Penerapan Text Mining dalam Pengkodean Penyakit Pasien Berdasarkan Kode ICD 10 Untuk Penyakit Dalam

Parjono

20917050

Tesis diajukan sebagai syarat untuk meraih gelar Magister Komputer

Konsentrasi Informatika Medis

Program Studi Informatika Program Magister

Fakultas Teknologi Industri

Universitas Islam Indonesia

2023

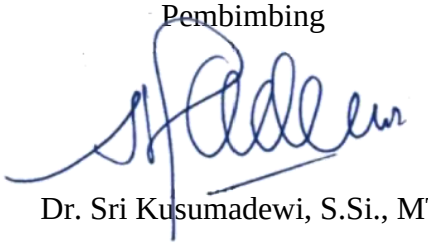
Lembar Pengesahan Pembimbing

Penerapan Text Mining dalam Pengkodean Penyakit Pasien Berdasarkan Kode ICD 10 Untuk Penyakit Dalam

Parjono
20917050

Yogyakarta, 24 Januari 2024

Pembimbing



Dr. Sri Kusumadewi, S.Si., MT

Lembar Pengesahan Penguji

Penerapan Text Mining dalam Pengkodean Penyakit Pasien Berdasarkan Kode ICD 10 Untuk Penyakit Dalam

Parjono

20917050

Yogyakarta, Februari 2024

Tim Penguji,

Dr. Sri Kusumadewi, S.Si., MT
Ketua

Izzati Muhimmah, ST., M.Sc, Ph.D
Anggota I

Irving Vitra Paputungan, S.T., M.Sc., Ph.D
Anggota II

05/03/2024

Mengetahui,
Ketua Program Studi Informatika Program Magister
Universitas Islam Indonesia

UNIVERSITAS ISLAM INDONESIA
PROGRAM STUDI INFORMATIKA
PROGRAM MAGISTER

Irving Vitra Paputungan, S.T., M.Sc., Ph.D.

Abstrak

Penerapan Text Mining dalam Pengkodean Penyakit Pasien Berdasarkan Kode ICD 10 Untuk Penyakit Dalam

Pengkodean penyakit yang lazim dilakukan di fasilitas layanan kesehatan memiliki 2 metode. Metode pertama koding penyakit dilakukan langsung oleh dokter dengan bantuan referensi master ICD10, metode ini memiliki kelebihan lebih cepat namun memiliki kekurangan yakni menambah beban kerja dokter. Metode kedua adalah koding dilakukan oleh petugas Rekam Medis terlatih, metode ini memiliki kelebihan koding yang lebih presisi, namun memiliki kekurangan yaitu lebih membutuhkan waktu dan tingkat *human error* yang tinggi karena diagnosis yang ditulis dokter perlu dikonfirmasikan ulang apabila ada keraguan didalamnya.Penelitian ini bertujuan membangun model klasifikasi penyakit dari diagnosis dokter berdasarkan kode ICD10 untuk memangkas waktu koding. Metodologi yang akan digunakan adalah Machine Learning klasifikasi *unstructure text* diagnosis dokter yang dikategorikan berdasarkan kode ICD10. Data pasien penyakit dalam yang dilakukan *pre-processing*, *text representing*, *modeling*, *evaluating*, dan pengujian. Pada tahapan *pre-processing* dilakukan pengenalan singkatan dalam diagnosis dengan acuan kamus singkatan yang sudah ditetapkan.Hasil evaluasi model didapatkan akurasi sebesar 91,8% yang berarti model dapat mengukur sebanyak 91,8% dari seluruh prediksi yang benar dibandingkan dengan total prediksi. Kemudian pengujian dengan data yang belum pernah dikenali model didapatkan akurasi sebesar 89,3% angka tersebut menunjukkan bahwa model memiliki kinerja yang relatif baik pada dataset pengujian.Model *Machine Learning* klasifikasi menggunakan Algoritma Neural Network dengan bantuan kamus singkatan mampu mengklasifikasikan *unstructure text* diagnosis dokter penyakit dalam ke dalam kode ICD10 dengan akurasi sebesar 91,8%. Selanjutnya model dari penelitian ini dapat digunakan secara terintegrasi dengan Sistem Management Rumah Sakit (SIMRS) sebagai alat bantu petugas koding dalam mengklasifikasikan diagnosis ke dalam kode ICD10.

Kata kunci

rekam medis, *text mining*, *machine learning*, *neural network*, ICD 10, diagnosis

Abstract

Application of Text Mining in Coding Patient Diseases Based on ICD 10 Code for Internist

Disease coding which is commonly carried out in health care facilities has 2 methods. The first method of disease coding is carried out directly by the doctor with the help of the ICD10 master reference. This method has the advantage of being faster but has the disadvantage of increasing the doctor's workload. The second method is coding carried out by trained Medical Records officers. This method has the advantage of more precise coding, but has the disadvantage of requiring more time and a high level of human error because the diagnosis written by the doctor needs to be reconfirmed if there is doubt in it. This research aims to build disease classification model of doctor diagnosis based on ICD10 codes to cut coding time. The methodology that will be used is Machine Learning unstructured text classification of doctor's diagnoses which are categorized based on ICD10 codes. Internal medicine patient data underwent pre-processing, text representing, modeling, evaluating and testing. At the pre-processing stage, abbreviations were introduced in the diagnosis using reference to a predetermined abbreviation dictionary. The results of the model evaluation showed an accuracy of 91.8%, which means the model can measure 91.8% of all correct predictions compared to the total predictions. Then testing with data that had never been recognized by the model, obtained an accuracy of 89.3%. This figure shows that the model has relatively good performance on the testing dataset. The machine learning classification model uses the Neural Network algorithm with the help of an abbreviation dictionary which is able to classify unstructured text of internal medicine doctors' diagnoses. into ICD10 codes with an accuracy of 91.8%. Furthermore, the model from this research can be used in an integrated manner with the Hospital Management System (HMS) as a tool to assist coding officers in classifying diagnoses into ICD10 codes.

Keywords

medical record, text mining, machine learning, neural network, ICD 10, diagnosis

Pernyataan Keaslian Tulisan

Dengan ini saya menyatakan bahwa tesis ini merupakan tulisan asli dari penulis, dan tidak berisi material yang telah diterbitkan sebelumnya atau tulisan dari penulis lain terkecuali referensi atas material tersebut telah disebutkan dalam tesis. Apabila ada kontribusi dari penulis lain dalam tesis ini, maka penulis lain tersebut secara eksplisit telah disebutkan dalam tesis ini.

Dengan ini saya juga menyatakan bahwa segala kontribusi dari pihak lain terhadap tesis ini, termasuk bantuan analisis statistik, desain survei, analisis data, prosedur teknis yang bersifat signifikan, dan segala bentuk aktivitas penelitian yang dipergunakan atau dilaporkan dalam tesis ini telah secara eksplisit disebutkan dalam tesis ini.

Segala bentuk hak cipta yang terdapat dalam material dokumen tesis ini berada dalam kepemilikan pemilik hak cipta masing-masing. Apabila dibutuhkan, penulis juga telah mendapatkan izin dari pemilik hak cipta untuk menggunakan ulang materialnya dalam tesis ini.

Yogyakarta, Februari 2024

Parjono, S.Kom

Daftar Publikasi

Parjono, P., & Kusumadewi, S. (2023). Pemodelan Text Mining dalam Pengkodean Penyakit Pasien Berdasar Kode ICD 10. Jurnal Nasional Teknologi dan Sistem Informasi, 9(2), 200-207. doi:https://doi.org/10.25077/TEKNOSI.v9i2.2023.200-207.

Publikasi yang menjadi bagian dari tesis

Publikasi berikut menjadi bagian dari Bab 1, Bab 2, Bab 3 dan Bab 4.

Sitasi publikasi 1

Kontributor	Jenis Kontribusi
Parjono	Mendesain eksperimen (30%) Menulis *paper* (80%)
Sri Kusumadewi	Mendesain eksperimen (60%) Menulis dan mengedit *paper* (30%)

Halaman Kontribusi

Manajemen salah satu Rumah Sakit Swasta di Surakarta, Departemen Teknologi Informasi, dan Unit Rekam Medis

Halaman Persembahan

Alhamdulillahi Rabbil Aalamin, sujud serta syukur kepada Allah SWT.
Terimakasih atas karunia-Mu yang telah memberikan kemudahan dan kelancaran sehingga tesis ini dapat terselesaikan dengan baik.
Halaman persembahan ini juga ditujukkan sebagai ungkapan terimakasih kepada keluarga saya, Ibu dan Istri tercinta yang telah mendoakan dan memberikan dukungan penuh selama perjuangan menempuh pendidikan.
Teruntuk untuk ananda Fathiya Aaira Adha dan Fabian Shaka Athallah terimakasih atas semangat yang diberikan, semoga tesis ini nantinya bisa menjadikan motivasi dan manfaat untuk kalian dalam menjalankan peran sebagai pribadi yang sholeh dan sholehah serta berguna untuk sesama.

“ You both could do better than me.”

Kata Pengantar

Assalamualaikum Warrahmatullahi Wabarakatuh

Segala puji dan syukur dipanjatkan kehadirat Allah SWT, karena atas berkat rahmat dan ridho-Nya tesis dengan judul "Penerapan Text Mining dalam Pengkodean Penyakit Pasien Berdasarkan Kode ICD 10" ini dapat diselesaikan.

Tesis ini disusun untuk memenuhi salah satu persyaratan memperoleh gelar Magister Komputer (M.Kom) dalam bidang Informatika Medis pada Program Studi Magister Teknik Informatika Universitas Islam Indonesia.

Penulis menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak penyusunan tesis ini tidak akan terselesaikan dengan baik. Maka itu pada kesempatan ini dengan penuh kerendahan hati dihaturkan ucapan terima kasih yang sebesar-besarnya kepada :

1. Bapak Dr. R. Teduh Dirgahayu, S.T., M.Sc selaku Ketua Program Pascasarjana Fakultas Teknologi Industri Universitas Islam Indonesia.
2. Ibu Dr. Sri Kusumadewi, S.Si., MT selaku pembimbing yang telah memberikan bimbingan, masukan, materi dan motivasi selama proses penelitian ini.
3. Segenap pimpinan, dosen dan karyawan Program Studi Magister Teknik Informatika Universitas Islam Indonesia, khususnya para dosen yang telah memberikan ilmunya kepada peneliti selama masa kuliah.
4. Seluruh Manajemen dan Civitas Hospitalia yang sudah banyak membantu hingga terselesainya Thesis ini

   Akhirnya sebuah harapan semoga tesis ini bermanfaat bagi kita semua.

   Amien yarobalalamin.....

   Wassalamualaikum Warrahmatullahi Wabarakatuh.

Surakarta, Desember 2023

Parjono, S.Kom

Daftar Isi

Daftar Tabel

Daftar Gambar

Glosarium

ICD	- International Classification of Diseases
ICOPIM	- International Classification of Procedures in Medicine
RM	- Rekam Medis
EHR	- Electronic Health Record
EMR	- Electronic Medical Record
BMC	- Bunda Medical Center
NLP	- Natural Language Processing
JST	- Jaringan Syaraf Tiruan
ROC	- Receiver Operator Characteristic

BAB 1
Pendahuluan

1.1 Latar Belakang

Menemukan pola dan hubungan dalam kumpulan data yang dihasilkan setiap hari menjadi tantangan tersendiri. Salah satu tantangan terbesar adalah text mining, yang melibatkan penerapan teknik untuk mengekstrak informasi yang relevan dari volume besar teks, umumnya dalam bahasa alami yang tidak terstruktur. Meskipun banyak kemajuan di bidang kesehatan, namun masih banyak bidang yang belum tereksplorasi dan masalah yang belum terpecahkan. Sebagai contoh masih terdapat keterbatasan dalam pengambilan informasi pasien dimana analisa dan pengambilan kesimpulan penyakit masih perlu ditingkatkan.

Penulisan rekam medis pasien sangat berbeda dengan standar penulisan lain seperti penulisan berita, buku, laporan dan lainnya, rekam medis pasien juga ditulis oleh para profesional pemberi asuhan lain yang terlibat dalam perawatan pasien dimana setiap profesional mempunyai kemampuan menulis yang berbeda-beda. Hal ini menjadikan pengambilan data atau informasi terkait pasien tersebut menjadi lebih sulit dikarenakan tidak adanya standar baku penulisan karena biasanya seperti diagnosa dokter ditulis secara deskriptif (Chen Y, 2017)

Di setiap Rumah Sakit biasanya terdapat unit khusus yang menangani dokumen-dokumen medis pasien yakni Unit Rekam Medis. Menurut Keputusan Menteri Kesehatan nomor 377/MenKes/SK/III/2007 tentang Standar Profesi Perekam Medis dan Informasi Kesehatan, klasifikasi dan kodefikasi penyakit, masalah kesehatan, dan tindakan medis adalah kompetensi yang harus dimiliki oleh perekam medis. Selama beberapa tahun, penggunaan prosedur dan istilah penyakit yang berbeda membuat data morbiditas dan mortalitas tidak akurat. Para ahli penyelenggara kesehatan menghasilkan perbendaharaan istilah medis klinis, sistem klasifikasi penyakit, dan nomenklatur penyakit dalam upaya mengorganisasikan dan menstandarkan bahasa medis.

Kualitas data terkode sangat penting bagi tenaga kerja manajemen informasi kesehatan, fasilitas asuhan kesehatan, dan profesional kesehatan. Membuat kode diagnosis dan tindakan medis ini dengan benar akan membantu asuhan keperawatan, efektivitas biaya klaim, meningkatkan kualitas pelayanan, membandingkan data morbiditas dan mortalitas,

menampilkan sepuluh penyakit utama, dan masalah lain yang berkaitan dengan pelayanan kesehatan. (Puspitasari, 2017)

Pengkodean penyakit yang dilakukan secara manual sangat bergantung pada kemampuan petugas koding. Petugas koding memiliki kemampuan dalam menginterpretasikan diagnosa dan data pasien berdasarkan pengalaman petugas tersebut. Proses koding juga membutuhkan waktu yang cukup lama serta memiliki potensi *human error* yang tinggi (Dalianis, 2018)

Pada (Nur Maimun, 2018) dilakukan evaluasi pengkodean diagnosis yang dilakukan oleh petugas koding Rekam Medis, ditemukan dari 463 berkas rekam medis rawat inap pada formulir ringkasan masuk dan keluar 93 kesalahan atau ketidaktepatan pengkodean diagnosis, penentuan diagnosa utama atau sekunder, serta ketidaktelitian petugas koding dalam pengkodingan penyakit ke dalam kode ICD-10 dimana hal tersebut dapat menghambat dalam proses klaim jaminan kesehatan nasional dan dapat merugikan rumah sakit.

Saat ini pengkodean penyakit menggunakan dua metode, yang pertama adalah klinisi/ dokter paham tentang pengkodean ICD-10 sehingga alih-alih menuliskan diagnosis secara *free text* dokter langsung menuliskan diagnosis dalam kode ICD-10 dengan bantuan aplikasi dari WHO berupa nomenklatur kode penyakit yang kemudian akan diaudit oleh petugas koding rekam medis. Metode ini tentu akan meningkatkan beban kerja dokter dimana dokter harus memeriksa pasien, memberikan tindakan pasien, meresepkan obat dan masih harus mencari kode penyakit berdasar ICD-10.

Metode kedua adalah petugas koding dari rekam medis yang memberikan kode penyakit berdasar diagnosis dokter. Pada metode kedua apabila ada hal yang sekiranya diragukan, petugas koding akan mengkonfirmasi ke dokter yang merawat pasien. Metode kedua ini akan memiliki kualitas yang lebih baik dibandingkan dengan metode pertama, namun metode pertama akan lebih efisien karena dokter langsung memberikan kode penyakit berstandar ICD-10 hanya saja perlu dilakukan review ulang oleh petugas koding rekam medik (Lingling Zhou, 2020)

Pada pelaksanaannya pengkodean penyakit masih menggunakan metode kedua, yakni mengandalkan petugas koding untuk mengkodekan penyakit. Dalam (M. K. Ross, 2014) disebutkan bahwa erat kaitanya ketepatan terminologi medis dengan keakuratan kode diagnosis rawat jalan oleh petugas koding, peluang terminologi tidak tepat akan menyebabkan ketidakakuratan kode diagnosis sebanyak 1,7 kali lebih besar dibanding dengan terminologi medis yang tepat.

Dari kebanyakan diagnosis medis yang berupa deskripsi *free text*, kemiripan makna serta istilah medis yang memiliki kekhususan daripada istilah umum akan menjadi tantangan tersendiri dalam mengekstrak informasi yang berada di dalamnya. Hal ini yang mendorong penulis untuk menerapkan *Text Mining* dalam membantu pengkodean penyakit melalui data diagnosis dokter sehingga diharapkan proses pengkodean penyakit menjadi lebih cepat dan mengurangi aspek human error khususnya untuk penyakit di Indonesia.

Tujuan dari pemodelan *text mining* untuk pendokedan penyakit ini adalah terbentuknya model machine learning untuk pengkodean penyakit berdasarkan kode ICD-10. Model tersebut diharapkan dapat membantu Rumah Sakit dalam pengkodean penyakit dengan lebih cepat, berkurangnya tingkat human error dan mengurangi beban petugas rekam medis. Pada akhirnya program integrasi data yang dicanangkan oleh Kementrian Kesehatan dapat direalisasikan terutama dari penyajian data fasyankes yang mendekati *realtime*. Masyarakat sebagai pemilik data dapat memperoleh haknya yakni rekam medis sebagai pasien, tepat setelah pasien meninggalkan fasyankes yang dipilihnya dan Kementrian Kesehatan sebagai pengatur regulasi memperoleh data agregasi dari seluruh fasyankes untuk kepentingan pengambilan kebijakan secara tepat dan cepat.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah diatas maka dapat dirumuskan masalah yang akan dibahas dalam penelitian ini adalah bagaimana membuat model untuk mengekstrak diagnosis dokter yang berbentuk *free text* deskriptif menjadi kategori Kode ICD 10 dan menerapkannya pada Unit Rekam Medis untuk membantu dalam pengkodean penyakit menggunakan metode *Machine Learning*.

1.3 Batasan Masalah

Agar pembahasan penulisan penelitian ini tidak meluas, maka perlu adanya batasan sebagai yakni sebagai berikut :

a. Penelitian ini hanya membatasi pada diagnosis penyakit pasien yang periksa ke Klinik Penyakit dalam
b. Data yang digunakan adalah 5 kategori kode ICD 10 yakni E11.9 Type 2 diabetes mellitus without complications, I10 Essential (primary) hypertension, I11.9 Hypertensive heart disease without (congestive) heart failure, K21.9 Gastro-oesophageal reflux disease without oesophagitis, K30 Functional dyspepsia
c. Penerapan akan diujicobakan pada petugas koding di Unit Rekam Medis

1.4 Tujuan Penelitian

Merujuk pada latar belakang masalah diatas adapun tujuan dari penelitian ini adalah sebagai berikut :

a. Membangun model klasifikasi untuk mengelompokkan diagnosis dokter yang berupa *free text* deskriptif ke dalam kode ICD 10 dengan bantuan kamus singakatan
b. Menerapkan model klasifikasi tersebut kepada petugas koding di Unit Rekam Medis untuk membantu koding penyakit menjadi lebih cepat

1.5 Manfaat Penelitian

Berdasarkan hasil yang didapatkan pada penelitian ini penulis dapat terwujud model klasifikasi yang dapat digunakan untuk aplikasi Sistem Manajemen Rumah Sakit (SIMRS) dalam mengelompokkan diagnosis penyakit berdasarkan kode ICD10.

Kemudian manfaat lain yang didapatkan yaitu membantu petugas koding di Unit Rekam Medis dalam mengkoding diagnosis penyakit dengan menggunakan model yang sudah dibuat dengan lebih cepat dan minimal *human error*.

