
442	1	1	1	1	0	0
443	1	1	1	1	0	0
444	1	1	1	1	0	0
445	1	1	1	1	0	0
446	1	1	1	1	0	0
447	1	1	1	1	0	0
448	1	1	1	1	0	0
449	1	1	1	1	0	0
450	1	1	1	1	0	0
451	1	1	1	1	0	0
452	1	1	1	1	0	0
453	1	1	1	1	0	0
454	1	1	1	1	0	0
455	1	1	1	1	0	0
456	1	1	1	1	0	0
457	1	1	1	1	0	0
458	1	1	1	1	0	0
459	1	1	1	1	0	0
460	1	1	1	1	0	0
461	1	1	1	1	0	0
462	1	1	1	1	0	0

463	1	1	1	1	0	0
464	1	1	1	1	0	0
465	1	1	1	1	0	0
466	1	1	1	1	0	0
467	1	1	1	1	0	0
468	1	1	1	1	0	0
469	1	1	1	1	0	0
470	1	1	1	1	0	0
471	1	1	1	1	0	0
472	1	1	1	1	0	0
473	1	1	1	1	0	0
474	1	1	1	1	0	0
475	1	1	1	1	0	0
476	1	1	1	1	0	0
477	1	1	1	1	0	0
478	1	1	1	1	0	0
479	1	1	1	1	0	0
480	1	1	1	1	0	0
481	1	1	1	1	0	0
482	1	1	1	1	0	0
483	1	1	1	1	0	0
484	1	1	1	1	0	0
485	1	1	1	1	0	0
486	1	1	1	1	0	0
487	1	1	1	1	0	0
488	1	1	1	1	0	0
489	1	1	1	1	0	0
490	1	1	1	1	0	0
491	1	1	1	1	0	0
492	1	1	1	1	0	0
493	1	1	1	1	0	0
494	1	1	1	1	0	0
495	1	1	1	1	0	0
496	1	1	1	1	0	0
497	1	1	1	1	0	0
498	1	1	1	1	0	0
499	1	1	1	1	0	0
500	1	1	1	1	0	0
501	1	1	1	1	0	0

502	1	1	1	1	0	0
503	1	1	1	1	0	0
504	1	1	1	1	0	0
505	1	1	1	1	0	0
506	1	1	1	1	0	0
507	1	1	1	1	0	0
508	1	1	1	1	0	0
509	1	1	1	1	0	0
510	1	1	1	1	0	0
511	1	1	1	1	0	0
512	1	1	1	1	0	0
513	1	1	1	1	0	0
514	1	1	1	1	0	0
515	1	1	1	1	0	0
516	1	1	1	1	0	0
517	1	1	1	1	0	0
518	1	1	1	1	0	0
519	1	1	1	1	0	0
520	1	1	1	1	0	0
521	1	1	1	1	0	0
522	1	1	1	1	0	0
523	1	1	1	1	0	0
524	1	1	1	1	0	0
525	1	1	1	1	0	0
526	1	1	1	1	0	0
527	1	1	1	1	0	0
528	1	1	1	1	0	0
529	1	1	1	1	0	0
530	1	1	1	1	0	0
531	1	1	1	1	0	0
532	1	1	1	1	0	0
533	1	1	1	1	0	0
534	1	1	1	1	0	0
535	1	1	1	1	0	0
536	1	1	1	1	0	0
537	1	1	1	1	0	0
538	1	1	1	1	0	0
539	1	1	1	1	0	0
540	1	1	1	1	0	0

