PREDIKSI KEBERLANJUTAN RAWAT INAP PASIEN RUMAH SAKIT MENGGUNAKAN XLNET-BiGRU-ATT PADA CATATAN KHUSUS KLINIS PASIEN

ANNISA DARWIS
D121201032

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
MAKASSAR
2024

PREDIKSI KEBERLANJUTAN RAWAT INAP PASIEN RUMAH SAKIT MENGGUNAKAN XLNET-BiGRU-ATT PADA CATATAN KHUSUS KLINIS PASIEN

ANNISA DARWIS

D121 20 1032

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

MAKASSAR

2024

PREDIKSI KEBERLANJUTAN RAWAT INAP PASIEN RUMAH SAKIT MENGGUNAKAN XLNET-BiGRU-ATT PADA CATATAN KHUSUS KLINIS PASIEN

ANNISA DARWIS
D121 20 1032

Skripsi

sebagai salah satu syarat untuk mencapai gelar sarjana

Program Studi Teknik Informatika

pada

PROGRAM STUDI TEKNIK INFORMATIKA
DEPARTEMEN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
MAKASSAR
2024

SKRIPSI

PREDIKSI KEBERLANJUTAN RAWAT INAP PASIEN RUMAH SAKIT MENGGUNAKAN XLNET-BiGRU-ATT PADA CATATAN KHUSUS KLINIS PASIEN

ANNISA DARWIS

D121201032

Skripsi,

telah dipertahankan di depan Panitia Ujian Sarjana Teknik Informatika pada tanggal bulan tahun dan dinyatakan telah memenuhi syarat kelulusan pada

Program Studi Teknik Informatika
Departemen Teknik Informatika
Fakultas Teknik
Universitas Hasanuddin
Makassar

Mengesahkan:
Pembimbing tugas akhir,

Anugrayani Bustamin, S.T., M.T.

NIP 19901201 201807 4 001

Mengetahui:
Ketua Program Studi,

Prof. Dr. Ir. Indrabayu, S. T., M. T., M. Bus. Sys., IPM, ASEAN. Eng.
NIP 19750716 200212 1 004

PERNYATAAN KEASLIAN SKRIPSI
DAN PELIMPAHAN HAK CIPTA

Dengan ini saya menyatakan bahwa, skripsi berjudul "Prediksi Keberlanjutan Rawat Inap Pasien Rumah Sakit Menggunakan XLNet-BiGRU-ATT pada Catatan Khusus Klinis Pasien" adalah benar karya saya dengan arahan dari pembimbing ibu Anugrayani Bustamin, S.T., M.T. Karya ilmiah ini belum diajukan dan tidak sedang diajukan dalam bentuk apa pun kepada perguruan tinggi mana pun. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka skripsi ini. Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini adalah karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut berdasarkan aturan yang berlaku.

Dengan ini saya melimpahkan hak cipta (hak ekonomis) dari karya tulis saya berupa skripsi ini kepada Universitas Hasanuddin.

Makassar, 11 September 2024

ANNISA DARWIS
NIM D121201032

Ucapan Terima Kasih

Assalamualaikum Warahmatullahi Wabarakatuh.

Alhamdulillahirabbil'aalamiin, puji dan syukur penulis panjatkan atas kehadirat Allah Subhanahu wa ta'ala yang telah mencurahkan rahmat serta karunia-Nya, sehingga penulis dapat menyelesaikan tugas akhir ini yang berjudul " Prediksi Keberlanjutan Rawat Inap Pasien Rumah Sakit Menggunakan XLNet-BiGRU-ATT pada Catatan Khusus Klinis Pasien" sebagai salah satu persyaratan yang harus dipenuhi dalam menyelesaikan jenjang Strata-1 pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Penulis menyadari bahwa dalam penyusunan dan penulisan laporan skripsi ini tidak lepas dari bantuan, bimbingan serta dukungan dari berbagai pihak, dari masa perkuliahan sampai dengan masa penyusunan skripsi. Oleh karena itu, penulis dengan senang hati menyampaikan terima kasih kepada :

1. Kedua Orang tua penulis, Bapak Darwis dan Ibu Nurlan, yang selalu memberikan dukungan dan doa yang tiada henti, memberi semangat kepada penulis, serta selalu sabar dalam mendidik penulis sejak kecil.
2. Seluruh Keluarga dan Saudara penulis, yang selalu memberikan semangat dan mendoakan penulis selama ini.
3. Ibu Anugrayani Bustamin, S.T., M.T. selaku pembimbing penulis yang selalu membimbing, menyediakan waktu, tenaga, pikiran, dan perhatian yang luar biasa untuk mengarahkan penulis dalam penyusunan tugas akhir ini.
4. Segenap Dosen dan Staf Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah banyak membantu selama perkuliahan hingga penyelesaian tugas akhir ini.
5. Ica, Echa dan Nabila yang telah memberikan semangat kepada penulis dalam menyelesaikan tugas akhir ini.
6. Teman - Teman Rezolver 20 atas dukungan, bantuan, dan semangat yang telah diberikan selama ini.
7. Serta seluruh pihak yang tak sempat penulis sebutkan satu per satu, tanpa sadar telah menjadi motivasi penulis dalam menyelesaikan tugas akhir.

Akhirnya dengan segala kerendahan hati, penulis menyadari bahwa tugas akhir ini masih terdapat kekurangan dan jauh dari kata sempurna baik dari isi maupun cara penyajian. Oleh karena itu penulis mengharapkan adanya saran maupun kritik yang bersifat membangun demi kesempurnaan tugas akhir ini. Semoga tugas akhir ini dapat memberikan manfaat bagi pembaca terlebih khusus bagi penulis.

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Penulis,

Annisa Darwis

ABSTRAK

ANNISA DARWIS**. Prediksi Keberlanjutan Rawat Inap Pasien Rumah Sakit Menggunakan XLNet-BiGRU-ATT pada Catatan Khusus Klinis Pasien** (Dibimbing oleh Anugrayani Bustamin, S.T., M.T.)

Latar Belakang. Penerimaan kembali pasien rumah sakit dalam jangka waktu singkat setelah keluar merupakan salah satu indikator penting dalam menilai kualitas perawatan kesehatan. Prediksi penerimaan kembali pasien berdasarkan catatan medis yang dihasilkan selama perawatan sebelumnya dapat membantu penyedia layanan kesehatan dalam mencegah kejadian ini. **Metode**. Penelitian ini mengembangkan model prediksi menggunakan arsitektur kombinasi XLNet, BiGRU (*Bidirectional Gated Recurrent Unit*), dan *attention mechanism*. Arsitektur ini memanfaatkan representasi konteks dari XLNet, kemampuan BiGRU dalam menangkap informasi sekuensial dua arah, serta *attention mechanism* untuk fokus pada bagian catatan medis yang relevan. *Hyperparameter* yang digunakan termasuk *learning rate* 2e-5, *batch size* 32, dan *max sequence length* 512. **Hasil**. Hasil terbaik diperoleh pada *epoch* ke-4, dengan nilai ROC-AUC, PR-AUC, dan PR80 yang mencapai 0.742, 0.723 dan 0.237. **Kesimpulan**. Arsitektur yang dikembangkan mampu memprediksi penerimaan kembali pasien berdasarkan data teks klinis yang kompleks. Model ini berpotensi digunakan sebagai alat bantu bagi penyedia layanan kesehatan dalam mengidentifikasi pasien dengan risiko tinggi untuk kembali dirawat, sehingga langkah pencegahan yang tepat dapat diambil.

Kata kunci: Penerimaan kembali pasien, XLNet, BiGRU, Mekanisme *Attention*, Prediksi, Catatan klinis.

ABSTRACT

ANNISA DARWIS**. *Predicting Hospital Readmissions Using XLNet-BiGRU-ATT on Patient Clinical Notes*** (*supervised by* Anugrayani Bustamin, S.T., M.T.)

***Background**. Hospital readmission within a short period after discharge is a key indicator of healthcare quality. Predicting patient readmission based on medical records generated during previous care can assist healthcare providers in preventing this event. **Methods**. This study developed a predictive model using a combination of XLNet, BiGRU (Bidirectional Gated Recurrent Unit), and attention mechanism architecture. This architecture leverages contextual representations from XLNet, the ability of BiGRU to capture bidirectional sequential information, and the attention mechanism to focus on relevant parts of the medical records. The hyperparameters used include a learning rate of 2e-5, a batch size of 32, and a max sequence length of 512. **Results**. The best results were obtained in the 4th epoch, with ROC-AUC, PR-AUC, and PR80 scores of 0.742, 0.723, and 0.237, respectively. **Conclusion**. The developed architecture successfully predicts hospital readmissions based on complex clinical text data. This model has the potential to be used as a tool to help healthcare providers identify patients at high risk of readmission, enabling appropriate preventive measures to be taken*.

***Keywords:** Patient readmission, XLNet, BiGRU, Attention mechanism, Prediction, Clinical records.*

DAFTAR ISI

Halaman

DAFTAR TABEL

Nomor urut Halaman

DAFTAR GAMBAR

Nomor urut Halaman

DAFTAR LAMPIRAN

Nomor urut Halaman

DAFTAR SINGKATAN, ISTILAH, DAN LAMBANG

Lambang/singkatan	Arti dan Penjelasan
NLP	*Natural Language Processing*
XLNet	*eXtreme Language Understanding NETwork*
BiGRU	*Bidirectional Gated Recurrent Unit*
ATT	*Attention Mechanism*
SOAP	*Subjective, Objective, Assessment, Plan*
MIMIC-III	*Medical Information Mart for Intensive Care III*
EHR	*Electronic Medical Record*
ICU	*Intensive Care Unit*
$P(X)$	*Permutation Languge Model*
$g_{zt}^{(m)}$	*Content Stream*
$h_{zt}^{(m)}$	*Query Stream*
h_t	*Recurrence mechanism*
$\odot$	*Hadamard Product*
x^t	*Current Node Input*
y^t	*Current Node Output*
h^t	*Hidden State*
σ	*Sigmoid*
r	*control reset gate*
z	*control update gate*
h_i	Representasi Kontekstual *Self-attention*
x	Urutan Kata dalam Bentuk Vektor
y	Label Sebenarnya
$\hat{y}$	Logist yang Dihasilkan Model
$\overrightarrow{h_t}$	*Forward Representation*
$\overleftarrow{h_t}$	*Backward Representation*
a_t	*Attention Score*
t	*Sequence Length*
T	Jumlah keseluruhan token dalam catatan medis
c	*Context Vector*
w	*Fully Connected Layer Weight*
$tanh$	*Hyperbolic tangen*
ROC	*Receiver Operating Characteristic*
AUC	*Area Under the Curve*
PR	*Precision-Recall*
TPR	*True Positive Rate*
FPR	*False Positive Rate*
TP	*True Positives*
FN	*False Negatives*
FP	*False Positives*

TN	*True Negatives*
PR80	*Recall at Precision 80*

BAB I
PENDAHULUAN

1.1 Latar Belakang

Intensive Care Unit (ICU) adalah ruang rawat di rumah sakit yang dilengkapi dengan staf dan peralatan khusus untuk merawat dan mengobati pasien dengan perubahan fisiologi yang cepat memburuk dan mempunyai intensitas defek fisiologi yang merupakan keadaan kritis dan dapat menyebabkan kematian. Perawatan intensif yang diberikan kepada setiap pasien kritis tersebut berkaitan erat dengan tindakan-tindakan yang memerlukan pencatatan medis yang berkesinambungan dan monitoring untuk memantau secara cepat perubahan fisiologis yang terjadi atau akibat dari penurunan fungsi organ-organ tubuh lainnya (Murwani, A., 2008).

Penerimaan kembali ke *Unit Perawatan Intensif* (ICU) di rumah sakit dapat menyebabkan beban yang tinggi pada sistem perawatan kesehatan, dengan efek sosial ekonomi pada pasien, keluarga, dan praktisi kesehatan. Penerimaan kembali ICU yang dini dan tidak direncanakan, dikaitkan dengan peningkatan risiko mortalitas, morbiditas, tinggal lebih lama di rumah sakit dan ICU, dan peningkatan biaya. Penelitian saat ini telah menunjukkan bahwa tingkat penerimaan kembali ICU dipengaruhi oleh faktor-faktor selain kualitas perawatan, seperti karakteristik pasien dan lama tinggal, dan secara umum semua sumber data yang memungkinkan (González-Nóvoa et al., 2023).

Peningkatan penggunaan teknologi dalam bidang kesehatan telah memberikan landasan yang kuat bagi pengembangan sistem yang mampu memproses dan menganalisis data medis dengan cepat dan efisien. *Natural Language Processing* (NLP) adalah bidang ilmu yang berfokus pada pengolahan bahasa alami oleh komputer. Dalam konteks pemulangan pasien, NLP dapat digunakan untuk menganalisis catatan medis dan dokumen terkait untuk mengidentifikasi faktor-faktor yang mungkin berkontribusi terhadap penerimaan kembali pasien.

Dibandingkan dengan gambaran terstruktur, catatan klinis memberikan gambaran pasien terkait gejala, alasan diagnosis, hasil radiologi, aktivitas sehari-hari, dan riwayat pasien. Dalam membuat prediksi klinis yang akurat diperlukan pembacaan data catatan klinis dalam jumlah besar, sementara dokter yang bekerja di unit perawatan intensif, perlu mengambil keputusan dalam waktu terbatas. Hal ini bisa saja menambah beban kerja dokter, sehingga sistem prediksi berdasarkan catatan klinis dapat berguna dalam praktiknya (Huang et al., 2020).

Di Indonesia rekam medis pasien mulai beralih menjadi berbasis elektronik dengan diterbitkannya Peraturan Menteri Kesehatan (PMK) nomor 24 tahun 2022 tentang rekam medis. Dimana fasilitas pelayanan kesehatan (fasyankes) diwajibkan menjalankan sistem pencatatan riwayat medis pasien secara elektronik. Berdasarkan aturan ini dituliskan bahwa Isi rekam medis paling sedikit terdiri atas identitas pasien, hasil pemeriksaan fisik dan penunjang, diagnosis, pengobatan, dan

rencana tindak lanjut pelayanan kesehatan, dan nama serta tanda tangan tenaga kesehatan pemberi pelayanan kesehatan.

Dikutip dari artikel infokes.co.id (Infokes Indonesia) yang terintegrasi dengan standar dan regulasi Kementerian Kesehatan Republik Indonesia (Kemenkes), format pencatatan rekam medis yang sering digunakan adalah format SOAP (*Subjective, Objective, Assessment, Plan*). Dalam SOAP, *Subjective* mencakup informasi yang dilaporkan oleh pasien atau keluarga mengenai keluhan atau gejala yang dialami serta riwayat penyakit pasien. *Objective* mencakup data yang diukur atau ditemukan secara objektif oleh tenaga medis. Contoh data yang dicatat di bagian ini termasuk hasil pemeriksaan fisik, laboratorium, atau hasil pemeriksaan penunjang lainnya. *Assessment* mencakup diagnosis, prognosis, dan rencana perawatan. *Plan* mencakup rencana tindakan yang akan dilakukan untuk mengatasi masalah kesehatan pasien seperti pemberian obat, terapi, tindakan medis, atau edukasi kepada pasien (*Pencatatan SOAP Rekam Medis Dan Contoh Penggunaannya - eClinic*, 2023).

Catatan SOAP adalah cara bagi petugas kesehatan untuk mendokumentasikan secara terstruktur dan terorganisir (Gogineni et al., 2019). Catatan SOAP membantu memandu petugas layanan kesehatan menggunakan data klinis untuk menilai, mendiagnosis, dan merawat pasien berdasarkan informasi yang mereka berikan. Catatan SOAP adalah informasi penting tentang status kesehatan pasien serta dokumen komunikasi antar profesional kesehatan (Podder et al., 2024).

Oleh karena itu, dataset yang akan digunakan dalam penelitian ini adalah MIMIC-III (*Medical Information Mart for Intensive Care III*) yang merupakan proyek database dari *Institutional Review Boards* dari Beth Israel Deaconess *Medical Center* (Boston, MA) dan *Massachusetts Institute of Technology* (Cambridge, MA). MIMIC III adalah database besar berisi catatan kesehatan elektronik (EHR) dari pasien ICU, yang mencakup informasi seperti diagnosis utama, prosedur yang dilakukan, dan rencana tindak lanjut, catatan terkait dengan pengelolaan obat-obatan, laporan konsultasi dari spesialis atau dokter, catatan mengenai arahan perawatan lanjutan, hasil laboratorium, dan lain-lain (Johnson et al., 2015), yang relevan dengan data rekam medis di Indonesia.

Asumsi *bag-of-words* telah digunakan untuk memodelkan teks klinis, selain model penyematan kata *log-bilinear* seperti *Word2Vec*. Namun catatan klinis yang panjang dan kata-kata yang saling bergantung, membuat metode ini tidak dapat menangkap ketergantungan jangka panjang yang diperlukan untuk menangkap makna klinis (Huang et al., 2020).

XLNet (*eXtreme Language Understanding NETwork*) – BiGRU (*Bidirectional Gated Recurrent Unit*) – ATT (*Attention Mechanism)* adalah salah satu jenis model dalam bidang *Natural Language Processing* atau NLP yang digunakan untuk berbagai tugas, seperti pemahaman bahasa alami, terjemahan mesin, dan lain-lain. Model ini menggabungkan beberapa teknik dan arsitektur yang efektif dalam penanganan teks. XLNet adalah model untuk *semantic understanding pre-training.* Model ini dibangun berdasarkan model sebelumnya seperti *mask language models* dan model bahasa *autoregresif* (AR), dan mengatasi tantangan spesifik dalam tahap

pre-training BERT. Salah satu keunggulan utama XLNet adalah kemampuannya menangani ketidakkonsistenan antara *mask flag* dan proses penyesuaian (Han et al., 2023).

Di dalam XLNet-BiGRU-ATT, ada tambahan lapisan-lapisan lain yang meningkatkan kemampuan model. Pertama, ada lapisan BiGRU, yang merupakan jenis arsitektur *recurrent neural network* (RNN) yang memungkinkan model untuk memproses data sekuensial dalam dua arah sekaligus: maju dan mundur. Ini membantu model dalam menangkap konteks yang lebih kaya dan bergantung pada sejarah lebih luas dalam teks. Dibandingkan dengan LSTM, GRU memiliki struktur yang lebih sederhana dengan performa serupa, dan *bidirectional* GRU (BiGRU) dapat menangkap informasi konteks lebih baik daripada RNN *unidirectional* (Han et al., 2023).

Kemudian, ada juga lapisan *attention* (perhatian) yang memberikan bobot pada bagian-bagian penting dari input. *Attention mechanism* memungkinkan model untuk fokus pada bagian-bagian penting yang paling relevan dari input saat membuat prediksi, yang sangat berguna dalam menangani data teks klinis yang panjang.
Gabungan dari XLNet sebagai *pre-trained language model*, BiGRU untuk *bidirectional sequential processing*, dan *attention layer* untuk memberikan penekanan pada informasi penting dalam teks, menjadikan XLNet-BiGRU-ATT sebagai model yang sangat kuat dalam penanganan berbagai tugas pemrosesan bahasa alami.

Dengan melihat permasalahan diatas, penulis mengusulkan judul “Prediksi Keberlanjutan Rawat Inap Pasien Rumah Sakit Menggunakan XLNet-BiGRU-ATT pada Catatan Khusus Klinis Pasien” untuk membangun model prediksi penerimaan kembali pasien rumah sakit dengan memanfaatkan rekam medis untuk membantu pertimbangan dokter atau petugas medis yang bekerja di unit perawatan intensif, yang perlu mengambil keputusan berdasarkan catatan klinis dalam jumlah besar, sehingga dapat menghemat waktu dan tenaga dokter.

1.2 Landasan Teori

1.2.1 Catatan Klinis

Catatan klinis atau rekam medis adalah berkas berisi catatan dan dokumen tentang pasien yang meliputi identitas, pemeriksaan, pengobatan, tindakan medis lain pada sarana pelayanan kesehatan untuk rawat jalan dan rawat inap baik dikelola pemerintah maupun swasta (permenkes nomor 209 / MENKES / PER / III / 2008). Rekam medis juga didefinisikan sebagai segala bentuk identifikasi dan dokumentasi pasien, baik dalam bentuk tertulis dan grafis. Sejarah rekam medis telah berlangsung selama ribuan tahun, dengan akar sejarah yang lebih awal dalam peradaban kuno. Hingga abad ke-19, rekam medis terutama digunakan untuk tujuan pendidikan, kemudian digunakan untuk tujuan lain seperti dalam prosedur asuransi atau hukum (Lorkowski & Pokorski, 2022).

Rekam medis tidak memiliki formalisasi selama ribuan tahun, dan pendekatan sistematis modern merupakan pencapaian 100–150 tahun terakhir.

Dalam aspek kualitas catatan, sering kali naskah medis yang membutuhkan banyak waktu untuk dibuat kemudian dianggap tidak berguna karena kurangnya keterbacaan. Banyak penulis mengatakan bahwa kualitas rekam medis tidak dapat dinilai ketika kasus dan pasien yang dijelaskan bersifat anonim. Oleh karena itu, kemampuan memilih dan menganalisis data dari catatan historis sangat penting. Riwayat penyakit yang tercatat dianggap sebagai alat pendidikan yang berharga, asalkan data dan deskripsinya seimbang dan saling melengkapi (Lorkowski & Pokorski, 2022).

1.2.2 *Electronic Medical Record*

Rekam medis mengalami perubahan revolusioner dari format berbasis kertas ke format elektronik, yang mencerminkan perkembangan sistem *eHealth*. Proses migrasi ke catatan kesehatan elektronik melibatkan penggunaan algoritma kecerdasan buatan atau *artificial intelligence* (AI) yang menyederhanakan layanan medis dengan menggunakan metode kerja yang lebih cepat dan lebih sederhana. AI menguntungkan pasien dan penyedia layanan karena meningkatkan manajemen dan komunikasi pasien di antara pusat-pusat medis, menghemat sumber daya, mengidentifikasi kontaminasi atau infeksi, dan membatasi biaya kesehatan (Lorkowski & Pokorski, 2022).

Torkman et al., (2024) dalam studi eksplorasinya mengenai rekam medis elektronik menyatakan bahwa *electronic medical record* (EMR) adalah salah satu aplikasi teknologi informasi kesehatan terpenting dalam perawatan kesehatan. EMR adalah sistem komputerisasi yang menyediakan pendekatan efisien untuk mengumpulkan, menyimpan, dan menampilkan informasi terkait kesehatan. EMR dikembangkan untuk mendukung tugas-tugas profesional perawatan kesehatan dalam praktik medis. Fungsi utama sistem ini meliputi tetapi tidak terbatas pada penyediaan catatan konsultasi medis, informasi tentang vaksinasi, rincian masalah kesehatan, resep, dan pembaruan, pengingat otomatis untuk berbagai tugas, dan peringatan janji temu medis. Meskipun manfaatnya sudah diakui, penerapan rekam medis elektronik di kalangan profesional kesehatan masih belum konsisten. Berbagai faktor memengaruhi keberhasilan adopsi dan pemanfaatan. Faktor organisasi seperti dukungan kepemimpinan, pelatihan yang memadai, dan infrastruktur teknis sangatlah penting. Misalnya, penelitian telah menunjukkan bahwa dukungan manajemen puncak yang kuat secara signifikan meningkatkan kemungkinan keberhasilan implementasi EMR, karena hal ini mendorong lingkungan yang mendukung bagi para profesional kesehatan untuk mengadopsi teknologi baru. Selain itu, keberadaan sistem pendukung TI di dalam rumah sakit sangat penting dalam mengatasi tantangan teknis dan mengurangi resistensi terhadap adopsi EMR. Kompleksitas dan kegunaan EMR yang dirasakan juga memainkan peran penting dalam penerimaan mereka. Profesional perawatan kesehatan lebih cenderung mengadopsi EMR jika sistemnya ramah pengguna dan selaras dengan alur kerja klinis mereka. Faktor-faktor seperti kemudahan penggunaan, waktu respons, dan kegunaan keseluruhan sistem EMR adalah yang terpenting. Sistem yang dianggap

rumit atau mengganggu pola kerja cenderung tidak diterima oleh penyedia layanan kesehatan. Oleh karena itu, memastikan bahwa EMR dirancang dengan masukan pengguna akhir dan dapat beradaptasi dengan lingkungan klinis dapat secara signifikan memengaruhi penerimaan dan pemanfaatannya.

Selain itu, atribut individu seperti literasi komputer, sikap terhadap teknologi, dan pengalaman sebelumnya dengan memengaruhi tingkat adopsi. Profesional perawatan kesehatan yang lebih akrab dengan sistem komputer dan yang memiliki sikap positif terhadap teknologi lebih cenderung menggunakan EMR secara efektif. Program pelatihan yang meningkatkan literasi komputer dan membiasakan staf dengan fungsi EMR dapat mengurangi resistensi dan mempromosikan sikap positif terhadap EMR. Studi telah menyoroti pentingnya pengembangan dan pelatihan profesional berkelanjutan dalam memastikan keberhasilan jangka panjang sistem EMR dalam pengaturan perawatan kesehatan.

Ada berbagai faktor yang dapat memengaruhi penggunaan EMR di sektor perawatan kesehatan. Misalnya, faktor-faktor berikut ditemukan menjadi pendorong niat perilaku mengenai penggunaan EMR: harapan kinerja, pengaruh sosial, dan kebiasaan menemukan bahwa fungsionalitas EMR dapat diperkuat dengan menambahkan fitur tambahan seperti kamus fasilitas penelitian dan entitas medis. Selain itu, ditemukan bahwa faktor individu seperti literasi komputer, norma pribadi, dan pengalaman individu dapat memainkan peran penting dalam penggunaan EMR oleh profesional perawatan kesehatan. Seperti halnya sistem informasi lainnya, keberhasilan implementasi EMR bergantung pada penggunaan pengguna.

Torkman et al. (2024) juga menjelaskan bahwa kesulitan yang dirasakan dalam penerapan EMR, seperti kompleksitas sistem dan dokumentasi yang memakan waktu, merupakan hambatan yang signifikan. Masalah kompatibilitas perangkat keras/perangkat lunak juga menjadi perhatian, karena beberapa sistem tidak terintegrasi dengan baik dengan alur kerja yang ada. Ketidakpastian kinerja pekerjaan, termasuk ketakutan kehilangan otonomi dan penghapusan data, semakin memengaruhi penerimaan. Kemudahan pengoperasian dan risiko yang dirasakan merupakan faktor penting, dengan pengguna menunjukkan bahwa sistem yang ramah pengguna lebih mungkin diadopsi. Dukungan sosial, seperti dukungan kolega dan *supervisor*, sangat penting untuk mengatasi tantangan adopsi. Kepercayaan pengguna dan dukungan organisasi, termasuk program pengembangan profesional, sangat penting untuk implementasi yang sukses. Akhirnya, dukungan teknologi, seperti bantuan dan pelatihan TI yang tepat waktu, diperlukan untuk mengatasi tantangan teknis.

Isu utama yang harus di atasi dalam implementasi EMR, yaitu: (1) Kebutuhan terhadap standar data di bidang terminologi klinik, (2) Aspek *privacy*, kerahasiaan dan keamanan data, (3) Pelaksanaan entri data oleh dokter dan tenaga medis lainnya, (4) Kesulitan integrasi sistem rekam medis dengan sumber informasi lain dalam pelayanan kesehatan (Berg, 2004).

Manfaat teknologi informasi dalam rekam medis elektronik, selain untuk efisiensi pencatatan dan pengolahan data, serta menyediakan informasi yang lebih akurat dan terpercaya, yaitu memiliki tujuan untuk mengurangi *medical error* dan

meningkatkan keamanan pasien (*patient safety*). Dengan adanya sistem aplikasi manajemen rekam medis, maka *medical error* dalam pengambilan keputusan oleh tenaga kesehatan dapat dikurangi, karena setiap pengambilan keputusan akan berdasarkan data rekam medis pasien yang telah ada dan sudah terintegrasi dengan unit pelayanan lainnya (Moody, L.E, et.al. 2004).

1.2.3 ICU *(Intensive Care Unit)*

Intensive care unit (ICU) adalah ruang rawat di rumah sakit yang dilengkapi dengan staf dan peralatan khusus untuk merawat dan mengobati pasien dengan perubahan fisiologi yang cepat memburuk yang mempunyai intensitas defek fisiologi satu organ ataupun mempengaruhi organ lainnya sehingga merupakan keadaan kritis yang dapat menyebabkan kematian. Perawatan intensif yang diberikan kepada setiap pasien kritis tersebut berkaitan erat dengan tindakan-tindakan yang memerlukan pencatatan medis yang berkesinambungan dan monitoring untuk memantau secara cepat perubahan fisiologis yang terjadi atau akibat dari penurunan fungsi organ-organ tubuh lainnya (Murwani, A., 2008).

ICU memainkan peran penting dalam sistem perawatan kesehatan dengan menyediakan perawatan khusus bagi pasien dengan kondisi yang parah atau mengancam jiwa. Salah satu aspek penting dalam mengelola sumber daya ICU dan memastikan perawatan pasien yang optimal adalah memperkirakan lamanya waktu perawatan di ICU (Rhazzafe et al., 2024).

ICU merupakan suatu bagian rumah sakit yang dilengkapi dengan staf khusus dan perlengkapan yang khusus yang ditujukan untuk observasi, perawatan dan terapi pasien-pasien yang menderita penyakit atau cedera yang mengancam jiwa atau potensial mengancam jiwa yang diharapkan masih dapat *reversible*. Umumnya pasien yang dirawat di ICU berada dalam keadaan tertentu, misalnya pasien dengan penyakit kritis yang menderita kegagalan satu atau lebih dari sistem organnya (Tabrani, 2007).