1.6 Sistematika Penulisan

Penulisan laporan tesis disusun dalam beberapa bab dan masing – masing bab terdiri dari sub bab dengan serangkaian pembahasan di dalamnya. Sistematika penulisan tesis ini susunannya sebagai berikut:

BAB 1 Pendahuluan

Bab ini menjelaskan latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB 2 Tinjauan Pustaka

Bab ini menjelaskan mengenai dasar teori yang digunakan dengan tambahan literature review dari penelitian sebelumnya untuk mendukung penerapan penelitian.

BAB 3 Metodologi Penelitian

Bab ini membahas gambaran umum sistem dan langkah-langkah dalam pelaksanaan penelitian yang terdiri dari perencanaan, akusisi pengetahuan, implementasi dan evaluasi.

BAB 4 Implementasi Dan Pengujian

Bab ini membahas mengenai hasil dari penerapan sistem pakar yang diteliti. Bagian ini juga membahas mengenai pengujian sistem baik dari segi desain maupun pengaruh sistem ketika diterapkan.

BAB 5 Kesimpulan dan Saran

Bab ini membahas tentang kesimpulan yang merupakan rangkuman dari hasil penelitian yang dilakukan, dan juga berisi saran-saran terhadap penelitian selanjutnya.

BAB 2
Tinjauan Pustaka

2.1 Penelitian Terdahulu

Beberapa penelitian yang telah dilakukan berkenaan dengan penerapan Machine Learning pada sektor kesehatan sebagaimana pada penelitian (Boytcheva, 2011) tentang pengkodean ICD-10 menggunakan berbagai metode telah dilakukan di beberapa negara, seperti di Bulgaria, Spanyol, Inggris, Perancis, Hungaria, dan Italia. Pada penelitian "Automatic Matching of ICD-10 codes to Diagnoses in Discharge Letters" oleh S. Boytcheva, pengkodean penyakit dilakukan secara otomatis menggunakan surat keluar pasien dengan bahasa Bulgaria yang diklasifikasikan ke dalam kode ICD 10 dengan menggunakan pendekatan klasifikasi multi kelas Support Vector Machine pengklasifikasian text memiliki kendala bahasa, dimana tidak terdapat standar penulisan surat keluar pasien secara internasional. Pada penelitian tersebut proses pre-processing dilakukan dengan mengubah Cyrillic ke bahasa latin agar lebih mudah diubah menjadi vector dan belum memasukkan sumber singkatan kata dalam bahasa Bulgaria.

Penelitian (Dalianis, 2018) "Clinical Text Mining: Secondary Use of Electronic Patient Records," Clinical Text Mining: Secondary Use of Electronic Patient Records, mengkodekan ICD-10 untuk penyebab kematian dengan menganalisis deskripsi teks bebas dalam sertifikat kematian, bersama dengan laporan otopsi terkait dan buletin klinis, dari Kementerian Kesehatan Portugis. Penelitian ini memanfaatkan jaringan saraf untuk mengeksplorasi sifat hierarkis dari data input. Skor akurasi lebih dari 89%, 81%, dan 76%, masing-masing untuk bab ICD-10, blok, dan kode lengkap. Pada penelitian ini teks bebas tidak bersumber pada deskripsi standar yang ada pada ICD, namun digunakan pada sertifikat kematian.

Kemudian penelitian yang dilakukan di Klinik Venderbilt, New York dengan judul "ICF based automation system for spinal cord injuries rehabilitation," bertujuan untuk menentukan apakah program Natural Language Processing (NLP) dapat secara otomatis mengkode status fungsional informasi sesuai dengan persyaratan International Classification of Functioning, Disability, and Health (ICF). Pengkodean sangat penting untuk tujuan pembiayaan dan pencatatan. Peneliti memperluas NLP MedLEE yang ada untuk membuat kode ringkasan pemulangan rehabilitasi. Sepuluh Kode ICD-9 telah dipilih yang ada hubungan dengan perubahan status fungsional. Evaluasi pengkodean dilakukan oleh sistem NLP, petugas koding ahli, dan petugas koding non-ahli hasilnya adalah sistem NLP dapat

mengkodekan hasil klasifikasi yang sangat mirip dengan hasil koding yang dilakukan oleh petugas (R. Mahmoud, 2014).

Sistem klasifikasi untuk ICD-9-CM pada diagnosis radiologi sebagaimana (Hanna Suominen, 2007) sistem klasifikasi multi-label untuk penetapan otomatis kode diagnostik pada laporan radiologi. Sistem ini adalah rangkaian pengayaan teks, fitur seleksi dan dua pengklasifikasi. Telah dievaluasi di Computational Medicine Center's 2007 Medical Natural Language Processing Challenge dan memperoleh nilai F1-score sebesar 87,7%. Terutama pengayaan teks dan fitur komponen seleksi terbukti berkontribusi besar pada keberhasilan klasifikasi.

Penelitian (Luis Pereira, 2013) "ICD9-based Text Mining Approach to Children Epilepsy Classification," mengelompokkan penyakit Epilepsy anak ke dalam kode ICD 9 dengan pendekatan model klasifikasi yaitu k-nearest neighbor multi class. Pendekatan yang diusulkan menggunakan catatan kesehatan sebagai sumber data, langkah preprocessing yang menggunakan NLP dilengkapi oleh definisi model yang dapat didefinisikan untuk dapat digunakan dalam skenario yang ada. Dari hasil penelitian masih diperoleh akurasi yang cukup baik antara 60-70% namun karena data training yang diperoleh kurang lengkap maka dokter kurang percaya diri saat menggunakannya.

Pada penelitian "Applying Text Mining for Classifying Disease from Symptoms," klasifikasi kode penyakit berdasarkan gejala yang timbul, lalu penelitian ini hanya memiliki model klasifikasi untuk mendukung bahasa Thailand dan bahasa Inggris, dimana penelitian hanya berfokus pada kasus Orthopedics. secara garis besar penelitian ini membandingkan model klasifikasi dengan membangun metode 10 fold cross validation, selanjutnya hasil akan dilihat berdasarkan efisiensi dan model klasifikasi yang paling cocok. Hasilnya ditemukan bahwa model text mining yang paling cocok adalah jaringan syaraf dimana memiliki True Positive Rate sebesar 89.03%. Penelitian ini tidak menyertakan usia, suhu badan, dan tekanan darah pada saat mengklasifikasikan gejala pasien (Pannaporn Ketpupong, 2018).

Dalam penelitian (Lingling Zhou, 2020) "Construction of a semi-automatic ICD-10 coding system," peneliti mengusulkan penggunaan ekspresi reguler (regexps) untuk membuat korespondensi antara kode diagnosis dan deskripsi diagnosis dalam pengaturan rawat jalan dan saat masuk dan keluar. Model deskripsi regexp disematkan dalam sistem pengkodean yang sudah ditingkatkan, yang menerima input deskripsi diagnosis dan menghasilkan kode diagnosis yang unik. Kekurangan dari metode ini adalah harus ada

penyesuaian kode terlebih dahulu dan dilakukan validasi atau penyesuaian setiap bertemu dengan kode yang tidak sesuai karena regexs berbaris aturan terstruktur.

Pada kajian literatur diatas didadapatkan berbagai kelebihan dan kekurangan dalam penggunaan *machine learning* dalam sektor kesehatan, sebagaimana penelitian untuk Cyrillic ke bahasa latin pada kasus pemulangan pasien dimana memiliki kelemahan tidak adanya standar penulisan pada surat pemulangan pasien. Lalu klasifikasi diagnosis semi automatis menggunakan regex dimana metode ini memiliki kelemahan tidak bisa dinamis dalam menentukan parameter regex yang digunakan. Dari beberapa metode tersebut metode machine learning menjadi cukup menjajikan dengan akurasi yang cukup baik di rentang 80-90%. Begitu juga pada metode machine learning yang akan dikerjakan oleh penulis adalah dengan menambah kamus data yang ditetapkan oleh Rumah Sakit.

Berdasarkan beberapa kajian literatur yang telah dilakukan, penelitian yang akan diusulkan memiliki posisi dan kemutakhiran yang dapat dijelaskan sebagai berikut:

1. Masukan yang diberikan oleh pengguna berupa kalimat dengan Bahasa Indonesia (bahasa sehari-hari) sehingga mudah dipahami dengan penambahan kamus singkatan yang berlaku pada Rumah Sakit tempat penelitian berlangsung.
2. Penelitian yang diusulkan menggunakan data diagnosis dari dokter Indonesia dengan pasien rawat jalan yang selanjutnya dikarenakan belum ada data diagnosis di Indonesia yang tersedia secara publik, maka akan dilakukan preprocessing terlebih dahulu sebelum masuk sebagai data yang akan diterima oleh machine learning.
3. Penyakit Dalam dipilih sebagai sampel utama mengingat penyakit ini paling banyak terdapat pada pasien di rumah sakit umum di Indonesia.
4. Machine learning sebagai metode yang mutakhir dalam sistem cerdas digunakan untuk keperluan pembelajaran model. Data rekam medik sebagai data primer digunakan sebagai pembelajaran sehingga objektivitas model terjaga.
5. Pengujian model melibatkan pakar (tidak sekedar diuji dari data set), sehingga pertimbangan atas ambiguitas data dapat diminimalisir.
6. Dalam membangun prototipe melibatkan calon pengguna sehingga diharapkan prototipe yang dihasilkan sesuai dengan kebutuhan pengguna

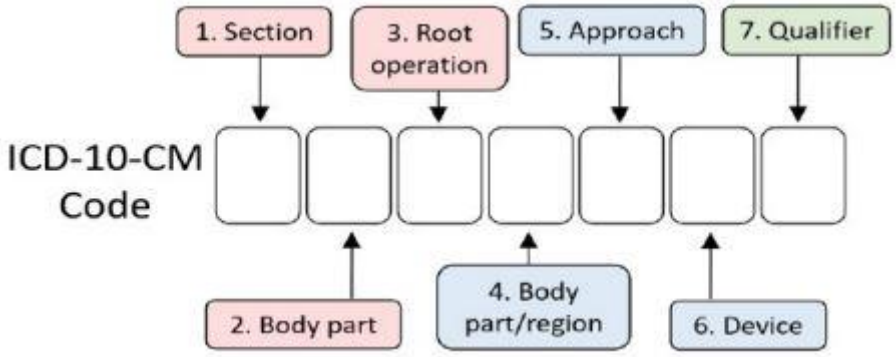
2.2 International Classification of Diseases 10th Revision (ICD 10)

ICD-10 (International Classification of Diseases, 10th Revision) adalah suatu sistem klasifikasi penyakit dan masalah kesehatan lainnya yang digunakan secara internasional untuk mencatat dan melaporkan statistik kesehatan. ICD-10 dikeluarkan oleh World Health

Organization (WHO) dan digunakan oleh banyak negara di seluruh dunia sebagai standar untuk mengkategorikan dan mengkode berbagai jenis penyakit dan kondisi medis. ICD-10 digunakan secara luas dalam dunia klaim asuransi kesehatan. Kode-kode ICD-10 membantu dalam mengidentifikasi dan mengklasifikasikan kondisi kesehatan pasien, yang sangat penting dalam proses klaim asuransi.

ICD-10 menyediakan kode-kode untuk berbagai penyakit, kondisi medis, dan masalah kesehatan lainnya. Dengan menggunakan kode ini, sensus dapat mengidentifikasi dan memetakan distribusi penyakit di antara populasi.Dengan menggunakan ICD-10, pihak yang bertanggung jawab atas sensus dapat memantau kondisi kesehatan populasi dari waktu ke waktu. Hal ini membantu dalam mengidentifikasi perubahan dalam pola penyakit dan mengevaluasi dampak program-program kesehatan masyarakat.Hasil dari sensus yang menggunakan ICD-10 dapat membantu dalam perencanaan sumber daya kesehatan, termasuk distribusi tenaga medis, peralatan medis, dan dukungan infrastruktur lainnya.

Kode ICD-10 dibagi menjadi dua kategori utama, CM dan PCS, CM menunjukkan "Modifikasi Klinis", sedangkan PCS menunjukkan "Sistem Pengkodean Prosedur". ICD10-CM adalah tentang diagnosis penyakit, dan strukturnya diilustrasikan pada Gambar 2. 1.



Gambar 2. 1. Struktur pengkodean pada ICD-10

Pada gambar diatas tersebut, tiga karakter pertama menunjukkan kategori dari diagnosis. Tiga karakter berikutnya sesuai dengan etiologi terkait. Karakter ketujuh menyediakan ekstensi. Dibandingkan dengan ICD-9-CM yang hanya memiliki 3-5 karakter, ICD 10-CM masing-masing memiliki 3-7 karakter. Oleh karena itu, ICD-10-CM yang menjelaskan informasi klinis terperinci dapat meningkat kerumitan dalam menentukan kode anatomi setiap blok dapat dijelaskan dengan tabel 2. 1 berikut.

Tabel 2. 1 Chapter ICD-10 CM

Ch. Blocks	**Title**	**Ch. Blocks**	**Title**
I. A00-B99	Certain infectious and parasitic diseases	XII. L00-L99	Diseases of the skin and subcutaneous tissue
II. C00-D48	Neoplasms	XIII. M00-M99	Diseases of the musculoskeletal system and connective tissue
III. D50-D89	Diseases of the blood and blood-forming organs and certain disorders involving the immune mechanism	XIV. N00-N99	Diseases of the genitourinary system
IV. E00-E90	Endocrine, nutritional and metabolic diseases	XV. O00-O99	Pregnancy, childbirth and the puerperium
V. F00-F99	Mental and behavioral disorders	XVI. P00-P96	Certain conditions originating in the perinatal period
VI. G00-G99	Diseases of the nervous system	XVII. Q00-Q99	Congenital malformations, deformations and chromosomal abnormalities
VII. H00-H59	Diseases of the eye and adnexa	XVIII. R00-R99	Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified
VIII. H60-H95	Diseases of the ear and mastoid process	XIX. S00-T98	Injury, poisoning and certain other consequences of external causes
IX. I00-I99	Diseases of the circulatory system	XX. V01-Y98	External causes of morbidity and mortality

X. J00-J99	Diseases of the respiratory system	XXI. Z00-Z99	Factors influencing health status and contact with health services
XI. K00-K93	Diseases of the digestive system	XXII. U00-U99	Codes for special purposes

Kode ICD-10 menggunakan lebih dari 10.000 kode yang berbeda pada dasar klasifikasi untuk membantu petugas koding dalam pengelompokan penyakit. Sampai saat ini, pengklasifikasikan penyakit masih bergantung pada petugas koding dengan membaca berkas tertulis, seperti diagnosis, keluhan utama, riwayat kesehatan, dan catatan operasi sebagai dasar untuk klasifikasi. Cara ini tentu kurang efektif dan memakan waktu, mempertimbangkan bahwa mengklasifikasikan penyakit dengan kemampuan profesional juga membutuhkan waktu rata-rata 20 menit. Penelitian ini bertujuan untuk membangun sistem pengkodean ICD-10 untuk klasifikasi teks deskriptif diagnosis dokter menjadi kode ICD-10 secara otomatis untuk menghemat waktu dan tenaga di rumah sakit.

Seiring berjalannya waktu, International Classification of Diseases (ICD) mengalami perkembangan dan pembaruan untuk mempertahankan relevansinya dengan kemajuan dalam ilmu kedokteran, penelitian kesehatan, dan kebutuhan sistem informasi kesehatan. Setelah ICD-10 yang diperkenalkan tahun 1994 selanjutnya ada ICD-11, yang diperkenalkan oleh World Health Organization (WHO), adalah revisi terbaru dari klasifikasi penyakit ini. ICD-11 diadopsi pada Majelis Kesehatan Dunia WHO pada tahun 2019. ICD-11 membawa perubahan signifikan dalam konsep, struktur, dan terminologi klasifikasi penyakit.Setiap revisi ICD bertujuan untuk meningkatkan kedalaman dan spesifikasi dalam menggambarkan kondisi kesehatan. Ini memungkinkan sistem kesehatan untuk lebih akurat mencatat dan melaporkan data kesehatan.ICD-11 mengalami peningkatan signifikan dalam kategori kesehatan mental dan menggambarkan dengan lebih baik spektrum gangguan mental.Pembaruan ICD mencerminkan perkembangan teknologi informasi dan integrasi sistem informasi kesehatan. Ini membantu dalam pengumpulan, analisis, dan pertukaran data kesehatan secara lebih efisien.

2.3 Machine Learning

Machine Learning dapat dipergunakan untuk menghasilkan prediksi serta memperbaiki sistem, dengan membuat keputusan yang lebih akurat, berdasarkan informasi

(execution, resources and requirement) yang tersedia. Machine Learning menggunakan teknik berdasarkan catatan keadaan informasi masa lalu untuk kemudian membuat suatu model yang tepat atas keadaan yang sering terjadi, selanjutnya mengenali anomali sistem hingga menghasilkan keputusan serta evaluasi terhadap sistem tersebut. Machine Learning merupakan teknik tata kelola infrastruktur Teknologi Informasi yang memungkinkan proses pengambilan keputusan dibuat dengan pemodelan terhadap sistem yang ada berdasarkan data-data sistem tersebut, sehingga model yang dihasilkan tersebut dapat diperbaharui, maupun membuat model baru lainnya yang disesuaikan dengan kebutuhan spesifik tertentu. Dengan kata lain, bahwa dengan penggunaan Machine Learning dalam tata kelola infrastruktur Teknologi Informasi, kita akan dapat menggali pengetahuan secara langsung atas perilaku sistem yang sedang berjalan.

Machine Learning dimulai dengan mengolah data-data yang ada, bertujuan untuk memperoleh informasi yang saling berhubungan dan untuk menentukan mana yang menjadi atribut, kemudian membuat suatu model yang dapat dipergunakan untuk menjelaskan bagaimana kondisi sistem yang ada, hingga akhirnya dibuatlah suatu keputusan berdasarkan model tersebut. Secara umum, cara kerja Machine Learning adalah dengan mengolah serangkaian data yang disebut sebagai data set, yang berasal dari suatu sistem dengan menentukan nilai-nilai dari sistem tersebut, menentukan mana yang atribut dan mana yang respon, kemudian membuat model berdasarkan nilai-nilai ini, sehingga ketika ada data yang baru, nilai yang diharapkan akan sesuai dengan ekspektasi atas model yang diperoleh. teknik implementasi Machine Learning dibagi dalam beberapa pendekatan, diantaranya adalah supervised learning dan unsupervised learning.