541	1	1	1	1	0	0
542	1	1	1	1	0	0
543	1	1	1	1	0	0
544	1	1	1	1	0	0
545	1	1	1	1	0	0
546	1	1	1	1	0	0
547	1	1	1	1	0	0
548	1	1	1	1	0	0
549	1	1	1	1	0	0
550	1	1	1	1	0	0
551	1	1	1	1	0	0
552	1	1	1	1	0	0
553	1	1	1	1	0	0
554	1	1	1	1	0	0
555	1	1	1	1	0	0
556	1	1	1	1	0	0
557	1	1	1	1	0	0
558	1	1	1	1	0	0
559	1	1	1	1	0	0
560	1	1	1	1	0	0
561	1	1	1	1	0	0
562	1	1	1	1	0	0
563	1	1	1	1	0	0
564	1	1	1	1	0	0
565	1	1	1	1	0	0
566	1	1	1	1	0	0
567	1	1	1	1	0	0
568	1	1	1	1	0	0
569	1	1	1	1	0	0
570	1	1	1	1	0	0
571	1	1	1	1	0	0
572	1	1	1	1	0	0
573	1	1	1	1	0	0
574	1	1	1	1	0	0
575	1	1	1	1	0	0
576	1	1	1	1	0	0
577	1	1	1	1	0	0
578	1	1	1	1	0	0
579	1	1	1	1	0	0

580	1	1	1	1	0	0
581	1	1	1	1	0	0
582	1	1	1	1	0	0
583	1	1	1	1	0	0
584	1	1	1	1	0	0
585	1	1	1	1	0	0
586	1	1	1	1	0	0
587	1	1	1	1	0	0
588	1	1	1	1	0	0
589	1	1	1	1	0	0
590	1	1	1	1	0	0
591	1	1	1	1	0	0
592	1	1	1	1	0	0
593	1	1	1	1	0	0
594	1	1	1	1	0	0
595	1	1	1	1	0	0
596	1	1	1	1	0	0
597	1	1	1	1	0	0
598	1	1	1	1	0	0
599	1	1	1	1	0	0
600	1	1	1	1	0	0
601	1	1	1	1	0	0
602	1	1	1	1	0	0
603	1	1	1	1	0	0
604	1	1	1	1	0	0
605	1	1	1	1	0	0
606	1	1	1	1	0	0
607	1	1	1	1	0	0
608	1	1	1	1	0	0
609	1	1	1	1	0	0
610	1	1	1	1	0	0
611	1	1	1	1	0	0
612	0	0	1	1	0	0
613	0	0	1	1	0	0
614	0	0	1	1	0	0
615	0	0	1	1	0	0
616	0	0	1	1	0	0
617	0	0	1	1	0	0
618	0	0	1	1	0	0

619	0	0	1	1	0	0
620	0	0	1	1	0	0
621	0	0	1	1	0	0
622	0	0	1	1	0	0
623	0	0	1	1	0	0
624	0	0	1	1	0	0
625	0	0	1	1	0	0
626	0	0	1	1	0	0
627	0	0	1	1	0	0
628	0	0	1	1	0	0
629	0	0	1	1	0	0
630	0	0	1	1	0	0
631	0	0	1	1	0	0
632	0	0	1	1	0	0
633	0	0	1	1	0	0
634	0	0	1	1	0	0
635	0	0	1	1	0	0
636	0	0	1	1	0	0
637	0	0	1	1	0	0
638	0	0	1	1	0	0
639	0	0	1	1	0	0
640	0	0	1	1	0	0
641	0	0	1	1	0	0
642	0	0	1	1	0	0
643	0	0	1	1	0	0
644	0	0	1	1	0	0
645	0	0	1	1	0	0
646	0	0	1	1	0	0
647	0	0	1	1	0	0
648	0	0	1	1	0	0
649	0	0	1	1	0	0
650	0	0	1	1	0	0
651	0	0	1	1	0	0
652	0	0	1	1	0	0
653	0	0	1	1	0	0
654	0	0	1	1	0	0
655	0	0	1	1	0	0
656	0	0	1	1	0	0
657	0	0	1	1	0	0