1.2.4 Prediksi Penerimaan Kembali Pasien

Prediksi penerimaan kembali pasien di rumah sakit dan lamanya rawat inap memberikan informasi tentang cara mengelola kapasitas tempat tidur rumah sakit dan jumlah staf yang dibutuhkan. Penerimaan kembali rumah sakit memainkan peran utama dalam pengeluaran rumah sakit. Baru-baru ini, fokus utama sistem perawatan kesehatan adalah pada pasien yang diterima kembali ke rumah sakit dalam jangka waktu yang singkat (sebagian besar pada penerimaan kembali yang terjadi dalam 30 hari) setelah keluar dari rumah sakit. Menurut laporan terbaru, beban sistem perawatan kesehatan Amerika Serikat adalah 41 miliar dolar, karena penerimaan kembali pasien diabetes di rumah sakit dalam waktu 30 hari. Sebuah studi di Spanyol mengungkapkan bahwa, sementara total biaya tahunan pasien diabetes adalah EUR 1803,6 per orang, biaya rawat inap untuk pasien ini adalah EUR 801,6. Studi lain yang dilakukan di Amerika Serikat menunjukkan bahwa biaya tahunan langsung

diabetes adalah sekitar USD 9595 per orang. Ada biaya langsung dan tidak langsung dari sistem perawatan kesehatan yang terkait dengan rawat inap pasien rawat inap. Biaya medis langsung mencakup biaya yang terkait dengan layanan yang diberikan di rumah sakit, seperti rawat inap, rawat inap ICU, tes laboratorium, dan jenis kunjungan rumah sakit lainnya (Tavakolian et al., 2023)

Penerimaan kembali ke ICU selama penerimaan di rumah sakit yang sama merupakan kejadian buruk yang jarang terjadi dan dapat menyebabkan beban yang tinggi pada sistem perawatan kesehatan, dengan efek sosial ekonomi yang sangat penting pada pasien, keluarga, dan praktisi kesehatan. Penerimaan kembali ICU yang dini dan tidak direncanakan, dengan tingkat penerimaan kembali berkisar antara 1,3% hingga 13,7%, dikaitkan dengan peningkatan risiko mortalitas, morbiditas, tinggal lebih lama di rumah sakit dan ICU, dan peningkatan biaya. Akibatnya, ada minat yang tinggi pada tingkat penerimaan kembali ICU sebagai indikator kualitas perawatan kritis. Namun demikian, penelitian saat ini telah menunjukkan bahwa tingkat penerimaan kembali ICU dipengaruhi oleh faktor-faktor selain kualitas perawatan, seperti karakteristik pasien dan lama tinggal, dan secara umum semua sumber data yang memungkinkan. Ini membuka masalah untuk penggunaan teknik kecerdasan buatan baru untuk mengeksploitasi semua informasi yang tersedia (González-Nóvoa et al., 2023)

Dalam beberapa tahun terakhir, penggunaan teknik pembelajaran mesin di bidang kesehatan telah meningkat untuk meningkatkan kualitas perawatan pasien dan untuk memfasilitasi pekerjaan tenaga kesehatan. Karena besarnya jumlah data yang dihasilkan di rumah sakit saat ini, sangat menarik untuk mengembangkan teknik untuk menganalisis data ini secara otomatis dan efisien, memfasilitasi pengambilan keputusan yang benar oleh tenaga kesehatan. Analisis manual dari semua data ini akan memerlukan waktu yang tidak tersedia dalam kerangka kerja sehari-hari rumah sakit, yang menyebabkan hanya sebagian kecil darinya yang dianalisis, kehilangan kesempatan untuk menganalisis data yang tersedia secara global. Pemantauan pasien yang berkelanjutan dan menyeluruh selama tinggal di ICU menghasilkan berbagai macam data biomedis dengan potensi besar untuk aplikasi (González-Nóvoa et al., 2023).

1.2.5 *Text Classification*

Text classification telah menjadi salah satu tugas fundamental dalam bidang *natural language processing* (NLP) dengan berbagai aplikasi penting dalam dunia nyata. Menurut Minaee et al. (2021) klasifikasi teks adalah proses mengategorikan dokumen teks ke dalam satu atau lebih kelas atau kategori yang telah ditentukan sebelumnya. Tugas ini memiliki berbagai aplikasi praktis, termasuk analisis sentimen, deteksi spam, kategorisasi berita, dan pengelompokan topik. Dalam beberapa tahun terakhir, kemajuan signifikan telah dicapai dalam bidang ini, terutama dengan munculnya teknik *deep learning*.

Metode *deep learning* telah mengungguli pendekatan *machine learning* tradisional dalam banyak *benchmark text classification*. Arsitektur *neural network*

seperti *convolutional neural networks* (CNN), *recurrent neural networks* (RNN), dan lebih baru lagi, model berbasis transformer seperti BERT dan XLNet, telah menunjukkan performa yang luar biasa dalam berbagai tugas klasifikasi teks. Keunggulan utama dari metode *deep learning* adalah kemampuannya untuk secara otomatis mempelajari representasi fitur yang relevan dari data teks mentah, menghilangkan kebutuhan akan ekstraksi fitur manual yang rumit.

Namun, tantangan masih tetap ada dalam domain ini. Salah satu isu utama adalah kebutuhan akan dataset berlabel yang besar untuk melatih model deep learning yang kompleks. Selain itu, interpretabilitas model *deep learning* seringkali lebih sulit dibandingkan dengan metode klasik seperti *naive bayes* atau *decision trees*. Terlepas dari tantangan ini, *text classification* terus menjadi area penelitian yang aktif dengan potensi besar untuk inovasi dan aplikasi praktis di masa depan (Minaee et al., 2021)

1.2.6 *Natural Language Processing*

Model *natural language processing* (NLP) adalah disiplin ilmu komputer yang bertujuan untuk memahami konsep dan maksud dari bahasa manusia. Sementara manusia cukup mahir memahami sintaks linguistik dan tata bahasa serta hubungan spasial tersirat, komputer memiliki kesulitan besar pengolahan *query* bahasa alami (Allen, 1995).

Teknologi *machine learning* dalam *natural language processing* (NLP) memberi komputer kemampuan untuk menginterpretasikan, memanipulasi, dan memahami bahasa manusia. Integrasi NLP dalam *software requirements engineering* (SRE) menawarkan solusi yang lebih efisien, meskipun semi-otomatis, untuk mendeteksi ambiguitas. Integrasi NLP dan AI ke dalam rekayasa perangkat lunak mewakili pergeseran paradigma yang signifikan, yang menjanjikan tidak hanya untuk mengotomatisasi tetapi juga untuk meningkatkan akurasi dan efektivitas proses ini. NLP, komponen penting AI, menjembatani kesenjangan antara bahasa manusia dan mesin, memungkinkan mesin untuk memproses, menganalisis, dan memahami data bahasa manusia, dan untuk menghasilkan respons seperti manusia. Integrasi ini sangat berdampak dimana NLP yang dikombinasikan dengan AI dapat meningkatkan kualitas persyaratan perangkat lunak dengan menganalisis atribut leksikal dan sintaksisnya, mengotomatiskan pemetaan ke representasi formal, dan menyempurnakan spesifikasi dengan mengekstraksi pengetahuan domain yang relevan. (Necula et al., 2024)

Necula et al. (2024), memetakan tren dalam NLP untuk bidang SRE dalam periode 1991–2023. Analisis ini mengungkapkan beberapa wawasan yang dikelompokkan berdasarkan periode waktu:

1. Periode 1: Hingga tahun 2000. Pada periode menjelang tahun 2000, "model matematika" menjadi titik fokus, yang menunjukkan tahap dasar pemodelan perangkat lunak, di mana model abstrak dan landasan teoritis sangat penting. Selama masa ini, landasan untuk kemajuan masa depan dalam rekayasa

perangkat lunak dan NLP sedang diinisialisasi, meskipun data menunjukkan tahap yang lebih baru untuk topik terkait NLP.

2. Periode 2: 2001–2010. Dari tahun 2001 hingga 2010, bidang ini mulai berkembang, dengan "linguistik komputasional" dan "pemrograman berorientasi objek" yang mendapat perhatian penting. Periode ini menandai pergeseran dari teori dasar ke metodologi yang lebih terapan dalam pengembangan perangkat lunak. Pertumbuhan "linguistik komputasional" selama dekade ini mengarah pada peningkatan minat dalam mengintegrasikan teknik pemrosesan bahasa ke dalam pengembangan perangkat lunak, khususnya di bidang seperti "*requirements analysis*" dan "pemilihan dan evaluasi perangkat lunak komputer".
3. Periode 3: 2011–2021. Dekade ini menyaksikan pertumbuhan signifikan dalam aplikasi NLP dalam SRE. Sistem pemrosesan bahasa alami, *software requirements*, dan *engineering requirements* mengalami peningkatan fokus yang substansial, yang menggambarkan dorongan industri untuk memanfaatkan NLP guna pengumpulan, analisis, dan pengelolaan yang lebih efektif. Munculnya *deep learning* menjelang akhir periode ini menandakan pergeseran ke arah teknik AI yang lebih canggih untuk mengatasi masalah kompleks dalam rekayasa perangkat lunak.
4. Periode 4: 2022–2023. Dalam beberapa tahun terakhir, terdapat fokus yang jelas pada teknologi AI mutakhir, dengan *deep learning, machine learning*, dan *learning systems* menjadi pusat perhatian. Tren ini mencerminkan kemajuan pesat yang terjadi dalam AI dan peningkatan penerapannya dalam pemodelan perangkat lunak dan *engineering requirements*. Penyebutan *language model* dan *requirement classification* selama periode ini menyoroti upaya berkelanjutan untuk menyempurnakan aplikasi NLP dalam pengembangan perangkat lunak, menjadikannya lebih akurat dan efisien.

Lebih jauh, Necula et al., (2024) mendalami evolusi tematik dalam ranah NLP dengan memberi gambaran tentang pergeseran signifikan dan kontinuitas fokus dalam bidang tersebut selama periode waktu yang berbeda. Dengan merangkum transisi dalam konsentrasi tematik dari satu era ke era lain, yang disorot oleh berbagai metrik seperti indeks inklusi tertimbang, indeks inklusi, jumlah kejadian, dan indeks stabilitas. Metrik ini menawarkan wawasan kuantitatif tentang tren yang berkembang dan pentingnya tema tertentu dari waktu ke waktu:

1. *Natural language* ke sistem NLP (1991–2000 ke 2001–2010): Transisi dari *natural language* ke *natural language processing system* dengan skor sempurna (1,00) pada indeks inklusi tertimbang dan indeks inklusi menunjukkan pergeseran signifikan ke arah operasionalisasi pemahaman bahasa alami dalam sistem komputasi, yang mencerminkan gerakan penting dalam penelitian NLP dengan hanya dua kejadian tetapi stabilitas tinggi.
2. Semantik dalam sistem NLP (2001–2010 hingga 2011–2020): Fokus pada “semantik” selama periode dengan skor maksimum ini menunjukkan minat yang mendalam dalam memahami dan memproses makna dalam bahasa, yang penting untuk aplikasi NLP yang efektif.

3. Sintaksis ke spesifikasi formal (2001–2010 ke 2011–2020): Pergeseran dari sintaksis ke spesifikasi formal dengan skor tertinggi menunjukkan minat yang semakin matang dalam mengintegrasikan analisis sintaksis ke dalam konteks yang lebih terstruktur dan formal, yang berpotensi memengaruhi area seperti spesifikasi dan desain perangkat lunak. Pergeseran ini dan yang sebelumnya menggarisbawahi pentingnya pemahaman makna dan struktur *natural language requirements* dalam literatur.
4. Dari otomatisasi ke ekstraksi (2001–2010 ke 2011–2020): Peralihan dari otomatisasi ke ekstraksi menunjukkan pergeseran fokus ke arah ekstraksi konsep berharga dari *software requirements*, yang penting untuk mengotomatisasi proses pengembangan *software,* meskipun indeks menunjukkan penekanan dan stabilitas sedang.
5. Kontinuitas ekstraksi (2011–2021 hingga 2022–2023): Fokus berkelanjutan pada ekstraksi dengan indeks inklusi tertimbang yang layak tetapi indeks inklusi yang lebih rendah menunjukkan minat yang berkelanjutan tetapi sedikit berkurang di area ini, yang mungkin mencerminkan kemajuan atau kejenuhan dalam metodologi ekstraksi.
6. Sistem NLP ke *user story* (2011–2021 ke 2021–2023): Transisi ke user story mungkin menunjukkan meningkatnya minat dalam menerapkan NLP untuk memahami dan memproses *requirements* dan narasi pengguna dalam proses pengembangan perangkat lunak yang tangkas.
7. Siklus hidup ke penilaian kualitas dan penambangan data (2011–2021 ke 2021–2023): Transisi ini mencerminkan meningkatnya minat dalam menilai kualitas dan mengekstraksi wawasan dari data di seluruh siklus hidup sistem atau proyek, yang menunjukkan penerapan konsep ini yang lebih luas dalam rekayasa perangkat lunak dan analisis data.
8. *Requirements engineering* ke sistem NLP (1991–2000 ke 2001–2010): Pergeseran ini, dengan indeks inklusi berbobot sedang tetapi indeks inklusi lebih rendah, menunjukkan meningkatnya kepentingan yang meskipun tidak terlalu menonjol seperti topik lainnya, mencerminkan integrasi prinsip-prinsip *engineering requirements* dengan sistem NLP.
9. Rekayasa perangkat lunak berbantuan komputer ke sistem NLP (1991–2000 ke 2001–2010): Transisi signifikan dengan indeks inklusi berbobot tinggi, mengarah pada integrasi alat rekayasa berbantuan komputer dengan NLP untuk meningkatkan proses pengembangan perangkat lunak.

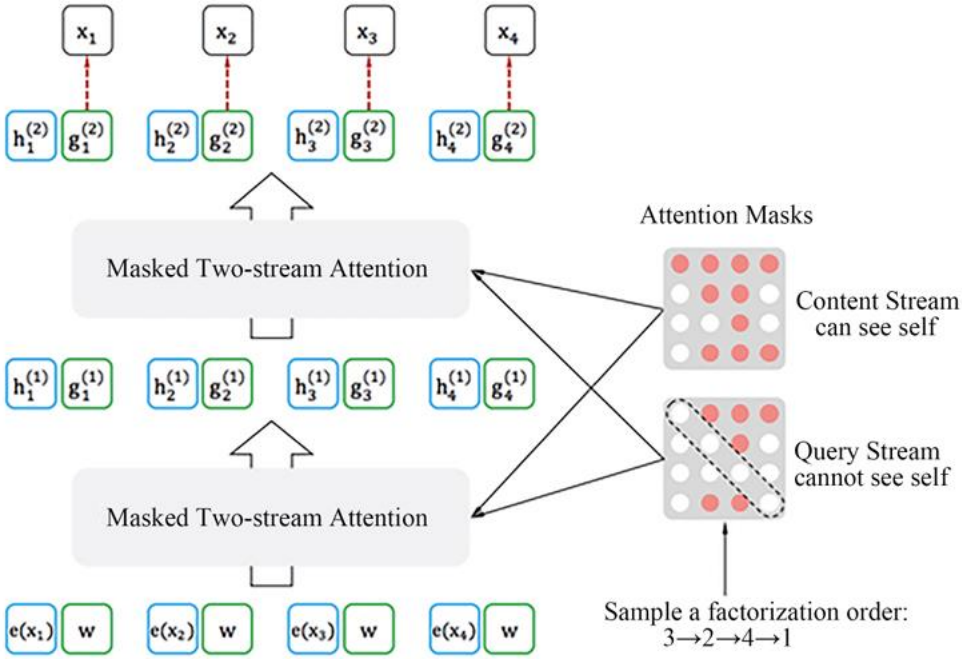
Meskipun penerapan NLP menunjukkan hasil yang menjanjikan, ada beberapa tantangan, termasuk keahlian teknis yang dibutuhkan untuk menggunakan model tersebut dan kebutuhan untuk mempertimbangkan isu etika dan mengatasi potensi bias. Diperlukan pengukuran yang ditetapkan untuk mengevaluasi etika penggunaan ML dan NLP. Oleh karena itu, komunitas akademis harus mengeksplorasi bagaimana teknologi ini dapat mendukung upaya dalam mengatasi tantangan ini untuk menghindari hasil negatif (Sarzaeim et al., 2023)

1.2.7 XLNet (e*Xtreme Language Understanding NETwork*)

XLNet merupakan model bahasa berbasis *transformation,* XLNet memperkenalkan pendekatan pelatihan berbasis permutasi yang memeriksa kata-kata dalam urutan yang berbeda dan dilatih dengan kumpulan data yang lebih besar. Model kasus dasar XLNet terdiri dari *12 transformer layers* dan 12 *attention heads* per-*layer* dengan 768 *hidden size* yang menghasilkan 110 juta parameter. Pendekatan ini memungkinkan XLNet untuk memahami teks dari kedua arah dengan lebih efisien, dan menangkap pola kompleks dalam data teks. Fitur ini berpotensi meningkatkan hasil analisis sentimen dalam skenario kompleks seperti percakapan klinis. XLNet menggunakan metode tokenisasi berbasis permutasi unik yang menyusun ulang kata-kata dalam kalimat dan menggunakan kata-kata terakhir untuk memprediksi kata berikutnya dalam urutan tersebut. (Chatzimina et al., 2024)

XLNet dapat memproses kalimat pendek dan memperoleh informasi semantik dengan menggunakan informasi konteks untuk mencapai representasi teks yang efektif. XLNet merupakan model mutakhir untuk *pre-training* dalam pemahaman semantik. Model ini dibangun berdasarkan model-model sebelumnya seperti *mask language models* dan *autoregressive* (AR) *language models*, model XLNet juga mengatasi tantangan-tantangan khusus dalam tahap *pre-training* BERT. Salah satu keuntungan utama XLNet adalah kemampuannya untuk menangani ketidakkonsistenan antara *masked flag* (penanda bagian yang di-mask dalam data) dan *refinement process* (proses penyempurnaan atau penyelarasan model selama pelatihan). Hal ini dicapai melalui pendekatan unik yang merekonstruksi *teks input* dengan cara permutasi dan komposisi. Hal ini memungkinkan integrasi fitur kontekstual ke dalam proses pelatihan model (Han et al., 2023)

XLnet menggunakan ide *random sorting* untuk memecahkan masalah model AR yang tidak dapat memperkenalkan informasi teks dua arah. Proses *random sorting* hanya mengurutkan nomor urut setiap posisi secara acak. Setelah pengurutan, kata-kata sebelum setiap kata dapat digunakan untuk memprediksi probabilitasnya. Di sini, kata-kata sebelumnya dapat diambil dari kata-kata sebelum kata saat ini dalam kalimat asli atau dari kata-kata setelahnya, yang setara dengan menggunakan informasi urutan dua arah (Han et al., 2023)



Gambar 1. *Two-stream self-attention mechanism structure*
(Sumber: (Han et al., 2023)

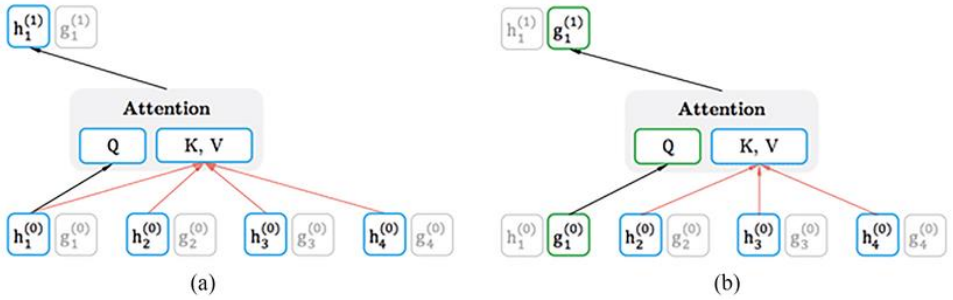
XLNet merupakan model bahasa *autoregresif* yang dapat memperoleh informasi kontekstual dua arah. Untuk memperoleh informasi kontekstual dua arah, XLNet terutama mengadopsi tiga metode, yaitu model bahasa permutasi, *two-stream self-attention*, dan mekanisme sirkulasi. Inti dari XLNet adalah model bahasa permutasi. Untuk mengekstrak informasi kontekstual dua arah, algoritma menggunakan urutan acak dari urutan input awal sambil mempertahankan model satu arah dari model *autoregresif*. XLNet mengintegrasikan model bahasa *autoregresif* dan memperkenalkan dua teknik utama *Transformer-XL* ke dalam XLNet, yaitu skema pengodean posisi relatif dan mekanisme pengulangan segmen (Shi et al., 2022)

Permutation Language Model. XLNet menggunakan model bahasa permutasi untuk mengatasi keterbatasan dari *masked language model* (MLM) yang digunakan di BERT. Dalam *masked language modeling*, beberapa token dalam urutan teks di-*mask* dan kemudian diprediksi oleh model. Namun, pendekatan ini mengganggu aliran alami teks dan memisahkan token yang di-*mask* dari konteks aslinya.

XLNet, sebaliknya, menggunakan permutasi urutan token yang memungkinkan model mempelajari probabilitas dari semua urutan token yang mungkin. Persamaan untuk memprediksi token berdasarkan urutan permutasi (1) adalah sebagai berikut:

$$P(x) = \prod_{t=1}^{T} P\left(x_{z_t} \middle| x_{z_1}, \dots, x_{z_{t-1}}\right) \quad (1)$$

Dimana, x_{z_1} adalah token yang berada di posisi ke-t dalam urutan permutasi z, dan T adalah jumlah total token dalam urutan permutasi tersebut. Dengan mempelajari semua urutan yang mungkin, XLNet dapat menangkap hubungan antar-token baik secara searah (kiri ke kanan) maupun terbalik (kanan ke kiri), yang menghasilkan pemahaman yang lebih kaya tentang konteks token. Studi yang dilakukan oleh (Yang et al., 2020) menyatakan bahwa pendekatan ini secara efektif meningkatkan performa pemahaman teks.



Gambar 2. (a) *Content stream* & (b) *Query stream*
(Sumber: (Han et al., 2023)

Two-Stream Self-Attention. Permutasi urutan token dapat membuat posisi asli token menjadi sulit dilacak. Untuk mengatasi masalah ini, XLNet memperkenalkan *two-stream self-attention*. Aliran ini terbagi menjadi dua jenis: *content stream* dan *query stream*, yang keduanya memiliki peran penting dalam memahami konten dan urutan token.

Content stream mengelola informasi tentang konten token dalam urutan, memperbarui representasi konten dengan memperhatikan semua token dalam sekuens. *Content stream* menggunakan informasi dari *query stream* sebelumnya untuk *key* dan *value*. Persamaan untuk representasi konten token dihitung melalui mekanisme *self-attention* (2) yaitu*:*

$$g_{z_t}^{(m)} \leftarrow Attention\,(Q = g_{z_t}^{(m-1)}, KV = h_{z<t}^{(m-1)};\, \theta \qquad (2)$$

Dimana:

a) $g_{z_t}^{(m)}$: *output content stream* untuk token pada *layer* ke-m
b) $Q = g_{z_t}^{(m-1)}$: *query* menggunakan representasi *content stream* dari *layer* sebelumnya
c) $KV = h_{z<t}^{(m-1)}$: *key* dan *value* menggunakan representasi *query stream* dari *layer* sebelumnya
d) θ : mewakili parameter yang dapat dilatih dalam operasi *attention*

Content stream menggunakan informasi dari kedua *stream* dengan tujuan mengumpulkan informasi kontekstual dari seluruh sekuens.

Query stream digunakan untuk memprediksi token berikutnya dalam urutan permutasi, memperbarui representasi *query* dengan hanya memperhatikan token-token sebelumnya (*autoregressive*). *Query stream* menggunakan informasi dari

query stream sendiri untuk *query, key,* dan *value .Stream* ini mempertahankan informasi tentang posisi permutasi dan diwakili oleh persamaan (3):

$$h_{z_t}^{(m)} \leftarrow Attention\,(Q = g_{z_t}^{(m-1)}, KV = h_{z \le t}^{(m-1)};\, \theta \tag{3}$$

Dimana:

a) $h_{z_t}^{(m)}$: *output query stream* untuk token pada *layer* ke-m
b) $Q = g_{z_t}^{(m-1)}$: *query* menggunakan representasi *query stream* dari layer sebelumnya
c) $KV = h_{z \le t}^{(m-1)}$: *key* dan *value* menggunakan representasi query stream dari layer sebelumnya
d) θ : mewakili parameter yang dapat dilatih dalam operasi *attention*

Query stream hanya menggunakan informasi dari dirinya sendiri dengan tujuan mempertahankan sifat *autoregressive* untuk prediksi token berikutnya.

Dengan memisahkan konten dan urutan posisi, XLNet memastikan bahwa model memahami dengan baik arti token maupun posisi token tersebut dalam konteks keseluruhan. Studi dari (Raffel et al., 2023) mendukung pentingnya model berbasis atensi untuk menangkap hubungan dependensi jarak jauh dalam teks.

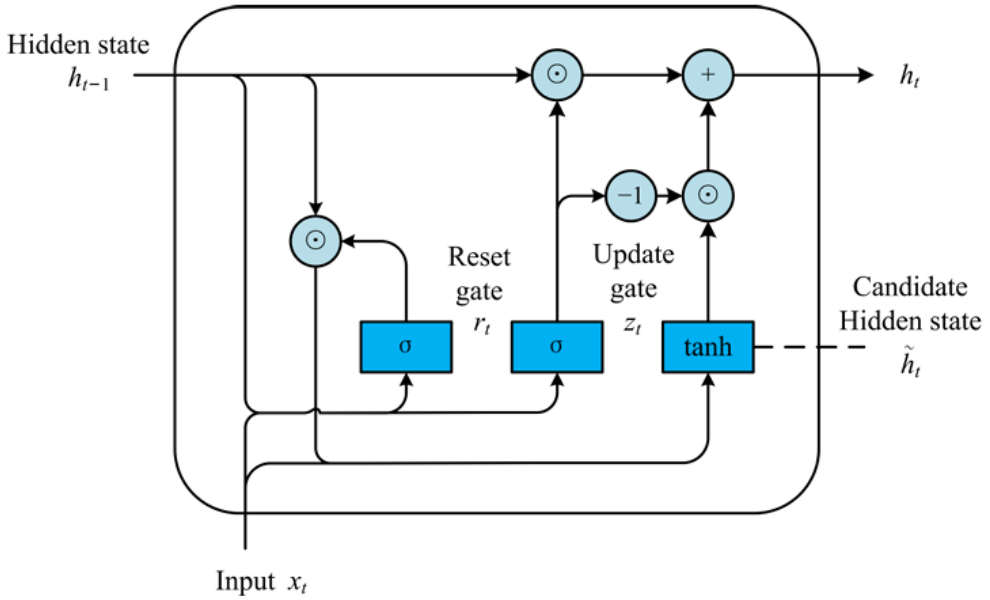
Mekanisme Sirkulasi (*Recurrence Mechanism*). Selain *self-attention*, XLNet menggabungkan prinsip dari *recurrent neural networks* (RNN) untuk memastikan bahwa informasi dari token-token sebelumnya disirkulasikan ke token-token berikutnya dalam urutan. Ini penting untuk memproses teks dalam urutan panjang tanpa kehilangan konteks. Persamaan dasar untuk mekanisme sirkulasi dalam XLNet (4) serupa dengan yang digunakan dalam RNN:

$$h_t = \sigma\,(W_h h_{t-1} + W_x x_t + b) \tag{4}$$

Dimana, h_t adalah *hidden state* yang mengandung informasi tentang token pada langkah t, sedangkan h_{t-1} membawa informasi dari token sebelumnya. Mekanisme ini membantu model mempertahankan memori jangka panjang dari urutan token, meningkatkan performa dalam tugas yang melibatkan konteks teks yang panjang.

1.2.8 BiGRU (*Bidirectional Gated Recurrent Unit*)

Gated recurrent unit (GRU) adalah jenis jaringan saraf rekursif yang berbeda dari jaringan saraf lainnya karena struktur gerbang internalnya. Struktur unik ini memungkinkan jaringan untuk menentukan data mana yang relevan dan data mana yang dapat dibuang berdasarkan hubungan mereka. Struktur ini memfasilitasi transmisi data yang efektif dalam jaringan sambil secara efektif mengendalikan informasi yang berlebihan. Sehingga, GRU dapat mengatasi masalah ketergantungan jangka panjang dalam jaringan saraf (Han et al., 2023)



Gambar 3. GRU *structure*
(Sumber: (Han et al., 2023)

Gated recurrent unit (GRU) menunjukkan efisiensi yang lebih tinggi dan kinerja yang lebih baik daripada RNN dan LSTM saat menangani dependensi jangka panjang. Dengan memperkenalkan dua *gating mechanisms* yaitu *reset gate dan update gate*. *Reset gate* menentukan seberapa banyak informasi masa lalu yang akan dipertahankan pada langkah waktu saat ini, sementara *update gate* memutuskan bagaimana informasi baru diintegrasikan ke dalam status saat ini. Desain struktural ini tidak hanya meningkatkan pemahaman model terhadap data sekuensial tetapi juga meningkatkan kemampuan generalisasinya dalam pembelajaran sampel kecil dengan mengurangi jumlah parameter. Atribut ini membuat GRU dapat digunakan dalam skenario dengan sumber daya terbatas atau di mana pelatihan cepat diperlukan (Zhao et al., 2024)

GRU menggabungkan *input node* saat ini atau *current node input* x^t dan *state* h^{t-1} ditransmisikan dari *node* sebelumnya untuk menghasilkan *output node* saat ini atau *current node's output* y^t dan *hidden state* h^t diteruskan ke-*node* berikutnya. Persamaan transmisi dan pembaruan parameter internal dalam jaringan (5) – (8) yaitu:

$$r^t = \sigma\left(w^{rx}x^t + w^{rh}h^{t-1} + b^r\right) \quad (5)$$

$$z^t = \sigma\left(w^{zx}x^t + w^{zh}h^{t-1} + b^z\right) \quad (6)$$

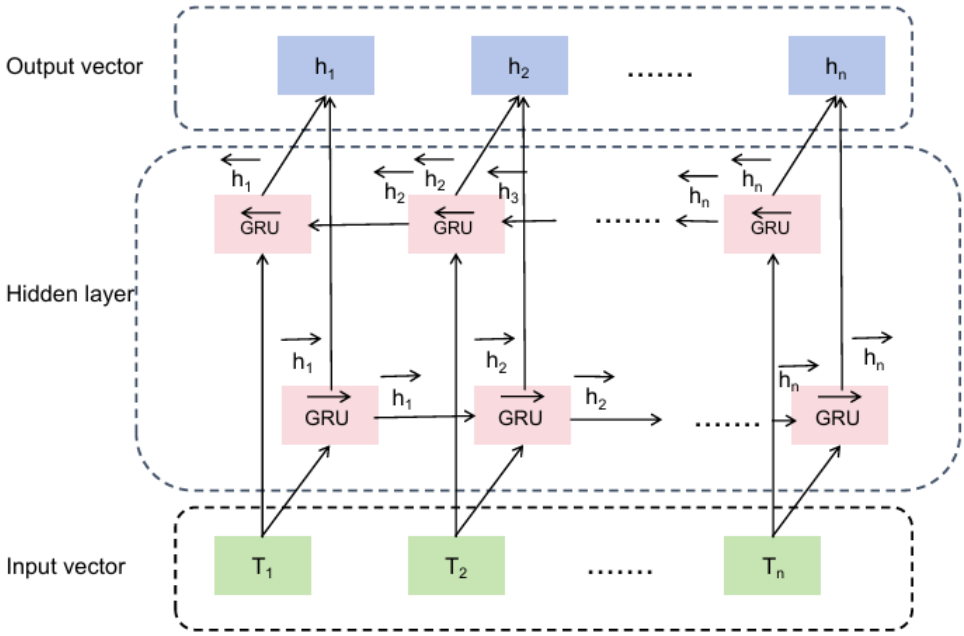
$$h' = tanh\left(w^{xh}x^t + r^t \odot w^{hh}h^{t-1}\right) \quad (7)$$

$$h^t = (1 - z^t) \odot h^{t-1} + h' \odot z^t \quad (8)$$

Dalam persamaan ini, σ menunjukkan fungsi $sigmoid$, yang berfungsi sebagai sinyal *gate control* untuk menjaga nilai dalam rentang [0, 1]. Semakin dekat sinyal *gate* ke-1, semakin banyak data yang diingat. Jika sebaliknya, semakin banyak data yang dilupakan. r adalah kontrol *reset gate* , dan z adalah kontrol *update gate*. h' menunjukkan keadaan tersembunyi dari kandidat *hidden state*. w^{rx}, w^{rh}, dan lainnya adalah *matriks weight*, sedangkan b^r, b^z, dan lainnya adalah *bias terms*. $\odot$ mewakili *Hadamard product,* yang berarti perkalian *element-wise* dari matriks (Zhao et al., 2024)