2.4 Clinical Text Mining

Text mining adalah suatu bidang baru yang sedang berkembang yang mencoba untuk mengumpulkan informasi yang memiliki arti dari teks Bahasa alami. Bidang ini mungkin lebih dikenali sebagai proses dalam menganalisis teks untuk mengekstrak informasi yang berguna untuk suatu tujuan tertentu. Dibandingkan dengan jenis data yang tersimpan dalam database, text mining menggunakan data teks yang tidak terstruktur, tidak memiliki bentuk yang jelas dan sulit untuk diuraikan dengan pendekatan algoritma. Namun, dalam budaya modern, teks adalah perantara yang paling umum untuk pertukaran secara formal dari informasi. Bidang text mining biasanya berkaitan dengan teks yang memiliki fungsi untuk komunikasi dari informasi yang faktual atau opini, dan keinginan untuk mencoba mengekstrak informasi dari sebuah teks secara otomatis merupakan hal yang menarik meskipun tingkat keberhasilan yang diperoleh hanyalah sebagian. Text mining umumnya

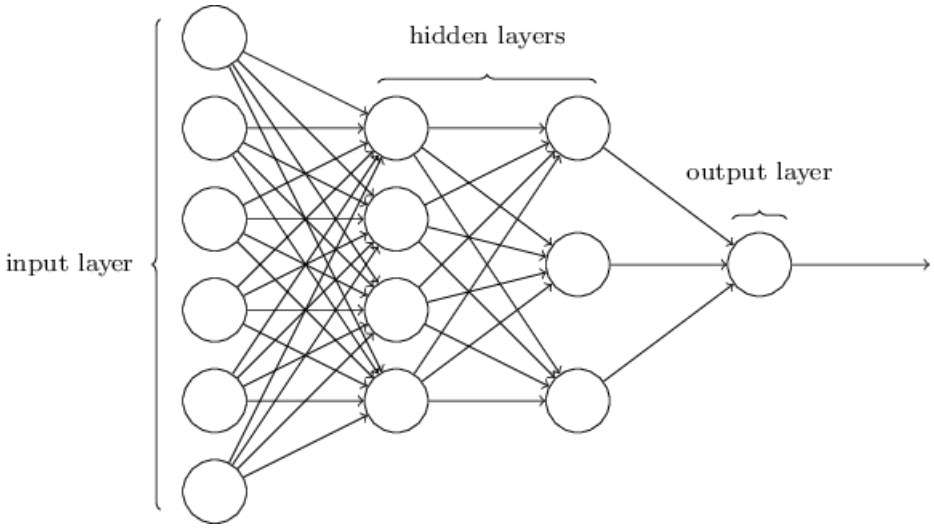
digunakan untuk menunjukkan sistem yang menganalisis sejumlah besar teks bahasa alami dan mendeteksi pola penggunaan leksikal atau bahasa dalam upaya untuk mengekstrak informasi yang mungkin berguna.

Clinical text mining adalah proses mengekstraksi informasi dan pengetahuan yang bermanfaat dari catatan kesehatan elektronik dan dokumen klinis lainnya menggunakan teknik Natural Language Processing (NLP) dan Machine Learning (ML). Natural Language Processing (NLP) merupakan salah satu cabang ilmu AI yang berfokus pada pengolahan bahasa natural. Bahasa natural adalah bahasa yang secara umum digunakan oleh manusia dalam berkomunikasi satu sama lain. Bahasa yang diterima oleh komputer butuh untuk diproses dan dipahami terlebih dahulu supaya maksud dari user bisa dipahami dengan baik oleh komputer. Ada berbagai terapan aplikasi dari NLP yaitu diantaranya adalah Chatbot yakni aplikasi yang membuat user bisa seolah-olah melakukan komunikasi dengan komputer, Translation Tools yaitu menterjemahkan Bahasa satu ke Bahasa yang lain dan aplikasi-aplikasi lain yang memungkinkan komputer mampu memahami instruksi bahasa yang diinputkan oleh user. Dalam bidang kesehatan pencatatan diagnosis pasien dicatat dalam rekam medis dalam unstructured text, berupa deskripsi tekstual yang sangat variatif. Salah satu pendekatan teknologi yang dapat diterapkan untuk mengekstrak informasi didalamnya adalah penggunaan Natural Language Processing.

2.5 Jaringan Syaraf Tiruan

Neural Network akan digunakan sebagai metode klasifikasi, adalah kumpulan neuron yang diatur dalam urutan beberapa lapisan, di mana neuron menerima sebagai input aktivasi neuron dari lapisan sebelumnya, dan melakukan perhitungan matematis sederhana (misalnya jumlah tertimbang dari input diikuti oleh aktivasi nonlinier). Neuron jaringan bersama-sama menerapkan pemetaan nonlinier yang kompleks dari input ke output. Pemetaan antara input dan output dipelajari dari data dengan mengadaptasi bobot setiap neuron menggunakan teknik yang disebut error backpropagation. Neural Network adalah arsitektur jaringan saraf multilayer feedforward atau jaringan saraf tiruan (JST) dengan banyak lapisan antara lapisan input dan output sebagaimana gambar 2. 2. Peneliti akan menggunakan set data latih berlabel untuk melakukan klasifikasi pada data yang serupa dan tidak berlabel.

Neural network ditentukan oleh tiga hal, yaitu pola hubungan antar-Neuron yang disebut jaringan. Metode untuk menentukan bobot penghubung yang disebut metode training / learning / algoritma, dan Fungsi aktivasi atau fungsi transfer.



Gambar 2. 2 Arsitektur Jaringan Syaraf Tiruan

Neuron pada lapisan input akan diisi dengan diagnosis hasil dari proses vectorisasi yang sudah dalam satuan angka akan diproses menggunakan fungsi aktivasi Sigmoid. Selanjutnya data hasil olahan dari layer tersembunyi dihubungkan oleh bobot-bobot tersembunyi menuju neuron pada layer output, sedangkan lapisan output adalah hasil dari pembobotan setiap layer yang menghasilkan kelas kode penyakit berdasarkan ICD 10 yang paling mendekati. Pada layer *output*, akan digunakan aktivasi *softmax*, yakni fungsi aktivasi yang umumnya digunakan pada lapisan output dari neural networks) untuk tugas klasifikasi multikelas. Fungsi softmax mengubah output numerik dari suatu layer menjadi distribusi probabilitas yang dapat diinterpretasikan.

Fungsi softmax memastikan bahwa setiap output mendapatkan nilai yang bersifat probabilitas dan tidak saling tumpang tindih. Pada lapisan output, neuron yang memiliki nilai softmax tertinggi akan mewakili kelas yang diprediksi oleh model.

Kemudian untuk fungsi *loss* akan digunakan fungsi *categorical_crossentropy* sebagaimana yang biasa digunakan untuk mengukur seberapa baik model klasifikasi memetakan input ke distribusi probabilitas kelas yang benar pada klasifikasi multikelas. Setiap sampel data dapat termasuk dalam salah satu dari beberapa kelas yang tidak saling tumpang tindih, categorical crossentropy memberikan ukuran seberapa dekat distribusi probabilitas prediksi model dengan distribusi probabilitas yang seharusnya (ground truth).

y_true : distribusi probabilitas kelas seharusnya (*one-hot encoded*).

y_pred : distribusi probabilitas kelas prediksi oleh model.

Maka, categorical crossentropy dihitung dengan rumus matematika berikut:

$$-\sum_i y_{true,i} \cdot \log(y_{pred,i}) \quad (1)$$

Rumus ini menjumlahkan produk dari nilai sebenarnya (y_true, i) dan logaritma nilai prediksi (y_pred,i) untuk setiap kelas kemudian dalam prosesnya akan dilakuakan pemantauan sejauh mana model yang dilatih memiliki nilai akurasi yang baik dan nilai loss seminimal mungkin dengan menampilkan proses pelatihan pada grafik *learning curve*, ini adalah representasi grafis dari kinerja model machine learning seiring waktu atau seiring dengan meningkatnya jumlah sampel latihan (data training) yang digunakan. Kurva pembelajaran memvisualisasikan bagaimana model berkembang dalam hal akurasi atau nilai kerugian saat dipelajari dari dataset.

Jika training loss rendah, validation loss juga rendah, dan kedua akurasi tinggi, ini menunjukkan model yang baik dengan kemampuan baik dalam memahami dan menggeneralisasi pola dalam data atau kondisi model adalah *good fitted*.

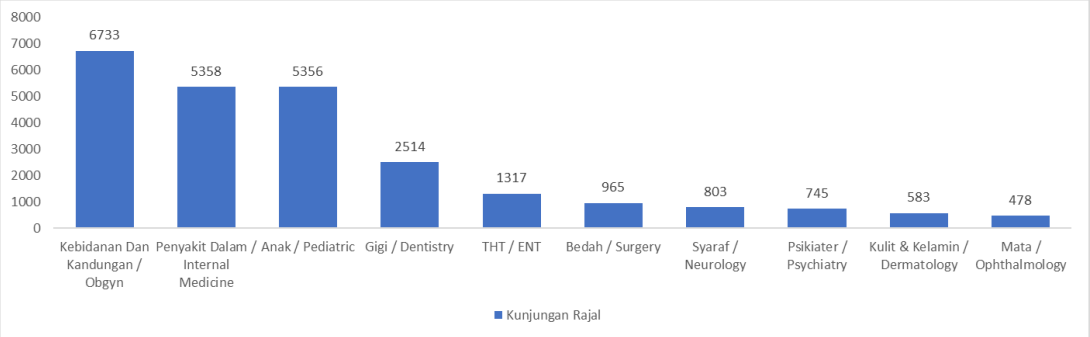
BAB 3
Metodologi

3.1 Pengumpulan Data Penelitian

Data penelitian yang digunakan adalah sejumlah data penyakit dari disiplin *Internist* atau Penyakit Dalam pada sebuah Rumah Sakit Swasta. Data berbentuk kumpulan penyakit dengan beberapa informasi kolom yang sudah disederhanakan oleh penulis untuk menjaga *privacy* pasien. Akhir data yang dipilih adalah data yang berisi kolom diagnosis dan kolom kode ICD10 yang sudah dikoding oleh petugas koding dari Unit Rekam Medis sebagai kelas data.

Dari keseluruhan data yang diambil adalah pasien dengan usia rata-rata 50 tahun yang melakukan kunjungan pemeriksaan di polikiklik Penyakit Dalam pada rentang waktu kunjungan bulan Januari hingga September. Sebagai variabel dependen adalah diagnosis pasien yang ditulis dokter dimana kasus tersebut diambil dari kelompok kasus baru dan kelompok kasus lama. Lokasi penelitian yang dipilih adalah Rumah Sakit Swasta yang memiliki praktik dokter spesialis penyakit dalam, dan memiliki petugas koding di Unit Rekam Medis.

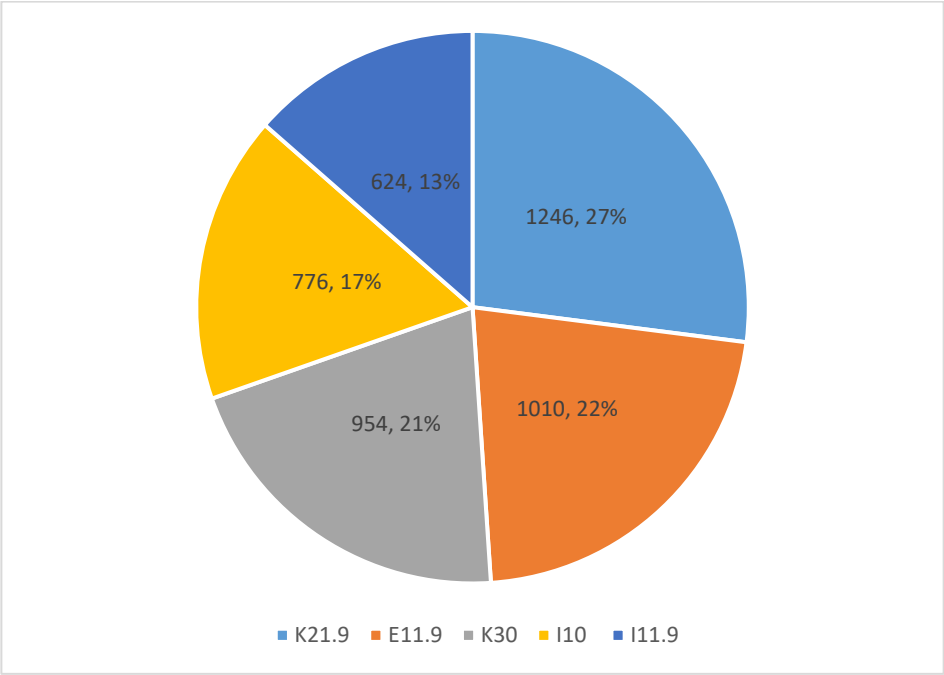
Kunjungan rawat jalan selama bulan Januari hingga Oktober 2020 digambarkan sebagaimana gambar 3. 1 pada grafik tersebut diambil kasus pada spesialis penyakit dalam dikarenakan data diagnosis lebih bervariasi dibandingkan dengan spesialis kebidandan dan kandungan dimana sebagian kasus adalah kasus kontrol ibu hamil yang biasanya 1 ibu hamil akan melakukan pemeriksaan berkala terhadap kandungannya 1 bulan 1 kali.



Gambar 3. 1 grafik kunjungan 10 besar spesialis

Jumlah total data yang diperoleh adalah sejumlah 4.610 diagnosis pasien, data tersebut diambil dari rentang waktu kunjungan pasien pada bulan Januari sampai dengan Oktober 2020. Kumpulan data tersebut diambil 5 kelas kode ICD10 dengan sebaran E11.9 Type 2 diabetes mellitus without complications, I10 Essential (primary) hypertension, I11.9

Hypertensive heart disease without (congestive) heart failure, K21.9 Gastro-oesophageal reflux disease without oesophagitis, K30 Functional dyspepsia dijelaskan oleh gambar 3. 2.

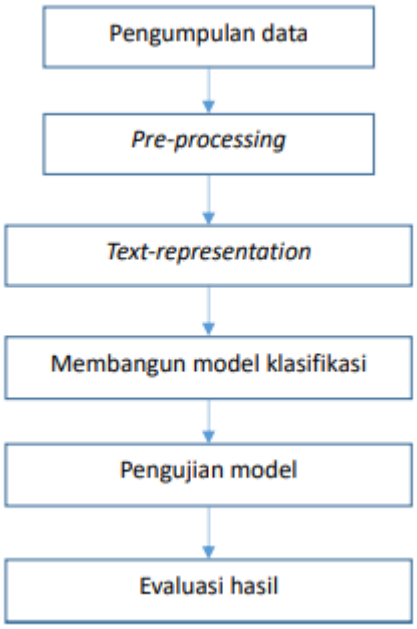


Gambar 3. 2 Sebaran kelas data diagnosa

Selain data diagnosis, juga dihimpun data berupa kamus singkatan untuk membantu menerjemahkan beberapa singkatan didalam diagnosa supaya lebih mudah dipahami maknanya.

3.2 Tahapan Penelitian

Penelitian yang akan digunakan adalah penelitian kuantitatif dimana tujuannya adalah membuat model text mining untuk klasifikasi kode penyakit dari data diagnosis yang tidak terstruktur ke dalam kode ICD10, berikut adalah gambar 3. 3 yakni tahapan yang akan dilakukan oleh peneliti



Gambar 3. 3 tahapan penelitian

3.2.1 Pengumpulan Data

Penetilitan ini menggunakan data yang diambil secara langsung pada objek penelitian yakni diagnosa pasien. Pengambilan data dilakukan dengan acak pada sejumlah pasien yang melakukan kunjungan pemeriksaan pada rentang waktu bulan Januari hingga Oktober 2020. Kumpulan pasien akan dilakukan penapisan data hingga diperoleh diagnosa yang paling sering muncul pada Rumah Sakit tersebut.

Diperoleh total data yang sejumlah 4.610 diagnosa pasien yang dikelompokkan ke dalam 5 kelas kode ICD10 dengan sebaran E11.9 Type 2 diabetes mellitus without complications, I10 Essential (primary) hypertension, I11.9 Hypertensive heart disease without (congestive) heart failure, K21.9 Gastro-oesophageal reflux disease without oesophagitis, K30 Functional dyspepsia.

Sebelum data tersebut akan masuk pada *pre-proccesing* ke 4.610 data akan dibagi beberapa bagian, yakni untuk data training, data testing dan data validasi. Data rentang Januari hingga September akan digunakan untuk proses training dan testing dengan komposisi pembagian adalah 70:30. Sedangkan untuk proses validasi akan menggunakan data pada bulan Oktober sebagai data di luar data training dan testing gambaran data dimuat sebagaimana tabel 3. 1.

Tabel 3. 1 data diagnosis dan kelas

Diagnosis	icd
IKTERIK EC PREHEPATAL (INFEKSI). DYSPEPSIA ORGANIK	K30
HT. OCASIONAL VES	I10
HT STG II DGN PAD A.TIBIALIS ANT-POST BILATERAL. DIZZINESS EC SPOND CERVICALIS. OBS ABD PAIN EC FATTY LIVER. DISPEPSIA TIPE DISMOTILITAS. DISLIPIDEMIA	I10
DM TIPE 2. SELULITIS PEDIS DX	E11.9
GERD, DELAYED GASTRIC EMPTING, GATRITIS ANTHRAL, GASTROENTEROPATHY, PYLORUS GAPPING, DUODENITIS NON SPESIFIK. PLANTAR FASCIITIS PEDIS S. DISLIPIDEMIA	K21.9
HTSTG II. VERTIGO DGN ASTHENIA	I10
GERD MIXED. SUSP.HEMORHOID INTERNA GR II. POST LAPARASKOPI CHOLE	K21.9
SELULITIS PEDIS D. HHD EC HT STG II. ANGINA PECTORIS STABIL. OA GENU BILATERAL. SPONDILOSIS LUMBALIS. DISPESIA - GERD	K30
HHD EC HT STG II. NEFROPATHY HT. DISLIPIDEMIA. TENSION HEADACHE CUM VERTIGO PERIFER EC KLINIS PARACERVICAL SPASM	I11.9
HT STG II. DISLIPIDEMIA. OBS POLYARTHRALGIA EC SPOND CERVICALIS. OA MANUS D	I10
HHD EC HT STG II . OBS GOUT ARTHRITITS. NEFROPATHY URAT	I11.9
…….	……

3.2.2 Pre-processing Data

Pre-processing adalah proses pembersihan data dari karakter maupun text yang akan mempengaruhi akurasi model nantinya, tahapan ini termasuk proses menyamakan besar kecil huruf untuk konsistensi penulisan, menghapus karakter yang tidak diperlukan, *expand contraction* dengan kamus singkatan, tokenisasi dan menghapus *stop words*. Pada *pre-processing* data diagnosis akan dilakukan normalisasi, identifikasi dan penanganan terhadap data yang hilang, duplikat, atau tidak relevan. Proses ini dapat mencakup pengisian nilai yang hilang, penghapusan duplikat, dan normalisasi format data.

Penekanan pada tahapan *pre-processing* adalah *contraction* atau penerjemahan kata yang berupa singkatan dengan mengacu pada kamus singkatan yang sudah ditetapkan oleh

Rumah Sakit. Tujuan dari *contraction* adalah beberapa penulisan dokter yang menggunakan istilah atau singkatan medis dapat dikenali secara utuh dengan menampilkan seluruh kepanjangan dari singkatan tersebut. Kamus singkatan sebagaimana tabel 3. 2 yang diperoleh dari Rumah Sakit dimana data diagnosis diambil dengan harapan dapat memberikan representasi lebih spesifik terhadap singkatan yang dipergunakan oleh dokter pada Rumah Sakit tersebut.

Tabel 3. 2 kamus singkatan

singkatan	**kepanjangan**
ec	et causa
dgn	Dengan
°C	derajat celcius
¼ p	¼ porsi
½ PS	½ porsi
2 j PP	2 jam post prandial
2D RT	2 dimensional radioterapy
…….	…….
URI	upper respiratory infection
Urin Cath/	urine catheter
URS	uretro renoscopi
URTI	upper respiratory tract infection
USG	Ultrasonography
Ur	Ureum

Pre-processing selanjutnya setelah proses *contraction* adalah pembersihan diagnosa dengan beberapa tahapan melalui metode *lower case, remove punctuation, remove stop words dan tokenization.*

3.2.3 Text Representation

Pada tahap ini, fokus utama penelitian adalah mengembangkan representasi yang efektif untuk teks yang akan digunakan sebagai input dalam proses pemodelan. Representasi teks merupakan kunci penting dalam analisis teks, terutama dalam konteks pengolahan bahasa alami dan machine learning.

Metode utama yang digunakan dalam representasi teks melibatkan konversi teks menjadi bentuk numerik agar dapat dimanfaatkan oleh algoritma *machine learning*. Dalam

konteks klasifikasi dan analisis diagnosis penyakit, pendekatan vectorisasi dengan menggunakan *Term Frequency-Inverse Document Frequency* (TF-IDF) menjadi pilhan untuk mewakili teks diagnosis dalam bentuk numerik.