658	0	0	1	1	0	0
659	0	0	1	1	0	0
660	0	0	1	1	0	0
661	0	0	1	1	0	0
662	0	0	1	1	0	0
663	0	0	1	1	0	0
664	0	0	1	1	0	0
665	0	0	1	1	0	0
666	0	0	1	1	0	0
667	0	0	1	1	0	0
668	0	0	1	1	0	0
669	0	0	1	1	0	0
670	0	0	1	1	0	0
671	0	0	1	1	0	0
672	0	0	1	1	0	0
673	0	0	1	1	0	0
674	0	0	1	1	0	0
675	0	0	1	1	0	0
676	0	0	1	1	0	0
677	0	0	1	1	0	0
678	0	0	1	1	0	0
679	0	0	1	1	0	0
680	0	0	1	1	0	0
681	0	0	1	1	0	0
682	0	0	1	1	0	0
683	0	0	1	1	0	0
684	0	0	1	1	0	0
685	0	0	1	1	0	0
686	0	0	1	1	0	0
687	0	0	1	1	0	0
688	0	0	1	1	0	0
689	0	0	1	1	0	0
690	0	0	1	1	0	0
691	0	0	1	1	0	0
692	0	0	1	1	0	0
693	0	0	1	1	0	0
694	0	0	1	1	0	0
695	0	0	1	1	0	0
696	0	0	1	1	0	0

697	0	0	1	1	0	0
698	0	0	1	1	0	0
699	0	0	1	1	0	0
700	0	0	1	1	0	0
701	0	0	1	1	0	0
702	0	0	1	1	0	0
703	0	0	1	1	0	0
704	0	0	1	1	0	0
705	0	0	1	1	0	0
706	0	0	1	1	0	0
707	0	0	1	1	0	0
708	0	0	1	1	0	0
709	0	0	1	1	0	0
710	0	0	1	1	0	0
711	0	0	1	1	0	0
712	0	0	1	1	0	0
713	0	0	1	1	0	0
714	0	0	1	1	0	0
715	0	0	1	1	0	0
716	0	0	1	1	0	0
717	0	0	1	1	0	0
718	0	0	1	1	0	0
719	0	0	1	1	0	0
720	0	0	1	1	0	0
721	0	0	1	1	0	0
722	0	0	1	1	0	0
723	0	0	1	1	0	0
724	0	0	1	1	0	0
725	0	0	1	1	0	0
726	0	0	1	1	0	0
727	0	0	1	1	0	0
728	0	0	1	1	0	0
729	0	0	1	1	0	0
730	0	0	1	1	0	0
731	0	0	1	1	0	0
732	0	0	1	1	0	0
733	0	0	1	1	0	0
734	0	0	1	1	0	0
735	0	0	1	1	0	0

736	0	0	1	1	0	0
737	0	0	1	1	0	0
738	0	0	1	1	0	0
739	0	0	1	1	0	0
740	0	0	1	1	0	0
741	0	0	1	1	0	0
742	0	0	1	1	0	0
743	0	0	1	1	0	0
744	0	0	1	1	0	0
745	0	0	1	1	0	0
746	0	0	1	1	0	0
747	0	0	1	1	0	0
748	0	0	1	1	0	0
749	0	0	1	1	0	0
750	0	0	1	1	0	0
751	0	0	1	1	0	0
752	0	0	1	1	0	0
753	0	0	1	1	0	0
754	0	0	1	1	0	0
755	0	0	1	1	0	0
756	0	0	1	1	0	0
757	0	0	1	1	0	0
758	0	0	1	1	0	0
759	0	0	1	1	0	0
760	0	0	1	1	0	0
761	0	0	1	1	0	0
762	0	0	1	1	0	0
763	0	0	1	1	0	0
764	0	0	1	1	0	0
765	0	0	1	1	0	0
766	0	0	1	1	0	0
767	0	0	1	1	0	0
768	0	0	1	1	0	0
769	0	0	1	1	0	0
770	0	0	1	1	0	0
771	0	0	1	1	0	0
772	0	0	1	1	0	0
773	0	0	1	1	0	0
774	0	0	1	1	0	0