Reset gate menghitung data setelah *reset* $r^t \odot h^{t-1}$ yang kemudian digabungkan dengan x^t dan melewati fungsi aktivasi $tanh$ untuk menjaga nilai *output* dalam [-1,1], menghasilkan *hidden state* h'. Secara bersamaan, *update gate* melakukan fungsi *forgetting* dan *selective memory*, di mana $(1 - z^t) \odot h^{t-1}$ secara selektif melupakan keadaan *node* sebelumnya, dan $h' \odot z^t$ secara selektif mengingat *hidden state*. Dalam *tasks named entity recognition*, informasi kontekstual sangat penting untuk memprediksi label entitas. Untuk mendapatkan informasi kontekstual dari masa lalu dan masa depan dalam teks, jaringan unit berulang dua arah (*bidirectional gated recurrent unit networks,* BiGRUs) biasanya digunakan. *Hidden state* h' dari GRU hanya dapat mengakses informasi dari konteks masa lalu, bukan masa depan (Zhao et al., 2024)

Untuk GRU, output hanya dipengaruhi oleh input saat ini dan *hidden state* sebelumnya, dan tidak terkait dengan *next state*, menjadikan GRU sebagai model searah. Namun, untuk mengekstraksi fitur yang efektif, perlu memperhatikan informasi saat ini, informasi sebelumnya, dan informasi berikutnya. Dapat dengan mudah dipahami bahwa menggabungkan konteks-konteks ini memudahkan pemahaman informasi tekstual. Berdasarkan ide ini, BiGRU diusulkan untuk menggabungkan *forward* GRU dan *backward* GRU, menjadikan hasil output sebagai kombinasi dari dua GRU berdasarkan *weight matrix* (Han et al., 2023)



Gambar 4. BiGRU *structure*
(Sumber: (Zhao et al., 2024)

BiGRU (*Bidirectional Gated Recurrent Unit*) memanfaatkan GRU *forward* dan *backward* untuk ekstraksi fitur dan bobot informasi kontekstual, menjumlahkan output, dan kemudian melewatinya melalui lapisan linier untuk memetakan vektor *d-dimensional* ke vektor *m-dimensional*, sehingga menghasilkan daftar vektor label *output* akhir dari jaringan BiGRU, $H = \{h_1, h_2, ..., h_n\}$, dengan $H \in \mathbb{R}^{n \times m}$, dimana n adalah panjang urutan teks dan m adalah jumlah *entity type labels*. (Zhao et al., 2024)

1.2.9 *Attention Layer*

Regularisasi ketika berhadapan dengan data tekstual para peneliti sering menghadapi tantangan berupa kalimat panjang yang memuat banyak sekali informasi. Meskipun Bi-GRU dapat mempertimbangkan informasi kontekstual teks secara komprehensif, jaringan ini memiliki keterbatasan dalam menekankan informasi spesifik yang secara langsung relevan dengan tugas target. Oleh karena itu, untuk meningkatkan fokus model pada bagian konteks yang relevan dengan tugas saat ini dan mengekstraksi fitur semantik yang lebih efektif, diperkenalkan *attention layer* di atas jaringan Bi-GRU. Secara khusus, *attention mechanism* yang ditambahkan ke jaringan Bi-GRU dapat menetapkan bobot yang berbeda untuk berbagai bagian informasi kontekstual. Untuk informasi yang terkait erat atau relevan dengan tugas target, model akan mengalokasikan lebih banyak perhatian, sedangkan untuk informasi yang kurang relevan, model akan memberikan lebih sedikit perhatian. Mekanisme ini memungkinkan model untuk memproses informasi yang terkait dengan tugas-tugas tertentu secara lebih efektif, sehingga mencapai kinerja yang lebih tinggi. Dengan menggabungkan Bi-GRU dengan *attention*

mechanism, model tidak hanya dapat menangkap ketergantungan jarak jauh tetapi juga menekankan pentingnya fitur yang relevan dengan memberikan bobot yang lebih tinggi (Zhao et al., 2024)

1.2.10 *PhysioNet*

PhysioNet adalah bagian dari proyek *Research Resource for Complex Physiologic Signals* yang dikelola oleh MIT *Lab for Computational Physiology*, didukung oleh *National Institutes of Health* (NIH). Didirikan pada tahun 1999, PhysioNet bertujuan untuk menyediakan data fisiologis kompleks dan alat untuk analisis sinyal dan sistem biologis. Ini adalah inisiatif terbuka yang memungkinkan peneliti di seluruh dunia untuk berbagi, mengakses, dan menganalisis data medis dan fisiologis.

A. Fitur Utama *PhysioNet*

1. Akses ke *Dataset* Medis dan Fisiologis Besar
   PhysioNet menyediakan berbagai dataset medis yang kaya dan kompleks, yang digunakan untuk berbagai tujuan penelitian di bidang biomedis, kecerdasan buatan, dan ilmu data kesehatan. Beberapa dataset paling terkenal di platform ini termasuk:
   a) MIMIC-III (*Medical Information Mart for Intensive Care*)**:** Kumpulan data besar dari catatan pasien ICU, termasuk data demografis, vitals, obat-obatan, dan catatan klinis. Digunakan dalam penelitian model prediksi, diagnosis, dan pengelolaan penyakit.
   b) *Cinc Challenge Datasets***:** Digunakan dalam kompetisi tahunan yang mengajak peneliti untuk memecahkan masalah dalam analisis sinyal fisiologis, seperti deteksi aritmia jantung dari sinyal EKG.
2. Data Beranotasi Lengkap dan Terperinci
   Dataset di *PhysioNet* sering kali dilengkapi dengan anotasi terperinci, termasuk label diagnostik, waktu kejadian medis, atau interpretasi klinis, yang sangat berguna untuk pengembangan model prediksi berbasis *machine learning* dan *deep learning.*
3. Alat dan Perangkat Lunak untuk Analisis Sinyal Fisiologis
   PhysioNet juga menyediakan alat dan perangkat lunak untuk analisis sinyal fisiologis, seperti sinyal EKG, EEG, atau tekanan darah. Perangkat lunak ini dirancang untuk membantu peneliti mengolah, memvisualisasikan, dan mengekstrak fitur dari sinyal biologis.
4. *Open Access*
   Semua data dan alat yang tersedia di *PhysioNet* dapat diakses secara terbuka, meskipun beberapa dataset tertentu memerlukan prosedur persetujuan karena adanya sensitivitas data pasien. Kebijakan akses terbuka ini memungkinkan kolaborasi lintas institusi dan negara, serta mempercepat kemajuan penelitian dalam berbagai bidang kesehatan.
5. Komunikasi Ilmiah Aktif
   PhysioNet telah menjadi pusat bagi komunitas global peneliti dan ilmuwan data yang bekerja di bidang pemrosesan sinyal fisiologis, kesehatan digital,

prediksi medis, dan AI dalam kesehatan. Platform ini secara rutin mengadakan kompetisi seperti *PhysioNet Challenges*, yang menantang komunitas untuk mengembangkan solusi inovatif dalam analisis sinyal fisiologis.

B. Penggunaan dalam Riset
PhysioNet mendukung berbagai penelitian, seperti:
1. Prediksi Readmisi Rumah Sakit: Banyak studi menggunakan *dataset* MIMIC-III untuk mengembangkan model prediksi berbasis *deep learning* atau *machine learning* untuk memprediksi r*eadmisi* pasien.
2. Analisis Sinyal Jantung: *Dataset* seperti EKG telah digunakan untuk mengembangkan algoritma yang mendeteksi aritmia, fibrilasi atrium, dan kondisi jantung lainnya.
3. Analisis Tren Klinis: *Dataset* ini sering digunakan untuk mempelajari pola klinis dalam pengobatan, seperti respon pasien terhadap obat-obatan atau prosedur ICU.

C. Penggunaan Dataset MIMIC-III
Salah satu *dataset* terpenting di *PhysioNet* adalah MIMIC-III, yang terdiri dari lebih dari 58.000 penerimaan pasien ICU dari 2001 hingga 2012 di Rumah Sakit Beth Israel Deaconess *Medical Center*. MIMIC-III mencakup berbagai data:
1. Catatan klinis: Ringkasan medis, pemeriksaan laboratorium, riwayat obat.
2. Data Fisiologis: Tekanan darah, EKG, dan sinyal vital lainnya.
3. Data demografis: Usia, jenis kelamin, dan diagnosis awal pasien. Dataset ini banyak digunakan dalam pengembangan model prediktif untuk hasil pasien, termasuk prediksi kematian, lama tinggal di ICU, serta potensi readmisi.

1.3 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, penelitian ini merumuskan masalah bagaimana cara mengimplementasikan dan tingkat performa model XLNet-BiGRU-ATT dalam memprediksi kemungkinan keberlanjutan rawat inap pasien setelah pemulangan dengan memanfaatkan catatan klinis dan data medis?

1.4 Tujuan dan Manfaat

1.4.1 Tujuan

Berdasarkan rumusan masalah yang ada, maka tujuan dari penelitian ini adalah untuk mengetahui cara mengimplementasikan dan tingkat performa model XLNet-BiGRU-ATT dalam memprediksi kemungkinan keberlanjutan rawat inap pasien setelah pemulangan dengan memanfaatkan catatan klinis dan data medis.

1.4.2 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Memberikan informasi sebagai dasar pengambilan keputusan atau pertimbangan dalam proses pemulangan pasien.
2. Peningkatan efisiensi waktu pembacaan (analisa) catatan klinis pasien, penelitian ini diharapkan dapat membantu pertimbangan dokter atau petugas medis yang bekerja di unit perawatan intensif, yang perlu mengambil keputusan berdasarkan catatan klinis dalam jumlah besar, sehingga dapat menghemat waktu dan tenaga dokter.

1.5 Ruang Lingkup

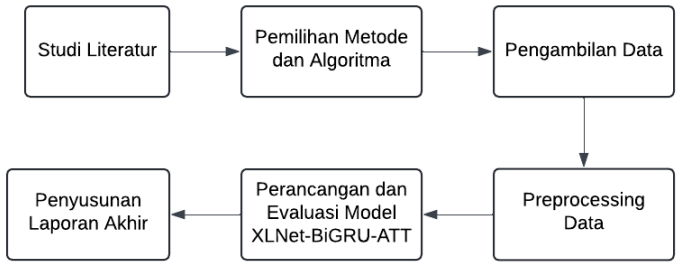
Adapun runang lingkup dalam penelitian ini adalah sebagai berikut:

1. Model klasifikasi untuk memprediksi pasien mana yang berisiko untuk masuk kembali ke rumah sakit dalam waktu 30 hari yang tidak direncanakan dengan menggunakan catatan klinis pasien dalam bentuk teks bebas.
2. Data yang digunakan berupa *database* MIMIC-III (*Medical Information Mart for Intensive Care III*). Basis data rumah sakit yang berisi data yang tidak teridentifikasi dari lebih dari 40.000 pasien yang dirawat di unit perawatan kritis di Beth Israel Deaconess *Medical Center* di Boston, Massachusetts dari tahun 2001 hingga 2012. *Database* ini mencakup informasi seperti demografi, pengukuran tanda vital yang dilakukan di samping tempat tidur (~1 titik data per jam), hasil tes laboratorium, prosedur, pengobatan, catatan perawat, laporan pencitraan, dan kematian (termasuk keluar dari rumah sakit).
   Sebelum data dimasukkan ke dalam *database* MIMIC-III, data tersebut terlebih dahulu diidentifikasi sesuai dengan standar Undang-Undang Portabilitas dan Akuntabilitas Asuransi Kesehatan atau *Health Insurance Portability and Accountability Act* (HIPAA) menggunakan pembersihan data terstruktur dan pergeseran tanggal.
   Proyek database ini disetujui oleh *Institutional Review Boards* dari Beth Israel Deaconess *Medical Center* (Boston, MA) dan Massachusetts *Institute of Technology* (Cambridge, MA).
3. Keluaran sistem berupa prediksi keberlanjutan rawat inap pasien rumah sakit dengan catatan klinis.

BAB II
METODE PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini akan berlangsung dalam beberapa alur tahapan yang terdapat pada Gambar 5.



Gambar 5. Tahapan penelitian

1. Studi Literatur

Pada tahap ini, dilakukan tinjauan literatur untuk mengumpulkan informasi dan penelitian yang relevan tentang prediksi penerimaan kembali pasien rumah sakit, teknik *Natural language processing* (NLP), model *deep learning* seperti XLNet, BiGRU (*Bidirectional Gated Recurrent Unit*), dan *attention mechanism*. Tujuannya adalah memahami dasar teori, metode yang telah ada, dan menemukan gap penelitian yang bisa dijadikan dasar skripsi.

2. Pemilihan Metode dan Algoritma

Setelah melakukan studi literatur, langkah berikutnya adalah memilih metode dan algoritma yang paling sesuai untuk menyelesaikan masalah prediksi penerimaan kembali pasien. Pada kasus ini, dipilih XLNet sebagai model *language representatio*n, BiGRU untuk menangani urutan data dengan memperhitungkan konteks sebelumnya dan sesudahnya, dan *attention mechanism* untuk memberikan bobot pada informasi penting dalam catatan klinis.

3. Pengambilan Data

Tahap ini melibatkan pengumpulan data catatan klinis pasien. Data yang dikumpulkan bisa berupa data tekstual dari rekam medis elektronik atau *electronic health records* (EHR) yang berisi informasi diagnosis, perawatan, riwayat medis, dan lainnya. Data harus mencakup informasi tentang pasien yang diterima kembali di rumah sakit setelah perawatan.

4. *Preprocessing Data*

Data yang telah dikumpulkan kemudian diproses untuk meningkatkan kualitas dan mempersiapkannya untuk analisis lebih lanjut. Tahap preprocessing meliputi pembersihan data (menghapus data yang tidak lengkap atau salah), normalisasi (menyamakan format data), tokenisasi (memecah teks menjadi kata-kata atau token)

dan encoding (mengubah teks menjadi representasi numerik yang bisa diproses oleh model deep learning).

5. Perancangan dan Evalusi Model XLNet-BiGRU-ATT

Tahap ini mencakup desain arsitektur sistem dan evaluasi model prediksi. Model XLNet digunakan untuk mengubah catatan klinis menjadi representasi vektor, kemudian BiGRU digunakan untuk menangani urutan data dengan konteks dari kedua arah (*forward* dan *backward*). *Attention mechanism* diterapkan untuk memfokuskan perhatian pada bagian penting dari catatan klinis. Sistem ini diimplementasikan menggunakan *framework deep learning* yaitu *PyTorch*, dan dilatih menggunakan data yang telah di-*preprocessing*.

6. Penyusunan Laporan Akhir

Langkah terakhir adalah menyusun laporan akhir yang mendokumentasikan seluruh proses penelitian, mulai dari studi literatur, pemilihan metode, pengambilan dan preprocessing data, hingga perancangan dan evaluasi sistem. Laporan akhir juga mencakup hasil eksperimen, analisis kinerja model, dan kesimpulan dari penelitian.

2.2 Waktu dan Lokasi Penelitian

Waktu penelitian dimulai sejak disetujuinya proposal penelitian ini, dimulai pada bulan Juni 2024. Penelitian ini dilaksanakan di Laboratorium *Artificial Intelligence* (AI), Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin, Kelurahan Romang Lompoa, Kecamatan Bontomarannu, Kabupaten Gowa.

Gambar 6. Lokasi penelitian

2.3 Instrumen Penelitian

Instrument penelitian yang digunakan dalam penelitian ini yaitu:

A. *Software*:
   1. *Windows* 10x64
   2. *Jupyter Notebook*
   3. *Google Colab*
   4. *Microsoft (Word, Excel* dan *Power Point)*
   5. *Python* 3.11.2
   7. *PyTorch*

B. *Hardware*:
   1. ASUS X540UBR Intel® Core™ i5-6200U CPU @2.30GHz 2.4GHz
   2. GPU NVIDIA A100 (colab hardware):
      a) Memori GPU (VRAM): 80 GB HBM2e (*High Bandwidth Memory*)
      b) RAM Sistem (CPU Memory)**:** 52 GB

2.4 Teknik Pengambilan Data

Data yang digunakan dalam penelitian ini berupa *database* MIMIC-III (*Medical Information Mart for Intensive Care III*). Basis data rumah sakit yang berisi data yang tidak teridentifikasi dari lebih dari 40.000 pasien yang dirawat di unit perawatan kritis di Beth Israel Deaconess *Medical Center* di Boston, Massachusetts dari tahun 2001 hingga 2012. *Database* ini mencakup informasi seperti demografi, pengukuran tanda vital yang dilakukan di samping tempat tidur (~1 titik data per jam), hasil tes laboratorium, prosedur, pengobatan, catatan perawat, laporan pencitraan, dan kematian (termasuk keluar dari rumah sakit). Sebelum data dimasukkan ke dalam *database* MIMIC-III, data tersebut terlebih dahulu diidentifikasi sesuai dengan standar Undang-Undang Portabilitas dan Akuntabilitas Asuransi Kesehatan atau *Health Insurance Portability and Accountability Act* (HIPAA) menggunakan pembersihan data terstruktur dan pergeseran tanggal. Proyek *database* ini disetujui oleh *Institutiona*l *Review Boards* dari Beth Israel Deaconess *Medical Center* (Boston, MA) dan Massachusetts *Institute of Technology* (Cambridge, MA).

Berikut adalah beberapa komponen utama dari data yang terdapat dalam MIMIC-III:

1. Demografi Pasien:
   Informasi dasar seperti usia, jenis kelamin, ras, tanggal lahir, dan tanggal kematian.
2. Data ICU:
   Informasi tentang rawat inap di ICU termasuk tanggal masuk dan keluar, jenis ICU, serta lama tinggal.
3. Catatan Klinik:
   Catatan teks bebas yang berisi catatan harian dokter, perawat, dan laporan radiologi.
4. Data Laboratorium:
   Hasil tes laboratorium seperti tes darah, tes urin, dan parameter biokimia lainnya.
5. Pengukuran Vital:
   Data vital *signs* seperti tekanan darah, denyut jantung, suhu tubuh, dan laju pernapasan yang dicatat secara berkala.
6. Pemeriksaan Pencitraan:
   Informasi tentang prosedur pencitraan seperti X-ray, CT scan, MRI, termasuk laporan interpretasi.
7. Obat-obatan:

Informasi tentang obat yang diberikan termasuk nama obat, dosis, waktu pemberian, dan *rute* administrasi.

8. Prosedur:
   Data tentang prosedur medis dan bedah yang dilakukan pada pasien selama di ICU.
9. Diagnosis:
   Kode diagnosis (misalnya ICD-9) yang diberikan kepada pasien selama rawat inap di ICU.
10. *Input/Output* Cairan:
    Catatan tentang cairan yang masuk (misalnya, infus) dan keluar (misalnya, urin) dari tubuh pasien.
11. *Scoring* dan Skala:
    Skor klinis dan skala penilaian seperti APACHE, SAPS, SOFA yang digunakan untuk menilai kondisi pasien.
12. Intervensi Mekanis:
    Informasi tentang penggunaan alat bantu mekanis seperti ventilator, dialisis, dan perangkat lain yang digunakan selama perawatan di ICU.

Data yang digunakan dalam penelitian ini adalah data dalam tabel *ADMISSIONS* dan *NOTEEVENTS* yang terdapat dalam *dataset* MIMIC III yang relevan dengan penelitian terkait prediksi penerimaan kembali pasien.

A. Tabel *ADMISSIONS*:

Tabel *admissions* mencatat informasi terkait setiap kali pasien masuk rumah sakit. Kolom-kolom penting dalam tabel ini meliputi:

Tabel 1. Atribut *admissions*

Attribute	*Information*
HADM_ID	ID unik untuk setiap rawat inap (*hospital admission*)
SUBJECT_ID	ID unik untuk setiap pasien
ADMITTIME	Waktu dan tanggal ketika pasien diterima di rumah sakit
DISCHTIME	Waktu dan tanggal ketika pasien dipulangkan dari rumah sakit
DEATHTIME	Waktu kematian, jika pasien meninggal selama rawat inap
ADMISSION_TYPE	Jenis penerimaan (misalnya, elective, emergency)
DIAGNOSIS	Diagnosis utama terkait penerimaan pasien

Tabel ini berfungsi untuk melacak waktu rawat inap, penyebab rawat inap, dan apakah pasien kembali dirawat dalam jangka waktu tertentu setelah dipulangkan. Berikut contoh data pasien yang terdapat pada tabel *admissions*:

ROW_ID	SUBJECT_ID	HADM_ID	ADMITTIME	DISCHTIME	DEATHTIME	ADMISSION_TYPE	ADMISSION_LOCATION	DISCHARGE_LOCATION	INSURANCE	LANGUAGE	RELIGION
21	22	165315	2196-04-09 12:26:00	2196-04-10 15:54:00		EMERGENCY	EMERGENCY ROOM ADMIT	DISC-TRAN CANCER/CHLDRN H	Private		UNOBTAINABLE
22	23	152223	2153-09-03 07:15:00	2153-09-08 19:10:00		ELECTIVE	PHYS REFERRAL/NORMAL DELI	HOME HEALTH CARE	Medicare		CATHOLIC
23	23	124321	2157-10-18 19:34:00	2157-10-25 14:00:00		EMERGENCY	TRANSFER FROM HOSP/EXTRAM	HOME HEALTH CARE	Medicare	ENGL	CATHOLIC
24	24	161859	2139-06-06 16:14:00	2139-06-09 12:48:00		EMERGENCY	TRANSFER FROM HOSP/EXTRAM	HOME	Private		PROTESTANT QUAKER
25	25	129635	2160-11-02 02:06:00	2160-11-05 14:55:00		EMERGENCY	EMERGENCY ROOM ADMIT	HOME	Private		UNOBTAINABLE
26	26	197661	2126-05-06 15:16:00	2126-05-13 15:00:00		EMERGENCY	TRANSFER FROM HOSP/EXTRAM	HOME	Medicare		CATHOLIC
27	27	134931	2191-11-30 22:16:00	2191-12-03 14:45:00		NEWBORN	PHYS REFERRAL/NORMAL DELI	HOME	Private		CATHOLIC
28	28	162569	2177-09-01 07:15:00	2177-09-06 16:00:00		ELECTIVE	PHYS REFERRAL/NORMAL DELI	HOME HEALTH CARE	Medicare		CATHOLIC

Gambar 7. Contoh data pasien

B. Tabel *NOTEEVENTS*

Tabel *noteevents* berisi catatan klinis (*clinical notes*) yang ditulis oleh tenaga kesehatan selama pasien menjalani perawatan. Kolom-kolom penting dalam tabel ini meliputi:

Tabel 2. Atribut *noteevents*

Attribute	*Information*
SUBJECT_ID	ID unik untuk setiap pasien
HADM_ID	ID unik untuk rawat inap yang terkait dengan catatan tersebut
CHARTTIME	Waktu dan tanggal catatan dibuat
CATEGORY	Kategori catatan (misalnya, *discharge summary, radiology report, nursing notes*)
TEXT	Isi catatan klinis dalam format teks yang mencakup informasi tentang status kesehatan pasien, tindakan medis, diagnosis, dan perawatan yang diberikan

Tabel ini berisi data yang kaya informasi berupa teks tidak terstruktur yang mencerminkan kondisi pasien selama dirawat. Berikut contoh data pasien yang terdapat pada tabel *noteevents*:

STORETIME	CATEGORY	DESCRIPTION	CGID	ISERROR	TEXT
	Discharge summary	Report			Admission Date: [**2151-7-16**] Discharge Date: [**2151-8-4**]Service:ADDENDUM:RADIOLOGIC STUDIES: Radiologic studies also included a chestCT, which confirmed cavitary lesions in the left lung apexconsistent with infectious process/tuberculosis. This alsomoderate-sized left pleural effusion.HEAD CT: Head CT showed no intracranial hemorrhage or masseffect, but old infarction consistent with past medicalhistory.ABDOMINAL CT: Abdominal CT showed lesions ofT10 and sacrum most likely secondary to osteoporosis. These canbe followed by repeat imaging as an outpatient. [**First Name8 (NamePattern2) **] [**First Name4 (NamePattern1) 1775**] [**Last Name (NamePattern1) **], M.D. [**MD Number(1) 1776**]Dictated By:[**Hospital 1807**]MEDQUIST36D: [**2151-8-5**] 12:11T: [**2151-8-5**] 12:21JOB#: [**Job Number 1808**]

Gambar 8. Catatan klinis pasien

Untuk memprediksi penerimaan kembali pasien, data dari tabel *ADMISSIONS* dan *NOTEEVENTS* dapat digunakan dalam kombinasi:

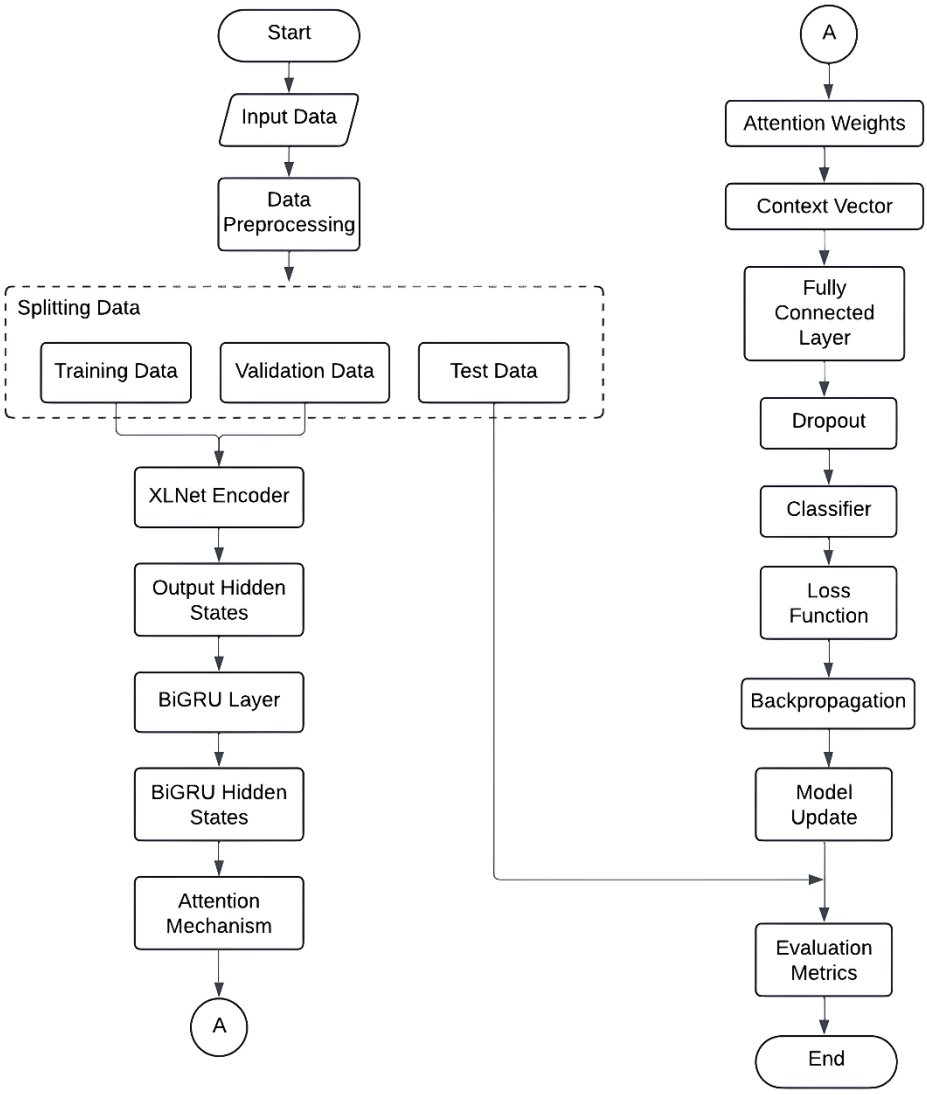
1. Mengidentifikasi Periode Waktu antara Rawat Inap:
   Menggunakan kolom *ADMITTIME* dan *DISCHTIME* dari tabel *ADMISSIONS*, sehingga dapat menghitung waktu sejak pasien dipulangkan hingga mereka kembali dirawat. Penerimaan kembali sering didefinisikan sebagai rawat inap yang terjadi dalam 30 hari setelah pasien dipulangkan.
2. Menggabungkan Catatan Klinis untuk Membuat Fitur:
   a) Tabel *NOTEEVENTS*, terutama catatan yang berada dalam kategori *discharge summary*, digunakan untuk mengekstraksi informasi teks penting mengenai status pasien pada saat dipulangkan.
   b) Model prediktif memanfaatkan catatan medis ini untuk mendeteksi pola atau kata-kata kunci yang terkait dengan risiko penerimaan kembali.
3. Membangun Model Prediksi:
   a) Catatan klinis dari *NOTEEVENTS* diolah menggunakan teknik pemrosesan bahasa alami atau *natural language processing* (NLP). Arsitektur model XLNet-BiGRU-ATT digunakan untuk memahami konteks teks klinis secara lebih baik.
   b) Selain itu, fitur numerik dari *ADMISSIONS* seperti diagnosis, jenis penerimaan (*emergency/elective*), dan informasi demografis juga dimasukkan sebagai input tambahan dalam model untuk meningkatkan akurasi prediksi.

Dengan menggabungkan data dari tabel ***ADMISSIONS*** yang memberikan informasi numerik dan waktu serta ***NOTEEVENTS*** yang berisi data teks tidak terstruktur, model prediksi penerimaan kembali pasien dapat dibuat dengan lebih

akurat. Model ini membantu memprediksi pasien mana yang memiliki risiko tinggi untuk kembali dirawat dalam waktu dekat setelah keluar dari rumah sakit, memungkinkan tindakan pencegahan lebih dini dilakukan.

2.5 Pelaksanaan Penelitian

Pelaksanaan penelitian terbagi ke dalam beberapa bagian, seperti *preprocessing data*, proses pelatihan model, proses evaluasi model & pengujian pada *test data.* Gambar 9 memperlihatkan seluruh alur pelaksanaan penelitian yang akan dilakukan.