Pendekatan ini memberikan bobot pada setiap kata berdasarkan frekuensinya dalam suatu dokumen dan seberapa umum kata tersebut di seluruh dokumen. Kata-kata yang muncul sering dalam dokumen tetapi jarang muncul secara keseluruhan mendapatkan bobot yang tinggi.

Term Frequency (TF) merupakan frekuensi kemunculan term i pada dokumen j dibagi dengan total term pada dokumen j. Formula untuk menghitung TF adalah sebagai berikut:

$$tf_{ij} = \frac{f\ d(i)}{\max_{j_{\epsilon d}} f\quad_d(j)} \qquad (2)$$

dengan $f_{t,d}$ adalah pencacahan mentah istilah dalam dokumen, yaitu jumlah kemunculan istilah t dalam dokumen d. Semakin sering suatu istilah muncul, semakin besar nilai tf-nya.

Inverse Document Frequency (IDF) adalah ukuran informasi yang diberikan oleh istilah t, yaitu seberapa sering atau jarang sebuah istilah muncul dalam seluruh dokumen. Semakin jarang suatu istilah di antara dokumen, semakin besar nilai idf-nya. Nilainya adalah logaritma dari kebalikan dari jumlah dokumen yang memiliki istilah t yang dibagi jumlah seluruh dokumen (N).

$$idf(t, D) = \log \frac{N}{\{d \in D : t\ \in d\}} \qquad (3)$$

Dalam hal ini, istilah yang jarang muncul pada koleksi keseluruhan istilah dinilai lebih berharga. Nilai kepentingan tiap istilah diasumsikan berbanding terbalik dengan jumlah teks yang mengandung istilah tersebut.

3.2.4 Membangun Model Klasifikasi

Tahapan inti dari setiap proses adalah membangun model klasifikasi yang dapat memberikan prediksi akurat berdasarkan fitur-fitur yang telah dipersiapkan sebelumnya. Proses ini melibatkan algoritma *Neural Network* sebagai pengolah model, data training dan testing akan menggunakan data kunjungan pemeriksaan pasien bulan Januari hingga September.

Data tersebut akan dibagi dengan komposisi 70:30 dan dilatih untuk mendapatkan akurasi yang tinggi dan nilai loss yang rendah, nilai *loss* adalah ukuran seberapa baik atau

seberapa buruk model memprediksi output yang benar untuk suatu masukan. Beberapa skenario pelatihan yang akan dijalankan adalah melakukan *tuning hyperparameter* meliputi penyesuaian jumlah layer, input size, batch size, epoch, maupun *dropout* untuk menangai model *overfitted.*

Dikarenakan klasifikasi yang akan dijalankan adalah klasifikasi multikelas, maka pada *layer output* menggunakan aktivasi *softmax*.

3.2.5 Pengujian Model

Tahapan pengujian model menggunakan Confusion Matrix dimana metode ini dapat memberikan gambaran rinci tentang seberapa baik model mampu mengklasifikasikan setiap kelas dan membantu mengidentifikasi jenis kesalahan yang dibuat oleh model. Beberapa parameter yang akan diukur adalah *accuracy, precission, recall,* dan *f-1 score*

Akurasi menyediakan indikasi umum seberapa baik model dapat mengklasifikasikan diagnosa dengan benar. Presisi (P) mengukur seberapa akurat model dalam memprediksi positif. Recall (R) mengukur seberapa baik model dapat menangkap semua kasus positif yang sebenarnya dan F1-score (F) adalah rata-rata harmonis antara presisi dan recall.

BAB 4
Hasil dan Pembahasan

Dari total 4610 data dijadikan 2 kolom yakni diagnosis dan kelas dimana keduanya memiliki tipe data Object yang ditunjukkan pada gambar 4. 1. Data tesebut adalah data murni dari pengambilan diagnosa pasien yang ditulis oleh dokter penyakit dalam dan belum dilakukan pengolahan data lebih lanjut. Dari hasil fungsi dataframe info didapatkan sebagai berikut, sedangkan gambaran hasil *load data* adalah sebagaimana ditunjukkan gambar 4. 2

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4610 entries, 0 to 4609
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   diagnosis  4610 non-null   object
 1   icd        4610 non-null   object
dtypes: object(2)
memory usage: 72.2+ KB
```

Gambar 4. 1 *type dataframe*

	diagnosis	icd
4333	KOLIK ABDOMEN PERBAIKAN. HT URGENSI. HHD. GASTRITIS ANTRUM. DISLIPIDEMIA	I10
2907	HT STG II. GASTROENTEROPATHY NSAID. IDIOSINKRASI LANSOPRAZOLE. STEROID WITHDRAWL EFFECT	I11.9
2971	HHD EC HT STG II DGN NEFRPATHY HT. GERD. OA GENU. PASCA TF	I11.9
426	OBS CHEST PAIN EC GERD. GASTRITIS ANTHRAL, PYLORUS GAPPING, DUODENITIS NON SPESIFIK. DISLIPIDEMIA	K21.9
621	GERD DGN GASTROENTEROPATHY CUM PANGASTRODUODENITIS EROSIVA NON SPESIFIK DGN H.PYLORI(+). OBS HIPERGLIKEMIA EC STRESS HIPERGLIKEMIA DD DM	K21.9
69	GERD. OBS.RECURRENT EPISTAKSIS EC SUSP.HIPERTROFI CHONCA. OBS. VERTIGO PERIFER	K21.9
2787	FEBRIS HARI KE 2 EC VIRAL DD BACTERIAL	K30
2599	OBS UNINVESTIGATED DYSPEPSIA WITH ALARM SYMPTOM DGN EPISODE KOLIK. OBS CHANGE BOWEL HABITS	K30
2557	UNINVESTIGATED DYSPEPSIA WITH ALARM SYMPTOM	K30
630	GERD CUM GASTRITS ANTHRAL EROSIVA	K21.9

Gambar 4. 2 muat data diagnosa dan kelas

4.1 Persiapan Environment

Dalam hal menyiapkan semua penunjang penelitian baik *hardware* dan *software* agar dalam proses hingga hasilnya dapat terkontrol, memastikan keakuratan data, dan mendukung pengembangan metodologi yang diterapkan dalam penelitian ini, maka pada penelitian ini penulis menggunakan beberapa peralatan yakni komputer *desktop* yang memiliki spesifikasi processor Intel(R) Core(TM) i5-7400 CPU @ 3.00GHz, Random Accessed Memory (RAM) 8 GB, dengan Operating System Windows 10 diatas Solid State Disk (SSD) 240GB untuk memperlancar saat proses komputasi.

Untuk *software* pendukung digunakan beberapa library pengolahan data dan *machine learning* seperti pandas, numpy, re, matplotlib, seaborn, nltk, string, sklearn, gensim untuk vektorisasi word2vec, serta tensorflow dan keras. Untuk melakukan semua proses penelitian, peneliti menggunakan bahasa pemrograman Python pada aplikasi web open-source Jupyter Notebook sebagai antar muka yang sudah dipasang pada komputer desktop.

4.2 Pre-processing Data

Beberapa tahapan *pre-processing* yang akan dilakukan adalah sebagaimana yang sudah dipaparkan dalam bab Metodologi sebelumnya.

4.2.1 Fungsi lowercase

Mengubah seluruh teks menjadi huruf kecil membantu mengatasi variasi casing yang mungkin ada dalam teks. Misalnya, kata yang ditulis dengan huruf besar di awal kalimat atau huruf besar secara acak akan dianggap sama setelah proses normalisasi ini. Ini membantu model mengenali kata-kata yang memiliki arti yang sama meskipun penulisan casing yang berbeda.

Sebagaimana kata “Gerd”, “GERD” dan “gerd” akan dibaca memiliki arti yang sama, juga untuk memastikan konsistensi dalam pemrosesan teks. Hal ini membantu mencegah model atau algoritma untuk menganggap dua kata yang sama tetapi ditulis dengan casing yang berbeda sebagai entitas yang berbeda.

Untuk proses *lowercase* dibuat fungsi berikut, dimana fungsinya nanti akan memuat data diagnose penyakit kemudian memilik kolom diagnose dan mengubah semua isi dalam kolom tersebut menjadi huruf kecil dengan koding pada *code snipped* 4. 1:

Code Snippet 4. 1 lowercase_text

```
def lowercase_text(text):
    return text.lower()
df['tidy_text']=df['diagnosis'].apply(lowercase_text)
```

Hasil dari fungsi lowercase dengan menampilkan kolom asli diagnosa dan hasil pengolahan pada kolom tidy_text sebagaimana gambar 4. 3

	diagnosis	icd	tidy_text
3411	OBS FEBRIS. HT STAGE II	I10	obs febris. ht stage ii
747	GERD. OBS BACK PAIN EC SPOND LUMBALIS. PPOK	K21.9	gerd. obs back pain ec spond lumbalis. ppok
1425	GERIATRI DNGAN LOW INTAKE . DM TIPE 2	E11.9	geriatri dngan low intake . dm tipe 2
1526	DM2OBESE -- NEUROPATI DIABETIKA. HHD EC HT STG II. DISPEPSIA	E11.9	dm2obese -- neuropati diabetika. hhd ec ht stg ii. dispepsia
1288	DM 2 OBESE __ NEUROPATI DIABETIKA. HHD EC HT STG II. KISTA GINJAL S	E11.9	dm 2 obese __ neuropati diabetika. hhd ec ht stg ii. kista ginjal s
2349	SUSP CHOLEYCSTITIS. DISPEPSIA TIPE ULCUS	K30	susp choleycstitis. dispepsia tipe ulcus
2132	SINUS ARITMIA. GDPT EC DM DD PREDIABETES. BRONKIITIS AKUT	E11.9	sinus aritmia. gdpt ec dm dd prediabetes. bronkiitis akut
2573	DISPEPSIA EC GASTROENTEROPATHY NSAIDS. SPOND CERVICALIS DGN PARACERVICAL SPASM CUM DIZZINESS. PRONE HT STG I	K30	dispepsia ec gastroenteropathy nsaids. spond cervicalis dgn paracervical spasm cum dizziness. prone ht stg i
3939	DYSPEPSIA MIX TYPE. CHOLECYSTITIS KRONIS. FATTY LIVER. DYSLIPIDEMIA	K30	dyspepsia mix type. cholecystitis kronis. fatty liver. dyslipidemia
56	HIPERESPONSIVE BRONKHUS DGN INF SKUNDER. GERD. HT STG II	K21.9	hiperesponsive bronkhus dgn inf skunder. gerd. ht stg ii

Gambar 4. 3 hasil *lowecase*

4.2.2 Remove punctuation

Hasil dari proses konversi menjadi huruf kecil akan disimpan ke dalam dataframe kolom tidy_text. Hasil konversi ini yang akan dilanjutkan pengolahannya ke dalam tahap *remove punctuation* yakni menghilangkan tanda baca dan angka yang dianggap tidak memiliki makna atau tidak mempengaruhi makna dari kalimat, dalam hal ini diagnosis yang dikerjakan dengan *code snippet* 4. 2.

Tanda baca seperti koma, titik, tanda tanya, dll. seringkali tidak memberikan informasi tambahan yang relevan untuk tugas klasifikasi atau analisis teks, dan menghapusnya dapat membantu menghilangkan noise dari teks. Namun pada diagnosa pasien tanda baca juga merupakan informasi yang tidak bisa begitu saja dihilangkan misalkan “suhu tubuh pasien adalah 36° c” jika kita pukul rata maka hasil akhir dari kalimat tersebut hanya “suhu tubuh pasien adalah c” hal ini tentu akan menghilangkan Informasi yang berharga, oleh karena itu penulis memberikan batasan tertentu agar makna dari diagnosa tidak menjadi bias dengan tetap mempertahankan tanda baca dan karakter tertentu.

Code Snippet 4. 2 remove punctuations

```
def remove_healthcare_punctuations(text):
    allowed_punctuations = "[]{}:,;/-"
    allowed_punctuations =""
    allowed_characters = string.ascii_letters + allowed_punctuations + " "

    # Menggunakan string.punctuation untuk mendapatkan semua karakter tanda baca
    all_punctuations = string.punctuation

    # Menghapus semua tanda baca kecuali yang diizinkan
    translator = str.maketrans('', '', all_punctuations.replace(allowed_punctuations, ''))

    # Menghapus tanda baca dari teks menggunakan translate()
    text_without_punctuations = ''.join(char if char in allowed_characters else '' for char in text)

    return text_without_punctuations
```

Hasilnya tersimpan pada kolom remove_punc pada gambar 4. 4

	diagnosis	icd	tidy_text	remove_punc
76	GERD. TFK. DISLIPIDEMIA	K21.9	gerd tfk dislipidemia	gerd tfk dislipidemia
2863	HHD DGN RIW VES. GERD. IBS TIPE KONSIPASI	I11.9	hhd dgn riw ves gerd ibs tipe konsipasi	hhd dgn riw ves gerd ibs tipe konsipasi
497	GERD MIXED. PARACERVICAL SPASM. TYPHOID FEVER. ALERGI LEVOFLOXACIN	K21.9	gerd mixed paracervical spasm typhoid fever alergi levofloxacin	gerd mixed paracervical spasm typhoid fever alergi levofloxacin
2342	DISPEPSIA. PRURITUS NON SP. G2P1A0 U.K 16MGG. POST CHOLELITHELTOMI 02012020	K30	dispepsia pruritus non sp gpa uk mgg post cholelitheltomi	dispepsia pruritus non sp gpa uk mgg post cholelitheltomi
2107	DM2 OBESE -- NEUROPATI DIABETIKA. HT STG II. HNP VL. PRURITUS DIABETES. PASCA C19 (B34.2)	E11.9	dm obese neuropati diabetika ht stg ii hnp vl pruritus diabetes pasca c b	dm obese neuropati diabetika ht stg ii hnp vl pruritus diabetes pasca c b
2496	DISPESIA TIPE ULCUS CUM DISMOTILITAS. CHOLEYCYSTITIS EC CHOLELITHIASIS. TRIGGER FINGER DIG II-III MANUS S. KISTA GINJAL BILATERAL. PROTRUSIO VU. VERTIGO. SUSP POST POWER SYNDROME	K30	dispesia tipe ulcus cum dismotilitas choleycystitis ec cholelithiasis trigger finger dig iiiii manus s kista ginjal bilateral protrusio vu vertigo susp post power syndrome	dispesia tipe ulcus cum dismotilitas choleycystitis ec cholelithiasis trigger finger dig iiiii manus s kista ginjal bilateral protrusio vu vertigo susp post power syndrome
2647	DISPEPSIA TIPE ULCUS CUM DISMOTILITAS	K30	dispepsia tipe ulcus cum dismotilitas	dispepsia tipe ulcus cum dismotilitas
951	GERD (ESOPHAGITIS LA GRADE B), GASTRITIS ANTHRAL EROSIVA, PSEUDODIVERTICLE ANTHRUM GASTER, GASTROENTEROPATHY NON SPESIFIK, DUODENITIS NON SPESIFIK. PASCA GASTROSKOPI BIOPSI DGN SEDASI. OBS DIZZIN...	K21.9	gerd esophagitis la grade b gastritis anthral erosiva pseudodiverticle anthrum gaster gastroenteropathy non spesifik duodenitis non spesifik pasca gastroskopi biopsi dgn sedasi obs dizzinees ec s...	gerd esophagitis la grade b gastritis anthral erosiva pseudodiverticle anthrum gaster gastroenteropathy non spesifik duodenitis non spesifik pasca gastroskopi biopsi dgn sedasi obs dizzinees ec s...
4109	ABDOMINAL PAIN. DISPEPSIA	K30	abdominal pain dispepsia	abdominal pain dispepsia
4222	HT STG II. OBS HIPERGLIKEMIA. OBS NAUSEA EC SUSP GERD	I10	ht stg ii obs hiperglikemia obs nausea ec susp gerd	ht stg ii obs hiperglikemia obs nausea ec susp gerd

Gambar 4. 4 hasil *remove punctuation*

4.2.3 Expand contraction

Proses berikutnya adalah menggunakan kamus singkatan yang sudah dihimpun dari Rumah Sakit untuk dapat diterapkan ke dalam data diagnosa. Sebagaimana pada data diagnose, dokter menulis “gerd” kemudian fungsi contraction akan mencari kepanjangan dari gerd pada kamus singkatan, sehingga hasil akhir setiap ditemukan kata “gerd” akan diubah sesuai kamus singkatan menjadi “gastroesophageal reflux disease”.

Hal ini diharapkan model dapat lebih mengenali data diagnose secara lengkap. Batasan dari proses ini adalah jika kamus singkatan memiliki 2 atau lebih kata yang sama dengan struktur huruf yang berbeda, maka akan diambil baris yang paling akhir. Kasus tersebut terjadi karena penulisan diagnosa tidak menggunakan kaidah penulisan yang sesuai dengan kamus singkatan yang sudah ditetapkan. Contoh dari kasus ini adalah penulisan “Ab” dan “AB” akan memiliki makna yang berbeda, dimana “Ab” menurut kamus singkatan berarti “Abortus” sedang “AB” adalah “Antibiotik” koding untuk *expand contraction* ditunjukkan dengan *code snippet* 4. 3.

Code Snippet 4. 3 expand contraction

```
mydic=mydic[[0,1]]

mydic=dict(zip(mydic[0], mydic[1]))

def handle_abbreviations(text, mydic):
    words = text.split()
    expanded_words = []

    for word in words:
```

```
        # Cek apakah kata merupakan singkatan yang perlu diuraikan
        expanded_word = mydic.get(word, word)
        expanded_words.append(expanded_word)

    expanded_text = ' '.join(expanded_words)
    return expanded_text

df['tidy_text']=df['tidy_text'].apply(lambda x: handle_abbreviations(x, mydic))
```

Hasil dari expand contraction sebagaimana dilihat kata “ht stg…” diuraikan sesuai kamus singkatan menjadi “hematokrit stage…”, kata “dm2 o hhd ec ht…” juga diuraikan sesuai kamus singkatan menjadi “diabetes militus tipe 2 objective hypersensitive hearth desease et causa…” sebagaimana gambar 4. 5.

	diagnosis	icd	tidy_text
4233	HT STG II DGN CEPHALGIA. DISLIPIDEMIA. RIW HIPERGLIKEMIA	I10	hematrokit stage ii dengan cephalgia dislipidemia riwayat hiperglikemia
1633	SINDROM HIPERGLIKEMIA AKUT (KAD CUM HHS/HYPERGLICEMIC HYPEROSMOLAR STATE) PERBAIKAN. DM2 NO -- NEUROPATI DIABETIKA. HHD. RIW VASCULITIS	E11.9	sindrom hiperglikemia akut kad cum hhshyperglicemic hyperosmolar state perbaikan diabetes mellitus tipe 2 nomor neuropati diabetika hypertensive heart disease riwayat vasculitis
195	GERD. SINDROM KELELAHAN KRONIK. SUSP FATTY LIVER	K21.9	gastroesophageal reflux disease sindrom kelelahan kronik suspect fatty liver
715	GERD. HHD DEKOMP EC HT STG II. DM 2 OBESE	K21.9	gastroesophageal reflux disease hypertensive heart disease dekomp et causa hematrokit stage ii diabetes mellitus 2 obese
4224	HIPERESPONSIVE BRONKHUS. HT STG II. DISLIPIDEMIA. CHOLELITHIASIS. NEFROLITHIASIS BILATERAL. BPH. OA GENU S. OBS BACK PAIN ECSPOND LUMBALIS CUM PARALUMBAL SPASM	I10	hiperesponsive bronkhus hematrokit stage ii dislipidemia cholelithiasis nefrolithiasis bilateral benigna prostat hypertropi osteoarhritis genu subjective obs back pain ecspond lumbalis cum paralum...
3293	SINDROME METABOLIK. HHD. DISLIPIDEMIA. GDPT (PREDIABETES). COLELITIASIS (0,5CM)	I11.9	sindrome metabolik hypertensive heart disease dislipidemia gdpt prediabetes colelitiasis 05cm
3958	DYSPEPSIA DISMOTILITI TYPE	K30	dyspepsia dismotiliti type
1893	DM2 O. HHD EC HT. VERTTIGO SENTRAL CUM PERIFER. KISTA SINUS ETMOID POSTERIOR DEKSTRA DGN SERUMEN PROOP	E11.9	diabetes mellitus tipe 2 objective hypertensive heart disease et causa hematrokit verttigo sentral cum perifer kista sinus etmoid posterior dekstra dengan serumen proop
159	GERD. VERIGO PERIFER	K21.9	gastroesophageal reflux disease verigo perifer
876	OBS NAUSEA VOMITUS INTRACTABLE EC DLM PELACAKAN. GERD, PANGASTRITIS DUODENITIS EROSIVA, MULTIPLE ULCER HEALING, GASTROENTEROPATHY EROSIVA. ANOREXIA PD GERIATRI. DM2 O -- NEUROPATI DIABETIKA. SUSP ...	K21.9	obs nausea vomitus intractable et causa dlm pelacakan gastroesophageal reflux disease pangastritis duodenitis erosiva multiple ulcer healing gastroenteropathy erosiva anorexia pd geriatri diabetes...

Gambar 4. 5 hasil *expand contraction*

4.2.4 Tokenization

Tokenisasi dalam konteks teks klasifikasi merujuk pada proses pemecahan teks menjadi unit-unit kecil yang disebut "token." Token bisa berupa kata, frasa, atau karakter, tergantung pada tingkat granularitas yang diinginkan. Tujuan tokenisasi adalah untuk mempersiapkan teks agar dapat diolah lebih lanjut oleh algoritma klasifikasi atau analisis teks yang dikerjakan dengan *code snippet* 4. 4.

Hasil dari teks yang sudah melalui expand_contraction akan dipecah kedalam setiap kata, dan hasil akhirnya nanti akan dilakukan vektorisasi dan *split* data training dan testing.

Code Snippet 4. 4 tokenization