775	0	0	1	1	0	0
776	0	0	1	1	0	0
777	0	0	1	1	0	0
778	0	0	1	1	0	0
779	0	0	1	1	0	0
780	0	0	1	1	0	0
781	0	0	1	1	0	0
782	0	0	1	1	0	0
783	0	0	1	1	0	0
784	0	0	1	1	0	0
785	0	0	1	1	0	0
786	0	0	1	1	0	0
787	0	0	1	1	0	0
788	0	0	1	1	0	0
789	0	0	1	1	0	0
790	0	0	1	1	0	0
791	0	0	1	1	0	0
792	0	0	1	1	0	0
793	0	0	1	1	0	0
794	0	0	1	1	0	0
795	0	0	1	1	0	0
796	0	0	1	1	0	0
797	0	0	1	1	0	0
798	0	0	1	1	0	0
799	0	0	1	1	0	0
800	0	0	1	1	0	0
801	0	0	1	1	0	0
802	0	0	1	1	0	0
803	0	0	1	1	0	0
804	0	0	1	1	0	0
805	0	0	1	1	0	0
806	0	0	1	1	0	0
807	0	0	1	1	0	0
808	0	0	1	1	0	0
809	0	0	1	1	0	0
810	0	0	1	1	0	0
811	0	0	1	1	0	0
812	0	0	1	1	0	0
813	0	0	1	1	0	0

814	0	0	1	1	0	0
815	0	0	1	1	0	0
816	0	0	1	1	0	0
817	0	0	1	1	0	0
818	0	0	1	1	0	0
819	0	0	1	1	0	0
820	0	0	1	1	0	0
821	0	0	1	1	0	0
822	0	0	1	1	0	0
823	0	0	1	1	0	0
824	0	0	1	1	0	0
825	0	0	1	1	0	0
826	0	0	1	1	0	0
827	0	0	1	1	0	0
828	0	0	1	1	0	0
829	0	0	1	1	0	0
830	0	0	1	1	0	0
831	0	0	1	1	0	0
832	0	0	1	1	0	0
833	0	0	1	1	0	0
834	0	0	1	1	0	0
835	0	0	1	1	0	0
836	0	0	1	1	0	0
837	0	0	1	1	0	0
838	0	0	1	1	0	0
839	0	0	1	1	0	0
840	0	0	1	1	0	0
841	0	0	1	1	0	0
842	0	0	1	1	0	0
843	0	0	1	1	0	0
844	0	0	1	1	0	0
845	0	0	1	1	0	0
846	0	0	1	1	0	0
847	0	0	1	1	0	0
848	0	0	1	1	0	0
849	0	0	1	1	0	0
850	0	0	1	1	0	0
851	0	0	1	1	0	0
852	0	0	1	1	0	0

853	0	0	1	1	0	0
854	0	0	1	1	0	0
855	0	0	1	1	0	0
856	0	0	1	1	0	0
857	0	0	1	1	0	0
858	0	0	1	1	0	0
859	0	0	1	1	0	0
860	0	0	1	1	0	0
861	0	0	1	1	0	0
862	0	0	1	1	0	0
863	0	0	1	1	0	0
864	0	0	1	1	0	0
865	0	0	1	1	0	0
866	0	0	1	1	0	0
867	0	0	1	1	0	0
868	0	0	1	1	0	0
869	0	0	1	1	0	0
870	0	0	1	1	0	0
871	0	0	1	1	0	0
872	0	0	1	1	0	0
873	0	0	1	1	0	0
874	0	0	1	1	0	0
875	0	0	1	1	0	0
876	0	0	1	1	0	0
877	0	0	1	1	0	0
878	0	0	1	1	0	0
879	0	0	1	1	0	0
880	0	0	1	1	0	0
881	0	0	1	1	0	0
882	0	0	1	1	0	0
883	0	0	1	1	0	0
884	0	0	1	1	0	0
885	0	0	1	1	0	0
886	0	0	1	1	0	0
887	0	0	1	1	0	0
888	0	0	1	1	0	0
889	0	0	1	1	0	0
890	0	0	1	1	0	0
891	0	0	1	1	0	0