Gambar 9. Pelaksanaan penelitian

2.5.1 *Preprocessing Data*

Preprocessing data adalah tahap awal dalam proses pengolahan data yang bertujuan untuk membersihkan, mengorganisir, dan mempersiapkan data agar siap untuk digunakan dalam analisis lebih lanjut. Dalam konteks prediksi penerimaan kembali pasien rumah sakit dengan catatan klinis menggunakan teknik NLP, beberapa langkah yang perlu dilakukan dalam pra-pemrosesan data antara lain:

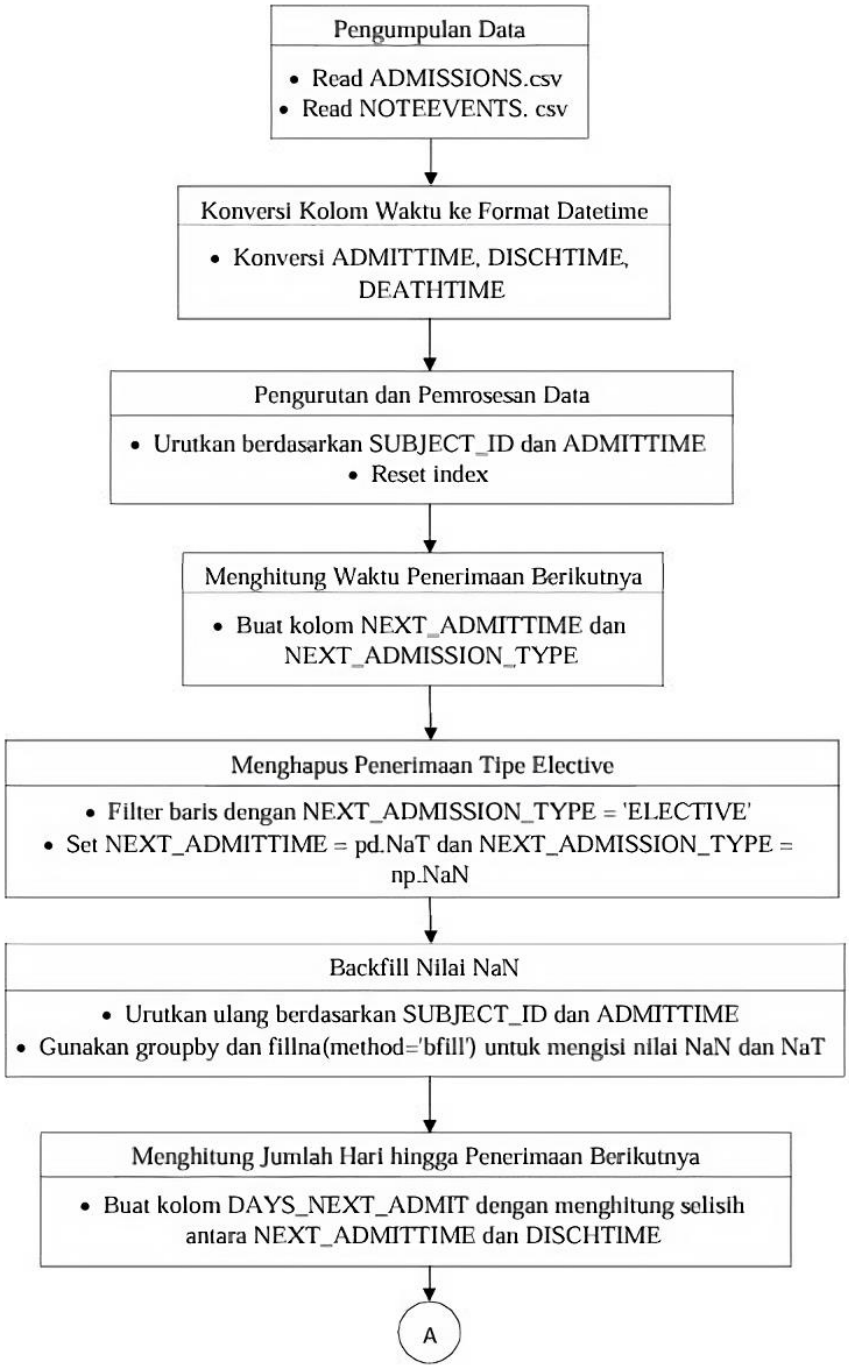
1. Normalisasi Data

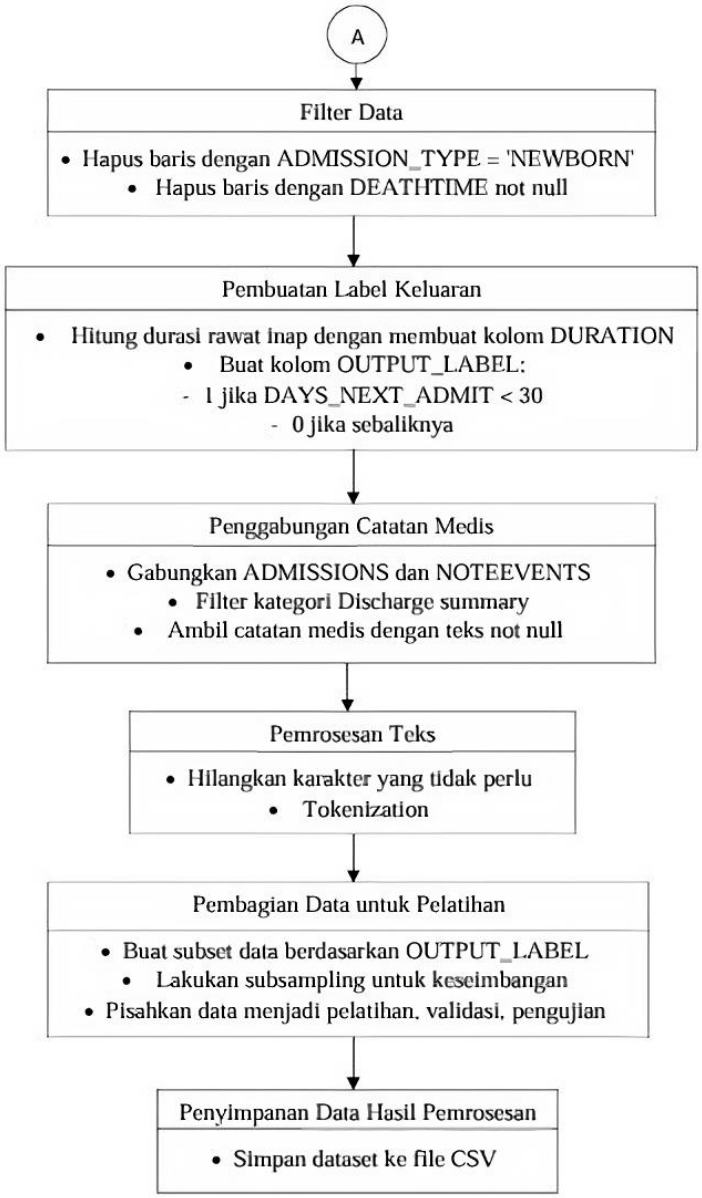
Normalisasi Data adalah proses mengubah data ke dalam format yang seragam atau standar. Dalam konteks klinis, catatan medis memiliki berbagai format dan representasi yang berbeda-beda. Oleh karena itu, normalisasi data bisa meliputi langkah-langkah seperti standarisasi format tanggal, penyesuaian skala, atau penanganan nilai yang hilang.

2. Pemilihan Atribut yang Paling Relevan untuk Analisis

Dalam analisis prediksi penerimaan kembali pasien, tidak semua informasi dalam catatan medis relevan. Oleh karena itu, penting untuk memilih atribut-atribut yang paling relevan atau informatif untuk memprediksi kemungkinan penerimaan kembali pasien. Misalnya, diagnosis awal, riwayat medis, prosedur medis yang dilakukan, dan instruksi tindak lanjut setelah pemulangan dapat menjadi atribut yang relevan untuk dipertimbangkan.

Gambar 10 menunjukkan proses pengumpulan dan pemrosesan data hingga penyimpanan dataset yang telah diproses untuk digunakan dalam pelatihan model prediksi penerimaan kembali pasien rumah sakit.


Pengumpulan Data
• Read ADMISSIONS.csv
• Read NOTEEVENTS. csv
Konversi Kolom Waktu ke Format Datetime
• Konversi ADMITTIME, DISCHTIME, DEATHTIME
Pengurutan dan Pemrosesan Data
• Urutkan berdasarkan SUBJECT_ID dan ADMITTIME
• Reset index
Menghitung Waktu Penerimaan Berikutnya
• Buat kolom NEXT_ADMITTIME dan NEXT_ADMISSION_TYPE
Menghapus Penerimaan Tipe Elective
• Filter baris dengan NEXT_ADMISSION_TYPE = 'ELECTIVE'
• Set NEXT_ADMITTIME = pd.NaT dan NEXT_ADMISSION_TYPE = np.NaN
Backfill Nilai NaN
• Urutkan ulang berdasarkan SUBJECT_ID dan ADMITTIME
• Gunakan groupby dan fillna(method='bfill') untuk mengisi nilai NaN dan NaT
Menghitung Jumlah Hari hingga Penerimaan Berikutnya
• Buat kolom DAYS_NEXT_ADMIT dengan menghitung selisih antara NEXT_ADMITTIME dan DISCHTIME
A



Gambar 10. *Preprocessing data*

1. Pengumpulan Data:
   Mengumpulkan data dari file `*ADMISSIONS*.csv` dan `*NOTEEVENTS*.csv` untuk mendapatkan informasi tentang pasien dan catatan medis.

	ROW_ID	SUBJECT_ID	HADM_ID	ADMITTIME	DISCHTIME	DEATHTIME	ADMISSION_TYPE	ADMISSION_LOCATION	DISCHARGE_LOCATION	INSURANCE	LA
0	21	22	165315	2196-04-09 12:26:00	2196-04-10 15:54:00	NaN	EMERGENCY	EMERGENCY ROOM ADMIT	DISC-TRAN CANCER/CHLDRN H	Private	
1	22	23	152223	2153-09-03 07:15:00	2153-09-08 19:10:00	NaN	ELECTIVE	PHYS REFERRAL/NORMAL DELI	HOME HEALTH CARE	Medicare	
2	23	23	124321	2157-10-18 19:34:00	2157-10-25 14:00:00	NaN	EMERGENCY	TRANSFER FROM HOSP/EXTRAM	HOME HEALTH CARE	Medicare	
3	24	24	161859	2139-06-06 16:14:00	2139-06-09 12:48:00	NaN	EMERGENCY	TRANSFER FROM HOSP/EXTRAM	HOME	Private	
4	25	25	129635	2160-11-02 02:06:00	2160-11-05 14:55:00	NaN	EMERGENCY	EMERGENCY ROOM ADMIT	HOME	Private	

Gambar 11. Data tabel *admissions*

```
df_notes.head()
```

	ROW_ID	SUBJECT_ID	HADM_ID	CHARTDATE	CHARTTIME	STORETIME	CATEGORY	DESCRIPTION	CGID	ISERROR	TEXT
0	174	22532	167853.0	2151-08-04	NaN	NaN	Discharge summary	Report	NaN	NaN	Admission Date: [**2151-7-16**] Dischar...
1	175	13702	107527.0	2118-06-14	NaN	NaN	Discharge summary	Report	NaN	NaN	Admission Date: [**2118-6-2**] Discharg...
2	176	13702	167118.0	2119-05-25	NaN	NaN	Discharge summary	Report	NaN	NaN	Admission Date: [**2119-5-4**] D...
3	177	13702	196489.0	2124-08-18	NaN	NaN	Discharge summary	Report	NaN	NaN	Admission Date: [**2124-7-21**] ...
4	178	26880	135453.0	2162-03-25	NaN	NaN	Discharge summary	Report	NaN	NaN	Admission Date: [**2162-3-3**] D...

Gambar 12. Data tabel *noteevents*

2. Konversi Kolom Waktu:
   Mengubah kolom yang berkaitan dengan waktu ke format *datetime* untuk memudahkan analisis waktu dan durasi.
3. Pengurutan dan Pemrosesan Data:
   Mengurutkan data berdasarkan `*SUBJECT_ID*` dan `*ADMITTIME*` untuk memastikan urutan kronologis dari penerimaan pasien.

	SUBJECT_ID	ADMITTIME	ADMISSION_TYPE
165	124	2160-06-24 21:25:00	EMERGENCY
166	124	2161-12-17 03:39:00	EMERGENCY
167	124	2165-05-21 21:02:00	ELECTIVE
168	124	2165-12-31 18:55:00	EMERGENCY

Gambar 13. Urutan kronologis penerimaan pasien

4. Menghitung Waktu Penerimaan Berikutnya:
   Menambahkan kolom yang mencatat waktu dan jenis penerimaan berikutnya untuk setiap pasien.

	SUBJECT_ID	ADMITTIME	ADMISSION_TYPE	NEXT_ADMITTIME	NEXT_ADMISSION_TYPE
165	124	2160-06-24 21:25:00	EMERGENCY	2161-12-17 03:39:00	EMERGENCY
166	124	2161-12-17 03:39:00	EMERGENCY	2165-05-21 21:02:00	ELECTIVE
167	124	2165-05-21 21:02:00	ELECTIVE	2165-12-31 18:55:00	EMERGENCY
168	124	2165-12-31 18:55:00	EMERGENCY	NaT	NaN

Gambar 14. Menambahkan *next admission type*

5. Menghapus Penerimaan Tipe *Elective:*
   Memfilter penerimaan yang bersifat *elective* (direncanakan) untuk fokus pada penerimaan yang tidak direncanakan.

	SUBJECT_ID	ADMITTIME	ADMISSION_TYPE	NEXT_ADMITTIME	NEXT_ADMISSION_TYPE
165	124	2160-06-24 21:25:00	EMERGENCY	2161-12-17 03:39:00	EMERGENCY
166	124	2161-12-17 03:39:00	EMERGENCY	NaT	NaN
167	124	2165-05-21 21:02:00	ELECTIVE	2165-12-31 18:55:00	EMERGENCY
168	124	2165-12-31 18:55:00	EMERGENCY	NaT	NaN

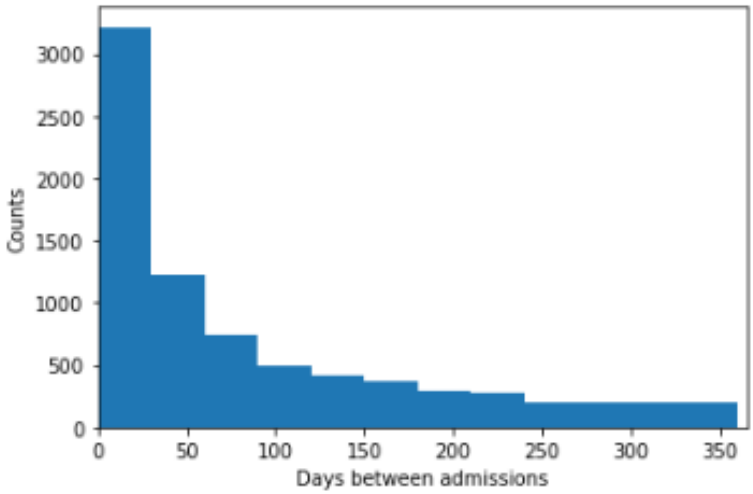
Gambar 15. Memfilter penerimaan *elective*

6. *Backfill* Nilai:
   Mengisi nilai yang hilang dengan informasi yang relevan untuk memastikan kelengkapan data.

	SUBJECT_ID	ADMITTIME	ADMISSION_TYPE	NEXT_ADMITTIME	NEXT_ADMISSION_TYPE
165	124	2160-06-24 21:25:00	EMERGENCY	2161-12-17 03:39:00	EMERGENCY
166	124	2161-12-17 03:39:00	EMERGENCY	2165-12-31 18:55:00	EMERGENCY
167	124	2165-05-21 21:02:00	ELECTIVE	2165-12-31 18:55:00	EMERGENCY
168	124	2165-12-31 18:55:00	EMERGENCY	NaT	NaN

Gambar 16. *Backfill* nilai

7. Menghitung Jumlah Hari hingga Penerimaan Berikutnya:
   Menentukan berapa hari hingga penerimaan berikutnya untuk memahami pola penerimaan ulang pasien.



Gambar 17. Plot histigram hari atara penerimaan kembali pasien

8. Filter Data:
   Menghapus data yang tidak relevan atau yang bisa mengganggu analisis, seperti data bayi baru lahir dan pasien yang meninggal.
9. Pembuatan Label Keluaran:

Membuat label keluaran (*output label*) untuk membedakan pasien yang akan diterima kembali dalam waktu kurang dari 30 hari dari yang tidak, guna keperluan model prediktif.

10. Penggabungan Catatan Medis:
    Menggabungkan informasi penerimaan pasien dengan catatan medis untuk analisis yang lebih komprehensif.
11. Pemrosesan Teks:
    Membersihkan dan menyiapkan teks catatan medis untuk analisis tekstual, seperti tokenisasi.
12. Pembagian Data:
    Membagi data menjadi set pelatihan, validasi, dan pengujian untuk membangun dan menguji model prediktif.
    a) Data Pelatihan (*Training*): Data pelatihan digunakan untuk melatih model prediktif. Model XLNet-BiGRU-ATT akan mempelajari pola dari data pelatihan ini untuk membuat prediksi di masa depan.
    b) Data Validasi: Setelah melatih model dengan data training, perlu mengukur kinerja model di luar sampel data tersebut. Data validasi digunakan untuk mengevaluasi kinerja model selama pelatihan.
    c) Data Pengujian (*Testing*): Setelah model dilatih menggunakan data pelatihan, data pengujian digunakan untuk menguji kinerja model. Data ini tidak digunakan dalam proses pelatihan agar dapat memberikan evaluasi yang obyektif terhadap kemampuan prediksi model. Evaluasi tersebut membantu memastikan bahwa model tidak hanya belajar dari data yang telah dilihat sebelumnya, tetapi juga mampu menggeneralisasi dan membuat prediksi yang akurat terhadap data baru.

    Hasil dari pembagian dataset terdapat di tabel 3.

    Tabel 3. Hasil pembagian *dataset*

Data	Hasil Pembagian *Dataset*	Persentase
Training	26245	80%
Validation	3037	10%
Test	3063	10%

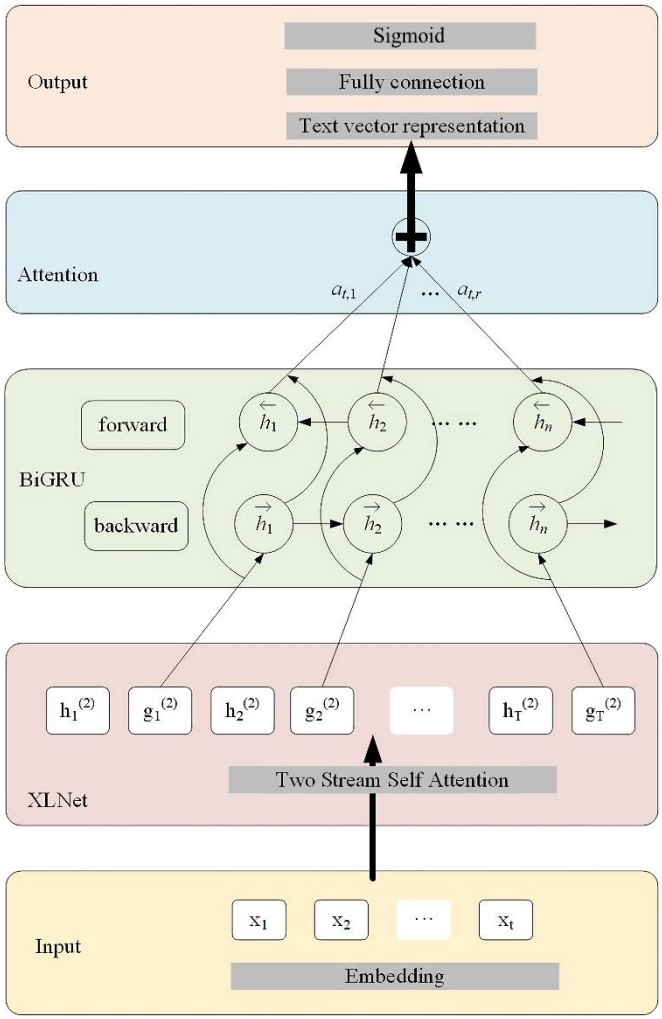
13. Penyimpanan Data Hasil Pemrosesan:
    Menyimpan dataset yang telah diproses untuk digunakan dalam analisis lebih lanjut atau pembangunan model.

Secara keseluruhan, tujuan dari proses ini adalah untuk mempersiapkan data yang bersih dan terstruktur agar dapat digunakan dalam membangun model prediktif untuk mengidentifikasi pasien yang berisiko tinggi kembali dirawat dalam waktu 30 hari setelah dipulangkan.

2.5.2 Training dan Implementasi Model

2.5.2.1 Model XLNet

Gambar 18. Menunjukkan arsitektur model XLNet-BiGRU-ATT.



Gambar 18. Aritektur XLNet-BiGRU-ATT

Arsitektur model XLNet-BiGRU-ATT dikembangkan berdasarkan penelitian Han et al., (2023) dengan modifikasi yang bertujuan untuk meningkatkan kinerja dalam tugas klasifikasi biner. Perubahan kunci yang diterapkan adalah penggantian fungsi aktivasi softmax pada lapisan output dengan sigmoid. Fungsi sigmoid lebih cocok untuk klasifikasi biner karena memprediksi probabilitas setiap kelas secara independen, menghasilkan output antara 0 hingga 1. Berbeda dengan softmax, yang membatasi total probabilitas antar kelas menjadi 1, sigmoid memberikan fleksibilitas lebih besar dalam memprediksi probabilitas setiap kelas, yang diharapkan dapat meningkatkan efektivitas model.

XLNet dikembangkan untuk menangkap relasi antara kata dalam urutan teks secara lebih kompleks dibandingkan metode *masked language model* konvensional seperti BERT. XLNet dilatih dengan teknik *permuted language modeling*, di mana model mempelajari urutan kata dari berbagai kombinasi permutasi, sehingga mampu menangkap konteks yang lebih kaya dari teks yang bersifat urutan panjang, seperti catatan medis.

Model XLNet menerima input berupa token yang dihasilkan dari proses *tokenization* menggunakan tokenizer yang sesuai dengan XLNet. Setiap catatan medis pasien diubah menjadi sekumpulan token ($x_1, x_2, \ldots, x_t$) yang mewakili kata atau frasa dalam catatan tersebut. Token ini kemudian diubah menjadi embedding vektor melalui lapisan *embedding* dari XLNet.

Arsitektur XLNet dilengkapi dengan beberapa lapisan *self-attention*, di mana setiap kata dalam teks dapat berinteraksi secara langsung dengan semua kata lainnya. Proses ini dilakukan untuk menghasilkan representasi kata yang memperhitungkan konteks sekitarnya, yang disebut sebagai *contextualized word embeddings.*

Alur kerja arsitektur xlnet yang menggunakan "Two-Stream Self-Attention":

1. Dua Aliran Informasi (*Two-Stream Information*): Terdapat dua jenis aliran (*stream*) informasi, yaitu *content stream* dan *query stream*. Kedua aliran ini dieksplorasi secara bersamaan untuk melakukan prediksi.
2. *Masked Two-Stream Attention*: Pada bagian utama, terdapat *attention layer* yang dirancang khusus untuk bekerja dengan dua aliran ini. Setiap token yang diinputkan, seperti x_1, x_2, dst., memiliki representasi yang diproses dalam *query stream* dan *content stream*. Pada aliran *content*, token bisa melihat informasi dari masa lalu dan dirinya sendiri, sedangkan pada *query stream*, token tidak bisa melihat dirinya sendiri (ini disebut *masking*).
3. *Positional Encoding and Factorization*: Ada juga pemilihan urutan faktorisasi, di mana contoh yang ditunjukkan dalam gambar 1 adalah "3 → 2 → 4 → 1". Ini menggambarkan urutan di mana informasi token akan diolah, yang memungkinkan model menangani *permutation-based language modeling.*
4. *Masked Self-Attention*: Model ini memanfaatkan mekanisme *masked self-attention,* di mana perhatian pada token tertentu dipengaruhi oleh posisi sebelumnya, sehingga bisa memprediksi dengan lebih fleksibel.

Persamaan dasar yang digunakan dalam XLNet (9) adalah sebagai berikut:

$$h_i = Attention\,(x_1, x_2, \ldots, x_i) \tag{9}$$

Dimana:

a) x adalah urutan kata dalam bentuk vektor yang dihasilkan dari tokenisasi teks.
b) h_i adalah representasi kontekstual dari kata i setelah melalui proses self-attention.

XLNet memiliki fungsi *feed-forward* untuk memproses *embedding* dari kata-kata secara bersamaan dengan kompleksitas waktu yang jauh lebih rendah

dibandingkan model sekuensial lainnya, seperti RNN atau GRU. Hasil dari XLNet kemudian diteruskan ke lapisan selanjutnya untuk diproses lebih lanjut.

Fungsi *loss* yang digunakan dalam arsitektur XLNet untuk klasifikasi biner ini adalah *binary cross-entropy with logits* (*BCEWithLogitsLoss*), yang cocok untuk klasifikasi dengan dua kemungkinan hasil, yaitu pasien diterima kembali atau tidak. Persamaan dari fungsi *loss* (10) adalah:

$$L(y, \hat{y}) = -(y \,.\log(\sigma(\hat{y})) + (1 - y).\log\,(1 - \sigma(\hat{y}))) \tag{10}$$

Dimana:

a) y adalah label sebenarnya (0 atau 1).
b) $\hat{y}$ adalah logits yang dihasilkan oleh model.
c) $\sigma(\hat{y})$ adalah fungsi aktivasi sigmoid yang mengubah logits menjadi probabilitas.

Persamaan untuk fungsi *sigmoid* $\sigma(z)$ (11) didefinisikan sebagai :

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{11}$$

2.5.2.2 Penambahan Lapisan BiGRU (*Bidirectional Gated Recurrent Unit*)

Meskipun XLNet sangat baik dalam memahami konteks global dari teks, penggunaan lapisan BiGRU bertujuan untuk menangkap informasi sekuensial lebih lanjut dari catatan medis pasien. BiGRU (*Bidirectional* GRU) adalah modifikasi dari arsitektur *recurrent neural network* (RNN) yang memungkinkan model untuk mengingat informasi masa lalu dengan lebih efektif sambil tetap menjaga efisiensi komputasi.

Keunggulan *BiGRU* dibandingkan RNN standar adalah kemampuannya untuk memproses sekuens dari dua arah sekaligus: maju (dari awal hingga akhir teks) dan mundur (dari akhir ke awal teks). Dengan demikian, informasi kontekstual dari kata di posisi sebelumnya dan sesudahnya dapat diambil secara lebih akurat. Pada lapisan BiGRU, keluaran dari arah maju dan mundur digabungkan untuk setiap waktu t, sehingga menghasilkan representasi tersembunyi h_t yang dirumuskan pada persamaan (12) sebagai berikut:

$$h_t = [\,\overrightarrow{h_t}, \overleftarrow{h_t}\,] \tag{12}$$

Dimana:

a) $\overrightarrow{h_t}$ adalah representasi dari arah maju.
b) $\overleftarrow{h_t}$ adalah representasi dari arah mundur.
c) h_t adalah gabungan dari keduanya.

Dengan menggunakan BiGRU, model mampu menangkap urutan kata yang lebih panjang secara lebih efektif, yang merupakan ciri khas dari catatan medis yang kompleks dan terkadang ambigu.

2.5.2.3 *Attention Mechanism*

Attention mechanism bertujuan untuk memfokuskan perhatian model pada kata atau frasa yang lebih penting dalam catatan medis. Dalam kasus prediksi penerimaan kembali pasien, tidak semua informasi dalam catatan medis sama pentingnya, beberapa kata atau bagian tertentu mungkin lebih relevan dalam menentukan hasil.

Attention mechanism memungkinkan model untuk "memperhatikan" hubungan antara setiap pasangan token dalam teks *input*, dan menentukan seberapa pentingnya setiap token dalam konteks keseluruhan teks. Dalam konteks prediksi penerimaan kembali pasien rumah sakit, *attention mechanism* akan membantu model untuk memahami hubungan antara berbagai bagian informasi pasien, seperti nama, usia, riwayat medis, gejala, dan diagnosis terakhir. Peran *attention mechanism:*

a. Pemahaman Konteks: *Attention mechanism* memungkinkan model untuk memahami konteks dari setiap token dalam teks *input*. Ini memungkinkan model untuk mengetahui bagaimana informasi tentang nama pasien, usia, riwayat medis, gejala, dan diagnosis terakhir berinteraksi dan saling mempengaruhi.
b. Pentingnya Informasi: *Attention mechanism* menentukan seberapa pentingnya setiap token dalam konteks teks keseluruhan. Ini memungkinkan model untuk mengetahui apakah informasi seperti riwayat medis yang mencakup diabetes tipe 2 dan hipertensi lebih relevan daripada gejala tertentu seperti sakit kepala dalam konteks prediksi penerimaan kembali pasien.
c. Menggabungkan Informasi: *Attention mechanism* memungkinkan model untuk menggabungkan informasi dari berbagai bagian teks *input,* untuk membuat representasi yang lebih kaya dan informatif dari teks keseluruhan. Ini membantu model untuk membuat prediksi yang lebih akurat tentang apakah pasien akan kembali ke rumah sakit atau tidak.

Dengan menggunakan *attention mechanism*, model dapat secara efektif memproses informasi dari teks input, dan menggunakan pemahaman yang diperolehnya untuk membuat prediksi yang lebih baik dalam kasus prediksi penerimaan kembali pasien rumah sakit.

Attention mechanism berfungsi dengan menghitung bobot untuk setiap representasi kata dalam teks. Bobot ini menunjukkan seberapa penting kata tersebut untuk keputusan akhir. Bobot *attention* a_t untuk waktu t dihitung dalam persamaan (13) sebagai:

$$a_t = \frac{\exp(e_t)}{\sum_{i=1}^{T} \exp(e_i)} \tag{13}$$

Dimana:

a) a_t adalah skor attention yang dihitung dari representasi tersembunyi di waktu t.
b) T adalah panjang sekuens (jumlah kata dalam catatan medis).

Setelah mendapatkan bobot *attention*, kemudian menghitung vektor konteks c dalam persaman (14) sebagai:

$$c = \sum_{t=1}^{T} a_t h_t \tag{14}$$

Dimana:

a) c adalah vektor konteks yang dihasilkan dari penjumlahan tertimbang representasi tersembunyi oleh bobot *attention*.
b) h_t adalah representasi tersembunyi di waktu t.

Dengan demikian, vektor konteks c mengandung informasi yang sudah dipilih secara selektif berdasarkan relevansinya terhadap tugas prediksi.

2.5.2.4 Lapisan Klasifikasi dan Sigmoid

Setelah mekanisme *attention* menghasilkan vektor konteks, vektor ini diteruskan ke lapisan *fully connected* untuk menghasilkan keluaran berupa *logits*. Proses ini diwakili oleh persamaan (15) berikut:

$$z = W \,.\, c + b \tag{15}$$

Dimana:

a) z adalah *logits*.
b) W adalah bobot lapisan *fully connected*.
c) c adalah vektor konteks dari mekanisme *attention*.
d) b adalah bias.

Nilai *logits* ini kemudian dilewatkan melalui fungsi sigmoid untuk mengubahnya menjadi probabilitas:

$$\hat{y} = \sigma(z) \tag{16}$$

Dimana $\sigma(z)$ adalah fungsi aktivasi sigmoid yang sudah dijelaskan sebelumnya. Nilai $\hat{y}$ ini kemudian digunakan untuk menentukan apakah pasien akan diterima kembali atau tidak, berdasarkan ambang batas tertentu, yang diinisialisasi ke 0.5. Secara keseluruhan inisialisasi model XLNet-BiGRU-ATT ditunjukkan pada tabel 4.