```
def tokenization(text):
    tokens = word_tokenize(text)
    return tokens

df['tidy_text']=df['tidy_text'].apply(tokenization)
```

Dengan hasil proses token sebagai berikut pada gambar 4. 6.

	diagnosis	icd	tidy_text
2418	OBS UNINVESTIGATED DYSPEPSIA WITH ALARM SYMPTOM	K30	[obs, uninvestigated, dyspepsia, with, alarm, symptom]
3277	HHD. DM TIPE II	I11.9	[hypertensive, heart, disease, diabetes, mellitus, tipe, ii]
1674	DM2 O. HHD EC HT. PASCA LAMINEKTOMI. GASTROENTEROPATHY NSAIDS	E11.9	[diabetes, mellitus, tipe, 2, objective, hypertensive, heart, disease, et, causa, hematrokit, pasca, laminektomi, gastroenteropathy, nsaids]
3816	DYSPEPSIA, TSK GERD. ISPA, TB DALAM PENGOBATAN. VIRAL INFECTION	K30	[dyspepsia, tsk, gastroesophageal, reflux, disease, infeksi, saluran, pernafasan, akut, tinggi, badan, dalam, pengobatan, viral, infection]
1383	DM2 NO -- NEUROPATI DIABETIKA. RIW KRISIS HIPERGLIKEMIA PD DM2NO. IDIOSINKRASI PIOGLITAZONE. SNH DGN VENTRIKULOMEGALI	E11.9	[diabetes, mellitus, tipe, 2, nomor, neuropati, diabetika, riwayat, krisis, hiperglikemia, pd, dm2no, idiosinkrasi, pioglitazone, stroke, non, haemorrhage, dengan, ventrikulomegali]
3476	HT STG I. OBS POLINEUROPATI EC SUSP SPOND LUMBALIS	I10	[hematrokit, stage, i, obs, polineuropati, et, causa, suspect, spond, lumbalis]
2217	DM 2 OBESE. VERTIGO DD PERIFER, CENTRAL. HHD EC HT STG II. DISPEPSIA	E11.9	[diabetes, mellitus, 2, obese, vertigo, different, diagnoses, perifer, central, hypertensive, heart, disease, et, causa, hematrokit, stage, ii, dispepsia]
205	GERD. SINDROM KELELAHAN KRONIK. SUSP FATTY LIVER	K21.9	[gastroesophageal, reflux, disease, sindrom, kelelahan, kronik, suspect, fatty, liver]
371	GERD DGN NCCP (NON CARDIAL CHEST PAIN). HHD EC HT STG II. DISLIPIDEMIA. MILD HIPOKALEMIA. COMMON COLD. IDIOSINKRASI AMLODIPINE (191219)	K21.9	[gastroesophageal, reflux, disease, dengan, nccp, non, cardial, chest, pain, hypertensive, heart, disease, et, causa, hematrokit, stage, ii, dislipidemia, mild, hipokalemia, common, cold, idiosink...
2104	DM2 NO -- NEUROPATI DIAB. DE	E11.9	[diabetes, mellitus, tipe, 2, nomor, neuropati, diab, de]

Gambar 4. 6 hasil tokenisasi

4.2.5 Remove Stopwords

Stopwords seperti "dan", "yang", "ke", dll memiliki frekuensi tinggi tetapi seringkali tidak memberikan nilai tambah pada pemahaman konten teks. Oleh karena itu, penghapusan stopwords menjadi langkah esensial untuk membersihkan data teks.

Penghapusan stopwords tidak hanya membantu memperbaiki kualitas dataset teks tetapi juga meningkatkan efisiensi komputasi dan interpretasi hasil analisis. Dengan mengeliminasi kata-kata yang tidak memberikan kontribusi penting, fokus ditempatkan pada informasi yang lebih bermakna, memudahkan pembentukan model dan analisis lebih lanjut dimana proses ini dikerjakan dengan *code snippet* 4. 5.

Code Snippet 4. 5 remove stopwords

```
stop_words_eng = set(stopwords.words("english"))
stop_words_indo = set(stopwords.words("indonesian"))

def remove_stopwords(text):
    text = [word for word in text if word not in stop_words_eng]
    text = [word for word in text if word not in stop_words_indo]
    return text

df['tidy_text']=df['tidy_text'].apply(remove_stopwords)
```

Dikarenakan kebanyak diagnosa medis menggunakan istilah campuran antara Bahasa Indonesia dan Bahasa inggris, maka dalam tahapan pengapusan stopwords menggunakan kamus stopwords Bahasa indoensia dan Bahasa inggris. Hasil *remove stopwords* ditunjukkan pada gambar 4. 7.

	diagnosis	icd	tidy_text
1073	GERD CUM GASTRITS ANTHRAL EROSIVA	K21.9	[gastroesophageal, reflux, disease, cum, gastrits, anthral, erosiva]
2749	PASCA FEBRIS 6 HR EC DHF . HEPATOPATI RELATED DHF. DISPEPSIA	K30	[pasca, febris, heart, rate, et, causa, dengue, haemorrhagic, fever, hepatopati, related, dengue, haemorrhagic, fever, dispepsia]
1374	DM2 NO -- NEUROPATI DIAB. ABSES PEDIS D	E11.9	[diabetes, mellitus, nomor, neuropati, diab, abses, pedis]
4482	PAROXYSMAL ATRIAL FLUTTER. HHD. POST PASANG DJ STENT	I11.9	[paroxysmal, atrial, flutter, hypertensive, heart, disease, posterior, pasang, diet, jantung, stent]
3822	OBS ABDOMINAL PAIN E.C DYSPEPSIA DD GERD	K30	[obs, abdominal, pain, et, causa, dyspepsia, different, diagnoses, gastroesophageal, reflux, disease]
787	GERD DGN NCCP	K21.9	[gastroesophageal, reflux, disease, nccp]
23	GERD - DISPEPSIA TIPE DISMOTILITAS. HIPOVIT D	K21.9	[gastroesophageal, reflux, disease, dispepsia, tipe, dismotilitas, hipovit]
26	GERD, GASTRITIS ANTHRAL EROSIVA, PYLORUS GAPPING, DUODENITIS NON SPESIFIK. KLINIS HEMORHOID INTERNA	K21.9	[gastroesophageal, reflux, disease, gastritis, anthral, erosiva, pylorus, gapping, duodenitis, non, spesifik, klinis, hemorhoid, interna]
100	GERD	K21.9	[gastroesophageal, reflux, disease]
1115	GERD MIXED . SUSP HN REN D	K21.9	[gastroesophageal, reflux, disease, mixed, suspect, hn, ren]

Gambar 4. 7 hasil *stopwords*

4.3 Text Representation

Hasil akhir dari *pre-processing* adalah bentuk token atau setiap diagnosa sudah dalam keadaan dipisahkan oleh tanda koma (,) untuk selanjutnya hasil tersebut akan direpresentasikan ke dalam bentuk vector sebagai input model. Pada tahap ini data akan dibagi menjadi 3 bagian yakni training, testing dan validasi.

4.3.1 Membagi data training dan testing

Pembagian data training dan testing menjadi 70:30 dilakukan dengan menggunakan fungsi train_test_split(). Sebelum data dibagi penulis mempopulasikan data keseluruhan berdasarkan kode ICD10 kelasnya seperti pada tabel 4. 1.

Tabel 4. 1 populasi data

K21.9	1246
E11.9	1010
K30	954
I10	776
I11.9	624
Kolom ICD dtype : int64	

Mendifinisikan kolom diagnose sebagai fitur dengan variable x dan kolom icd sebagai kelas dengan variable y dengan *code snippet* 4. 6.

Code Snippet 4. 6 split data traning dan testing

```
X=df['tidy_text']
y=df['icd']
X_train, X_test, y_train_, y_test_ = train_test_split(X, y, test_size =0.30, random_state=42)
y_train=pd.get_dummies(y_train_).values
y_test=pd.get_dummies(y_test_).values
```

Hasil pembagian dari x_train dan y_train adalah 3227 : 1383 dengan 5 kelas, begitu juga untuk hasil x_test dan y_test mendapatkan pembagian yang sama.

4.3.2 Vectorisasi data

Tahap terakhir dari *text-representation* adalah merepresentasikan setiap kata menjadi bentuk vector sebagai masukan model yang akan dibuat. Proses vektorisasi yang akan dijalankan dengan menggunakan metode TF-IDF pada *code snippet* 4 7.

Code Snippet 4. 7 vectorisasi data TF-IDF

```
def dummy_fun(doc):
    return doc

tfidf_vect=TfidfVectorizer(analyzer='word', tokenizer=dummy_fun, preprocessor=dummy_fun,
token_pattern=None, smooth_idf=False)
tfidf_vect.fit(X_train)
X_train=tfidf_vect.transform(X_train)
X_test=tfidf_vect.transform(X_test)
```

4.4 Membangun Model

Tahapan membangun model adalah mencari konfigurasi yang paling bagus baik dari sisi akurasi, recall, precision dan diharapkan model memiliki tingkat loss seminimal mungkin. Dalam tahapan ini dijalankan beberapa skenario sebagaimana berikut tabel 4. 2 :

Tabel 4. 2 skenario pembuatan model

NO	hyperparameter	keterangan
1.	Menambah hidden layer	1 layer input-output dengan 2 hidden layer
2.	Penyesuaian nilai *epoch*	6, 12, 48
3.	Penyesuaian nilai *batch_size*	50, 100, 200

Dari setiap skenario akan dicatat dan dibandingkan hasil akurasi, recall, precission dan testing loss dari learning curve.

Sebagai acuan parameter awal adalah bahwa model yang akan dibangun akan menggunakan algoritma Neural Network dimana output layer sesuai dengan jumlah kelas data yang diolah yaitu 5 output layer. Kemudian fungsi aktivasi yang digunakan setiap layer input dan hidden layer adalah aktivasi relu, dan untuk layer output dikarenakan data merupakan klasifikasi multikelas maka akan digunakan aktivasi softmax untuk

menghasilkan distribusi probabilitas dari keluaran model. Fungsi softmax memetakan nilai-nilai output ke dalam distribusi probabilitas yang jumlahnya sama dengan 1, yang dapat digunakan untuk menentukan kelas prediksi.

Parameter terakhir adalah menggunakan fungsi loss categorical_crossentropy biasanya digunakan dalam tugas klasifikasi multikelas di mana setiap sampel data dapat termasuk dalam lebih dari satu kelas.

1. Percobaan dengan 3 hidden layer dengan *code snippet* 4. 8

Code Snippet 4. 8 percobaan 1

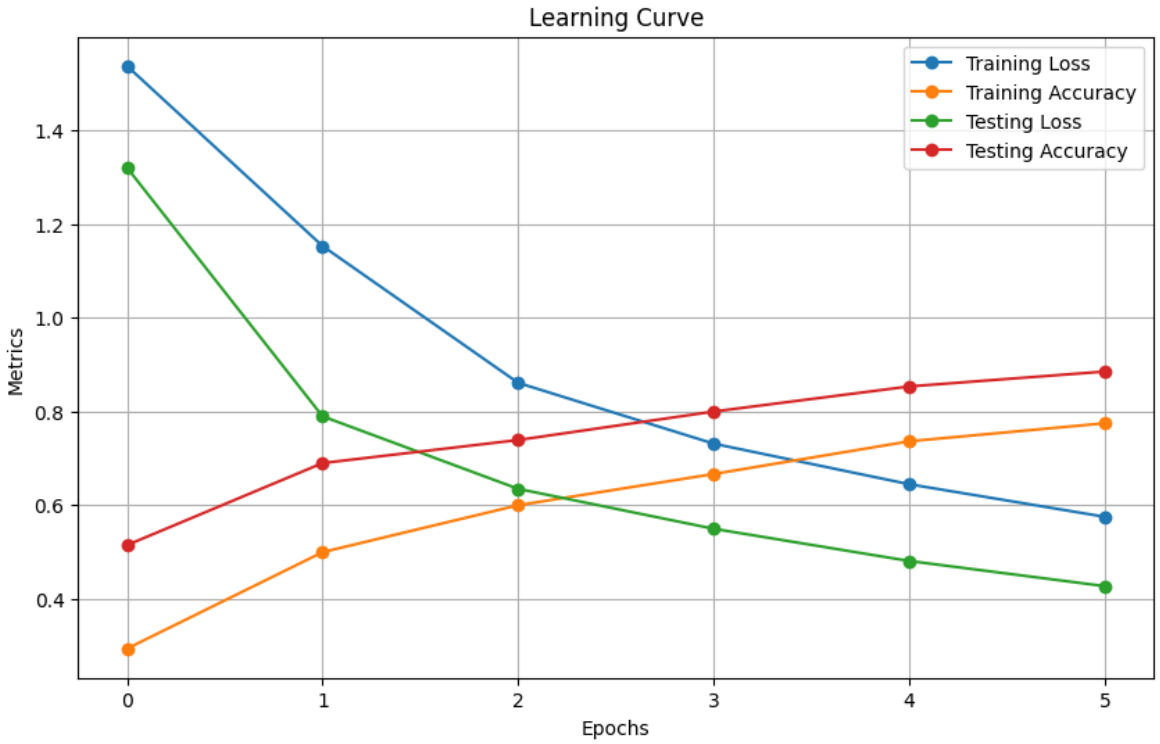
```
def create_model(X):
   model = Sequential()
   model.add(Dense(64, input_dim = X.shape[1], activation = 'relu'))
   model.add(Dropout(0.5))
   model.add(Dense(32, activation = 'relu'))
   model.add(Dense(17, activation = 'relu'))
   model.add(Dense(8, activation = 'relu'))
   model.add(Dropout(0.2))
   model.add(Dense(5,activation='softmax'))
      model.compile(loss='categorical_crossentropy',optimizer='adam', metrics=['accuracy'])
   print(model.summary())