892	0	0	1	1	0	0
893	0	0	1	1	0	0
894	0	0	1	1	0	0
895	0	0	1	1	0	0
896	0	0	1	1	0	0
897	0	0	1	1	0	0
898	0	0	1	1	0	0
899	0	0	1	1	0	0
900	0	0	1	1	0	0
901	0	0	1	1	0	0
902	0	0	1	1	0	0
903	0	0	1	1	0	0
904	0	0	1	1	0	0
905	0	0	1	1	0	0
906	0	0	1	1	0	0
907	0	0	1	1	0	0
908	0	0	1	1	0	0
909	0	0	1	1	0	0
910	0	0	1	1	0	0
911	0	0	1	1	0	0
912	0	0	1	1	0	0
913	0	0	1	1	0	0
914	0	0	1	1	0	0
915	0	0	1	1	0	0
916	0	0	1	1	0	0
917	0	0	1	1	0	0
918	0	0	1	1	0	0
919	0	0	1	1	0	0
920	0	0	1	1	0	0
921	0	0	1	1	0	0
922	0	0	1	1	0	0
923	0	0	1	1	0	0
924	0	0	1	1	0	0
925	0	0	1	1	0	0
926	0	0	1	1	0	0
927	0	0	1	1	0	0
928	0	0	1	1	0	0
929	0	0	1	1	0	0
930	0	0	1	1	0	0

931	0	0	1	1	0	0
932	0	0	1	1	0	0
933	0	0	1	1	0	0
934	0	0	1	1	0	0
935	0	0	1	1	0	0
936	0	0	1	1	0	0
937	0	0	1	1	0	0
938	0	0	1	1	0	0
939	0	0	1	1	0	0
940	0	0	1	1	0	0
941	0	0	1	1	0	0
942	0	0	1	1	0	0
943	0	0	1	1	0	0
944	0	0	1	1	0	0
945	0	0	1	1	0	0
946	0	0	1	1	0	0
947	0	0	1	1	0	0
948	0	0	1	1	0	0
949	0	0	1	1	0	0
950	0	0	1	1	0	0
951	0	0	1	1	0	0
952	0	0	1	1	0	0
953	0	0	1	1	0	0
954	0	0	1	1	0	0
955	0	0	1	1	0	0
956	0	0	1	1	0	0
957	0	0	1	1	0	0
958	0	0	1	1	0	0
959	0	0	1	1	0	0
960	0	0	1	1	0	0
961	0	0	1	1	0	0
962	0	0	1	1	0	0
963	0	0	1	1	0	0
964	0	0	1	1	0	0
965	0	0	1	1	0	0
966	0	0	1	1	0	0
967	0	0	1	1	0	0
968	0	0	1	1	0	0
969	0	0	1	1	0	0

970	0	0	1	1	0	0
971	0	0	1	1	0	0
972	0	0	1	1	0	0
973	0	0	1	1	0	0
974	0	0	1	1	0	0
975	0	0	1	1	0	0
976	0	0	1	1	0	0
977	0	0	1	1	0	0
978	0	0	1	1	0	0
979	0	0	1	1	0	0
980	0	0	1	1	0	0
981	0	0	1	1	0	0
982	0	0	1	1	0	0
983	0	0	1	1	0	0
984	0	0	1	1	0	0
985	0	0	1	1	0	0
986	0	0	1	1	0	0
987	0	0	1	1	0	0
988	0	0	1	1	0	0
989	0	0	1	1	0	0
990	0	0	1	1	0	0
991	0	0	1	1	0	0
992	0	0	1	1	0	0
993	0	0	1	1	0	0
994	0	0	1	1	0	0
995	0	0	1	1	0	0
996	0	0	1	1	0	0
997	0	0	1	1	0	0
998	0	0	1	1	0	0
999	0	0	1	1	0	0
1000	0	0	1	1	0	0
1001	0	0	1	1	0	0
1002	0	0	1	1	0	0
1003	0	0	1	1	0	0
1004	0	0	1	1	0	0
1005	0	0	1	1	0	0
1006	0	0	1	1	0	0
1007	0	0	1	1	0	0
1008	0	0	1	1	0	0