Tabel 4. Inisialisasi model XLNet-BiGRU-ATT

Layer/Block	*Details*
transformer	*a) Embedding: (32000, 768)*
(XLNetModel)	*b) Layer: 12 x XLNetLayer*
rel_attn	*c) LayerNorm: LayerNorm(768)*
(XLNetRelativeAttention)	*d) Dropout: p=0.1*
ff (XLNetFeedForward)	*e) LayerNorm: LayerNorm(768)*
	f) Linear 1: in_features=768, out_features=3072
	g) Linear 2: in_features=3072, out_features=768
	h) Dropout: p=0.1
	i) Activation: GELU
dropout	*Dropout: p=0.1, inplace=False*
sequence_summary	*j) Linear: in_features=768, out_features=768*
	k) Activation: Tanh
	l) Dropout: p=0.1
logits_proj	*Linear: in_features=768, out_features=1*
bigru	*GRU: in_features=768, hidden_size=384, batch_first=True, bidirectional=True*
attention	*Linear: in_features=768, out_features=1*
fc	*Linear: in_features=768, out_features=768*
classifier	*Linear: in_features=768, out_features=1*
loss_fct	*BCEWithLogitsLoss*

Alur proses yang terjadi pada tabel diatas yaitu:

1. *Embedding Layer*
   a) *Layer*: *XLNetModel -> Embedding*(32000, 768)
   b) Fungsi: *Embedding layer* berfungsi untuk mengonversi *input* token (dalam bentuk *indeks integer*) menjadi vektor berdimensi 768. *Embedding* ini merupakan representasi distribusi dari token yang mengandung informasi semantik, yang nantinya digunakan sebagai *input* untuk lapisan-lapisan selanjutnya dalam model.
2. XLNet *Layers* (*Self-Attention* dan *Feedforward*)
   a) *Layer: XLNetLayer*(12 *layers*) yang terdiri dari:
      1) *XLNetRelativeAttention*: Menggunakan *two-stream self-attention*, yaitu teknik *autoregressive* yang memungkinkan pemrosesan token secara kontekstual. Attention yang digunakan juga memperhitungkan posisi relatif antar token.
      2) *LayerNorm*: Normalisasi dilakukan dengan parameter *LayerNorm*(768) untuk menjaga stabilitas distribusi nilai dalam jaringan.
      3) *Dropout*: Digunakan untuk regularisasi dengan nilai p=0.1 guna mencegah *overfitting*.

b) XLNetFeedForward:

1) *LayerNorm*: Sama seperti pada lapisan *attention*, normalisasi dilakukan untuk stabilitas.
2) *Linear Layer* 1: Mengubah dimensi dari 768 menjadi 3072, memperbesar representasi internal dari fitur *input*.
3) *Linear Layer* 2: Mengembalikan dimensi dari 3072 kembali ke 768.
4) *Activation* (GELU): Fungsi aktivasi yang digunakan adalah GELU (*Gaussian Error Linear Unit*). GELU secara efektif memberikan aktivasi *non-linear* yang lebih halus dibandingkan ReLU, khususnya berguna dalam pemrosesan model berbasis NLP.
5) *Dropout*: *Dropout* sebesar p=0.1 kembali diterapkan sebagai mekanisme regularisasi untuk mengurangi *overfitting*.

3. *Sequence Summary Layer*

Layer: SequenceSummary

a) *Linear Layer*: Mengambil *output* dari lapisan XLNet dengan dimensi 768, kemudian diproyeksikan kembali ke dimensi yang sama menggunakan *Linear*(*in_features*=768, *out_features*=768).
b) *Activation* (Tanh): Aktivasi Tanh digunakan untuk memampatkan *output* ke dalam rentang (-1, 1), membantu dalam normalisasi output.
c) Dropout: Setelah aktivasi, *dropout* sebesar p=0.1 diterapkan.

4. BiGRU *Layer*

Layer: GRU(768, 384, *batch_first=True, bidirectional=True*)

a) Fungsi: Lapisan ini memiliki *hidden size* 384, yang berarti setelah digabungkan (*forward* + *backward*), ukuran *output* akan menjadi 768.
b) BiGRU berguna untuk memproses data sekuensial seperti teks medis yang memiliki informasi penting di awal atau akhir catatan.

5. *Attention Mechanism*

Layer: Attention -> Linear(*in_features=768, out_features=1*)

a) Fungsi: Lapisan *attention* bertujuan untuk memprioritaskan informasi penting dalam sekuens input. Setiap vektor output BiGRU akan diberi bobot relevansi melalui fungsi linear dengan proyeksi ke output bernilai 1. Ini memungkinkan model untuk memfokuskan perhatiannya pada fitur-fitur tertentu dalam catatan medis pasien.
b) Setelah bobot attention dihitung, *context vector* diperoleh melalui proses penggandaan (*dot-product*) antara bobot *attention* dan *output* dari BiGRU.

6. *Fully Connected Layer* (FC)

Layer: Linear(in_features=768, out_features=768)

a) Fungsi: Lapisan *fully connected* (FC) ini bertujuan untuk memampatkan representasi *context vector* yang diperoleh dari

mekanisme *attention*. Proyeksi linear ini digunakan untuk mengurangi dimensi representasi fitur sebelum menuju ke lapisan *output*.

b) *Dropout*: *Dropout* sebesar p=0.1 diterapkan kembali untuk regularisasi sebelum klasifikasi akhir dilakukan.

7. *Classification Layer*

Layer: Linear(in_features=768, out_features=1)

a) Fungsi: Lapisan ini merupakan *linear classifier* yang bertugas untuk menghasilkan *logits*. *Output* dari lapisan ini adalah skor prediksi, yang dalam kasus ini merupakan nilai real yang menunjukkan kemungkinan apakah pasien akan kembali dirawat atau tidak.

b) Output berupa satu nilai (logit), yang akan diproses lebih lanjut oleh fungsi loss.

8. *Loss Function*

Layer: BCEWithLogitsLoss

a) Fungsi: Fungsi *loss* yang digunakan adalah *Binary Cross-Entropy with Logits (BCEWithLogitsLoss).* Fungsi ini secara otomatis menggabungkan fungsi sigmoid ke dalam perhitungan *loss*, yang berarti *logits* dari *classifier* akan diubah menjadi probabilitas antara 0 dan 1 sebelum dihitung selisihnya dengan label yang sebenarnya (0 atau 1).

b) Loss ini menghitung perbedaan antara prediksi model dan label sebenarnya, serta digunakan untuk mengoptimalkan parameter selama proses pelatihan.

Kesimpulan proses:

1. *Input* token diubah menjadi *embedding vector*. *Embedding* diproses oleh 12 lapisan XLNet, yang terdiri dari *self-attention* dan *feedforward* dengan *LayerNorm* dan GELU sebagai aktivasi.
2. Setelah input token di-*embed,* vektor *embedding* ini diproses oleh XLNet yang dilengkapi dengan *two-stream self-attention*. XLNet berbeda dari model-model transformer lain karena memproses data secara *autoregressive* dan *autoregressive permuted*, yang memungkinkan model ini untuk menangkap informasi kontekstual secara lebih efektif.
3. Output dari XLNet berupa representasi vektor sekuensial $h_1^{(2)}, h_2^{(2)}, \ldots, h_r^{(2)}$ yang digunakan sebagai input bagi lapisan selanjutnya.

```
Output XLNet (hidden states dari token):
tensor([[[-0.8984, -0.5233, -0.9776,  ..., -0.4563,  0.3893,  0.2800],
         [-0.8479, -0.4829, -0.9435,  ..., -0.4333,  0.3550,  0.2761],
         [-0.7998, -0.4382, -0.9690,  ..., -0.4226,  0.3180,  0.2627],
         ...,
         [-1.0260, -1.2186, -2.4152,  ..., -0.7135,  0.3664, -0.7094],
         [-0.4430, -1.0411, -1.2441,  ..., -0.6757,  0.1345,  0.0515],
         [-0.4250, -1.0077, -1.1801,  ..., -0.7421,  0.1171,  0.0188]]])
```

Gambar 19. Output XLNet

4. Vektor representasi sekuensial dari XLNet kemudian diproses oleh lapisan BiGRU. BiGRU (*Bidirectional* GRU) merupakan varian dari GRU (*Gated Recurrent Unit*) yang memproses data sekuensial dalam dua arah (*forward* dan *backward*). Ini memungkinkan model untuk menangkap informasi dari masa lalu dan masa depan dalam catatan medis pasien.
5. Pada proses ini, keluaran dari BiGRU adalah representasi dua arah yang digabungkan dalam satu vektor, yaitu: $\overrightarrow{h_1}, \overleftarrow{h_1}$, $\overrightarrow{h_2}, \overleftarrow{h_2}$,…, $\overrightarrow{h_n}, \overleftarrow{h_n}$

```
Output BiGRU (representasi sekuensial dari token):
tensor([[[-0.5338,  0.3906, -0.5204,  ..., -0.7470, -0.6333, -0.4927],
         [-0.6989,  0.4273, -0.5134,  ..., -0.7590, -0.6528, -0.5038],
         [-0.7525,  0.4180, -0.4643,  ..., -0.7715, -0.6707, -0.5226],
         ...,
         [-0.6273,  0.8097, -0.6945,  ..., -0.9253, -0.6011, -0.4749],
         [-0.5657,  0.7313, -0.8762,  ..., -0.9644, -0.9237, -0.3388],
         [-0.4928,  0.6710, -0.9175,  ..., -0.8861, -0.8439, -0.2094]]],
       grad_fn=<TransposeBackward1>)
```

Gambar 20. Output BiGRU

6. Setelah melalui BiGRU, mekanisme *attention* diterapkan untuk memberi bobot pada informasi penting dalam catatan medis.
7. Nilai *attention* ini kemudian digunakan untuk menghitung *context vector* dengan mengalikan *attention* dengan *output* BiGRU.

```
Output Attention (vektor konteks berdasarkan bobot perhatian):
tensor([[-0.8103,  0.6798, -0.4622,  0.8542, -0.3122,  0.8693,  0.2371, -0.4482,
         -0.3588,  0.8155,  0.6928,  0.7847,  0.5619,  0.3382,  0.4381,  0.6227,
         -0.5449,  0.8357, -0.0524,  0.7968,  0.9858,  0.1926, -0.3837,  0.4631,
          0.5473,  0.5408,  0.0501,  0.3142,  0.8622,  0.4217,  0.2322, -0.2657,
          0.8239, -0.6904, -0.8183,  0.5167,  0.6485,  0.9631,  0.5325, -0.5526,
          0.5068,  0.1828, -0.4083,  0.9776,  0.4431,  0.7389,  0.8741,  0.9439,
          0.8448,  0.1249, -0.8908,  0.0506,  0.4034, -0.8902, -0.8177, -0.3561,
         -0.8673,  0.7097, -0.2838, -0.1083,  0.1515,  0.8525,  0.1434, -0.2255,
          0.5652,  0.4329,  0.2162,  0.7963, -0.3109, -0.9163, -0.6978, -0.2045,
```

Gambar 21. Output *Attention Mechanism*

8. Setelah *context vector* diperoleh, representasi teks ini kemudian diproses melalui lapisan *fully connected* yang mengubahnya menjadi vektor yang lebih ringkas dan ditambah dengan lapisan *dropout* untuk mencegah *overfitting*.
9. Vektor representasi teks akhir ini kemudian diumpankan ke lapisan *output* (*linear classifier*) untuk menghasilkan *logits*, yang merupakan skor prediksi apakah pasien akan diterima kembali atau tidak.
10. Untuk pelatihan, model menggunakan fungsi *loss binary cross-entropy* dengan *logits* (*BCEWithLogitsLoss*), yang menghitung perbedaan antara nilai prediksi dengan label sebenarnya.

2.5.2.5 Optimasi Model

Pelatihan model dilakukan menggunakan optimasi AdamW yang dikombinasikan dengan penjadwalan *learning rate* berbasis *warmup*. AdamW adalah variasi dari algoritma optimisasi Adam yang menggabungkan regulasi *weight decay* untuk mencegah *overfitting.* Secara keseluruhan optimasi model XLNet-BiGRU-ATT ditunjukkan pada tabel 5.

Tabel 5. Optimasi model

Step	*Description*
Pengelompokan Parameter	Mendefinisikan dua grup parameter untuk optimizer: 1. *no_decay*: Bias & bobot *LayerNorm* dikecualikan dari *weight decay*. 2. decay: Parameter model lainnya dengan *weight decay* sebesar 0.01.
Pengaturan *Optimizer*	Inisialisasi optimizer AdamW menggunakan parameter yang dikelompokkan dan *learning rate*
Perhitungan Langkah *Warmup*	Menghitung jumlah langkah *warmup* (*warmup_steps* = *warmup_proportion* x *num_train_steps*).
Pengaturan *Scheduler*	Mengatur *scheduler learning rate linear* dengan *warmup* Parameter scheduler: - *num_warmup_steps*: Jumlah langkah *warmup* yang telah dihitung. - *num_training_steps*: Total jumlah langkah pelatihan (*num_train_steps*).

Alur proses yang terjadi pada tabel 5 yaitu:

A. Inisialisasi Variabel *no_decay*

Variabel *no_decay* adalah daftar yang berisi nama-nama parameter yang tidak akan dikenakan *weight decay*. Parameter ini biasanya mencakup bias dari neuron dan parameter dari *layer normalization* seperti *LayerNorm.bias* dan *LayerNorm.weight*. *Weight decay* merupakan salah satu bentuk regularisasi L2 yang digunakan untuk mencegah model overfitting dengan menambahkan penalti terhadap nilai besar dari bobot model. Namun, dalam arsitektur *neural network*, bias dan parameter *normalization layer* dikecualikan dari penalti ini, karena *weight decay* tidak relevan untuk arsitektur tersebut. Weight decay membantu mengurangi kompleksitas model dengan mendorong model untuk memiliki bobot yang lebih kecil, sehingga membuat model lebih general dan mengurangi risiko overfitting pada data latih.

B. Membuat *optimizer_grouped_parameters*

Parameter model dibagi menjadi dua kelompok:

1. Kelompok pertama mencakup semua parameter yang tidak ada dalam daftar *no_decay*. Kelompok ini akan dikenakan *weight decay* dengan nilai 0.01. Penambahan *weight decay* dilakukan melalui optimasi yang bertujuan untuk mengurangi magnitudo dari bobot model.

   Persamaan Regularisasi L2 atau *Weight Decay* (17) yaitu:

$$L_{total} = L_{original} + \lambda \sum_i w_i^2 \quad (17)$$

Dimana:

a) $L_{original}$ adalah fungsi *loss* asli.
b) λ adalah *hyperparameter* yang mengontrol tingkat regularisasi (*weight decay*).
c) w_i adalah bobot dari parameter model.

2. Kelompok kedua mencakup parameter yang ada di dalam daftar *no_decay*, seperti bias dan *LayerNorm*. Untuk parameter ini, *weight decay* diatur ke 0 karena tidak memerlukan penalti regularisasi.

C. Membuat *Optimizer* AdamW

Mendefinisikan *optimizer* yang akan digunakan untuk memperbarui bobot model selama proses pelatihan. *Optimizer* yang digunakan adalah **AdamW** (*Adam with Weight Decay*), yang merupakan variasi dari algoritma optimasi Adam yang telah dioptimalkan untuk menerapkan *weight decay* secara lebih efisien.

AdamW menggabungkan keuntungan dari Adam dan penambahan *weight decay* secara eksplisit, tanpa mempengaruhi perhitungan gradien seperti yang terjadi pada regularisasi L2 dalam Adam standar.

AdamW mengombinasikan perhitungan gradien dan momentum dalam persamaan (18) – (21) sebagai berikut:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (18)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (19)$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \ \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (20)$$

$$w_t = w_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} - \eta \lambda w_{t-1} \quad (21)$$

Dimana:

a) m_t adalah estimasi gradien momentum.
b) v_t adalah estimasi gradien kuadrat (momentum kedua).
c) $\hat{m}_t$ dan $\hat{v}_t$ adalah *bias-corrected* momentum.
d) g_t adalah gradien.
e) η adalah *learning rate*.
f) λ adalah *weight decay*.
g) w_t adalah bobot pada waktu t.

Berbeda dengan Adam standar, AdamW memisahkan *weight decay* dari perhitungan gradien, sehingga *weight decay* hanya diterapkan langsung ke parameter model setelah gradien diperhitungkan.

D. Menghitung $Warmup\_steps$

$Warmup\_steps$ adalah jumlah langkah (*steps*) selama fase *warmup* pada saat pelatihan. *Warmup* adalah strategi di mana *learning rate* secara bertahap ditingkatkan selama beberapa langkah awal pelatihan, kemudian secara bertahap diturunkan pada langkah-langkah berikutnya.

$Warmup\_steps$ dihitung berdasarkan proporsi dari total langkah pelatihan (num_train_steps). Jika $warmup\_proportion$ adalah 0.1, berarti 10% dari seluruh langkah pelatihan digunakan untuk fase *warmup*.

Persamaan untuk menghitung $warmup\_steps$ (22) yaitu:

$$warmup\_steps = warmup\_proportion \times num\_train\_steps \quad (22)$$

Dimana:

a) $warmup\_proportion$ adalah persentase langkah pelatihan yang dialokasikan untuk *warmup*.
b) num_train_steps adalah jumlah total langkah pelatihan.

E. Membuat *Scheduler*

Membuat penjadwal (*scheduler*) yang mengatur nilai *learning rate* selama pelatihan menggunakan pendekatan *linear schedule with warmup*. Fungsi ini melakukan dua hal:

1. Pada fase *warmup*, *learning rate* akan meningkat secara *linear* dari 0 hingga mencapai nilai puncak (*learning rate* awal yang diatur dalam *optimizer*).
2. Setelah fase *warmup*, *learning rate* akan menurun secara *linear* hingga akhir pelatihan.

Persamaan untuk penjadwalan *linear* dengan *warmup* (23) – (24) yaitu:

Untuk $t \leq warmup\_steps$:

$$\eta_t = \frac{t}{warmup\_steps} \cdot \eta_{max} \quad (23)$$

Untuk $t > warmup\_steps$:

$$\eta_t = \eta_{max} \cdot (1 - \frac{t - warmup\_steps}{num\_train\_steps - warmup\_steps}) \quad (24)$$

Dimana:

a) η_t adalah *learning rate* pada langkah t.
b) η_{max} adalah *learning rate* maksimum (yang diatur dalam *optimizer*).
c) t adalah langkah pelatihan saat ini.
d) $warmup\_steps$ adalah jumlah langkah dalam fase *warmup*.
e) num_train_steps adalah total langkah pelatihan.

Pada fase awal, *learning rate* dimulai dari 0 dan meningkat hingga mencapai nilai puncak selama langkah-langkah *warmup*. Setelah fase *warmup* selesai, *learning rate* akan menurun secara *linear* hingga mendekati nol pada akhir pelatihan.

Sehingga secara keseluruhan proses yang terjadi yaitu:

1. Inisialisasi: Model diinisialisasi dengan bobot *pre-trained* XLNet, kemudian menambahkan lapisan BiGRU dan *attention mechanism*.
2. Proses *Forward Pass*: Data catatan medis pasien yang telah di-tokenisasi melewati arsitektur XLNet, BiGRU, dan *attention mechanism*. Hasil akhirnya adalah nilai *logits*.
3. Perhitungan *Loss*: *Logits* yang dihasilkan dibandingkan dengan label sebenarnya menggunakan fungsi *BCEWithLogitsLoss*.
4. *Backward Pass*: *Gradien loss* dihitung dan digunakan untuk memperbarui bobot model.
5. Optimasi: Bobot model diperbarui menggunakan AdamW, dengan *weight decay* diterapkan untuk mencegah *overfitting*.
6. Penjadwalan *Learning Rate*: *Learning rate* diatur agar menurun secara bertahap selama pelatihan, dengan *warmup* awal untuk stabilisasi.

2.5.3 Evaluasi Model

1. Uji Model

   Menguji Model pada Data Pengujian untuk Mengevaluasi Kinerjanya:

   a. Setelah melatih model dengan data pelatihan, langkah selanjutnya adalah menguji kinerja model pada data pengujian yang belum pernah dilihat sebelumnya.
   b. Data pengujian memberikan gambaran objektif tentang seberapa baik model dapat menggeneralisasi pola yang telah dipelajari dari data pelatihan untuk membuat prediksi yang akurat terhadap data baru.
   c. Uji model pada data pengujian memastikan bahwa model tidak hanya "menghafal" data pelatihan tetapi juga mampu mengidentifikasi pola umum yang dapat diterapkan pada situasi baru.

2. Hitung Metrik Evaluasi

   Evaluasi model dilakukan menggunakan metrik berikut:

   A. ROC-AUC (*Area Under the ROC Curve*):

      1. **ROC (*Receiver Operating Characteristic*):**
         ROC adalah kurva yang menggambarkan performa sebuah model klasifikasi biner dengan memplot *True Positive Rate* (TPR) di sumbu y melawan *False Positive Rate* (FPR) disumbu x pada berbagai *threshold* klasifikasi. TPR dikenal juga sebagai *recall*, dan FPR adalah persentase negatif yang salah diklasifikasikan sebagai positif. ROC digunakan untuk mengevaluasi keseimbangan antara sensitivitas (*recall*) dan 1-spesifisitas (FPR).
      2. **AUC (Area Under the Curve):**
         Mengukur performa sebuah model klasifikasi dengan menghitung luas di bawah kurva evaluasi. AUC saja tidak bisa digunakan untuk mengukur performa model tanpa adanya kurva seperti ROC atau *Precision-Recall* (PR). AUC adalah metrik yang menghitung luas di

bawah kurva, sehingga ia memerlukan kurva seperti ROC atau PR untuk diukur. Artinya, AUC adalah ringkasan dari performa model yang dinilai berdasarkan kurva tersebut. AUC memberikan gambaran keseluruhan dari performa model pada berbagai *threshold*, tetapi tanpa kurva (ROC atau PR), tidak ada yang diukur. ROC atau PR adalah alat visual yang memetakan kinerja model pada berbagai ambang batas keputusan, dan AUC hanya menghitung area yang berada di bawah kurva tersebut.

3. **ROC-AUC (Area Under the ROC Curve):**
   AUC adalah ukuran luas di bawah kurva ROC, yang memberikan nilai tunggal untuk menilai performa model klasifikasi. Nilai AUC berkisar antara 0 dan 1. Semakin besar nilai AUC (mendekati 1), semakin baik model dalam membedakan antara kelas positif dan negatif.
   ROC Curve sendiri tidak memberikan satu nilai tunggal untuk menilai kinerja model, tapi memberikan gambaran tentang bagaimana model berperforma pada berbagai *threshold*. Namun, ROC *Curve* digunakan untuk menghasilkan AUC, yang merupakan nilai tunggal yang dapat digunakan untuk menilai kinerja model secara keseluruhan.
   AUC memetakan performa model ke dalam satu angka yang menunjukkan keseimbangan antara *True Positive Rate* (TPR) dan *False Positive Rate* (FPR) atau antara *Precision* dan *Recall*, tergantung pada jenis kurva yang digunakan (ROC atau *Precision-Recall*).

Fungsi ini menghitung skor prediksi berdasarkan beberapa metrik, menggabungkan prediksi individual dari satu pasien, dan kemudian mengevaluasinya menggunakan kurva ROC.

Alur proses perhitungan:

1. Mengurutkan Data Berdasarkan ID:
   Data diurutkan berdasarkan ID. Setiap ID mewakili satu pasien, sehingga jika ada beberapa contoh data untuk satu pasien, mereka akan dikelompokkan bersama.
2. Menghitung Skor Agregat:
   Skor agregat untuk setiap pasien dihitung sebagai gabungan dari skor maksimal dan rata-rata skor untuk semua contoh data dari pasien tersebut. Bagian ini mengagregasi skor prediksi per ID dengan menghitung rata-rata yang melibatkan maksimum, jumlah, dan jumlah nilai per ID. Persamaan untuk menghitung Skor Agregat (25) yaitu:

$$Agregat\ Skor = \frac{\max(s_i) + \frac{1}{2}\sum_{i=1}^{n} s_i}{1 + \frac{1}{2}n} \tag{25}$$

   Di mana s_i adalah skor prediksi individual dan n adalah jumlah contoh data untuk satu pasien.

Penjelasan:

a) $\max(s_i)$: Mengambil nilai maksimum dari semua prediksi untuk satu pasien. Ini menganggap bahwa skor prediksi tertinggi adalah yang paling relevan.

b) $\sum_{i=1}^{n} s_i$: Mengambil jumlah dari semua prediksi untuk pasien tersebut. Bagian ini memberi kontribusi dari semua prediksi, tetapi dibagi dua untuk menyeimbangkan kontribusinya dengan nilai maksimum.

c) $1 + \frac{1}{2}n$: Merupakan faktor pembagi yang menormalkan skor akhir, sehingga hasilnya dipengaruhi oleh jumlah prediksi yang ada (semakin banyak prediksi, semakin besar kontribusi dari rata-rata prediksi).

Skor agregat ini memberikan keseimbangan antara menggunakan prediksi maksimum dan rata-rata prediksi pasien untuk menghasilkan satu nilai representatif yang akan digunakan untuk evaluasi model.

3. Mengambil Nilai Label Terendah:
Nilai label terendah untuk setiap pasien diambil menggunakan fungsi agregat `min`. Ini memastikan bahwa jika ada label yang berbeda untuk pasien yang sama, maka label negatif (0) lebih diutamakan. Label "0" berarti pasien tidak dirawat kembali, sementara "1" berarti dirawat kembali.

4. Plot kurva ROC dan AUC:

a) *False Positive Rate* (FPR) dan *True Positive Rate* (TPR) dihitung menggunakan fungsi *roc_curve* dengan persamaan (26) dan (27) berikut:

$$TPR = \frac{TP}{TP + FN} \tag{26}$$

$$FPR = \frac{FP}{FP + TN} \tag{27}$$

Dimana:

TPR : Merupakan *recall* atau sensitivitas, yang menunjukkan seberapa baik model mengenali kelas positif.

TP : *True Positives* (prediksi positif yang benar).

FN : *False Negatives* (prediksi negatif yang salah).

FPR : Menunjukkan seberapa sering model salah memprediksi negatif sebagai positif.

FP : *False Positives* (prediksi positif yang salah).

TN : *True Negatives* (prediksi negatif yang benar).

ROC (*Receiver Operating Characteristic*) mengukur performa model klasifikasi dengan memplot *True Positive Rate* (TPR) melawan *False Positive Rate* (FPR) di berbagai nilai *threshold.*

Fungsi $roc\_curve()$ akan menghasilkan nilai FPR, TPR, dan threshold berdasarkan:

1. *True Labels***:** Ini digunakan untuk membandingkan prediksi model dengan label sebenarnya (0 atau 1). Dalam rumus TPR dan FPR:
   a) TP (*True Positives*): Diukur dengan berapa banyak prediksi positif yang benar-benar positif sesuai dengan x.
   b) FP (*False Positives*): Diukur dengan berapa banyak prediksi positif yang sebenarnya negatif sesuai dengan x.

   x digunakan untuk menghitung nilai-nilai TP**,** FN**,** FP**,** dan TN yang dibutuhkan untuk menghitung TPR dan FPR**.**
2. *Temp values* (*Predicted Scores*): Ini adalah skor probabilitas atau *logits* yang dihasilkan oleh model. Dalam setiap iterasi *threshold*, *temp values* digunakan untuk menentukan apakah suatu sampel harus diklasifikasikan sebagai positif atau negatif. ROC curve dihitung dengan menggunakan beberapa threshold berbeda pada *temp values* , untuk menghitung berapa banyak TP**,** FP**,** FN**,** dan TN di berbagai ambang batas tersebut.
   a) Misalnya, jika *threshold* adalah 0.5, maka:
      Jika *temp values* > 0.5, prediksi dianggap positif.
      Jika *temp values* <= 0.5, prediksi dianggap negatif.
   b) Dengan *threshold* yang berbeda-beda, dapat menghasilkan berbagai nilai TPR dan FPR, yang kemudian diplot sebagai kurva ROC.

   True Labels: Menyediakan informasi *actual labels* yang digunakan untuk menghitung TP, FP, FN, dan TN, yang diperlukan untuk mendapatkan TPR dan FPR pada setiap *threshold*.

   temp values (*Predicted Scores*): Memberikan skor probabilitas yang dihasilkan model, yang dibandingkan dengan berbagai *threshold* untuk memutuskan klasifikasi (positif/negatif). Ini digunakan untuk menentukan prediksi pada setiap *threshold*, yang kemudian digunakan untuk menghitung TPR dan FPR.

   Dengan demikian, kedua variabel (*true labels* dan *temp values*) bersama-sama digunakan untuk menghitung ROC melalui perhitungan TPR dan FPR di berbagai *threshold*.

b) AUC (*Area Under Curve*) dihitung dari FPR dan TPR menggunakan fungsi auc.

Persamaan AUC (28) yaitu:

$$AUC = \int_0^1 TPR(FPR)d(FPR) \tag{28}$$

AUC mengukur performa model dalam membedakan antara kelas positif dan negatif. Nilai AUC berkisar antara 0 dan 1, di mana 1 menunjukkan model yang sempurna dan 0.5 menunjukkan model acak.

B. PR-AUC (*Area Under the Precision-Recall Curve*):

a) PR (*Precision-Recall Curve*):
Precision-Recall (PR) *curve* adalah kurva yang memplot *Precision* (presisi) di sumbu y melawan *Recall* (atau *True Positive Rate*) di sumbu x. *Precision* adalah proporsi prediksi positif yang benar-benar benar, sedangkan *Recall* adalah proporsi positif aktual yang terprediksi dengan benar. PR *Curve* lebih fokus pada performa pada kelas minoritas (positif) dan sangat berguna ketika dataset tidak seimbang.

b) PR-AUC (Area Under the Precision-Recall Curve):
PR-AUC mengukur luas di bawah kurva *Precision-Recall*, memberikan gambaran performa keseluruhan model dalam konteks presisi dan recall. PR-AUC berguna terutama dalam kasus di mana kelas positif lebih jarang. Nilai yang lebih besar menunjukkan performa yang lebih baik, dengan PR-AUC maksimal adalah 1.

Fungsi ini menghasilkan kurva *Precision-Recall* dari hasil prediksi.

Alur proses perhitungan:

1. Menghitung *Precision* dan *Recall*:
Persamaan untuk menghitung Precision (29) yaitu:

$$Precision = \frac{TP}{TP + FP} \tag{29}$$

Persamaan untuk menghitung Recall (30) yaitu:

$$Recall = \frac{TP}{TP + FN} \tag{30}$$

2. Menghitung AUC untuk Precision-Recall:
AUC untuk kurva Precision-Recall dihitung menggunakan fungsi auc.

C. PR80 (*Recall at Precision* 80):

RP80 adalah metrik yang menunjukkan *Recall* pada titik di mana Precision mencapai nilai 80%. Ini menilai seberapa baik model dapat menangkap kelas positif (*recall*) dengan tingkat presisi yang tinggi (minimal 80%). Metrik ini berguna jika tujuan evaluasi model adalah untuk mempertahankan tingkat presisi yang cukup tinggi.