   return model
```

Code Snippet 4. 9 epoch 6 batch_size 50

```
history = model.fit(X_train.toarray(), y_train, epochs=6, batch_size=50, verbose=1,
validation_data=(X_test.toarray(), y_test))
```

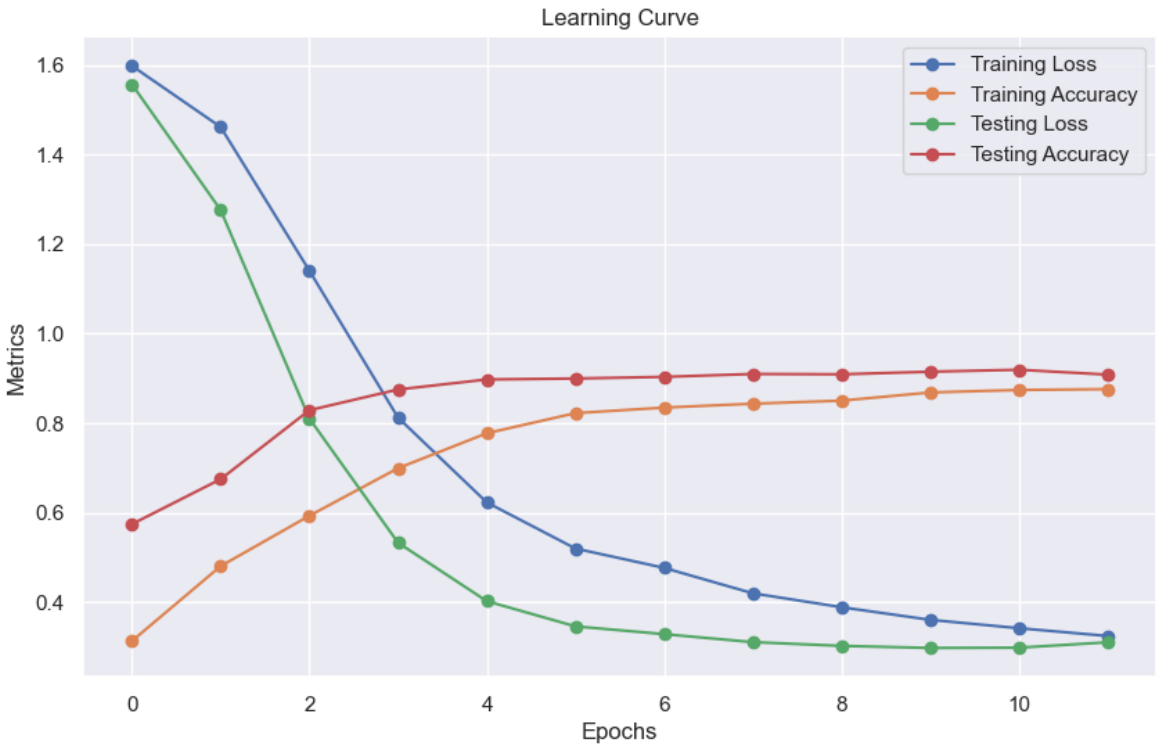
Hasil percobaan dengan nilai epoch 6 dan batch_size 50 ditampilkan pada learning curve berikut dengan hasil testing accuracy 0,885 dan nilai testing loss 0,427 sesuai yang ditunjukkan pada gambar 4. 8. Grafik pembelajaran menunjukkan kurva positif dimana dapat dilihat bahwa training, testing accuracy naik beriringan dan training, testing loss turun beriringan yang membuktikan model dapat mempelajari data secara baik dengan akurasi 88% namun pada epoch terakhir garis akurasi dan loss belum bertemu sehingga masih ada potensi kenaikan nilai akurasi dan penurunan loss jika epoch ditambah.



Gambar 4. 8 percobaan 1a 3 hidden layer 6 epoch 50 batch_size

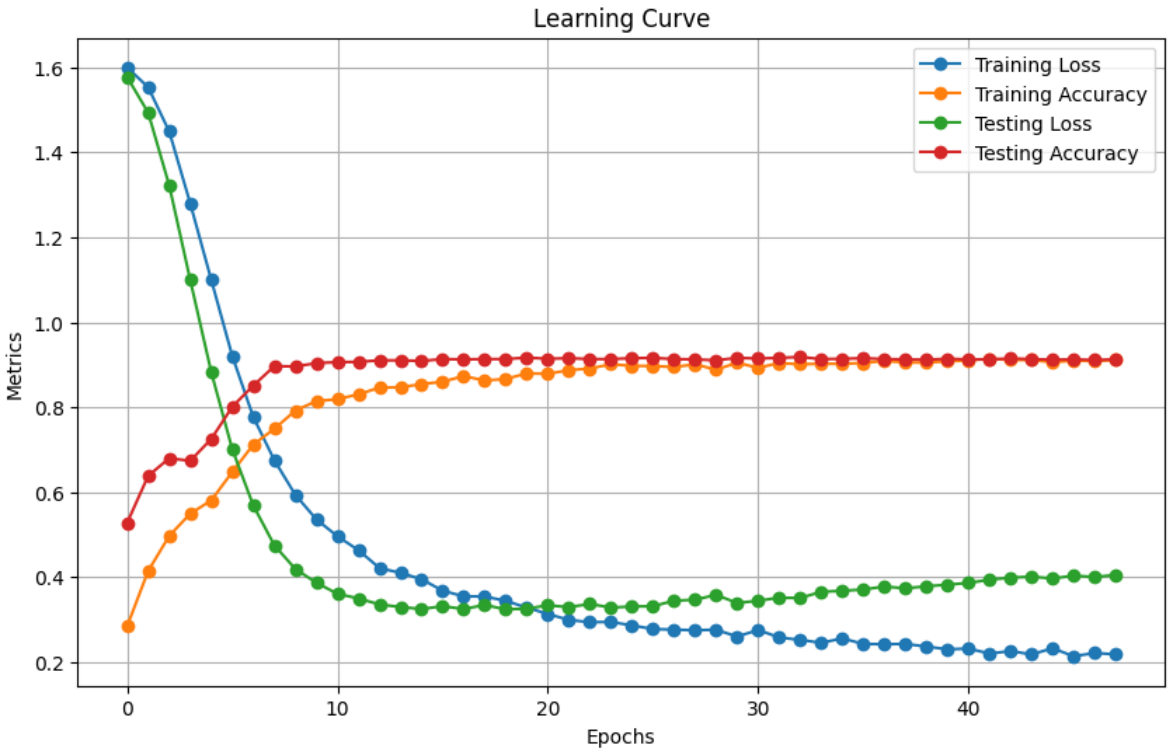
Selanjutnya penyesuaian nilai epoch 12 dan batch_size 100 pada hasil learning curve didapatkan nilai sebagai berikut, dengan hasil testing accuracy lebih tinggi yaitu 0,908 dan nilai testing loss lebih rendah menjadi 0,310 sebagaimana ditunjukkan gambar 4 9. Pada gambar 4 9 didapatkan garis training dan testing loss bertemu pada titik epoch terakhir menyentuh nilai 0,310 lebih rendah dibandingkan dengan epoch 6 pada percobaan 1a gambar 4. 8.

Kemudian garis akurasi meningkat dan hampir bertemu pada satu titik pada epoch terakhir, akurasi meningkat dari percobaan 1a dengan nilai akurasi semula 88% menjadi 90% meningkat cukup signifikan.



Gambar 4. 9 percobaan 1b 3 hidden layer 12 epoch 100 batch_size

Percobaan terakhir adalah menambah nilai epoch dan batch_size menjadi 48 dan 200 sebagaimana hasil dibawah, didapatkan hasil testing accuracy lebih tinggi yaitu 0.912 namun nilai testing loss lebih tinggi menjadi 0.405 seperti ditunjukkan pada gambar 4 10. Pada skenario epoch tertinggi gambar 4. 10 menunjukkan kenaikan garis akurasi yang lebih baik namun tidak signifikan hanya sebesar 0,004 saja, sedangkan garis loss semakin terpisah antara training dan testing yang menggambarkan bahwa semakin banyak epoch, model akan memiliki nilai loss yang semakin tinggi yakni pada gambar 4. 10 diperoleh nilai loss sebesar 0.405 lebih besar dari percobaan 1b dengan 12 epoch.



Gambar 4. 10 percobaan 1c 3 hidden layer 48 epoch 200 batch_size

Hasil rekap hasil percobaan 1 adalah sebagaimana dipaparkan pada tabel 4. 3 berikut, dapat disimpulkan bahwa nilai epoch besar tidak menjamin nilai loss kecil sebagaimana didapatkan nilai loss paling baik adalah dengan konfigurasi hyperparameter pada jumlah epoch 12 dengan batch_size 100.

Tabel 4. 3 hasil percobaan 1

	epoch:batch_size					
	6:50		12:100		48:200	
	val_loss	val_accuracy	val_loss	val_accuracy	val_loss	val_accuracy
3 hidden layer	0,427	0,885	0,31	0,908	0,405	0,912

2. Percobaan dengan 2 hidden layer

   Percobaan ke-2 adalah mencoba mengurangi jumlah hidden layer pada model, kemudian untuk skenario hyperparameter masih sama dengan percobaan pertama, dikerjakan dengan *code snippet* 4. 10.

Code Snippet 4. 10 percobaan 2

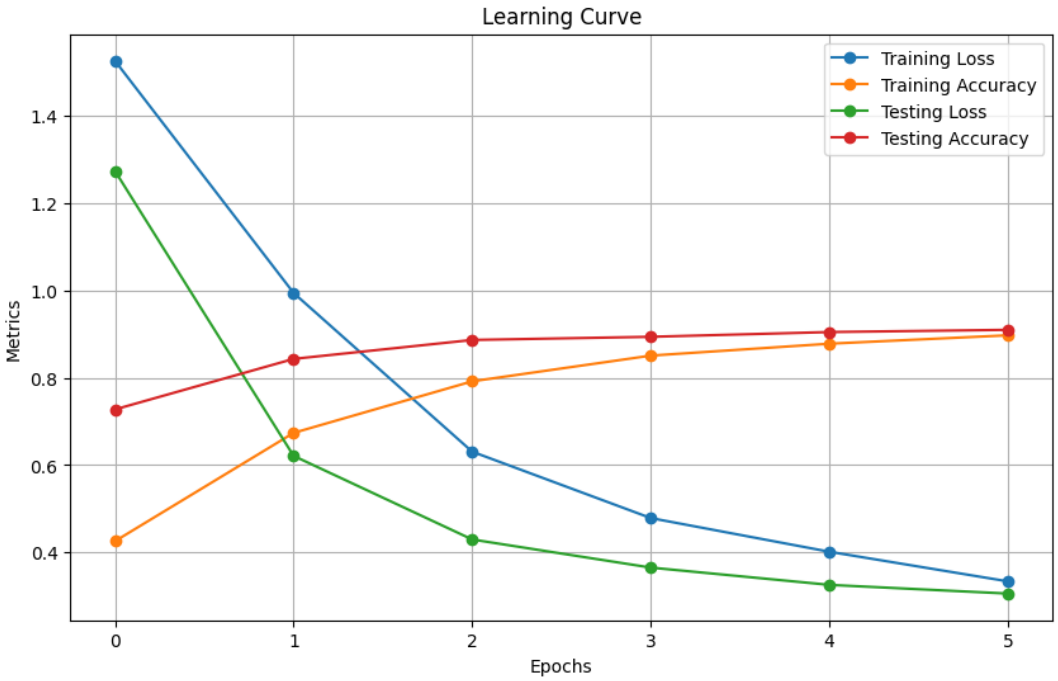
```
def create_model(X):
    model = Sequential()
    model.add(Dense(64, input_dim = X.shape[1], activation = 'relu'))
    model.add(Dropout(0.5))
    model.add(Dense(32, activation = 'relu'))
    model.add(Dense(17, activation = 'relu'))
    model.add(Dropout(0.2))
    model.add(Dense(5,activation='softmax'))

    model.compile(loss='categorical_crossentropy',optimizer='adam', metrics=['accuracy'])
    print(model.summary())

    return model
history = model.fit(X_train.toarray(), y_train, epochs=6, batch_size=50, verbose=1, validation_data=(X_test.toarray(), y_test))
```

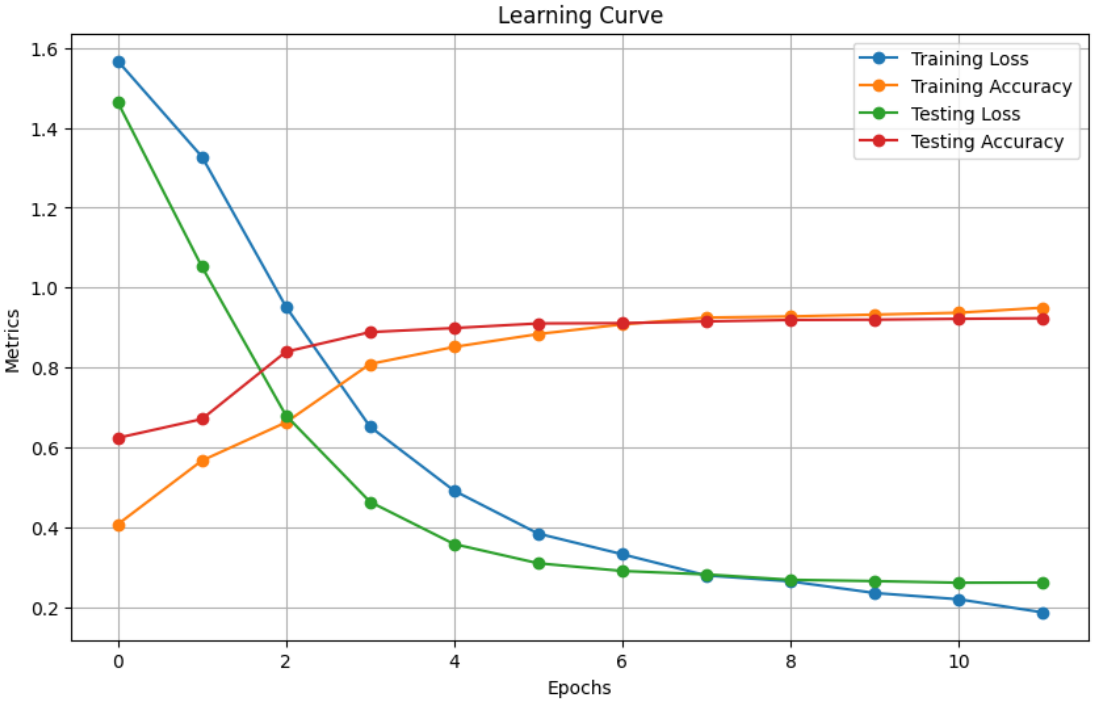
hasil percobaan ke-2 dengan pengurangan jumlah hidden layer dari 3 menjadi 2 dengan nilai parameter epoch 6 dan batch_size 50 memiliki performa yang lebih baik dibandingkan dengan 3 hidden layer yakni nilai testing loss 0,305 dan akurasi naik menjadi 0,909 sebagaimana berikut gambar 4. 11.

Dapat dilihat bila dibandingkan dengan percobaan 1a dengan konfigurasi hyperparameter sama namun jumlah hidden layer dari 3 menjadi 2 garis akurasi dan loss cenderung bertemu pada satu titik yang mana dapat diartikan sebagai titik maksimal pada epoch konfigurasi. Hasil konfigurasi penurunan jumlah hidden layer mendapatkan nilai akurasi yang lebih baik yakni 0,909 dengan nilai loss 0,305 hampir menyamai dengan konfigurasi hyperparameter percobaan 1b dengan jumlah hidden layer 3, nilai epoch 12 dan nilai batch_size 100.



Gambar 4. 11 percobaan 2a 2 hidden layer 6 epoch 50 batch_size

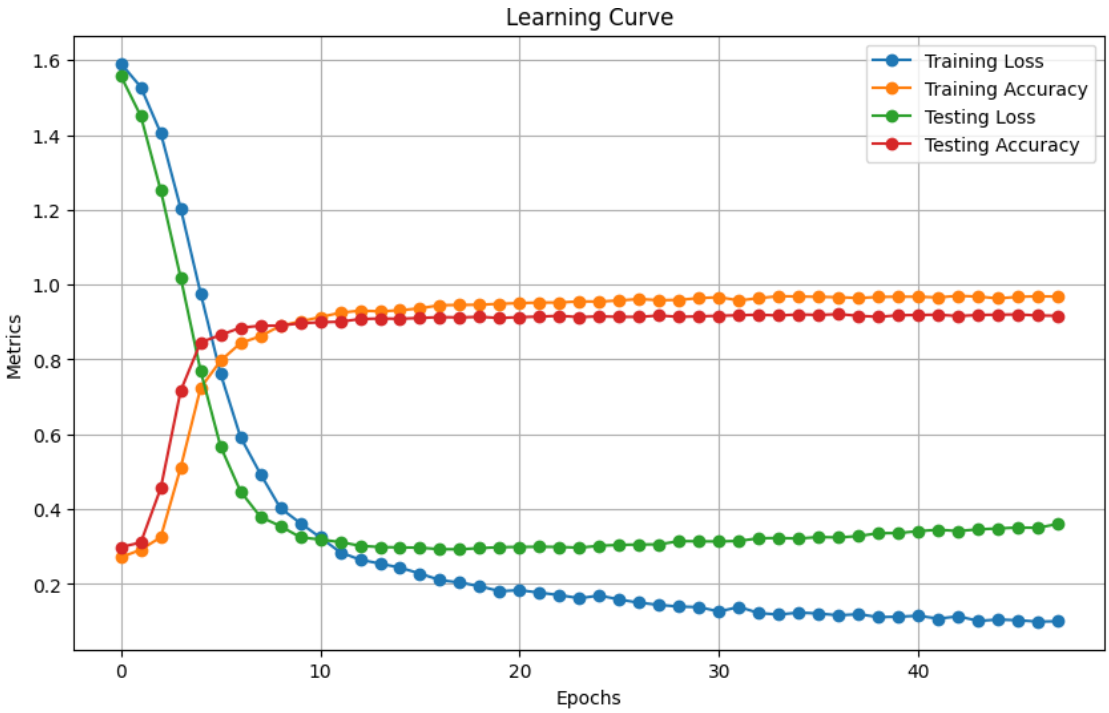
Selanjutnya adalah bertahap menambah nilai parameter epoch menjadi 12 dan batch_size 100, hasil dari penambahan tersebut didapatkan nilai testing loss 0,262 lebih baik dibandingkan epoch dan batch_size kecil sebagaimana digambarkan pada garis grafik gambar 4. 12 dapat dilihat dimana traning dan testing loss bertemu pada 1 titik di sekitar epoch ke 7-8 dan mulai berpisah pada epoch ke 8 yang menggambarkan nilai loss akan menjadi tidak stabil pada epoch diatas 8. Pada percobaan 2b nilai akurasi meningkat menjadi 0,923 sebagaimana ditampilkan pada gambar 4 12 dibandingkan dengan percobaan pada nilai epoch kecil yakni 6 epoch dan 50 batch_size.



Gambar 4. 12 percobaan 2b 2 hidden layer 12 epoch 100 batch_size

Terakhir adalah menaikkan nilai parameter epoch menjadi 48 dan batch_size menjadi 200, nilai maksimal ini menghasilkan training loss sebesar 0,360 dan akurasi menjadi 0,916 dimana dapat dilihat performa model lebih rendah daripada nilai epoch 12 dan batch_size 100 yang ditunjukkan gambar 4. 13.

Nilai loss paling stabil pada titik epoch ke-10 sedangkan diatas nilai tersebut hasil dari training dan testing loss menjadi menyebar sebagaimana digambarkan oleh gambar 4. 13 garis training loss dan testing loss. Sebagaimana nilai loss, nilai akurasi menjadi melebar pada epoch lebih dari 10 dan didapatkan nilai akurasi tidak lebih baik daripada nilai akurasi pada percobaan 2b dengan konfiguasi 12 epoch dan batch_size 100.



Gambar 4. 13 percobaan 2c 2 hidden layer 48 epoch 200 batch_size

Hasil rekap data dari percobaan 2 diatas sebagai berikut tabel 4 4.

Tabel 4. 4 hasil percobaan 2

	epoch:batch_size					
	6:50		12:100		48:200	
	val_loss	val_accuracy	val_loss	val_accuracy	val_loss	val_accuracy
2 hidden layer	0,305	0,909	0,262	0,923	0,36	0,916

3. Percobaan dengan 1 hidden layer

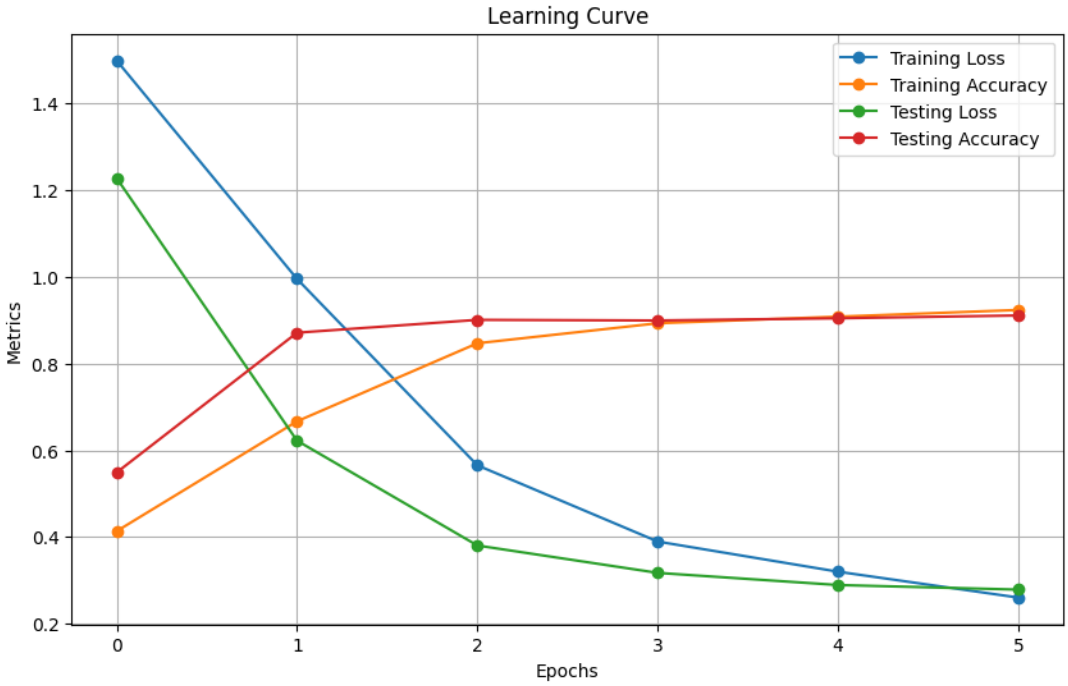
Percobaan terakhir adalah meninggalkan hidden layer hingga 1 layer, namun nilai parameter epoch dan batch_size masih menggunakan kombinasi yang sama akan dikerjakan dengan *code snippet* 4. 11.

Code Snippet 4. 11 percobaan 3