1009	0	0	1	1	0	0
1010	0	0	1	1	0	0
1011	0	0	1	1	0	0
1012	0	0	1	1	0	0
1013	0	0	1	1	0	0
1014	0	0	1	1	0	0
1015	0	0	1	1	0	0
1016	0	0	1	1	0	0
1017	0	0	1	1	0	0
1018	0	0	1	1	0	0
1019	0	0	1	1	0	0
1020	0	0	1	1	0	0
1021	0	0	1	1	0	0
1022	0	0	1	1	0	0
1023	0	0	1	1	0	0
1024	0	0	1	1	0	0
1025	0	0	1	1	0	0
1026	0	0	1	1	0	0
1027	0	0	1	1	0	0
1028	0	0	1	1	0	0
1029	0	0	1	1	0	0
1030	0	0	1	1	0	0
1031	0	0	1	1	0	0
1032	0	0	1	1	0	0
1033	0	0	1	1	0	0
1034	0	0	1	1	0	0
1035	0	0	1	1	0	0
1036	0	0	1	1	0	0
1037	0	0	1	1	0	0
1038	0	0	1	1	0	0
1039	0	0	1	1	0	0
1040	0	0	1	1	0	0
1041	0	0	1	1	0	0
1042	0	0	1	1	0	0
1043	0	0	1	1	0	0
1044	0	0	1	1	0	0
1045	0	0	1	1	0	0
1046	0	0	1	1	0	0
1047	0	0	1	1	0	0

1048	0	0	1	1	0	0
1049	0	0	1	1	0	0
1050	0	0	1	1	0	0
1051	0	0	1	1	0	0
1052	0	0	1	1	0	0
1053	0	0	1	1	0	0
1054	0	0	1	1	0	0
1055	0	0	1	1	0	0
1056	0	0	1	1	0	0
1057	0	0	1	1	0	0
1058	0	0	1	1	0	0
1059	0	0	1	1	0	0
1060	0	0	1	1	0	0
1061	0	0	1	1	0	0
1062	0	0	1	1	0	0
1063	0	0	1	1	0	0
1064	0	0	1	1	0	0
1065	0	0	1	1	0	0
1066	0	0	1	1	0	0
1067	0	0	1	1	0	0
1068	0	0	1	1	0	0
1069	0	0	1	1	0	0
1070	0	0	1	1	0	0
1071	0	0	1	1	0	0
1072	0	0	1	1	0	0
1073	0	0	1	1	0	0
1074	0	0	1	1	0	0
1075	0	0	1	1	0	0
1076	0	0	1	1	0	0
1077	0	0	1	1	0	0
1078	0	0	1	1	0	0
1079	0	0	1	1	0	0
1080	0	0	1	1	0	0
1081	0	0	1	1	0	0
1082	0	0	1	1	0	0
1083	0	0	1	1	0	0
1084	0	0	1	1	0	0
1085	0	0	1	1	0	0
1086	0	0	1	1	0	0