Fungsi ini menghitung *precision-recall* untuk prediksi yang dihasilkan dan kemudian mengevaluasi *recall* pada *precision* yang lebih dari 80%.

Menghitung *Recall* pada *Precision* > 0.8:

a) *Threshold* untuk *Precision* ditetapkan pada 80%. Jika *precision* lebih dari 0.8, maka recall pada titik tersebut dihitung.

b) Jika tidak ada sampel yang memenuhi syarat ini, maka RP80 (*Recall at Precision* 80) dianggap nol.

Interpretasi:

a) *Recall at Precision* 80 (PR80) berarti seberapa banyak contoh positif yang berhasil diidentifikasi model (*recall*) dengan presisi yang setidaknya 80%.
b) PR80 adalah metrik yang sangat berguna dalam situasi di mana ingin memastikan bahwa ketika model memprediksi positif, model ini 80% yakin bahwa prediksi tersebut benar, sekaligus memaksimalkan kemampuan model untuk mendeteksi sebanyak mungkin kasus positif.

PR80 dihitung dengan mencari nilai *recall* dari *precision* yang lebih dari 0.80. Jika precision lebih besar dari 80%, maka recall-nya disebut sebagai PR80.

Sehingga secara keseluruhan proses evaluasi yang dilakukan yaitu:

a) ROC-AUC (*Area Under the ROC Curve)*:
Menghitung skor agregat prediksi dan mengevaluasi performa model menggunakan ROC dan AUC.
b) PR-AUC (Area Under the Precision-Recall Curve):
Menghasilkan kurva *Precision-Recall* untuk memvisualisasikan *trade-off* antara *Precision* dan *Recall*.
c) PR80 (Recall at Precision 80):
Menggabungkan perhitungan *Precision-Recall* dan mengevaluasi *recall* pada *precision* di atas 80%, yang dikenal sebagai RP80.

Hubungan Antar Metrik:

a. **ROC-AUC** dan **PR-AUC**: Kedua metrik ini menggambarkan performa model, namun fokusnya berbeda. ROC-AUC lebih cocok digunakan ketika terdapat keseimbangan antara kelas positif dan negatif, sementara PR-AUC lebih fokus pada kinerja untuk mendeteksi kelas positif, khususnya ketika data imbalanced (jumlah kelas negatif lebih banyak).
    1) Jika menggunakan ROC, maka AUC-ROC akan menunjukkan keseimbangan antara:
        a. *True Positive Rate* (TPR): Proporsi positif yang benar-benar terdeteksi oleh model (juga dikenal sebagai *recall*).
        b. *False Positive Rate* (FPR): Proporsi negatif yang salah terdeteksi sebagai positif oleh model.

        AUC-ROC menunjukkan seberapa baik model dapat memisahkan antara kelas positif dan negatif dengan mempertimbangkan bagaimana model mengelola TPR dan FPR di berbagai ambang batas (*thresholds*).
    2) Jika menggunakan *Precision-Recall* (PR) *curve*, maka AUC-PR akan menunjukkan keseimbangan antara:
        a. *Precision*: Proporsi prediksi positif yang benar-benar positif.
        b. *Recall* (TPR): Proporsi positif yang benar-benar terdeteksi oleh model.

AUC-PR lebih fokus pada evaluasi bagaimana model menjaga keseimbangan antara *precision* dan *recall*, terutama saat menangani dataset yang tidak seimbang (di mana kelas positif lebih sedikit dari kelas negatif).

b. **RP80** adalah metrik yang spesifik digunakan untuk mengevaluasi recall ketika presisi diatur pada tingkat tertentu (80%), yang sangat penting dalam skenario di mana kesalahan prediksi positif lebih kritis.

Kombinasi metrik ini memberi gambaran yang lebih komprehensif tentang performa model dalam berbagai skenario evaluasi.

BAB III
HASIL DAN PEMBAHASAN

3.1 Hasil

Pada penelitian ini, model XLNet-BiGRU-ATT digunakan untuk memprediksi penerimaan kembali pasien rumah sakit berdasarkan catatan klinis pasien. Pelatihan model dilakukan dengan beberapa parameter penting, yaitu *learning rate* sebesar 2e-5, *batch size* sebesar 32, dan panjang urutan maksimum (*max sequence length*) 512. Dataset dibagi menjadi tiga, dengan 80% digunakan untuk data pelatihan, sementara 20% sisanya digunakan untuk validasi dan pengujian.

3.1.1 Pemrosesan *Text* Medis

Chunk Text. *Chunking* adalah teknik pemecahan teks panjang menjadi potongan-potongan yang lebih kecil. Dalam penelitian ini, chunking diterapkan untuk memotong catatan medis yang panjang menjadi segmen-segmen dengan jumlah kata yang lebih kecil. Pada penelitian ini, setiap catatan teks dibagi menjadi potongan berisi 318 kata

Proses *Chunking*:

a. Input Teks Panjang: Teks yang berasal dari catatan medis sering kali sangat panjang, dan tidak bisa langsung diolah oleh model seperti XLNet, yang memiliki batasan panjang input (biasanya 512 token).
b. Pembagian Teks Menjadi *Chunk*: Dalam kode yang digunakan, teks dipecah menjadi potongan dengan panjang 318 kata. Jika teks lebih panjang dari itu, maka teks tersebut dipisahkan menjadi beberapa *chunk*.
   Misalnya, jika ada teks dengan 1000 kata, maka teks tersebut akan dibagi menjadi 3 *chunk*: dua *chunk* pertama berisi masing-masing 318 kata, dan *chunk* terakhir berisi kata-kata sisa (misalnya 364 kata, di mana hanya sebagian yang digunakan sesuai aturan potongan minimal).
c. Mengelola sisa kata: Potongan teks yang lebih kecil dari 318 kata, tetapi memiliki lebih dari 10 kata, maka sisa tersebut juga dijadikan sebagai potongan tersendiri. Hal ini penting untuk tidak membuang bagian teks yang relevan, walaupun tidak mencapai ukuran 318 kata.
d. Penggabungan Kembali Hasil *Chunking*: Potongan-potongan teks ini kemudian digunakan sebagai *input* baru dengan label yang sama, sehingga setiap chunk merepresentasikan informasi dari catatan pasien yang terpisah.

Alasan di balik pembagian ini dijelaskan sebagai berikut:

a. Batas Panjang Input Model: Model seperti XLNet memiliki batasan panjang *input*. Batas ini biasanya diukur dalam jumlah token, bukan kata. Pada tahap awal, pemecahan teks dilakukan dalam 318 kata sebagai strategi untuk membatasi panjang teks sebelum tokenisasi. Setelah tokenisasi, teks dapat berubah menjadi lebih banyak token karena ada kata yang terpecah menjadi beberapa token. Jika teks terlalu panjang, model tidak dapat memproses seluruh teks secara efisien atau akan memotong sebagian teks, sehingga

mengurangi informasi penting. *Chunking* membantu memastikan bahwa seluruh teks diproses tanpa ada bagian yang hilang.

b. Efisiensi Komputasi: *Chunking* membantu mengurangi waktu pemrosesan dan kebutuhan memori. Teks yang terlalu panjang membutuhkan lebih banyak sumber daya komputasi, dan chunking memecah masalah ini menjadi bagian-bagian kecil yang bisa diproses dengan lebih efisien.
c. Menghindari Kehilangan Informasi: Proses *chunking* memastikan bahwa setiap bagian teks tetap dipertahankan dan diolah, tanpa ada informasi penting yang diabaikan. Bahkan potongan kecil dengan kata-kata sisa yang cukup signifikan (lebih dari 10 kata) tetap diikutkan untuk analisis.
d. Penanganan Dokumen Panjang: Catatan medis sering kali sangat panjang, dan jika seluruh dokumen diberikan ke model sekaligus, ada risiko sebagian besar informasi yang ada di awal atau akhir dokumen akan diabaikan oleh model. *Chunking* memungkinkan model untuk memproses bagian-bagian berbeda dari dokumen dan memastikan semua bagian mendapat perhatian yang proporsional.

WordPieces Tokenizer. Setelah teks dibagi menjadi chunk yang lebih kecil, langkah selanjutnya adalah tokenisasi. Pada penelitian ini, tokenisasi dilakukan menggunakan XLNet*Tokenizer* dari model XLNet**.** *WordPieces* adalah teknik tokenisasi berbasis sub-kata yang digunakan oleh model XLNet untuk menangani kata-kata yang tidak ada dalam kosakata model atau kata-kata yang kompleks. Proses Tokenisasi *WordPieces*:

a. *Input* Teks: Setelah teks dipecah menjadi *chunk*, setiap *chunk* diubah menjadi token menggunakan XLNet*Tokenizer*. XLNet menggunakan pendekatan *WordPieces* untuk memecah kata menjadi unit yang lebih kecil, terutama ketika kata tersebut tidak dikenal atau tidak umum dalam kosakata model.
    3) Misalnya, jika *input* teks adalah "*unbelievable*", *tokenizer* XLNet akan memecahnya menjadi "*un*", "*believe*", dan "*able*".

    Pendekatan ini berguna untuk menangani kata-kata yang jarang atau kompleks, tanpa harus membuang informasi.
b. Penanda Sub-Kata (▁): Dalam hasil tokenisasi, terdapat penanda karakter khusus seperti ▁, yang menandakan bahwa token tersebut merupakan awal dari kata baru. Misalnya, dalam tokenisasi teks asli:
    4) *▁date, ▁of, ▁birth*
    5) *▁lethargic* menjadi *▁let, har, gic*

    Simbol ini berguna untuk membantu model memahami struktur kata dan konteks kalimat.
c. Pemetaan ke Token Angka: Setiap sub-kata atau token diubah menjadi angka unik (*INPUT_ID*) dalam kosakata XLNet, yang kemudian digunakan sebagai input bagi model. Token angka ini digunakan untuk mewakili kata-kata dalam format yang dapat diolah oleh model pembelajaran mesin.

Alasan penggunaan WordPieces:

a. Menangani Kata yang Tidak Ada dalam Kosakata: *WordPieces* memungkinkan model untuk menangani kata-kata yang tidak ada dalam kosakata (*Out-Of-Vocabulary*, OOV). Dengan memecah kata menjadi bagian yang lebih kecil, bahkan kata baru atau jarang dapat diwakili oleh kombinasi sub-kata yang sudah dikenal oleh model.
b. Efisiensi Representasi: Daripada mengandalkan kosakata yang sangat besar untuk setiap kata yang mungkin muncul, *WordPieces* memungkinkan model untuk menggunakan kosakata yang lebih kecil dengan memecah kata-kata kompleks menjadi sub-kata yang umum. Ini membuat pemodelan teks menjadi lebih efisien dan fleksibel.
c. Menangkap Makna Secara Kontekstual: *WordPieces* memecah kata menjadi unit yang lebih kecil yang masih bisa mempertahankan makna. Model kemudian dapat memprediksi kata-kata yang tidak dikenal berdasarkan bagian-bagian kata tersebut. Misalnya, meskipun model tidak tahu kata "antifungal", ia masih memahami arti "anti" dan "fungal".
d. Generalisasi yang Lebih Baik: Dengan memecah kata menjadi sub-kata yang lebih kecil, model dapat melakukan generalisasi yang lebih baik untuk kata-kata baru atau kombinasi kata yang tidak sering muncul. Ini membuat model lebih tangguh dalam menghadapi data yang bervariasi.

Tabel 6. Hasil tokenisasi teks medis

Text	***Tokens***
may recommend lisinopril low dose as an outpatient, follow up in 1 month with cardiology . pna- patient received levaquin, vanc, flagyl and ceftriaxone in the ed. imaging concerning for pneumonia. - trend leukocytosis and fever curve - change unasyn for coverage of aspiration pna to augmentin 500mg po tid today() - blood cultures: pending - urine cultures: ngtd final . delta ms: per patient's pcp, . , and pt's sister- patient is at baseline. lethargy may have been infection. will continue infectious w/u and treatment. tox screen negative except for benzos. . elevated liver enzymes: most likely shock liver, hypotension during v-tach and cardioversion; kub was normal . - trend lft's: improving over time - ruq ultrasound: liver and gallbladder normal with right pleural effusion and ascites; . anemia: stable over hospital course - will monitor daily - patient has a hx of anemia. . elevated d-dimer cta- negative, may be acute phase reactant.	*_may _recommend _ lis in op ril _low _dose _as _an _outpatient , _follow _up _in _1 _month _with _cardio logy _ . _p na - _patient _received _ le va quin , _van c , _flag yl _and _ ce ft ria x one _in _the _ ed . _imaging _concerning _for _pneumonia . _ - _trend _ le uk oc y t osis _and _fever _curve _ - _change _ una syn _for _coverage _of _as piration _p na _to _augment in _500 mg _po _ tid _today ( ) _ - _blood _cultures : _pending _ - _urine _cultures : _ ng t d _final _ . _delta _ m s : _per _patient ' s _ pc p , _ . _ , _and _ pt ' s _sister - _patient _is _at _baseline . _let har gy _may _have _been _infection . _will _continue _infectious _ w / u _and _treatment . _to x _screen _negative _except _for _be nz os . _ . _elevated _liver _enzymes : _most _likely _shock _liver ,*

Lanjutan tabel 6. Hasil tokenisasi teks medis

Text	***Tokens***
*. fen -replete lytes, aspiration precautions. ppx: pneumoboots, heparin sc and ppi . fc confirmed with sister . brother: (c), (h) francetta medications on admission: none discharge medications: amiodarone 200 mg tablet sig: one (1) tablet po daily (daily). disp:*30 tablet(s)* refills:*2* amoxicillin-pot clavulanate 250-5 mg/5 ml suspension for reconstitution sig: ten (10) ml po tid (3 times a day) for 7 days. disp:*210 ml* refills:*0* discharge disposition: home with service facility: discharge diagnosis: supraventricular tachycardia requiring cardioversion pneumonia discharge condition: stable and improving discharge instructions: you will be discharged home today after being in the hospital for both a fast heart rate and a pneumonia. both of these were controlled in the hospital, and you will be sent home with two new medications. the antibiotic augmentin is to help treat your pneumonia. you will take this for the next 7 days. you are also prescribed amiodarone. this medicatoin is to help control your heart rate. you must take this everyday.*	*_hypo tension _during _v - tach _and _cardio version ; _ k ub _was _normal _ . _ - _trend _ lf t ' s : _improving _over _time _ - _ ru q _ultrasound : _liver _and _ gall bla d der _normal _with _right _ ple ural _ ef fusion _and _as cit es ; _ . _an emia : _stable _over _hospital _course _ - _will _monitor _daily _ - _patient _has _a _ h x _of _an emia . _ . _elevated _ d - di mer _c ta - negative , _may _be _acute _phase _react ant . _ . _ fen _ - re ple te _ ly tes , _as piration _precautions . _ . _pp x : _p ne um o boo t s , _he par in _sc _and _ ppi _ . _f c _confirmed _with _sister _ . _brother : _ (c) , _ (h) _ franc etta _medications _on _admission : _none _discharge _medications : _ ami oda rone _200 _mg _tablet _ s ig : _one _ (1) _tablet _po _daily _ (dai ly) . _dis p : * 30 _tablet (s) * _refill s : * 2 * _am oxi cil lin - pot _ cla vul an ate _250 - 5 _mg / 5 _ ml _suspension _for _recon s titu tion _ s ig : _ten _ (10) _ ml _po _ tid _ (3 _times _a _day) _for _7 _days . _dis p : * 2 10 _ ml * _refill s : * 0 * _discharge _disposition : _home _with _service _facility : _discharge _diagnosis : _supra vent r icular _ tach y card ia _requiring _cardio version _pneumonia _discharge _condition : _stable _and _improving _discharge _instructions : _you _will _be _discharged _home _today _after _being _in _the _hospital _for _both _a _fast _heart _rate _and _a _pneumonia . _both <sep> <cls>*

Chunking diperlukan untuk memastikan bahwa teks panjang dapat diproses oleh model yang memiliki batas panjang input. Ini memungkinkan teks panjang dibagi menjadi segmen-segmen yang lebih kecil yang lebih mudah diproses oleh model tanpa kehilangan informasi penting. **WordPieces** pada XLNet diperlukan untuk menangani kata-kata yang tidak ada dalam kosakata, dengan cara memecah kata

menjadi bagian-bagian yang lebih kecil (sub-kata). Ini memungkinkan model untuk tetap memahami konteks dan makna dari kata yang jarang atau baru, serta membuat pemrosesan teks menjadi lebih efisien.

3.1.2 Hasil Pelatihan Model

Selama proses pelatihan, model dievaluasi menggunakan metrik ROC-AUC, PR-AUC dan PR80. Hasil pelatihan menunjukkan bahwa model mencapai kinerja terbaik pada *epoch* ke-4, di mana nilai ROC-AUC mencapai 0.7424 dan PR-AUC sebesar 0.7227, sedangkan PR80 mencapai 0.237. Hasil ini mengindikasikan bahwa model memiliki performa yang baik dalam memprediksi kemungkinan pasien akan kembali ke rumah sakit.

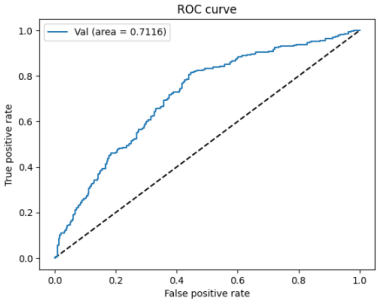
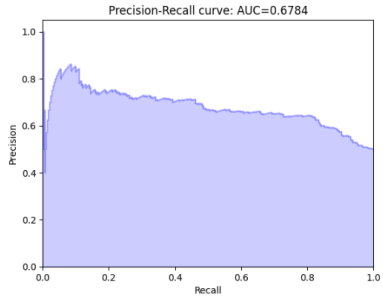
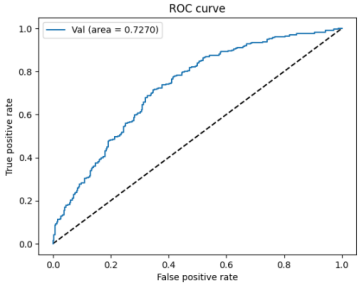
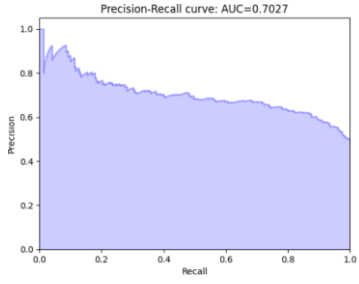
Tabel berikut merangkum performa model pada setiap *epoch* yang diuji:

Tabel 7. Performa model pada setiap *epoch*

Epoch	ROC-AUC	PR-AUC	PR80
1	0.7116	0.6784	0.1103
2	0.7270	0.7027	0.1793
3	0.7424	0.7060	0.1517
4	0.7424	0.7227	0.2379
5	0.7262	0.7005	0.2068

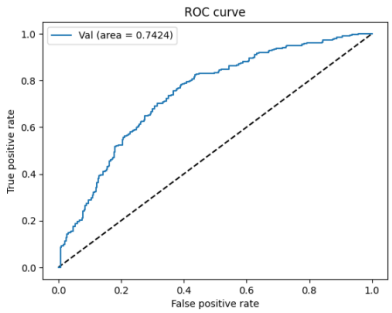
Tabel berikut menunjukkan *curva* ROC-AUC dan *curva* PR-AUC pada setiap *epoch*. Dimana pada *epoch* ke-4, model menunjukkan hasil terbaik dengan nilai ROC-AUC sebesar 0.7424 dan PR-AUC sebesar 0.7227.

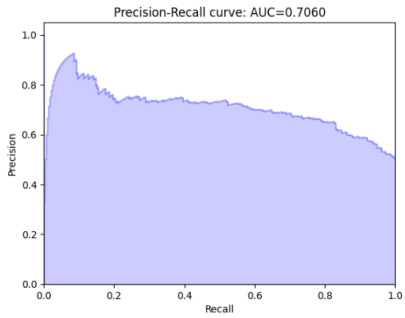
Tabel 8. *Curva* performa model pada setiap *epoch*

***Epoch* 1**		
***Epoch* 2**		

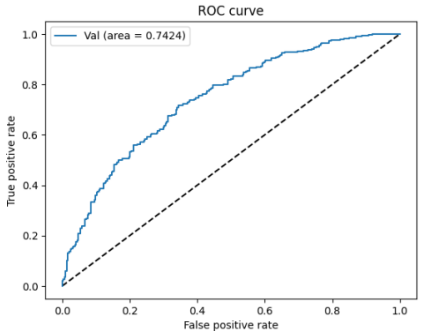
Lanjutan tabel 8. *Curva* performa model pada setiap *epoch*

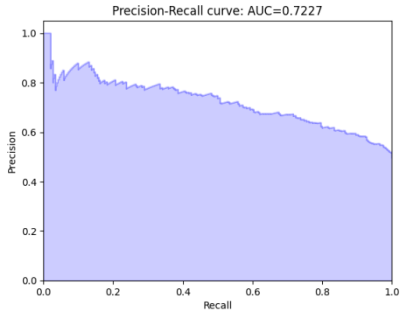
***Epoch* 3**



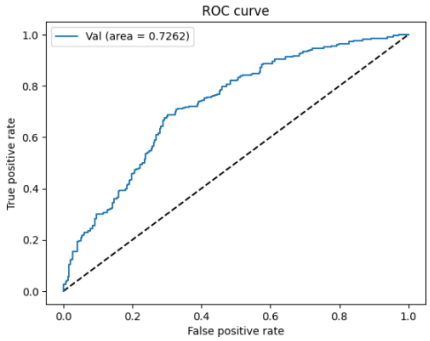


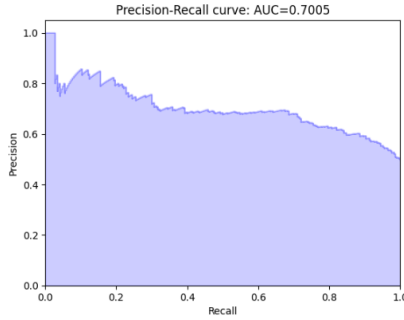
***Epoch* 4**





***Epoch* 5**





ROC *Curve (Receiver Operating Characteristic Curve)* – Kiri

a. Sumbu X (*False Positive Rate* - FPR): Ini menunjukkan proporsi kesalahan positif, yaitu seberapa sering model salah memprediksi positif pada data yang sebenarnya negatif.
b. Sumbu Y (*True Positive Rate* - TPR): Ini menunjukkan seberapa baik model memprediksi positif yang benar pada data yang benar-benar positif.
c. Area di bawah kurva (AUC): Nilai AUC sebesar 0.7424 menandakan bahwa model memiliki kemampuan yang cukup baik dalam membedakan antara kelas positif dan negatif. Semakin tinggi AUC (mendekati 1), semakin baik performa model.

d. Garis putus-putus (random guessing): Garis diagonal ini menunjukkan performa tebakan acak. Semakin jauh kurva dari garis ini, semakin baik performa model.

Precision-Recall Curve – Kanan

a. Sumbu X (*Recall*): Ini menunjukkan proporsi dari *instance* positif yang benar-benar diidentifikasi oleh model (*True Positives* / (*True Positives* + *False Negatives*)).
b. Sumbu Y (*Precision*): Ini menunjukkan seberapa baik prediksi positif model (*True Positives* / (*True Positives* + *False Positives*)).
c. AUC *Precision-Recall* (0.7227): Menunjukkan performa keseluruhan dalam hal *precision* dan *recall*.

3.2 Pembahasan

Hasil penelitian menunjukkan bahwa model **XLNet-BiGRU-ATT** dapat memberikan performa yang kompetitif dalam memprediksi penerimaan kembali pasien rumah sakit. Dengan nilai **ROC-AUC**, **PR-AUC**, dan **PR80** yang masing-masing mencapai 0.742, 0.723, dan 0.237 pada **epoch ke-4**, model ini mampu memanfaatkan informasi yang terkandung dalam catatan medis pasien secara efisien. Hasil ini menunjukkan kemampuan model dalam menangani tantangan prediksi yang melibatkan data klinis dengan teks panjang dan kompleks.

3.2.1 Pemilihan Model XLNet-BiGRU-ATT

Pemilihan arsitektur model XLNet-BiGRU-ATT didasarkan pada kombinasi keunggulan model berbasis transformer seperti XLNet dan kemampuan model BiGRU (*Bidirectional Gated Recurrent Unit*) dalam menangkap informasi sekuensial dari teks klinis. XLNet digunakan untuk menangkap representasi teks yang lebih kaya, karena XLNet mampu memproses dependensi antar kata secara bidirectional tanpa keterbatasan arsitektur yang hanya memperhitungkan urutan kata secara searah.

Penambahan lapisan BiGRU memberikan kemampuan model untuk menangkap hubungan temporal dalam catatan klinis, yang berguna untuk memahami konteks dari kata-kata atau frasa-frasa penting yang tersebar dalam teks. Mekanisme *attention* juga ditambahkan untuk membantu model dalam memfokuskan pada bagian-bagian teks yang lebih relevan dengan tugas prediksi penerimaan kembali pasien. Dengan demikian, model diharapkan mampu memaksimalkan pemahaman konteks penting dalam teks panjang yang terkandung dalam catatan klinis.

3.2.2 Perbandingan dengan Studi Sebelumnya

Dalam studi sebelumnya yang dilakukan oleh Huang et al., dengan judul "*ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission*" yang menggunakan arsitektur model seperti *Clinical*BERT, *Bag-of-words*, Bi-LSTM dan BERT, nilai ROC-AUC yang dicapai berkisar antara 0.684 hingga 0.714, nilai PR-

AUC yang dicapai berkisar antara 0.674 hingga 0.701 sedangkan nilai PR80 yang dicapai berkisar antara 0.172 hingga 0.242. Peningkatan AUC yang diperoleh dalam penelitian ini dapat dikaitkan dengan penggunaan XLNet yang mampu memanfaatkan *contextualized embeddings* lebih baik dibandingkan model *recurrent neural networks* konvensional. Selain itu, penggunaan lapisan BiGRU memungkinkan model untuk memahami urutan data secara dua arah, dan penambahan *attention mechanism* membantu model dalam memfokuskan pada bagian penting dari catatan medis.

Tabel 9. Perbandingan performa model dengan studi sebelumnya

Model	ROC-AUC	PR-AUC	PR80
ClinicalBERT[1]	0.714	0.701	0.242
Bag-of-words[1]	0.684	0.674	0.217
BiLSTM[1]	0.694	0.686	0.223
BERT[1]	0.692	0.678	0.172
XLNet-BiGRU-ATT[2]	**0.742**	**0.723**	**0.237**

Keterangan:

1. Hasil model ClinicalBERT, Bag-of-words, Bi-LSTM, dan BERT merujuk pada Huang et al. (2020), *'ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission*.'
2. Hasil model *XLNet-BiGRU-Attention* merupakan hasil penelitian ini.

3.2.3 Interpretasi Data Hasil Pengujian Model

Pada bagian ini, dilakukan interpretasi terhadap hasil pengujian model prediksi penerimaan kembali pasien yang telah dibangun. Interpretasi ini mencakup perbandingan antara nilai aktual (*ground truth*) dan nilai prediksi (*predicted output*) dari model yang digunakan.

Model yang digunakan dalam penelitian ini adalah XLNet-BiGRU-ATT, yang telah dilatih menggunakan data catatan klinis pasien dari dataset MIMIC III. Pengujian dilakukan dengan menggunakan beberapa metrik evaluasi, seperti ROC-AUC, PR-AUC, dan RP80. Metrik ini digunakan untuk menilai seberapa baik model dapat memprediksi kemungkinan seorang pasien akan diterima kembali setelah keluar dari rumah sakit berdasarkan catatan klinis.

Dalam tahap pengujian, model menghasilkan *logits* yang kemudian diubah menjadi probabilitas menggunakan fungsi aktivasi sigmoid untuk mendapatkan probabilitas penerimaan kembali. Probabilitas ini dibandingkan dengan label aktual yang menunjukkan apakah pasien benar-benar diterima kembali atau tidak.