```
def create_model(X):
   model = Sequential()
   model.add(Dense(64, input_dim = X.shape[1], activation = 'relu'))
   model.add(Dropout(0.5))
   model.add(Dense(32, activation = 'relu'))
   model.add(Dropout(0.2))
   model.add(Dense(5,activation='softmax'))
      model.compile(loss='categorical_crossentropy',optimizer='adam', metrics=['accuracy'])
   print(model.summary())
```

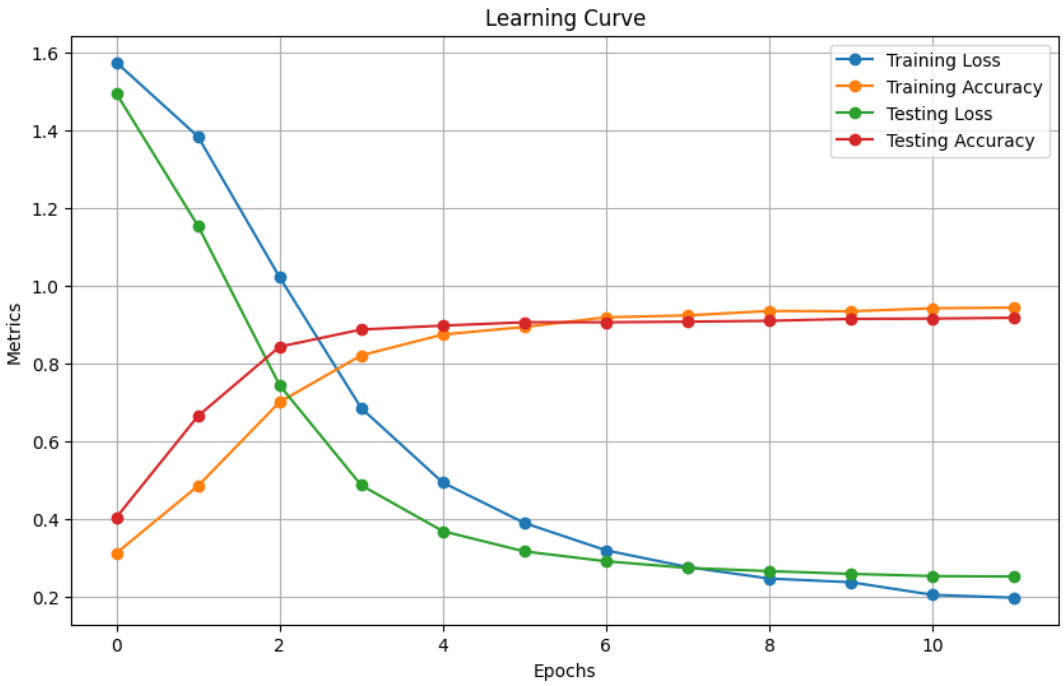
```
    return model
history = model.fit(X_train.toarray(), y_train, epochs=6, batch_size=50, verbose=1,
validation_data=(X_test.toarray(), y_test))
```

Hasil pertama untuk model dengan 1 hidden layer dengan nilai parameter epoch 6 dan batch_size 50 didapatkan nilai testing loss 0,278 dan nilai akurasi sebesar 0,911 ditunjukkan detail pada gambar 4. 14. Pada garis grafik akurasi training dan testing sudah bertemu pada epoch ke 3-4 namun pada epoch tersebu nilai loss masih terlihat tinggi ditunjukkan pada gambar 4. 14 grafik hijau dan biru.



Gambar 4. 14 percobaan 3a 1 hidden layer 6 epoch 50 batch_size

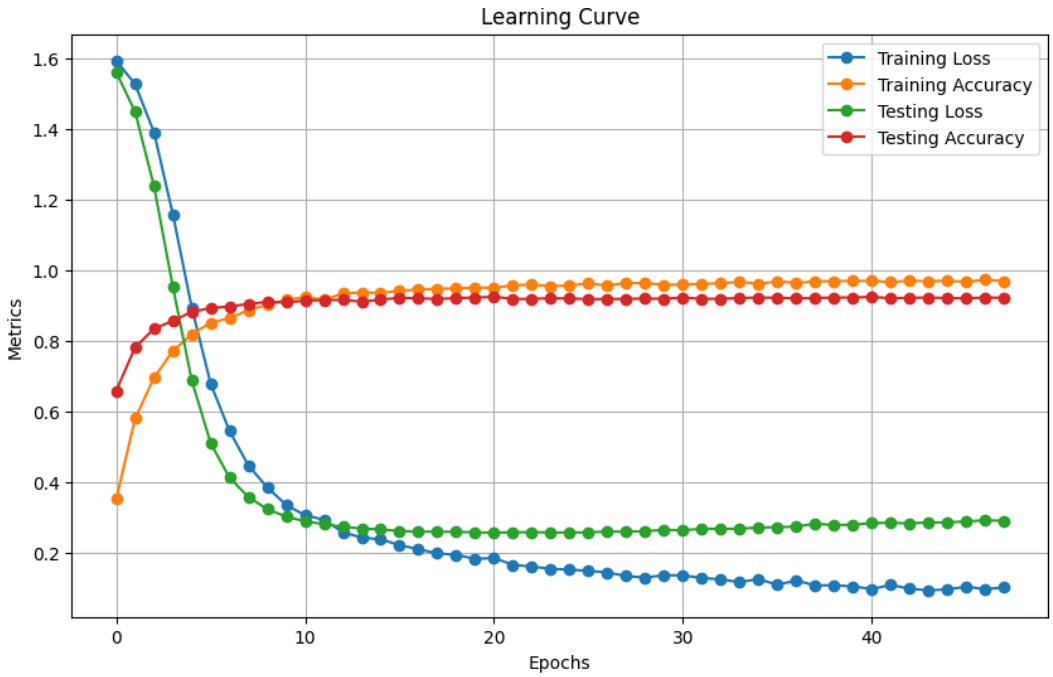
Selanjutnya nilai hyperparameter epoch dan batch_size akan ditingkatkan menjadi 12 epoch dan nilai batch_size adalah 100, disini didapatkan nilai testing loss lebih baik namun tidak signifikan yaitu 0,251 dan nilai akurasi menjadi 0,918 yang ditampilkan pada gambar 4. 15 dibawah.



Gambar 4. 15 percobaan 3b 1 hidden layer 12 epoch 100 batch_size

Percobaan terakhir adalah meningkatkan nilai parameter paling tinggi dari 12 menjadi 48 dan nilai batch_size dari 100 menjadi 200, didapatkan hasil akhir dari tuning dengan nilai paling tinggi yakni nilai testing loss sebesar menjadi lebih besar pada 0,291 dan sedangkan nilai akurasi meningkat menjadi 0,922 sebagaimana dipaparkan pada gambar 4. 16.

Pada grafik traning dan testing loss bertemu pada sekitar epoch 12 menunjukkan nilai loss yang paling stabil pada titik epoch tersebut. Sedangkan epoch lebih besar dari 12 akan diperoleh nilai training dan testing loss yang semakin melebar artinya testing loss semakin naik setelah epoch 12.



Gambar 4. 16 percobaan 3c 1 hidden layer 48 epoch 200 batch_size

Dari ketiga percobaan tersebut didapatkan hasil rekap pada tabel berikut. Pada tabel tersebut dapat dilihat bahwa nilai testing loss yang paling kecil atau palin baik adalah pada percobaan dengan 1 hidden layer dan nilai hyperparameter di rentang 12 untuk nilai epoch dan 100 untuk nilai batch_size.

Sedangkan untuk nilai akurasi yang paling tinggi diperoleh dari percobaan dengan 1 hidden layer dan nilai hyperparameter di rentang 48 untuk jumlah epoch dan 200 untuk nilai batch_size, secara detail direkap pada tabel 4. 5.

Tabel 4. 5 Rekap hasil percobaan

	epoch:batch_size					
	6:50		12:100		48:200	
	val_loss	val_accuracy	val_loss	val_accuracy	val_loss	val_accuracy
3 hidden layer	0,427	0,885	0,31	0,908	0,405	0,912
2 hidden layer	0,305	0,909	0,262	0,923	0,36	0,916
1 hidden layer	0,278	0,911	0,251	0,918	0,291	0,922

Dari beberapa skenario percobaan yang sudah dilakukan, sebagaimana tercermin pada gambar 4. 15 garis grafik training dan testing loss turun beriringan pada titik terendah diantara ketiga percobaan dengan nilai validation loss 0,251 sedangkan nilai training dan testing akurasi tertinggi di angka 0,918. Walaupun nilai akurasi tertinggi pada konfigurasi dengan 2 hidden layer, konfigurasi dengan 1 hidden layer dipilih dengan pertimbangan untuk mencegah model overfitting dimana model terlalu menyesuaikan data training, overfitting terjadi ketika model terlalu baik dalam menyesuaikan diri dengan data training. Pertimbangan lain adalah perihal efektifitas waktu pelatihan dan komputasi. Semakin besar jumlah layer, semakin kompleks dan lambat waktu pelatihan dan inferensi model. Ini karena setiap layer memerlukan lebih banyak komputasi untuk memproses data.

Dari hasil ini, dapat disimpulkan bahwa penggunaan 1 hidden layer memberikan keseimbangan yang baik antara kompleksitas model dan kinerja, dengan val_loss yang rendah dan val_accuracy cukup tinggi.

4.5 Pengujian Model

Dari hasil model yang sudah dilatih pada tahap sebelumnya, model akan masuk proses pengujian yang memungkinkan kita untuk mengukur sejauh mana model mampu menggeneralisasi pada data yang belum pernah dilihat sebelumnya.

Pengukuran hasil model machine learning dalam tugas klasifikasi multikelas melibatkan beberapa metrik evaluasi yang dapat memberikan gambaran menyeluruh tentang kinerja model. Dalam hal ini penulis akan menggunakan confusion matrix dan mengukur sensitivitas model dengan library matplotlib yang dikerjakan dengan *code snippet* 4. 12.

Code Snippet 4. 12 pengujian model

```
y_pred = []
for val in y_predict:
    y_pred.append(np.argmax(val))

def evaluateModel(y_test, y_pred):
    cf_matrix = confusion_matrix(y_test,y_pred)
    sensitivity = recall_score(y_test,y_pred, average=None)
    sns.set_theme(rc={'figure.figsize':(8,8)})
    ax = sns.heatmap(cf_matrix,annot=True,cmap="Reds",fmt="g",xticklabels=['E11.9', 'I10', 'I11.9',
'K21.9', 'K30'],
                yticklabels=['E11.9', 'I10', 'I11.9', 'K21.9', 'K30'],cbar=False)
    ax.set_ylabel('True Labels')
    ax.set_xlabel('Predicted Labels');
    plt.title("Confusion Matrix On Test Data")
    plt.show()
    print("Classification Report:")
    print(classification_report(y_test, y_pred, target_names =['E11.9', 'I10', 'I11.9', 'K21.9', 'K30'],
                            digits=3))
```

```
    target_names =['E11.9', 'I10', 'I11.9', 'K21.9', 'K30']
    sensitivities_dict = dict(zip(target_names, sensitivity))

    print("Sensitivity setiap kelas :")
    for kelas, sensitivity in sensitivities_dict.items():
        rounded_sensitivity = round(sensitivity, 3)
        print(f"{kelas.ljust(5)} : {rounded_sensitivity}".ljust(15))

evaluateModel(np.argmax(y_test, axis=1), y_pred)
```

pada hasil pengujian model didapatkan nilai precission, recall, f-1 score dan support setiap kelas sebagaimana berikut :

```
Classification Report:
              precision    recall  f1-score   support

       E11.9      0.934     0.947     0.941       301
         I10      0.850     0.846     0.848       208
       I11.9      0.869     0.874     0.872       175
       K21.9      0.952     0.945     0.949       402
         K30      0.929     0.926     0.927       297

    accuracy                          0.918      1383
   macro avg      0.907     0.908     0.907      1383
weighted avg      0.918     0.918     0.918      1383
```

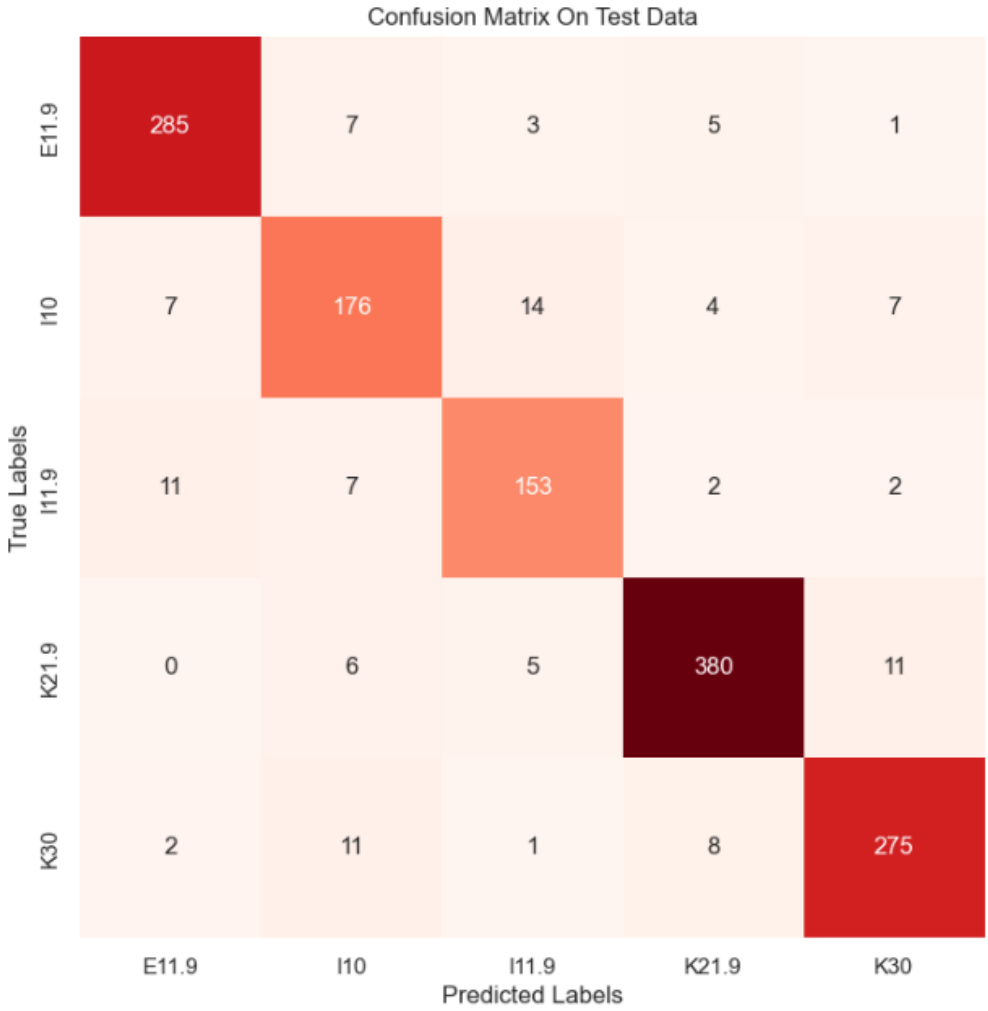
Nilai *precision* mengukur seberapa banyak dari kelas yang diprediksi benar oleh model dibandingkan dengan total prediksi positif untuk kelas tersebut. Contoh nilai precision untuk kelas E11.9 adalah 0.934, yang berarti 93.4% dari prediksi positif untuk kelas ini benar-benar termasuk dalam kelas tersebut. Lalu nilai recall mengukur seberapa banyak dari seluruh kasus positif yang berhasil diidentifikasi oleh model dibandingkan dengan total kasus positif yang sebenarnya.

Nilai *recall* untuk kelas E11.9 adalah 0.947, yang berarti 94.7% dari seluruh kasus positif kelas ini berhasil diidentifikasi. Kemudian terakhir dapat dilihat dari akurasi model secara keseluruhan memiliki nilai akurasi sebesar 0.918 atau 91.8% yang berarti model dapat mengukur sebanyak 91.8% dari seluruh prediksi yang benar oleh model dibandingkan dengan total prediksi.

Lalu nilai *support* menggambarkan seberapa banyak *instance* atau sampel yang terdapat dalam setiap kelas sebagaimana nilai kelas E11.9 adalah 301, I10 adalah 208, I11.9 adalah 175, K21.9 adalah 402 dan K30 adalah 287.

C*onfussion matrix* hasil pengujian model ditunjukkan dengan *heatmap plot* gambar 4 17 dimana semakin gelap warna akan menunjukkan nilai yang semakin besar dari jumlah

data yang diprediksi dari total sampel. Pada plot diagonal dapat dilihat bahwa kelas K21.9 memiliki *true positive* sebesar 383 dari banyaknya sampel kelas tersebut yaitu 402. Sedangkan pada plot horizontal dimana dapat dilihat sejauh mana ketidaktepatan model dalam melakukan prediksi kelas, sebagai contoh kelas K21.9 yang diprediksi sebagai kelas E11.9 adalah 0, kelas K21.9 yang diprediksi sebagai I10 adalah 6, kelas K21.9 yang diprediksi sebagai I11.9 adalah 5 dan kelas K21.9 yang diprediksi sebagai kelas K30 adalah 11.



Gambar 4. 17 *confussion matrix* pengujian model

Selanjutnya pada tabel 4. 6 ditampilkan distibusi probabilitas pada setiap kelas hasil prediksi dalam persentase. Pada tabel dapat dilihat bahwa kelas atau kode ICD-10 E11.9 diprediksi benar oleh model sebanyak 95%, diprediksi sebagai I10 dan K21.9 masing-masing sebanyak 2%, diprediksi sebagai I11 sebanyak 1% dan diprediksi sebagai K30 sebanyak 0 atau model tidak pernah memprediksi kelas E11.9 sebagai kelas K30.

Jika dilihat lebih jauh nilai prediksi yang tidak sesuai dengan kelas sebenarnya diatas 5% adalah kelas E11.9 yang diprediksi model sebagai I11.9 sebanyak 7% dan kelas

I11.9 yang diprediksi sebagai E11.9 sebanyak 6%. Pada ketiga kelas atau kode diagnosis tersebut yakni E11.9 Type 2 diabetes mellitus without complications, I11.9 Hypertensive heart disease without (congestive) heart failure, I10 Essential (primary) hypertension memiliki diagnosis yang hampir sama sehingga ketidaktepatan model dalam mengklasifikasikan diagnosis tersebut sedikit lebih besar daripada kelas yang lain.

Tabel 4. 6 rangking penyakit berdasarkan probabilitas kejadian

	JUMLAH SAMPEL	E11.9	I10	I11.9	K21.9	K30
E11.9	301	95%	2%	1%	2%	0%
I10	208	3%	85%	7%	2%	3%
I11.9	175	6%	4%	87%	1%	1%
K21.9	402	0%	1%	1%	95%	3%
K30	297	1%	4%	0%	3%	93%

4.6 Evaluasi Model

Setelah selesai dengan membangun model dan menguji model yang secara keseluruhan memiliki nilai akurasi sebesar 0.918 dengan nilai loss sebesar 0.251 selanjutnya model akan dievaluasi menggunakan data baru yang belum pernah digunakan untuk pelatihan data. Dalam hal evaluasi model, penulis mencoba membagi data tunggal dan dataframe untuk proses evaluasi.

4.6.1 Testing data Tunggal

Sebelum dilakukan testing dengan data berjumlah banyak, model diberikan masukan data tunggal dengan mengambil data dari kunjungan pasien bulan Oktober 2020 yang belum pernah dikenali oleh model. Pada evaluasi model dengan data tunggal didapatkan model dapat memberikan klasifikasi yang benar dengan nilai akurasi atau keyakinan lebih dari 80%.

Code Snippet 4. 13 data input tunggal

```
data_input = ["HIPERTENSI STAGE II (PERBAIKAN). DYSLIPIDEMIA. DM TIPE 2 OBESE GD . TERKONTROL BURUK. DYSPEPSIA. ISK (PERBAIKAN)",
"GERD MIXED DGN PALPITASI. HT STG I. KISTA REN S",
"HT STG II. OBS ABD PAIN EC FATTY LIVER. DISPEPSIA TIPE DISMOTILITAS. DISLIPIDEMIA",
"HT STG I. DISLIPIDEMIA. HIPERURICEMIA. POST ROI CLAVICULA S",
"OBS HEARTBURN CUM NON CARDIAC CHEST PAIN EC KLINIS GERD MIXED. HT STG I. SPONDILOSIS LUMBALIS"]

label_target = "I10,K21.9,I10,I10,K21.9"
```

Code Snippet 4. 14 pre-processing evaluasi