1087	0	0	1	1	0	0
1088	0	0	1	1	0	0
1089	0	0	1	1	0	0
1090	0	0	1	1	0	0
1091	0	0	1	1	0	0
1092	0	0	1	1	0	0
1093	0	0	1	1	0	0
1094	0	0	1	1	0	0
1095	0	0	1	1	0	0
1096	0	0	1	1	0	0
1097	0	0	1	1	0	0
1098	0	0	1	1	0	0
1099	0	0	1	1	0	0
1100	0	0	1	1	0	0
1101	0	0	1	1	0	0
1102	0	0	1	1	0	0
1103	0	0	1	1	0	0
1104	0	0	1	1	0	0
1105	0	0	1	1	0	0
1106	0	0	1	1	0	0
1107	0	0	1	1	0	0
1108	0	0	1	1	0	0
1109	0	0	1	1	0	0
1110	0	0	1	1	0	0
1111	0	0	1	1	0	0
1112	0	0	1	1	0	0
1113	0	0	1	1	0	0
1114	0	0	1	1	0	0
1115	0	0	1	1	0	0
1116	0	0	1	1	0	0
1117	0	0	1	1	0	0
1118	0	0	1	1	0	0
1119	0	0	1	1	0	0
1120	0	0	1	1	0	0
1121	0	0	1	1	0	0
1122	0	0	1	1	0	0
1123	0	0	1	1	0	0
1124	0	0	1	1	0	0
1125	0	0	1	1	0	0

1126	0	0	1	1	0	0
1127	0	0	1	1	0	0
1128	0	0	1	1	0	0
1129	0	0	1	1	0	0
1130	0	0	1	1	0	0
1131	0	0	1	1	0	0
1132	0	0	1	1	0	0
1133	0	0	1	1	0	0
1134	0	0	1	1	0	0
1135	0	0	1	1	0	0
1136	0	0	1	1	0	0
1137	0	0	1	1	0	0
1138	0	0	1	1	0	0
1139	0	0	1	1	0	0
1140	0	0	1	1	0	0
1141	0	0	1	1	0	0
1142	0	0	1	1	0	0
1143	0	0	1	1	0	0
1144	0	0	1	1	0	0
1145	0	0	1	1	0	0
1146	0	0	1	1	0	0
1147	0	0	1	1	0	0
1148	0	0	1	1	0	0
1149	0	0	1	1	0	0
1150	0	0	1	1	0	0
1151	0	0	1	1	0	0
1152	0	0	1	1	0	0
1153	0	0	1	1	0	0
1154	0	0	1	1	0	0
1155	0	0	1	1	0	0
1156	0	0	1	1	0	0
1157	0	0	1	1	0	0
1158	0	0	1	1	0	0
1159	0	0	1	1	0	0
1160	0	0	1	1	0	0
1161	0	0	1	1	0	0
1162	0	0	1	1	0	0
1163	0	0	1	1	0	0
1164	0	0	1	1	0	0

1165	0	0	1	1	0	0
1166	0	0	1	1	0	0
1167	0	0	1	1	0	0
1168	0	0	1	1	0	0
1169	0	0	1	1	0	0
1170	0	0	1	1	0	0
1171	0	0	1	1	0	0
1172	0	0	1	1	0	0
1173	0	0	1	1	0	0
1174	0	0	1	1	0	0
1175	0	0	1	1	0	0
1176	0	0	1	1	0	0
1177	0	0	1	1	0	0
1178	0	0	1	1	0	0
1179	0	0	1	1	0	0
1180	0	0	1	1	0	0
1181	0	0	1	1	0	0
1182	0	0	1	1	0	0
1183	0	0	1	1	0	0
1184	0	0	1	1	0	0
1185	0	0	1	1	0	0
1186	0	0	1	1	0	0
1187	0	0	1	1	0	0
1188	0	0	1	1	0	0
1189	0	0	1	1	0	0
1190	0	0	1	1	0	0
1191	0	0	1	1	0	0
1192	0	0	1	1	0	0
1193	0	0	1	1	0	0
1194	0	0	1	1	0	0
1195	0	0	1	1	0	0
1196	0	0	1	1	0	0
1197	0	0	1	1	0	0
1198	0	0	1	1	0	0
1199	0	0	1	1	0	0
1200	0	0	1	1	0	0
1201	0	0	1	1	0	0
1202	0	0	1	1	0	0
1203	0	0	1	1	0	0