Tabel 10. Data hasil pengujian model

Catatan Klinis	Label Aktual	Nilai Prediksi (0-1)	Label Prediksi
laboratories on admission were a white count 1, hematocrit 4, hematocrit 57, platelet count 57, chem-7 with 131/3/97/23/16/8 and alt 14, ast 40, alk phos 112, total bilirubin 9, albumin of 6, and amylase of	1	0.6166	1 (TP)

Lanjutan Tabel 10. Data hasil pengujian model

Catatan Klinis	Label Aktual	Nilai Prediksi (0-1)	Label Prediksi
the patient's inr was hospital course: on hospital day five it was discovered the patient was mrsa bacteremic. infectious disease was consulted to evaluate patient. the patient was continued to be worked up, getting a bone scan and _ to evaluate for possible sources of infection. her vancomycin level was titrated. she was transferred to the medical intensive care unit on the for close monitoring, of pa catheter and for acute renal impairment with a creatinine that went from 8 to the patient's renal function improved over a period of time returning to a baseline of the patient was transferred to the floor and prepped for an orthotopic liver transplant. on hospital day 17 the patient was being pre-op'd for orthotopic liver transplant and was given the appropriate preoperative medications. on hospital day 17 and postoperative day one, the patient did not receive her liver secondary to development of a large clot intraoperatively. as such, the patient was taken out of the operating room and failed to receive her transplant. the patient went back to the unit for close monitoring immediately postoperatively and was then transferred to the floor. the patient was finally transferred to the floor on the hospital day on the floor patient had a fairly unremarkable course. on hospital day 29 patient was to be discharged to an extended care facility where she will receive physical therapy and await a potential new liver for transplant. discharge medications: ketaconazole cream. acetaminophen 325 mg two tabs p.o. q. 4-6h. p.r.n. morphine sulfate 2 mg/ml syringe one to two injections q. 4h. miconazole powder. ciprofloxacin 250 mg tabs p.o. b.i.d. protonix 40 mg one tab.	1	0.6166	1 (TP)
his dry wt. was 79kg before all of these hospitalizations. beta blocker and lisinopril were held initially due to concern for acute decompensated heart failure and increasing creatinine, but were restarted when his cr stabilized around baseline and he was diuresing well.. # shock: resolved. pt off of pressors >48hr before admission to the floor. suspect primary primary cause was sepsis due to ecoli bacteremia. wbc trended down & pt afebrile on abx since leaving icu.	0	0.6148	1(FP)

Lanjutan Tabel 10. Data hasil pengujian model

Catatan Klinis	Label Aktual	Nilai Prediksi (0-1)	Label Prediksi
suspect also, evidence of cardiogenic component to shock w/ low svo however, echo w/o evidence of acute mi & ef stable. we changed his ceftazidime to ceftriaxone and then to oral cefpodoxime to finish out 2 weeks course of antibiotics. some question as to whether his urosepsis originated from chronic prostatitis and so he was referred to urology.. # hematuria: was due to foley trauma originally, but then he continued to have hematuria intermittently. he has been scheduled with urology doctor for for evaluation of his difficulty urinating, hematuria and history of urosepsis, question of chronic prostatitis as he appeared to have a somewhat positive ua (wbc's) after being on antibiotics for several days with a moderately tender prostate. pt. was explicitly instructed that he needs to keep this appointment as he may need a workup to rule out malignancy. hct remained stable and was at a on day of d/c. # anemia: baseline hct varies, though predominantly in 30s. was 2 on day of d/c iron studies indicative of anemia of chronic disease. # crf: stable. pt.'s cr remained approximately at baseline for the duration of his admission and was 5 on day of d/c.. # s/p r hip replacement: appears to be healing well. staples removed. pt. needs to remain anticoagulated for dvt prophylaxis until on warfarin. pt. was discharged on 5mg warfarin daily for one more dose and then back to 3mg daily.	0	0.6148	1(FP)
date of birth: sex: f service: cardiothoracic allergies: heparin agents attending: chief complaint: abdominal pain major surgical or invasive procedure: replacement of ascending aorta with a 28-mm gelweave dacron graft. bentall procedure with a composite st. mechanical graft, size 21 mm, reference number. tunneled hemodialysis catheter placement- history of present illness: this patient is a 48 year old female who complains of abdominal pain. patient presents with abdominal pain to an outside hospital. patient reports having intermittent abdominal pain became more constant over last day. patient underwent a ct scan which showed a descending aortic aneurysm. patient transferred to. ct reviewed and found to show type a dissection starting at the root and extending to the iliac	0	0.7868	0(TN)

Lanjutan Tabel 10. Data hasil pengujian model

Catatan Klinis	Label Aktual	Nilai Prediksi (0-1)	Label Prediksi
bifurcation. she was brought emergently to the operating room for repair. past medical history: mild mental retardation, hypertension social history: lives independently with husband works at stop and shop cigarettes: no etoh: no family history: family history: heart disease; htn physical exam: general: awake, somewhat anxious skin: dry intact heent: perrla eomi neck: supple full rom chest: lungs clear bilaterally heart : rrr irregular murmur grade abdomen: soft non-distended non-tender bowel sounds + extremities: warm, well-perfused no edema neuro: grossly intact pulses: femoral right: 2 left: 2 dp right: 2 left:2 pt : 2 left:2 radial right: 2 left:2 carotid bruit right: no left: no pertinent results: 07:10am blood wbc-2 rbc-49 hgb-5* hct-5* mcv-88 mch-2 mchc-5 rdw-4 plt ct-434 05:50am blood wbc-3 rbc-54* hgb-6* hct-5* mcv-86 mch-1 mchc-0 rdw-4 plt ct-381 06:10am blood wbc-2 rbc-26* hgb-9* hct-1* mcv*	0	0.7868	0(TN)
retained hardware, 8 weeks of iv antibiotic therapy was recommended by the id service, followed by life-long oral suppressive therapy. his antibiotic course will be: cefazolin 2 grams every 8 hours x 8 weeks post-debridement(to), followed by lifelong suppressive therapy for retained infected pacing wires. he continued to progress and was ready for discharge to rehab on. medications on admission: atorvastatin 10 mg daily furosemide 20 mg daily ipratropium-albuterol lisinopril 5 mg tablet - 5 (one half) tablet(s) by mouth daily metoprolol succinate - 25 mg daily aspirin 81 mg tablet - one tablet(s) by mouth daily cholecalciferol (vitamin d3) docusate sodium 100 mg twice a day lactobacillus rhamnosus gg 1 capsule(s) by mouth daily multivitamin discharge medications: acetaminophen 325 mg tablet sig: 1-2 tablets po q4h (every 4 hours) as needed for pain. ranitidine hcl 150 mg tablet sig: one (1) tablet po bid (2 times a day). docusate sodium 100 mg capsule sig: one (1) capsule po bid (2 times a day). atorvastatin 10 mg tablet sig: one (1) tablet po daily (daily). heparin (porcine) 5,000 unit/ml solution sig: one (1) ml injection tid (3 times a day). lisinopril 10 mg tablet sig: one (1) tablet po daily (daily). metoprolol	1	0.7055	0(FN)

Lanjutan Tabel 10. Data hasil pengujian model

Catatan Klinis	Label Aktual	Nilai Prediksi (0-1)	Label Prediksi
tartrate 50 mg tablet sig: 5 tablets po tid (3 times a day): 75 mg tid. nystatin 100,000 unit/ml suspension sig: five (5) ml po qid (4 times a day). aspirin 325 mg tablet, delayed release (e.c.) po daily (daily). sodium chloride 65 % aerosol.	1	0.7055	0(FN)

Dari tabel di atas, dapat dilihat bahwa nilai aktual yang dimiliki oleh masing-masing pasien (0 atau 1) dibandingkan dengan hasil prediksi dari model. Model mengeluarkan probabilitas penerimaan kembali, dan berdasarkan ambang batas yang telah ditetapkan, probabilitas tersebut diklasifikasikan menjadi 0 (tidak diterima kembali) atau 1 (diterima kembali).

Untuk memahami mengapa kesalahan klasifikasi terjadi pada ***False Negative*** **(FN)** dan ***False Positive*** **(FP)**, perlu mempertimbangkan konteks dari topik prediksi penerimaan kembali pasien menggunakan catatan klinis dengan model **XLNet-BiGRU-ATT** pada data **MIMIC III**. Beberapa kecenderungan kesalahan disebabkan oleh:

1. *False Negative* (FN): Label 1 yang diklasifikasikan sebagai 0

Ini berarti model mengklasifikasikan pasien yang **seharusnya dirawat kembali** sebagai **tidak dirawat kembali**.

a) Fitur Penting Tidak Ditangkap oleh Model: Ada faktor penting dalam catatan klinis yang menggambarkan kondisi pasien yang serius atau berisiko tinggi, tetapi tidak cukup jelas dalam representasi token atau *embedding* dari model. Misalnya, informasi medis penting yang tersirat dalam teks atau ada istilah medis yang jarang muncul sehingga tidak tertangkap dengan baik oleh model.

b) Kondisi tidak umum di Catatan Klinis Tertentu: Beberapa catatan klinis tidak memberikan gambaran yang cukup kuat atau pola yang jelas bahwa pasien membutuhkan perawatan ulang, meskipun dalam kenyataannya pasien itu memang dirawat kembali. Kasus-kasus ini dapat dianggap sebagai *outlier*, karena data ini menyimpang dari pola umum yang diharapkan oleh model prediksi. Misalnya, ada pasien dengan gejala yang tidak mencolok atau tanpa riwayat penyakit kritis, tetapi tetap dirawat kembali karena perkembangan kondisi yang tiba-tiba atau akibat pengaruh eksternal tambahan di luar riwayat catatan klinis pasien. *Outlier* semacam ini dapat mempersulit model dalam mendeteksi pola prediktif yang relevan dan akurat.

2. *False Positive* (FP): Label 0 yang diklasifikasikan sebagai 1

Ini berarti model mengklasifikasikan pasien yang **seharusnya tidak dirawat kembali** sebagai **dirawat kembali**.

a) *Overfitting* pada Fitur Spesifik: Model terlalu sensitif terhadap fitur-fitur tertentu yang dianggapnya penting untuk memprediksi penerimaan kembali, meskipun fitur tersebut tidak selalu menandakan hal itu. Misalnya, adanya istilah medis atau pengobatan tertentu yang sering muncul pada pasien yang dirawat kembali dapat disalahartikan sebagai faktor penting, meskipun itu tidak selalu berarti pasien akan dirawat kembali.

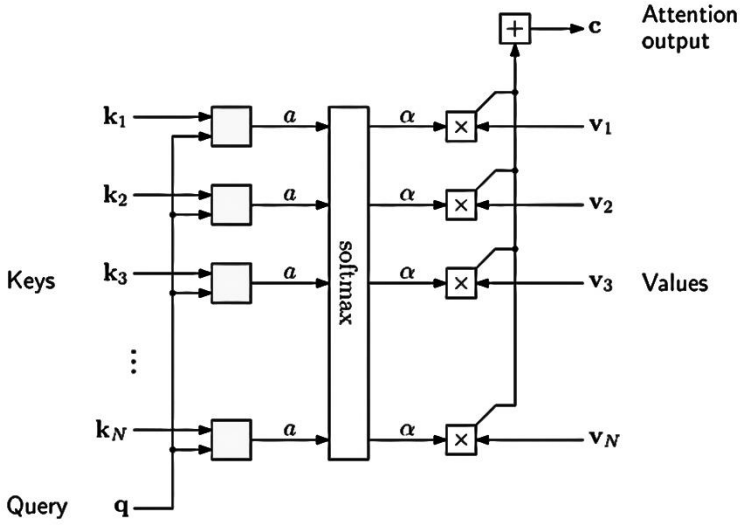
3.2.4 *Sample* Alur Sistem dan Visualisasi Model

Sample alur sistem dan visualisasi model berfungsi untuk memahami dan menggambarkan alur atau proses kerja sistem. Adapun pada visualisasi model dilakukan visualisasi *attention* yang berfungsi untuk memahami keputusan model dengan menampilkan skor perhatian (*attention scores*) yang digunakan model untuk fokus pada bagian teks tertentu.

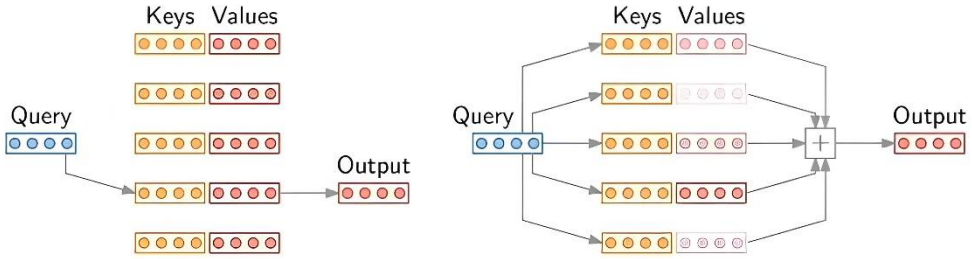
Tahap 1: *Preprocessing* dan Tokenisasi

a) *Input* Teks: "*He has experienced acute on chronic diastolic heart failure in the setting of volume overload due to his sepsis*."
b) Proses Tokenisasi: Teks diubah menjadi representasi numerik (token *IDs*) yang dapat dipahami oleh model.
    1. Contoh Tokenisasi:
    $Token\ IDs$: $[103, 125, 2315, 1312, 1123, 3214, 6532, 1200, 2316, 110, 120, 3335, 50, 2600, 2761, 999, 75, 122, 4321]$
    2. *Embedding*: Setiap token ID diterjemahkan menjadi vektor *embedding* dengan dimensi tetap, misalnya 768. Proses ini menghasilkan vektor *embedding* awal untuk tiap kata dalam kalimat.

Tahap 2: XLNet *Encoding*



Gambar 22. *Computational graph*



Gambar 23. *Attention recap*

1. Perhitungan *Attention* untuk *Content Stream*

a) Perhitungan *attention* untuk *content stream* diwakili oleh persamaan berikut:

$$g_{z_t}^{(m)} \leftarrow Attention\,(Q = g_{z_t}^{(m-1)}, KV = h_{z<t}^{(m-1)};\, \theta \tag{31}$$

Dimana:

a) $g_{z_t}^{(m)}$: *output content stream* untuk token pada *layer* ke-m

b) $Q = g_{z_t}^{(m-1)}$: *query* menggunakan representasi *content stream* dari *layer* sebelumnya

c) $KV = h_{z<t}^{(m-1)}$: *key* dan *value* menggunakan representasi *query stream* dari *layer* sebelumnya

d) θ : mewakili parameter yang dapat dilatih dalam operasi *attention*

Dalam XLNet, embedding token melewati mekanisme *self-attention*, di mana tiap kata "memperhatikan" kata lain dalam kalimat untuk menangkap konteks yang lebih luas. Tiap token embedding diproyeksikan menjadi tiga vektor utama: *Query* (Q), *Key* (K), dan *Value* (V). Misalnya, untuk token "*heart*" dan "*failure*".

Query (Q) dari $g^{(m-1)}$ untuk "*heart*":

$$Q = [0.5,0.3,0.1,0.1]$$

Key (K) dari $h^{(m-1)}$ untuk "*failure*":

$$K = [0.3,0.4,0.2,0.1]$$

Value (V) dari $h^{(m-1)}$ untuk "*failure*":

$$V = [0.3,0.4,0.2,0.1]$$

b) Hitung *Attention Score*

Menghitung skor perhatian antara token menggunakan:

$$Attention\,(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{32}$$

Dimana d_k = 4 (dimensi vector)

Untuk nilai *Query* (Q) dan *Key* (K), misalnya dipetakan sebagai berikut:

Tabel 11. Pemetaan nilai *Query* (Q) dan *Key* (K)

	Q_1	Q_2	Q_3	…	Q_n
K_1	$\frac{Q_1K_1}{\sqrt{d_k}}$	$\frac{Q_2K_1}{\sqrt{d_k}}$	$\frac{Q_3K_1}{\sqrt{d_k}}$	…	$\frac{Q_nK_1}{\sqrt{d_k}}$
K_2	$\frac{Q_1K_2}{\sqrt{d_k}}$	$\frac{Q_2K_2}{\sqrt{d_k}}$	$\frac{Q_3K_2}{\sqrt{d_k}}$	…	$\frac{Q_nK_2}{\sqrt{d_k}}$
K_3	$\frac{Q_1K_3}{\sqrt{d_k}}$	$\frac{Q_2K_3}{\sqrt{d_k}}$	$\frac{Q_3K_3}{\sqrt{d_k}}$	…	$\frac{Q_nK_3}{\sqrt{d_k}}$
…	…	…	…	…	…

Maka, dapat dihitung skor perhatian antara token "*heart*" dengan token lain, misalnya "*failure*", sebagai berikut:

$$\begin{aligned} Q \cdot K^T &= (0.5 \times 0.3) + (0.3 \times 0.4) + (0.1 \times 0.2) + (0.1 \times 0.1) \\ &= 0.15 + 0.12 + 0.02 + 0.01 \\ &= 0.30 \end{aligned}$$

$$Score = \frac{0.30}{\sqrt{4}} = 0.15$$

c) Aplikasi *softmax*

$$\alpha = softmax(0.15) = \frac{e^{0.15}}{e^{0.15}} = 1.0$$

d) Output final

$$g_{z_t}^{(m)} = \alpha \times V = 1.0 \times [0.3,0.4,0.2,0.1] = [0.3,0.4,0.2,0.1]$$

2. Perhitungan *Attention* untuk *Query Stream*

a) Perhitungan *attention* untuk *content stream* diwakili oleh persamaan berikut:

$$h_{z_t}^{(m)} \leftarrow Attention\,(Q = g_{z_t}^{(m-1)}, KV = h_{z \le t}^{(m-1)};\, \theta \qquad (33)$$

Dimana:

a) $h_{z_t}^{(m)}$: *output query stream* untuk token pada *layer* ke-m
b) $Q = g_{z_t}^{(m-1)}$: *query* menggunakan representasi *query stream* dari layer sebelumnya
c) $KV = h_{z \le t}^{(m-1)}$: *key* dan *value* menggunakan representasi query stream dari layer sebelumnya
d) θ : mewakili parameter yang dapat dilatih dalam operasi *attention*

Persiapan Vector:

Query (Q) dari $g^{(m-1)}$ untuk "*heart*":

$$Q = [0.5,0.3,0.1,0.1]$$

Key (K) dari $h^{(m-1)}$ untuk "*failure*":

$$K = [0.3,0.4,0.2,0.1]$$

Value (V) dari $h^{(m-1)}$ untuk "*failure*":

$$V = [0.3,0.4,0.2,0.1]$$

b) Hitung *Attention Score*
Dimana d_k = 4 (dimensi vector)

$$\begin{aligned} Q \cdot K^T &= (0.4 \times 0.4) + (0.3 \times 0.4) + (0.2 \times 0.1) + (0.1 \times 0.1) \\ &= 0.16 + 0.12 + 0.02 + 0.01 \\ &= 0.31 \end{aligned}$$

$$Score = \frac{0.31}{\sqrt{4}} = 0.155$$

c) Aplikasi *softmax*

$$\alpha = softmax(0.15) = \frac{e^{0.155}}{e^{0.155}} = 1.0$$

d) Output final

$$h_{z_t}^{(m)} = \alpha \times V = 1.0 \times [0.4,0.4,0.1,0.1] = [0.4,0.4,0.1,0.1]$$

Keterangan:

1. Perhitungan di atas hanya untuk 1 pasang kata ("*heart*" dan "*failure*".)
2. Dalam praktik nyata:
   - Perhitungan dilakukan untuk semua pasangan kata dalam kalimat
   - Dimensi *vector* lebih besar (contoh: 512 atau 768)
   - Ada *multiple attention heads* yang bekerja paralel
3. $\sqrt{d_k}$ adalah scaling factor untuk menstabilkan gradien
4. θ adalah parameter yang dapat dilatih dalam model

Tahap 3: BiGRU Layer

Pada lapisan ini, output dari XLNet diteruskan ke BiGRU untuk menangkap informasi konteks dari dua arah: maju dan mundur.

Proses *Forward* dan *Backward*:

a) *Forward Pass*: BiGRU mengolah kata dari kiri ke kanan. Misalnya, untuk token "*heart*":

$$h_{forward} = GRU_{forward}\ (XLNet\ Output_{heart}, h_{forward,previous}) \qquad (34)$$

Misalnya, $h_{forward} = [0.2, 0.5, -0.3, \ldots]$.

b) *Backward Pass*: Proses yang sama diterapkan dari kanan ke kiri, menghasilkan:

$$h_{backward} = GRU_{backward}\ (XLNet\ Output_{heart}, h_{backward\ previous}) \quad (35)$$

Misalnya, $h_{backward} = [-0.1, 0.4, 0.6, \ldots]$.

c) Penggabungan *Forward* dan *Backward*: *Output* dari dua arah digabungkan:

$$h_{BiGRU} = [h_{forward}\ ;\ h_{backward}] \quad (36)$$

Misalnya, $h_{BiGRU} = [0.2, 0.5, -0.3, -0.1, 0.4, 0.6, \ldots]$, yang mewakili hubungan yang lebih baik antar kata dengan konteks yang luas.

Tahap 4: *Attention Mechanism*

Attention Mechanism memberikan bobot lebih tinggi pada kata-kata yang relevan.

a) Menghitung Skor Attention: Untuk setiap *timestep*, misalnya *timestep* ke-7 ("*heart*"), dihitung skor *attention* dengan:

$$e_7 = \tanh(W \cdot\ h_7 + b) \quad (37)$$

Dimana:

a) e_t adalah nilai output (atau aktivasi) pada waktu t
b) tanh adalah fungsi hiperbolik tangen, yang merupakan fungsi aktivasi nonlinear
c) W adalah matriks bobot (*weight matrix*) yang digunakan untuk menghitung kombinasi linier dari input h_t
d) h_t adalah vektor masukan (input) pada waktu t
e) b adalah vektor bias (*bias vector*) yang ditambahkan ke hasil kombinasi linier

Misalnya, $e_7 = 0.65$

b) Normalisasi (*Softmax*), dimana : α_t adalah skor attention yang dihitung dari representasi tersembunyi diwaktu t.

$$\alpha_7 = \frac{\exp(e_7)}{\sum_{i=1}^{T} \exp(e_i)} \quad (38)$$

Jika totalnya adalah 4.2, maka:

$$\alpha_7 = \frac{\exp(0.65)}{4.2} \approx 0.23$$

c) *Weighted sum* untuk *context vector*, dimana vektor konteks yang dihasilkan dari penjumlahan tertimbang representasi tersembunyi oleh bobot *attention* dan h_t adalah representasi tersembunyi diwaktu t.

$$context = \sum_{t=1}^{T} \alpha_t h_t \quad (39)$$

Jika $h_7 = [0.2, 0.5, -0.3, \dots]$ dan $\alpha_7 = 0.23$, maka kontribusi *timestep* ke-7 adalah:

$$\alpha_7 \cdot h_7 = 0.23 \cdot [0.2, 0.5, -0.3, \dots] = [0.046, 0.115, -0.069, \dots]$$

Tahap 5: Prediksi Akhir

Context vector ini digunakan pada lapisan sigmoid untuk menghasilkan probabilitas:

$$probabilitas = \sigma\,(W_{output} \cdot context + b_{output}) \quad (40)$$

Misalnya, hasil akhirnya adalah 0.72, berarti prediksi adalah positif untuk rawat inap lanjutan.

Contoh Perhitungan Metrik Evaluasi: ROC-AUC, PR-AUC, dan PR80:

Misalkan setelah tahap akhir dari prediksi (setelah melalui XLNet, BiGRU, dan *Attention Mechanism*), model menghasilkan skor probabilitas untuk beberapa sampel pasien apakah mereka akan dirawat inap kembali atau tidak. Probabilitas ini digunakan untuk menghitung beberapa metrik.

1. Mendapatkan Prediksi dari Model

   Misalkan beberapa sampel data dengan skor probabilitas yang dihasilkan oleh model, dan ditentukan *threshold* (ambang batas) tertentu untuk mengklasifikasikan prediksi sebagai positif atau negatif. Berikut adalah tabel simulasi prediksi model:

   Tabel 12. Sampel data dengan skor probabilitas yang dihasilkan model

Sample	Skor Probabilitas	Threshold	Prediksi Model	Label Aktual
1	0.95	0.5	Positif	Positif
2	0.80	0.5	Positif	Positif
3	0.60	0.5	Positif	Negatif
4	0.40	0.5	Negatif	Negatif
5	0.30	0.5	Negatif	Positif
6	0.20	0.5	Negatif	Negatif

   Dari tabel ini, bisa dihitung TP, FP, TN, dan FN pada *threshold* 0.5.

   a) *True Positives* (TP): Prediksi Positif yang sesuai dengan label Positif = 2 (*Sampel* 1 dan 2).
   b) *False Positives* (FP): Prediksi Positif yang tidak sesuai dengan label (seharusnya Negatif) = 1 (*Sampel* 3).
   c) *True Negatives* (TN): Prediksi Negatif yang sesuai dengan label Negatif = 2 (*Sampel* 4 dan 6).
   d) *False Negatives* (FN): Prediksi Negatif yang tidak sesuai dengan label (seharusnya Positif) = 1 (*Sampel* 5).

2. Menghitung Metrik Dasar pada Berbagai *Threshold*
   Dari nilai TP, FP, TN, dan FN, bisa dihitung beberapa metrik dasar seperti *True Positive Rate* (TPR) dan *False Positive Rate* (FPR), yang diperlukan untuk menghitung ROC-AUC, serta *Precision* dan *Recall* untuk menghitung PR-AUC dan PR80.
   Misalkan dilakukan pengujian pada tiga *threshold*: 0.3, 0.5, dan 0.7.
   1) *Threshold* 0.3
      a) Prediksi model pada *threshold* 0.3:
         Semua skor di atas 0.3 diklasifikasikan sebagai positif.
         Prediksi: [Positif, Positif, Positif, Positif, Positif, Negatif]
      b) Confusion matrix pada *threshold* 0.3:
         TP = 3 (*Sample* 1, 2, dan 5)
         FP = 2 (*Sample* 3 dan 4)
         TN = 1 (*Sample* 6)
         FN = 0
      c) Metrik pada *threshold* 0.3:

$$True\ Positive\ Rate\ (TPR)/\ Recall\ = \frac{TP}{TP + FN} = \frac{3}{3} = 1.0$$

$$False\ Positif\ Rate\ (FPR) = \frac{FP}{FP + TN} = \frac{2}{3} = 0.67$$

$$Precision = \frac{TP}{TP + FP} = \frac{3}{5} = 0.6$$

   2) *Threshold* 0.5
   a) *Confusion matrix* pada *threshold* 0.5:
      TP = 2 (*Sample* 1 dan 2)
      FP = 1 (*Sample* 3)
      TN = 2 (*Sample* 4 dan 6)
      FN = 1 (*Sample* 5)
   b) Metrik pada *threshold* 0.5:

$$TPR = \frac{2}{3} = 0.67$$

$$FPR = \frac{1}{3} = 0.33$$

$$Precision\ = \frac{2}{3} = 0.67$$

   3) *Threshold* 0.7
      a) *Confusion matrix* pada *threshold* 0.7:
      TP = 1 (*Sample* 1)
      FP = 0

TN = 3 (*Sample* 3, 4, dan 6)
FN = 1 (*Sample* 2 dan 5)

c) Metrik pada *threshold* 0.7:

$$TPR = \frac{1}{3} = 0.33$$

$$FPR = \frac{0}{3} = 0.0$$

$$Precision = \frac{1}{1} = 1.0$$

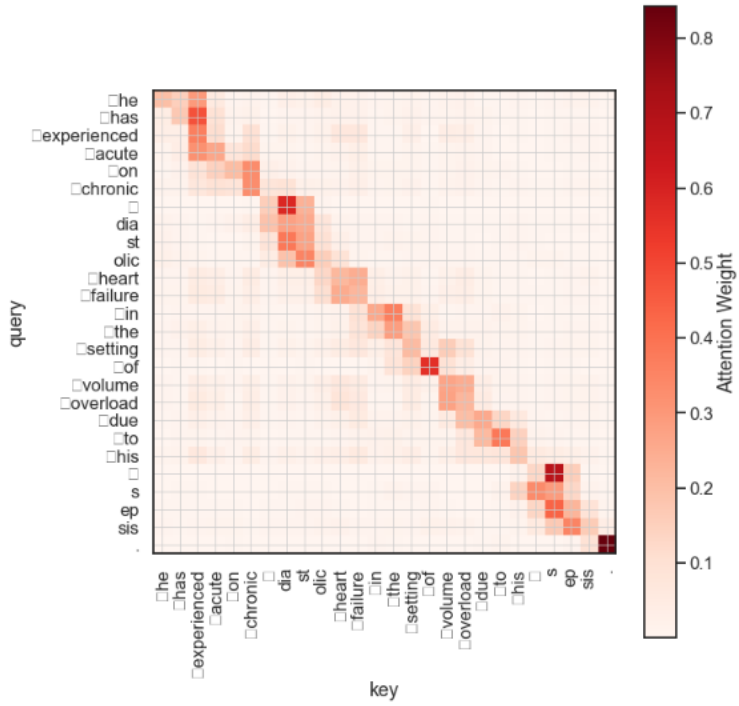
3. Menghitung ROC-AUC
   ROC-AUC adalah luas di bawah kurva yang memplot TPR vs FPR pada berbagai *threshold*. Berdasarkan tiga titik dari threshold di atas:
   a) *Threshold* 0.3: TPR = 1.0, FPR = 0.67
   b) *Threshold* 0.5: TPR = 0.67, FPR = 0.33
   c) *Threshold* 0.7: TPR = 0.33, FPR = 0.0
   Dengan menghubungkan titik-titik ini dan menghitung area di bawahnya, dapat dihitung nilai AUC, yang pada contoh ini dapat dihitung sebagai 0.78. Ini menunjukkan performa yang cukup baik.
4. Menghitung PR-AUC
   PR-AUC adalah luas di bawah kurva *Precision-Recall*, yang memplot *Precision* vs *Recall* pada berbagai *threshold*. Dari contoh di atas:
   a) *Threshold* 0.3: *Precision* = 0.6, *Recall* = 1.0
   b) *Threshold* 0.5: *Precision* = 0.67, *Recall* = 0.67
   c) *Threshold* 0.7: *Precision* = 1.0, *Recall* = 0.33
   Dengan menghitung area di bawah kurva ini, bisa didapatkan nilai PR-AUC. Pada contoh ini, PR-AUC kira-kira bernilai 0.72.
5. Menghitung PR80
   PR80 mengukur *Recall* saat *Precision* mencapai 80%. Dari tabel di atas, pada *threshold* antara 0.5 dan 0.7, mendekati atau melebihi *Precision* 80%:
   a) Pada *Threshold* 0.5: *Precision* 67%
   b) Pada *Threshold* 0.7: *Precision* 100%, *Recall* 33%
   Sehingga PR80 pada nilai *Recall* 33% (dengan *Precision* yang memenuhi syarat di *threshold* 0.7).

Visualisasi *Attention:*

Dalam konteks XLNet-BiGRU-ATT, berikut langkah-langkah visualisasi *attention*:

a) Menghitung Skor Perhatian: Setelah model XLNet dan BiGRU memproses teks, *output* berupa *hidden states* dilewatkan ke mekanisme perhatian (*attention mechanism*). Mekanisme ini memberikan bobot ke setiap token, menunjukkan pentingnya token tersebut dalam konteks keseluruhan teks.

b) Visualisasi dengan *Heatmap*: Setelah mendapatkan *attention scores*, dibuat *heatmap* yang memperlihatkan interaksi antar token berdasarkan *query* dan *key*. Dalam konteks ini, *query* adalah token yang sedang diperiksa, sedangkan *key* adalah token yang dilihat untuk memutuskan seberapa relevan atau penting. Heatmap memungkinkan untuk melihat distribusi perhatian antar token dalam teks.



Gambar 24. Visualisasi *attention*

c) Interpretasi *Heatmap*: Jika model memprediksi *readmission* pasien, bisa dilihat bagian mana dari teks (seperti nama obat atau dosis) yang model anggap penting untuk prediksi. Token dengan nilai perhatian tinggi (warna lebih terang pada *heatmap*) adalah token yang dianggap penting oleh model untuk pengambilan keputusan. Misalnya, dalam visualisasi attention yang ditunjukkan pada gambar 25, contoh kalimat yang digunakan yaitu: "*he has experienced acute on chronic diastolic heart failure in the setting of volume overload due to his sepsis*" dimana **"***sepsis***"** dan **"***chronic diastolic***"** diberi perhatian lebih, ini menunjukkan bahwa model menganggap informasi tersebut sangat relevan untuk memprediksi readmission pasien.

Visualisasi ini bertujuan agar manusia (dokter, peneliti) dapat memahami proses pengambilan keputusan model, memberikan transparansi pada model berbasis *deep learning* yang biasanya bersifat *black-box*.

3.2.5 Interpretabilitas Model

Dalam konteks penerapan model prediksi berbasis *deep learning* di bidang klinis, seperti untuk memprediksi *readmission* pasien, interpretabilitas menjadi isu yang sangat penting. Hal ini dikarenakan:

a) Keputusan klinis yang kritis: Dokter mengandalkan informasi yang jelas dan dapat dibenarkan untuk membuat keputusan terkait perawatan pasien. Model *deep learning* seperti XLNet yang digunakan adalah *black-box model*, artinya sulit untuk memahami bagaimana model mencapai kesimpulan tertentu. Ini memicu skeptisisme, terutama dalam lingkungan yang membutuhkan akuntabilitas seperti perawatan kesehatan.

b) Mekanisme *Attention* untuk Interpretabilitas: Mekanisme *attention* memungkinkan interpretasi yang lebih transparan karena *attention scores* mengungkapkan bagian mana dari data yang model fokuskan. Dalam model seperti XLNet-BiGRU-ATT, dapat dilihat token-token mana yang memengaruhi prediksi. Dengan memvisualisasikan *attention*, model *deep learning* dapat memberi pengguna (dokter, peneliti) alasan yang lebih kuat tentang mengapa prediksi tertentu dibuat, mengurangi ketidakpercayaan pada model, dan membantu dalam pengambilan keputusan yang berbasis data.

c) Alur Interpretasi: Mekanisme ini juga memungkinkan evaluasi lebih lanjut oleh dokter. Misalnya, jika model secara konsisten memberikan perhatian tinggi pada aspek-aspek yang tidak relevan, ini bisa mengindikasikan masalah pada model atau data, yang dapat diperbaiki di masa mendatang.