```
data_input_lower = [text.lower() for text in data_input]
def framework_preprocessing(text):
```

```
    text = lowercase_text(text)
    text = remove_healthcare_punctuations(text)
    text = handle_abbreviations(text, mydic)
    text = tokenization(text)
    text = remove_stopwords(text)
    return text

X_new = [framework_preprocessing(text) for text in data_input_lower ]

print(X_new)
```

Code Snippet 4. 15 data evaluasi

```
X_new_vectorized = tfidf_vect.transform(X_new)
predictions_new = model.predict(X_new_vectorized.toarray())
dataset_labels = ["E11.9","I10", "I11.9", "K21.9","K30"]
predicted_labels_new = np.argmax(predictions_new, axis=1)
predicted_labels_dataset = [dataset_labels[idx] for idx in predicted_labels_new]
result_df_new = pd.DataFrame({'Original_Text': X_new, 'Predicted_Labels': predicted_labels_dataset})
print(result_df_new.to_markdown(index=False))

for i, pred in enumerate(predictions_new):
    probabilities_str = ", ".join([f"{prob * 100:.2f}%" for prob in pred])
    print(f"Sample {i + 1} - Probabilities: [{probabilities_str}]")
```

Kemudian hasilnya adalah dipaparkan pada tabel 4. 6

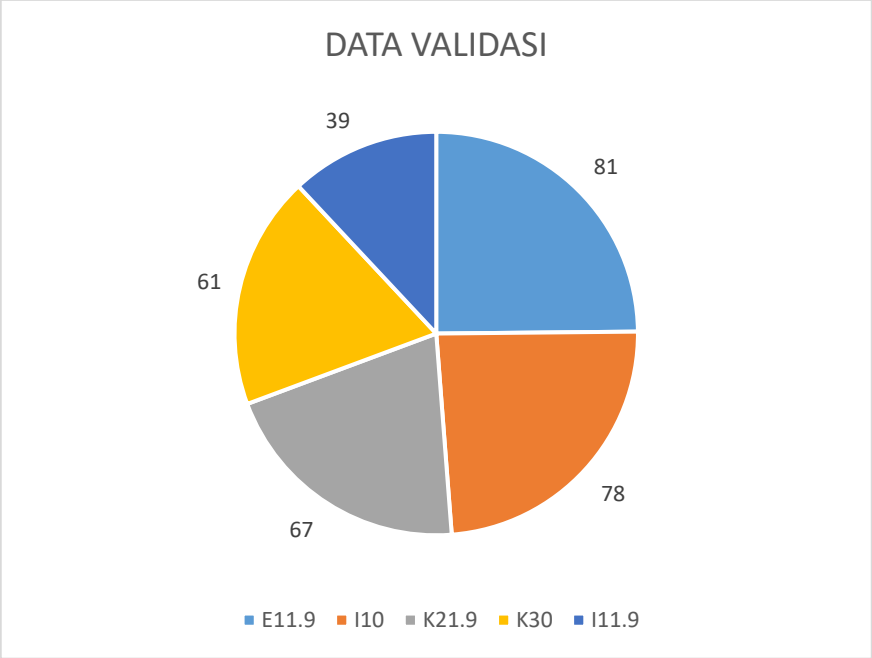
Tabel 4. 7 hasil pengujian data tunggal

	TARGET	HASIL MODEL				
		E11.9	I10	I11.9	K21.9	K30
HIPERTENSI STAGE II (PERBAIKAN). DYSLIPIDEMIA. DM TIPE 2 OBESE GD . TERKONTROL BURUK. DYSPEPSIA. ISK (PERBAIKAN)	I10	1.59%	97.90%	0.14%	0.10%	0.27%
GERD MIXED DGN PALPITASI. HT STG I. KISTA REN S	K21.9	0.19%	6.53%	0.28%	92.82%	0.17%
HT STG II. OBS ABD PAIN EC FATTY LIVER. DISPEPSIA TIPE DISMOTILITAS. DISLIPIDEMIA	I10	1.29%	82.71%	1.03%	0.77%	14.20%
HT STG I. DISLIPIDEMIA. HIPERURICEMIA. POST ROI CLAVICULA S	I10	0.48%	98.44%	0.58%	0.36%	0.15%
OBS HEARTBURN CUM NON CARDIAC CHEST PAIN EC KLINIS GERD MIXED. HT STG I. SPONDILOSIS LUMBALIS	K21.9	0.16%	8.76%	0.25%	90.48%	0.35%

4.6.2 Testing data frame

Pengujian model dengan data baru adalah langkah kritis dalam mengevaluasi kinerja model pada situasi yang sebeneranya. Saat menguji model dengan data baru, penulis ingin memastikan bahwa model dapat menggeneralisasi dengan baik dan memberikan prediksi yang akurat untuk data yang tidak pernah dilihat sebelumnya.

Sebelum data baru masuk menjadi input dari model, data baru akan melalui *pre-processing* yang sama sebagaimana model tersebut dibuat. Hal ini dilakukan untuk menjaga konsistensi pengujian. Data validasi yang akan digunakan adalah data kunjungan pasien dengan periode bulan Oktober yang sudah disiapkan, data tersebut memiliki karakter yang sama dengan data testing dan training yaitu data pasien dengan usia rata-rata 50 tahun dari pasien laki-laki dan perempuan. Data dalam bentuk excel dimuat dengan library pandas. Sebaran populasi setiap kelas dari data baru tersebut adalah sebagai berikut dengan total data 326 data sebagaimana gambar 4. 18.



Gambar 4. 18 sebaran data evaluasi

Code Snippet 4. 16 evaluasi dengan dataframe baru

```
data_uji = pd.read_excel('data-testing-fresh.xlsx', sheet_name="master-testing")
def lowercase_text(text):
    return text.lower()

data_uji['lower']=data_uji['diagnosa-test'].apply(lowercase_text)
def framework_preprocessing(text):

    text = remove_healthcare_punctuations(text)
    text = handle_abbreviations(text, mydic)
    text = tokenization(text)
    text = remove_stopwords(text)

    return text

data_uji['data_clean'] = data_uji['lower'].apply(framework_preprocessing)
data_uji.sample(5)
```

	diagnosa-test	icd-test	lower	data_clean
53	HIPERTENSI STAGE 2	I10	hipertensi stage 2	[hipertensi, stage, 2]
25	TYPE 2 DIABETES MELLITUS STABLE	E11.9	type 2 diabetes mellitus stable	[type, 2, diabetes, mellitus, stable]
70	DM TIPE 2. HHD. LBP EC POST JATUH. DISLIPIDEMIA	E11.9	dm tipe 2. hhd. lbp ec post jatuh. dislipidemia	[diabetes, mellitus, tipe, 2, hypertensive, heart, disease, low, back, pain, et, causa, posterior, jatuh, dislipidemia]
62	HIPERTENSI . TTH. DYSLIPIDEMIA	I10	hipertensi . tth. dyslipidemia	[hipertensi, tth, dyslipidemia]
208	DM . FATTY LIVER. SUSP CHOLEYCYSTITIS. PRURITUS DIAB. PENYINTAS C 19. OBS AKUT URTICARIA	E11.9	dm . fatty liver. susp choleycystitis. pruritus diab. penyintas c 19. obs akut urticaria	[diabetes, mellitus, fatty, liver, suspect, choleycystitis, pruritus, diab, penyintas, celcius, 19, obs, akut, urticaria]
126	HHD EC HT. ACUT ON CKD. NEFROPATHY URAT. OA GENU BILATERAL. SSUP HNP VC DGN SPOND THORACO LUMBALIS DGN POROTIC	I11.9	hhd ec ht. acut on ckd. nefropathy urat. oa genu bilateral. ssup hnp vc dgn spond thoraco lumbalis dgn porotic	[hypertensive, heart, disease, et, causa, hematrokit, acut, chronic, kidney, disease, nefropathy, urat, osteoarhritis, genu, bilateral, ssup, herniated, nucleus, pulposus, vital, capasity, spond, ...
201	DISPEPSIA	K30	dispepsia	[dispepsia]
287	SUSP HNP VC DGN DIZZY. HHD EC HT URGENCY. GTG. GASTROENTEROPATHY. AKI DGN HIPERURICEMIA. RHINOBRONKHITIS AKUT DGN INF SKUNDER. FATTY LIVER. DISLIPIDEMIA	I11.9	susp hnp vc dgn dizzy. hhd ec ht urgency. gtg. gastroenteropathy. aki dgn hiperuricemia. rhinobronkhitis akut dgn inf skunder. fatty liver. dislipidemia	[suspect, herniated, nucleus, pulposus, vital, capasity, dizzy, hypertensive, heart, disease, et, causa, hematrokit, urgency, gtg, gastroenteropathy, aki, hiperuricemia, rhinobronkhitis, akut, inf...
323	GERD	K21.9	gerd	[gastroesophageal, reflux, disease]
318	CHF EC HHD DGN RIW VES. POST BR PN. HIPERKOAGULOPATHY. DM2 OBESE DGN DKD. GASTROENTEROPATHY. CHOLECYTITIS. DISLIPIDEMIA	I11.9	chf ec hhd dgn riw ves. post br pn. hiperkoagulopathy. dm2 obese dgn dkd. gastroenteropathy. cholecytitis. dislipidemia	[congestive, heart, failure, et, causa, hypertensive, heart, disease, riwayat, ventricular, extra, systole, posterior, br, primary, nurse, hiperkoagulopathy, diabetes, mellitus, tipe, 2, obese, dk...

Gambar 4. 19 hasil *pre-processing* data evaluasi

Hasil dari klasifikasi data baru dengan model machine learning yang sudah dilatih akan ditampilkan hasil metrik evaluasinya berupa Tingkat precision, recall, dan akurasi dengan *code snippet* 4. 17.

Code Snippet 4. 17

```
y_pred = model.predict(X_uji_vec.toarray())

cf_matrix = confusion_matrix(y_uji_vec.argmax(axis=1), y_pred.argmax(axis=1))
sensitivity = recall_score(y_uji_vec.argmax(axis=1), y_pred.argmax(axis=1), average=None)

class_labels = ["E11.9","I10", "I11.9", "K21.9","K30"]
plt.figure(figsize=(8, 6))
sns.heatmap(cf_matrix, annot=True, cmap="Blues", fmt="g",
            xticklabels=class_labels, yticklabels=class_labels)
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title("Confusion Matrix With Data Baru")
plt.show()

sensitivities_dict = dict(zip(class_labels, sensitivity))

classification_rep = classification_report(y_uji_vec.argmax(axis=1), y_pred.argmax(axis=1),
                                           target_names = class_labels, digits =3)
print("Classification Report Data Baru:")
print(classification_rep)

print("Sensitivity setiap kelas :")
for kelas, sensitivity in sensitivities_dict.items():
    rounded_sensitivity = round(sensitivity, 3)
    print(f"{kelas.ljust(5)} : {rounded_sensitivity}".ljust(15))
```

Dari hasil confussion matrix diperoleh nilai akurasi sebesar 0.893 atau 89.3% dimana nilai akurasi sebesar 0.893 menunjukkan bahwa model memiliki kinerja yang relatif baik pada dataset evaluasi/ pengujian. Akurasi mengukur sejauh mana model mampu membuat

prediksi yang benar secara keseluruhan. Hasil secara keseluruhan tidak terlalu jauh dari hasil pengujian model saat training dan testing. Sebagaimana data berikut

Classification Report Data Baru:

	precision	recall	f1-score	support
E11.9	0.942	0.802	0.867	81
I10	0.816	0.910	0.861	78
I11.9	0.800	0.923	0.857	39
K21.9	0.983	0.881	0.929	67
K30	0.923	0.984	0.952	61
accuracy			0.893	326
macro avg	0.893	0.900	0.893	326
weighted avg	0.900	0.893	0.893	326

BAB 5
Kesimpulan dan Saran

5.1 Kesimpulan

Model klasifikasi untuk menangani input teks diagnosa dokter yang tidak terstruktur ke dalam kode penyakit berdasarkan ICD10 pada disiplin penyakit dalam yang dibuat mampu mengklasifikasikan data baru dengan akurasi sebesar 89.3%.

Rekomendasi tuning hyperparameter model untuk mendapatkan konfigurasi terbaik dengan mempertimbangkan beban komputasi dan potensi model terjadinya overfitted yakni dengan konfigurasi jumlah 1 hidden layer 12 epoch dan 100 batch_size. Dengan konfigurasi tersebut diperoleh nilai validasi loss paling minimun yaitu 0,251 dengan nilai akurasi model sebesar 0,918 atau 91,8%.

Pada beberapa kode ICD-10 seperti E11.9 Type 2 diabetes mellitus without complications, I11.9 Hypertensive heart disease without (congestive) heart failure, I10 Essential (primary) hypertension memiliki diagnosis yang hampir sama sehingga diaharapkan penulisan diagnosis oleh dokter bisa lebih spesifik untuk mengurangi ketidaktepatan prediksi atau klasifikasi model.

Algoritma Artificial Neural Network (ANN) memiliki nilai akurasi yang cukup baik dalam mengklasifikasikan penyakit berdasarkan kode ICD-10 yakni sebesar 89.3% model mampu mencapai tingkat akurasi yang tinggi dalam pengklasifikasian penyakit ke dalam kode ICD-10. Ini mengurangi risiko kesalahan petugas koding dalam proses pengkodean penyakit, yang dapat memiliki dampak besar pada pengelompokan diagnosis dan pengelolaan pasien.

Pengujian pada lingkungan klinis yang nyata dimana penelitian ini akan digunakan untuk membantu petugas rekam medis dalam pengkodean penyakit pada Rumah Sakit dimana data tersebut diambil dan keberhasilan penggunaan model dalam pengklasifikasian penyakit ke dalam kode ICD-10 memerlukan kolaborasi erat antara profesional medis yang memahami konteks klinis dan ilmuwan data yang memiliki keahlian dalam pengembangan model.

Dengan demikian, penggunaan deep learning dalam pengklasifikasian penyakit ke dalam kode ICD-10 menjanjikan berbagai manfaat dalam meningkatkan akurasi, efisiensi, dan produktivitas dalam sektor perawatan kesehatan. Namun, perlu perhatian yang cermat terhadap implementasi yang aman dan kebutuhan untuk pemeliharaan serta pembaruan model seiring berjalannya waktu.

5.2 Saran

Model yang dihasilkan oleh penulis masih perlu diujikan kepada petugas koding yang dalam hal ini sebagai pakar atau ahli dalam mengelompokkan diagnosis dokter ke dalam kode ICD-10 agar dalam penggunaanya menjadi lebih terarah.

Saran untuk penelitian selanjutnya adalah peneliti dapat membuat interface yang terintegrasi dengan Sistem Manajemen Rumah Sakit (SIMRS) sehingga proses klasifikasi dapat berjalan realtime untuk kebutuhan data pola penyakit. Selanjutnya dalam proses word embedding peneliti dapat melakukan kombinasi dengan metode lain untuk meningkatkan akurasi model.

Penulisan diagnosa perlu dilakukan penetapan standar dimana ketika dokter menuliskan istilah yang berupa singkatan harus merujuk pada kamus singkatan yang sudah ditetapkan oleh Rumah Sakit. Hal ini penting dikarenakan dalam *pre-processing* terdapat tahapan *expand contraction* dimana tahapan ini akan membaca utuh apapun yang diketikkan oleh dokter pada diagnosa dan akan melakukan pencocokan pada kamus singkatan yang sudah ditetapkan.

Daftar Pustaka

Boytcheva, S. (2011). Automatic Matching of ICD-10 codes to Diagnoses in Discharge Letters. *Proceedings of the Workshop on Biomedical Natural Language Processing.* Bulgaria.

Chen Y, L. H. (2017). Automatic ICD-10 coding algorithm using an improved longest common subsequence based on semantic similarity. *PLoS One*.

Dalianis, H. (2018). *Clinical Text Mining Secondary Use Of Electronic Patient Records.* Springer Open.

Defa Miftara Agustine, R. D. (2017). Hubungan Ketepatan Terminologi Medis dengan Keakuratan Kode Diagnosis Rawat Jalan oleh Petugas Kesehatan di Puskesmas Bambanglipuro Bantul. *Jkesvo (Jurnal Kesehatan Vokasional), 1*.

Hanna Suominen, S. P. (2007). Machine Learning to Automate the Assignment of Diagnosis Codes to Free-text Radiology Reports: a Method Description.

Lingling Zhou, C. C. (2020). Construction of a Semi-automatic ICD-10 Coding System. *BMC Medical Informatics and Decision Making*.

Luis Pereira, R. R. (2013). ICD9-based Text Mining Approach to Children Epilepsy Classification. *Procedia Technology*, 1351-1360.

M. K. Ross, W. W.-M. (2014). "Big Data" and the Electronic Health Record. *IMIA Yearbook of Medical Informatics*.

Mardi, Y. (2018). Data Mining Rekam Medis Untuk Menentukan Penyakit Terbanyak Menggunakan Decision Tree C4.5. *JURNAL SAINS DAN INFORMATIKA, v4.ii*, 40-53.

Nabila Nanda Widyastuti, A. B. (2018). Analisis Word2vec untuk Perhitungan Kesamaan Semantik antar Kata. *e-Proceeding of Engineering*, (p. 7603).

Nielsen, J. (2012). *Usability 101: Introduction to Usability*. Retrieved Januari 2022, from https://www.nngroup.com/articles/usability-101-introduction-to-usability/

Nur Maimun, J. N. (2018). Pengaruh Kompetensi Coder terhadap Keakuratan dan Ketepatan Pengkodean Menggunakan ICD 10 di Rumah Sakit X Pekanbaru Tahun 2016. *Jurnal Kesmars*, 31-43.

Pannaporn Ketpupong, K. P. (2018). Applying Text Mining for Classifying Disease from Symptoms. *The 18th International Symposium on Communications and Information Technologies, 18*, 467-472.

Percha, B. (2021). Modern Clinical Text Mining: A Guide and Review. *Annual Review of Biomedical Data Science, 4*(1).

Prajapat, D. R. (2020, Februari 14). *Text Classification: BERT vs DNN*. Retrieved Desember 31, 2021, from https://eng.zemosolabs.com/text-classification-bert-vs-dnn-b226497c9de7

Puspitasari, N. (2017). Evaluasi Tingkat Ketidaktepatan Pemberian Kode Diagnosis Dan Faktor Penyebab Di Rumah Sakit X Jawa Timur. *Jurnal Manajemen Kesehatan Yayasan RS.Dr. Soetomo*, 158-168.

R. Mahmoud, N. E. (2014). ICF Based Automation System for Spinal Cord Injuries Rehabilitation. *Proceedings in Computer Engineering and Systems (ICCES), 2014 9th International Conference.* Egypt, Cairo: IEEE Publishing.

Raja, U. &. (2008). Text Mining In Healthcare: Applications And Opportunities. *Journal of healthcare information management : JHIM*.

su-Ming Wang, Y.-H. C.-C.-N.-Y.-W.-W. (2020). Using Deep Learning for Automatic Icd-10 Classification from Free-Text Data. *European Journal for Biomedical Informatics, 16*(1).

Wang, Y. &.-M. (2020). MedSTS: a resource for clinical semantic textualsimilarity. *Language Resources and Evaluation, 54*.

Zhou L, C. C. (2020). Construction of a semi-automatic ICD-10 coding system. *BMC Med Inform Decision Making*.

LAMPIRAN