1204	0	0	1	1	0	0
1205	0	0	1	1	0	0
1206	0	0	1	1	0	0
1207	0	0	1	1	0	0
1208	0	0	1	1	0	0
1209	0	0	1	1	0	0
1210	0	0	1	1	0	0
1211	0	0	1	1	0	0
1212	0	0	1	1	0	0
1213	0	0	1	1	0	0
1214	0	0	1	1	0	0
1215	0	0	1	1	0	0
1216	0	0	1	1	0	0
1217	0	0	1	1	0	0
1218	0	0	1	1	0	0
1219	0	0	1	1	0	0
1220	0	0	1	1	0	0
1221	0	0	1	1	0	0
1222	0	0	1	1	0	0
1223	0	0	1	1	0	0
1224	0	0	1	1	0	0
1225	0	0	1	1	0	0
1226	0	0	1	1	0	0
1227	0	0	1	1	0	0
1228	0	0	1	1	0	0
1229	0	0	1	1	0	0
1230	0	0	1	1	0	0
1231	0	0	1	1	0	0
1232	0	0	1	1	0	0
1233	0	0	1	1	0	0
1234	0	0	1	1	0	0
1235	0	0	1	1	0	0
1236	0	0	1	1	0	0
1237	0	0	1	1	0	0
1238	0	0	1	1	0	0
1239	0	0	1	1	0	0
1240	0	0	1	1	0	0
1241	0	0	1	1	0	0
1242	0	0	1	1	0	0

1243	0	0	1	1	0	0
1244	0	0	1	1	0	0
1245	0	0	1	1	0	0
1246	0	0	1	1	0	0
1247	0	0	1	1	0	0
1248	0	0	1	1	0	0
1249	0	0	1	1	0	0
1250	0	0	1	1	0	0
1251	0	0	1	1	0	0
1252	0	0	1	1	0	0
1253	0	1	0	0	1	1
1254	0	1	0	0	1	1
1255	0	1	0	0	1	1
1256	0	1	0	0	1	1
1257	0	1	0	0	1	1
1258	0	1	0	0	1	1
1259	0	1	0	0	1	1
1260	0	1	0	0	1	1
1261	0	1	0	0	1	1
1262	0	1	0	0	1	1
1263	0	1	0	0	1	1
1264	0	1	0	0	1	1
1265	0	1	0	0	1	1
1266	0	1	0	0	1	1
1267	0	1	0	0	1	1
1268	0	1	0	0	1	1
1269	0	1	0	0	1	1
1270	0	1	0	0	1	1
1271	0	1	0	0	1	1
1272	0	1	0	0	1	1
1273	0	1	0	0	1	1
1274	0	1	0	0	1	1
1275	0	1	0	0	1	1
1276	0	1	0	0	1	1
1277	0	1	0	0	1	1
1278	0	1	0	0	1	1
1279	0	1	0	0	1	1
1280	0	1	0	0	1	1
1281	1	0	0	0	1	1

1282	1	0	0	0	1	1	
1283	1	0	0	0	1	1	
1284	1	0	0	0	1	1	
1285	1	0	0	0	1	1	
1286	1	0	0	0	1	1	
1287	1	0	0	0	1	1	
1288	1	0	0	0	1	1	
1289	1	0	0	0	1	1	
1290	1	0	0	0	1	1	
1291	1	0	0	0	1	1	
1292	1	0	0	0	1	1	
1293	1	0	0	0	1	1	
1294	1	0	0	0	1	1	
1295	1	0	0	0	1	1	
1296	1	0	0	0	1	1	
1297	1	0	0	0	1	1	
1298	1	0	0	0	1	1	
1299	1	0	0	0	1	1	
1300	1	0	0	0	1	1	
1301	1	0	0	0	1	1	
1302	1	0	0	0	1	1	
1303	1	0	0	0	1	1	
Jumlah				1252	1252	51	51