BAB IV
KESIMPULAN DAN SARAN

4. 1 Kesimpulan

Penelitian ini berhasil membuktikan bahwa model XLNet-BiGRU-ATT memiliki potensi besar dalam memprediksi kemungkinan pasien diterima kembali di rumah sakit berdasarkan catatan medis mereka. Arsitektur ini memanfaatkan keunggulan masing-masing komponen, yaitu XLNet, BiGRU, dan *attention mechanism*, untuk menangkap informasi sekuensial secara lebih mendalam. Hasil terbaik diperoleh pada *epoch* ke-4 dengan nilai ROC-AUC 0,742, PR-AUC 0,723, dan PR80 0,237, menunjukkan kemampuan yang baik dalam membedakan pasien berisiko tinggi untuk diterima kembali.

Parameter model diinisialisasi berdasarkan penelitian "*ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission*" oleh Huang et al., (2020) yang menggunakan objek penelitian serupa, yaitu prediksi penerimaan kembali pasien rumah sakit. Pengaturan *learning rate* sebesar 2e-5 terbukti memberikan performa yang optimal pada *epoch* ke-4. Pemilihan panjang urutan maksimum 512 juga relevan dengan catatan medis yang cenderung panjang. Model ini menunjukkan keunggulan signifikan dibandingkan model sebelumnya, terutama berkat penggunaan XLNet *pre-trained* yang unggul dalam menangkap konteks melalui *permuted language modeling* dan *two-stream self-attention*, membantu memahami pola dalam catatan medis yang kompleks. BiGRU menambah kemampuan dalam memahami hubungan temporal secara *bidirectional*, memungkinkan model menangkap pola kronologis penting seperti riwayat penyakit dan perkembangan kondisi pasien. Penerapan *attention mechanism* setelah BiGRU meningkatkan kinerja dengan memfokuskan perhatian pada informasi yang lebih relevan, sehingga prediksi menjadi lebih akurat.

Dalam praktik klinis, model ini dapat digunakan sebagai bahan pertimbangan yang mendukung pengambilan keputusan rumah sakit, khususnya manajemen pasien berisiko tinggi. Prediksi risiko memungkinkan intervensi yang lebih tepat, seperti konsultasi tambahan, penyesuaian pengobatan, atau pemberian instruksi perawatan, yang pada akhirnya meningkatkan kualitas layanan dan menurunkan biaya operasional akibat penerimaan kembali pasien.

Secara keseluruhan, kombinasi model XLNet-BiGRU-ATT terbukti efektif dalam memprediksi penerimaan kembali pasien berdasarkan catatan medis. Hasil penelitian ini menunjukkan potensi besar *deep learning* dalam meningkatkan kualitas dan efisiensi layanan kesehatan. Penelitian lebih lanjut dapat diarahkan pada optimalisasi model dengan arsitektur lebih kompleks, augmentasi data, atau integrasi dengan data *non*-tekstual atau genetik.

4. 2 Saran

Terdapat beberapa saran yang dapat dilakukan untuk penelitian selanjutnya berdasarkan hasil penelitian yang ada, diantaranya:

1. Melakukan implementasi hasil pemodelan ke dalam bentuk aplikasi nyata sebagai alat bantu yang memberikan kemudahan dalam melakukan prediksi penerimaan kembali pasien dan sebagai sarana untuk mengevaluasi kinerja model dalam keadaan yang sebenarnya.
2. Melakukan *tuning hyperparameter*, dimana pelatihan ini hanya dilakukan pada satu *set hyperparameter* yang diambil dari penelitian sebelumnya tanpa eksplorasi lebih lanjut terhadap variasi *hyperparameter* lainnya, karena keterbatasan sumber daya GPU yang disebabkan ukuran model yang besar dan penggunaan sequence length catatan medis yang panjang. Meskipun performa model XLNet-BiGRU-ATT menunjukkan hasil yang lebih tinggi dibandingkan dengan model lain, masih ada potensi peningkatan jika dilakukan *tuning hyperparameter* lebih lanjut. Eksplorasi terhadap berbagai kombinasi *hyperparameter* seperti *learning rate, batch size*, atau jumlah *hidden units* mungkin dapat meningkatkan performa model lebih jauh.
3. Pengembangan konsep *multilingual*, yang memungkinkan model prediksi memiliki kemampuan memproses teks dari berbagai bahasa. Hal ini akan meningkatkan generalisasi model, terutama dalam konteks data klinis lintas negara atau wilayah dengan populasi yang berbicara berbagai bahasa. Dengan demikian, model dapat dioptimalkan untuk aplikasi yang lebih luas dan relevan di berbagai lokasi geografis.

DAFTAR PUSTAKA

Chatzimina, M. E., Papadaki, H. A., Pontikoglou, C., & Tsiknakis, M. (2024). A Comparative Sentiment Analysis of Greek Clinical Conversations Using BERT, RoBERTa, GPT-2, and XLNet. *Bioengineering*, *11*(6), Article 6. https://doi.org/10.3390/bioengineering11060521

Gogineni, H., Aranda, J. P., & Garavalia, L. S. (2019). Designing professional program instruction to align with students' cognitive processing. *Currents in Pharmacy Teaching and Learning*, *11*(2), 160–165. https://doi.org/10.1016/j.cptl.2018.11.015

González-Nóvoa, J. A., Campanioni, S., Busto, L., Fariña, J., Rodríguez-Andina, J. J., Vila, D., Íñiguez, A., & Veiga, C. (2023). Improving Intensive Care Unit Early Readmission Prediction Using Optimized and Explainable Machine Learning. *International Journal of Environmental Research and Public Health*, *20*(4), Article 4. https://doi.org/10.3390/ijerph20043455

Han, T., Zhang, Z., Ren, M., Dong, C., Jiang, X., & Zhuang, Q. (2023). Text Emotion Recognition Based on XLNet-BiGRU-Att. *Electronics*, *12*(12), Article 12. https://doi.org/10.3390/electronics12122704

Huang, K., Altosaar, J., & Ranganath, R. (2020). *ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission* (arXiv:1904.05342). arXiv. http://arxiv.org/abs/1904.05342

Johnson, A., Pollard, T., & Mark, R. (2015). *MIMIC-III Clinical Database* (Version 1.4) [Dataset]. PhysioNet. https://doi.org/10.13026/C2XW26

Lorkowski, J., & Pokorski, M. (2022). Medical Records: A Historical Narrative. *Biomedicines*, *10*(10), Article 10. https://doi.org/10.3390/biomedicines10102594

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). *Deep Learning Based Text Classification: A Comprehensive Review* (arXiv:2004.03705). arXiv. https://doi.org/10.48550/arXiv.2004.03705

Necula, S.-C., Dumitriu, F., & Greavu-Șerban, V. (2024). A Systematic Literature Review on Using Natural Language Processing in Software Requirements Engineering. *Electronics*, *13*(11), Article 11. https://doi.org/10.3390/electronics13112055

Pencatatan SOAP Rekam Medis dan Contoh Penggunaannya—eClinic. (2023, November 17). https://www.eclinic.id/pencatatan-soap-rekam-medis/

Podder, V., Lew, V., & Ghassemzadeh, S. (2024). SOAP Notes. In *StatPearls*. StatPearls Publishing. http://www.ncbi.nlm.nih.gov/books/NBK482263/

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2023). *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer* (arXiv:1910.10683). arXiv. https://doi.org/10.48550/arXiv.1910.10683

Rhazzafe, S., Caraffini, F., Colreavy-Donnelly, S., Dhassi, Y., Kuhn, S., & Nikolov, N. S. (2024). Hybrid Summarization of Medical Records for Predicting Length

of Stay in the Intensive Care Unit. *Applied Sciences*, *14*(13), Article 13. https://doi.org/10.3390/app14135809

Sarzaeim, P., Mahmoud, Q. H., Azim, A., Bauer, G., & Bowles, I. (2023). A Systematic Review of Using Machine Learning and Natural Language Processing in Smart Policing. *Computers*, *12*(12), Article 12. https://doi.org/10.3390/computers12120255

Shi, F., Kai, S., Zheng, J., & Zhong, Y. (2022). XLNet-Based Prediction Model for CVSS Metric Values. *Applied Sciences*, *12*(18), Article 18. https://doi.org/10.3390/app12188983

Tavakolian, A., Rezaee, A., Hajati, F., & Uddin, S. (2023). Hospital Readmission and Length-of-Stay Prediction Using an Optimized Hybrid Deep Model. *Future Internet*, *15*(9), Article 9. https://doi.org/10.3390/fi15090304

Torkman, R., Ghapanchi, A. H., & Ghanbarzadeh, R. (2024). A Framework for Antecedents to Health Information Systems Uptake by Healthcare Professionals: An Exploratory Study of Electronic Medical Records. *Informatics*, *11*(3), Article 3. https://doi.org/10.3390/informatics11030044

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2020). *XLNet: Generalized Autoregressive Pretraining for Language Understanding* (arXiv:1906.08237). arXiv. https://doi.org/10.48550/arXiv.1906.08237

Zhao, D., Chen, X., & Chen, Y. (2024). Named Entity Recognition for Chinese Texts on Marine Coral Reef Ecosystems Based on the BERT-BiGRU-Att-CRF Model. *Applied Sciences*, *14*(13), Article 13. https://doi.org/10.3390/app14135743

LAMPIRAN

Lampiran 1. *Dataset* Penelitian: Kelas Positif

Positif:
Contoh 1:
Teks: lv cavity size. normal regional lv systolic function. hyperdynamic lvef >75%. mid-cavitary gradient. no vsd. right ventricle: normal rv chamber size and free wall motion. aorta: mildly dilated ascending aorta. aortic valve: three aortic valve leaflets. moderately thickened aortic valve leaflets. mild as (area 2-9cm2). trace ar. mitral valve: mildly thickened mitral valve leaflets. severe mitral annular calcification. mod functional ms due to mac. mild to moderate (+) mr. tricuspid valve: mildly thickened tricuspid valve leaflets. no ts. mild tr. indeterminate pa systolic pressure. pulmonic valve/pulmonary artery: no ps. pericardium: no pericardial effusion. conclusions the left atrium is normal in size. no atrial septal defect is seen by 2d or color doppler. the estimated right atrial pressure is 0-5 mmhg. there is mild symmetric left ventricular hypertrophy. the left ventricular cavity size is normal. regional left ventricular wall motion is normal. left ventricular systolic function is hyperdynamic (ef>75%). a mid-cavitary gradient is identified. there is no ventricular septal defect. right ventricular chamber size and free wall motion are normal. the ascending aorta is mildly dilated. there are three aortic valve leaflets. the aortic valve leaflets are moderately thickened. there is mild aortic valve stenosis (valve area 2-9cm2). trace aortic regurgitation is seen. the mitral valve leaflets are mildly thickened. there is severe mitral annular calcification. there is moderate functional mitral stenosis (mean gradient 11 mmhg) due to mitral annular calcification. mild to moderate (+) mitral regurgitation is seen. the tricuspid valve leaflets are mildly thickened. the pulmonary artery systolic pressure could not be determined
---
Contoh 2:
Teks: 4 mg tablet, sublingual sig: one (1) tablet sublingual as directed as needed for chest pain. outpatient lab work please check inr on monday and call results to doctor at. acetaminophen 325 mg tablet sig: 1-2 tablets po q6h (every 6 hours) as needed for pain. disp:*6 vils* refills:*0* discharge disposition: home with service facility: vna discharge diagnosis: ventricular tachycardia non st elevation myocardial infarction discharge condition: stable. discharge instructions: you had a dangerous heart rhythm called ventricular tachycardia and was started on sotolol, a medicine to prevent this rhythm. in addition, an internal defibrillator (icd) was placed that will shock you out of this rhythm. you cannot get the icd dressing wet for one week. no showers of baths. you may wash your hair in a sink. you are scheduled in the device clinic in 1 week, they will check the function of the icd and take off the dressing. no lifting more than 5 pounds with your left arm for 6 weeks, no lifting your left arm over your head for 6 weeks. you will be on antibiotics to prevent an infection at the icd site for 3 days. you also had a cardiac catheterization that showed extensive blockages in your coronary artery. your medicines were adjusted to help your heart function. medication changes: sotolol: to prevent ventricular tachycardia restart your coumadin at 2 mg, you will need to check your inr on monday. decrease your aspirin to 81mg, continue taking plavix.. please call doctor if your icd fires, if you have any redness, swelling, tenderness or bleeding at the icd site, if you have any chest pain, fevers, chills or trouble breathing. weigh yourself every morning, md if weight > 3 lbs in 1 day or 6 pounds in 3 days. adhere to 2 gm sodium diet: information was given to you about this at discharge.. followup instructions: cardiology: doctor phone:

Contoh 3:
Teks: date of birth: sex: m service: medicine allergies: pneumovax 23 / allopurinol / hydralazine attending: chief complaint: transferred from osh with abdominal pain major surgical or invasive procedure: ercp with stent; cholecystostomy; hemodialysis history of present illness: the pt is a 74m with hx of cad s/p cabg in, dm, gout, ckd who presented to yesterday with ruq pain and noted to have t=4 with wbc of 13k (17k today), elevated ast/alt/alk phos. u/s w/ gallstones. noted to have elevated troponins although ekg unrevealing. was given cefoxitin and iv hep. is pain free this am. he denies any chest pain or shortness of breath beyond his baseline. he has baseline doe, pnd, orthopnea, ef is 40% in. he was transferred for ercp and placement of biliary stent. on arrival, his vs were stable and he was comfortable. he denied chest pain, sob beyond his basline, fevers, chills. all other review of systems was negative. past medical history: cad s/p cabg, redo cab g s/p aaa repair iddm ckd gout chronic systolic chf h/o gib social history: retired managment consultant. has 5 children. nonsmoker, quit over 20 years ago. no alcohol use. family history: nc physical exam: appearance: nad vitals: t: 9 bp: 115/71 hr: 88 rr: 18 o2: 90% ra eyes: eomi, perrl, conjunctiva clear, noninjected, anicteric, no exudate ent: dry neck: no jvd, no carotid bruits cardiovascular: rrr, nl s1/s2, no m/r/g respiratory: cta bilaterally, comfortable, no wheezing, mild bibasilar crackles gastrointestinal: soft, ruq
---

Lampiran 2. *Dataset* Penelitian: Kelas Negatif

Negatif:
Contoh 1:
Teks: metoprolol tartrate 25 mg nifedipine sr 90 mg daily sevelamer carbonate 1600 mg tid w/ meals simvastatin 80 mg daily iron 325 mg daily discharge medications: metoprolol tartrate 25 mg tablet sig: one (1) tablet po bid (2 times a day). furosemide 40 mg tablet sig: two (2) tablet po bid (2 times a day). nifedipine 90 mg tablet sustained release sig: one (1) tablet sustained release po daily (daily). insulin glargine 100 unit/ml solution sig: eight (8) units subcutaneous once a day. calcitriol 25 mcg capsule sig: one (1) capsule po once a day. sevelamer hcl 800 mg tablet sig: two (2) tablet po three times a day. simvastatin 80 mg tablet sig: one (1) tablet po once a day. discharge disposition: home discharge diagnosis: primary diagnosis: angioedema secondary diagnosis: diabetes mellitus end-stage renal disease discharge condition: mental status: clear and coherent. level of consciousness: alert and interactive. activity status: ambulatory - independent. discharge instructions: you were admitted to the hospital with a reaction called angioedema. we think that this was due to your lisinopril dosing. it may have been related to your iron as well. you should not take lisinipril ever or this may cause a life-threatening reaction. the following changes were made to your medications: stop lisinopril stop glyburide you should follow-up with your pcp 2 weeks. weigh yourself every morning, md if weight goes up more than 3 lbs. you should monitor your blood pressure at home and call your primary care doctor values over 160 systolic. you should check your blood sugars regularly and call your primary care doctor for values over 200 or less than followup instructions: scheduled appointments: provider: phone: date/time: 3:30 provider:, md phone: date/time: 4:00 provider:, md phone: date/time: 8:30 please see your primary care doctor within 2

Contoh 2:
Teks: fasciitis. placement of a suprapubic tube. medications on discharge: sarna lotion applied topically four times per day as needed (for rash). sinemet 25/100-mg tablets one tablet by mouth four times per day. acetaminophen 325-mg tablets one to two tablets by mouth q.4-6h. as needed. metoprolol 25-mg tablets one-half tablet by mouth twice per day. pantoprazole 40-mg tablets one tablet by mouth q.24h. miconazole powder one application topically as needed (for yeast at skin folds). polyvinyl alcohol 4 percent-drops 1 to 2 drops in the eyes as needed. percocet 5/325-mg tablets one to two tablets by mouth q.4- 6h. as needed (for pain). discharge instructions and followup: the patient was instructed to follow up with doctor in plastic surgery in one to two weeks. the patient was to call telephone number for an appointment. the patient was also instructed to follow up with doctor in urology in two weeks; please call telephone number for an appointment. the patient will need wet-to-dry dressing changes twice per day for her left thigh wound and a dry dressing on the right arm wound., dictated by: medquist36 d: 10:24:45 t: 11:31:39 job#:

Contoh 3:
Teks: ventricular wire was placed with good capture. he was taken on for a permanent pacemaker placement. see procedure note for full details. he was being 100% paced with good capture and weaned off neosynephrine after placement. of note, the patient has a history of chronic left pleural effusion. thoracic surgery was consulted and recommended a formal decortication. he needs to follow up with doctor in weeks to determine the timing for the decortication. he continued to be followed by the renal team and dialyzed 3 times per week via left upper extremity fistula. transplant surgery was consulted for a concern in the exam of his lue fistula. reportedly prior to surgery the graft had a strong thrill and postoperatively found to have only a weak pulse. access was found to be patent. last hemodialysis treatment was. chest tubes and pacing wires were removed per cardiac surgery protocol. he was transferred to the step down unit on post operative day 8 in stable condition. physical therapy continued to work with him for increased strength and endurance. a bedside swallowing was performed due to coughing while taking thin liquids. it was suggested a diet of thin liquids and soft consistency solids with 1:1 supervision during meals secondary to mental status. of note, patient did have episodes of sundowning, for which he received haldol with good results. once on the step down unit, mr. well. he was working with physical therapy, tolerating a full po diet and his incisions were healing well. it was felt that he was safe for transfer to rehab at this time. he was discharged to rehab following hemodialysis on pod medications on admission: asa 81mg daily, insulin sliding scale, imdur 60mg daily, lopressor 25mg daily, prilosec 20mg daily, renvela 8mg tid, travatan opthalmic solution, vitamin b complex/vitamin c/folic acid 1 capsule daily, crestor 20mg daily, repaglinide 2mg daily, filgrastim 300 mcg sc saturdays allergies:sulfa, niacin,
---

Lampiran 3. *Source Code Program: Preprocessing*

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from tqdm import tqdm, trange
import pickle

df_adm = pd.read_csv('M:/Dataset/mimic/mimic-iii-clinical-database-1.4/ADMISSIONS.csv')
df_adm.head()
df_adm.ADMITTIME = pd.to_datetime(df_adm.ADMITTIME, format='%Y-%m-%d %H:%M:%S', errors='coerce')
df_adm.DISCHTIME = pd.to_datetime(df_adm.DISCHTIME, format='%Y-%m-%d %H:%M:%S', errors='coerce')
df_adm.DEATHTIME = pd.to_datetime(df_adm.DEATHTIME, format='%Y-%m-%d %H:%M:%S', errors='coerce')
# urutan berdasarkan subject_ID dan tanggal masuk
df_adm = df_adm.sort_values(['SUBJECT_ID','ADMITTIME'])
df_adm = df_adm.reset_index(drop = True)
df_adm.loc[df_adm.SUBJECT_ID == 124,['SUBJECT_ID','ADMITTIME','ADMISSION_TYPE']]
df_adm = df_adm.sort_values(['SUBJECT_ID', 'ADMITTIME'])
df_adm = df_adm.reset_index(drop=True)
df_adm['NEXT_ADMITTIME'] = df_adm.groupby('SUBJECT_ID').ADMITTIME.shift(-1)
df_adm['NEXT_ADMISSION_TYPE'] = df_adm.groupby('SUBJECT_ID').ADMISSION_TYPE.shift(-1)
df_adm.loc[df_adm.SUBJECT_ID == 124,['SUBJECT_ID','ADMITTIME','ADMISSION_TYPE','NEXT_ADMITTIME','NEXT_ADMISSION_TYPE']]
rows = df_adm.NEXT_ADMISSION_TYPE == 'ELECTIVE'
df_adm.loc[rows,'NEXT_ADMITTIME'] = pd.NaT
df_adm.loc[rows,'NEXT_ADMISSION_TYPE'] = np.NaN
df_adm.loc[df_adm.SUBJECT_ID == 124,['SUBJECT_ID','ADMITTIME','ADMISSION_TYPE','NEXT_ADMITTIME','NEXT_ADMISSION_TYPE']]
# urutan berdasarkan subject_ID dan tanggal masuk
df_adm = df_adm.sort_values(['SUBJECT_ID','ADMITTIME'])

df_adm[['NEXT_ADMITTIME','NEXT_ADMISSION_TYPE']] = df_adm.groupby(['SUBJECT_ID'])
[['NEXT_ADMITTIME','NEXT_ADMISSION_TYPE']].fillna(method = 'bfill')
df_adm.loc[df_adm.SUBJECT_ID == 124,['SUBJECT_ID','ADMITTIME','ADMISSION_TYPE','NEXT_ADMITTIME','NEXT_ADMISSION_TYPE']]
# hitung jumlah hari antara keluar dan masuk berikutnya
df_adm['DAYS_NEXT_ADMIT']=  (df_adm.NEXT_ADMITTIME - df_adm.DISCHTIME).dt.total_seconds()/(24*60*60)
```

```
print('Number with a readmission:', (~df_adm.DAYS_NEXT_ADMIT.isnull()).sum())
print('Total Number:', len(df_adm))

df_adm = df_adm.loc[df_adm.ADMISSION_TYPE != 'NEWBORN']
df_adm = df_adm.loc[df_adm.DEATHTIME.isnull()]

df_adm['OUTPUT_LABEL'] = (df_adm.DAYS_NEXT_ADMIT < 30).astype('int')
df_adm['DURATION'] = (df_adm['DISCHTIME']-df_adm['ADMITTIME']).dt.total_seconds()/(24*60*60)

df_notes = pd.read_csv('M:\\Dataset\\mimic\\mimic-iii-clinical-database-1.4\\NOTEEVENTS.csv')
# Sort by subject_ID, HAD_ID then CHARTDATE
df_notes = df_notes.sort_values(by=['SUBJECT_ID','HADM_ID','CHARTDATE'])
# Merge notes table to admissions table
df_adm_notes = pd.merge(df_adm[['SUBJECT_ID','HADM_ID','ADMITTIME','DISCHTIME','DAYS_NEXT_ADMIT','NEXT_ADMITTIME',
                                'ADMISSION_TYPE','DEATHTIME','OUTPUT_LABEL','DURATION']],
                        df_notes[['SUBJECT_ID','HADM_ID','CHARTDATE','TEXT','CATEGORY']],
                        on = ['SUBJECT_ID','HADM_ID'],
                        how = 'left')

# Grab date only, not the time
df_adm_notes.ADMITTIME_C = df_adm_notes.ADMITTIME.apply(lambda x: str(x).split(' ')[0])

df_adm_notes['ADMITTIME_C'] = pd.to_datetime(df_adm_notes.ADMITTIME_C, format = '%Y-%m-%d', errors = 'coerce')
df_adm_notes['CHARTDATE'] = pd.to_datetime(df_adm_notes.CHARTDATE, format = '%Y-%m-%d', errors = 'coerce')

# Gather Discharge Summaries Only
df_discharge = df_adm_notes[df_adm_notes['CATEGORY'] == 'Discharge summary']
df_discharge = (df_discharge.groupby(['SUBJECT_ID','HADM_ID']).nth(-1)).reset_index()
df_discharge=df_discharge[df_discharge['TEXT'].notnull()]

def less_n_days_data(df_adm_notes, n):
    df_less_n = df_adm_notes[
        ((df_adm_notes['CHARTDATE'] - df_adm_notes['ADMITTIME_C']).dt.total_seconds() / (24 * 60 * 60)) < n]
    df_less_n = df_less_n[df_less_n['TEXT'].notnull()]
```

```
    # concatenate first
    df_concat = pd.DataFrame(df_less_n.groupby('HADM_ID')['TEXT'].apply(lambda x: "%s" % ' '.join(x))).reset_index()
    df_concat['OUTPUT_LABEL'] = df_concat['HADM_ID'].apply(
        lambda x: df_less_n[df_less_n['HADM_ID'] == x].OUTPUT_LABEL.values[0])

    return df_concat

import re

def preprocess1(x):
    y = re.sub('\\[(.*?)\\]', '', x)  # remove de-identified brackets
    y = re.sub('[0-9]+\.', '', y)  # remove 1.2. since the segmenter segments based on this
    y = re.sub('dr\.', 'doctor', y)
    y = re.sub('m\.d\.', 'md', y)
    y = re.sub('admission date:', '', y)
    y = re.sub('discharge date:', '', y)
    y = re.sub('--|__|==', '', y)
    return y

def preprocessing(df_less_n):
    df_less_n['TEXT'] = df_less_n['TEXT'].fillna(' ')
    df_less_n['TEXT'] = df_less_n['TEXT'].str.replace('\n', ' ')
    df_less_n['TEXT'] = df_less_n['TEXT'].str.replace('\r', ' ')
    df_less_n['TEXT'] = df_less_n['TEXT'].apply(str.strip)
    df_less_n['TEXT'] = df_less_n['TEXT'].str.lower()

    df_less_n['TEXT'] = df_less_n['TEXT'].apply(lambda x: preprocess1(x))

    # to get 318 words chunks for readmission tasks
    df_len = len(df_less_n)
    want = pd.DataFrame({'ID': [], 'TEXT': [], 'Label': []})
    for i in tqdm(range(df_len)):
        x = df_less_n.TEXT.iloc[i].split()
        n = int(len(x) / 318)
        for j in range(n):
            want = want.append({'TEXT': ' '.join(x[j * 318:(j + 1) * 318]), 'Label': df_less_n.OUTPUT_LABEL.iloc[i],
                                'ID': df_less_n.HADM_ID.iloc[i]}, ignore_index=True)
        if len(x) % 318 > 10:
            want = want.append({'TEXT': ' '.join(x[-(len(x) % 318):]), 'Label': df_less_n.OUTPUT_LABEL.iloc[i],
                                'ID': df_less_n.HADM_ID.iloc[i]}, ignore_index=True)

    return want

df_discharge = preprocessing(df_discharge)
df_discharge.to_pickle("M:/Informatika/End Game/Skripsi/Program/pickle/df_discharge.pkl")
df_discharge = pd.read_pickle('M:\\Informatika\\End Game\\Proposal Skripsi\\4\\18-4-2024\\pickle\\df_discharge.pkl')
df_discharge.shape

readmit_ID = df_adm[df_adm.OUTPUT_LABEL == 1].HADM_ID
not_readmit_ID = df_adm[df_adm.OUTPUT_LABEL == 0].HADM_ID
# subsampling to get the balanced pos/neg numbers of patients for each dataset
not_readmit_ID_use = not_readmit_ID.sample(n=len(readmit_ID), random_state=1)
id_val_test_t = readmit_ID.sample(frac=0.2, random_state=1)
id_val_test_f = not_readmit_ID_use.sample(frac=0.2, random_state=1)

id_train_t = readmit_ID.drop(id_val_test_t.index)
id_train_f = not_readmit_ID_use.drop(id_val_test_f.index)

id_val_t = id_val_test_t.sample(frac=0.5, random_state=1)
id_test_t = id_val_test_t.drop(id_val_t.index)

id_val_f = id_val_test_f.sample(frac=0.5, random_state=1)
id_test_f = id_val_test_f.drop(id_val_f.index)

# test if there is overlap between train and test, should return "array([], dtype=int64)"
(pd.Index(id_test_t).intersection(pd.Index(id_train_t))).values

id_test = pd.concat([id_test_t, id_test_f])
test_id_label = pd.DataFrame(data=list(zip(id_test, [1] * len(id_test_t) + [0] * len(id_test_f))),
                             columns=['id', 'label'])
```

```
id_val = pd.concat([id_val_t, id_val_f])
val_id_label = pd.DataFrame(data=list(zip(id_val, [1] * len(id_val_t) + [0] * len(id_val_f))), columns=['id', 'label'])

id_train = pd.concat([id_train_t, id_train_f])
train_id_label = pd.DataFrame(data=list(zip(id_train, [1] * len(id_train_t) + [0] * len(id_train_f))),
                              columns=['id', 'label'])

discharge_train = df_discharge[df_discharge.ID.isin(train_id_label.id)]
discharge_val = df_discharge[df_discharge.ID.isin(val_id_label.id)]
discharge_test = df_discharge[df_discharge.ID.isin(test_id_label.id)]

df = pd.concat([not_readmit_ID_use, not_readmit_ID])
df = df.drop_duplicates(keep=False)
# check to see if there are overlaps
(pd.Index(df).intersection(pd.Index(not_readmit_ID_use))).values

# for this set of split with random_state=1, we find we need 400 more negative training samples
not_readmit_ID_more = df.sample(n=400, random_state=1)
discharge_train_snippets = pd.concat([df_discharge[df_discharge.ID.isin(not_readmit_ID_more)], discharge_train])

# shuffle
discharge_train_snippets = discharge_train_snippets.sample(frac=1, random_state=1).reset_index(drop=True)

# check if balanced
discharge_train_snippets.Label.value_counts()

discharge_train_snippets.to_csv('M:\\Informatika\\End Game\\Proposal Skripsi\\4\\18-4-2024\\data\\discharge\\train.csv')
discharge_val.to_csv('M:\\Informatika\\End Game\\Proposal Skripsi\\4\\18-4-2024\\data\\discharge\\val.csv')
discharge_test.to_csv('M:\\Informatika\\End Game\\Proposal Skripsi\\4\\18-4-2024\\data\\discharge\\test.csv')
```

Lampiran 4. *Source Code Program: Modeling* XLNet-BiGRU-ATT

```
import sys
from transformers import XLNetTokenizer
import torch
import torch.nn as nn
from transformers import XLNetForSequenceClassification, XLNetTokenizer, AdamW, get_linear_schedule_with_warmup
from torch.utils.data import TensorDataset, DataLoader, RandomSampler, DistributedSampler
from tqdm import tqdm
import matplotlib.pyplot as plt
import math

tokenizer = XLNetTokenizer.from_pretrained('xlnet-base-cased')
data_dir = 'M:\\Informatika\\End Game\\Proposal Skripsi\\4\\18-4-2024\\data\\discharge'
train_examples = None
num_train_steps = None
if do_train:
    train_examples = processor.get_train_examples(data_dir)
    num_train_steps = int(
        len(train_examples) / train_batch_size / gradient_accumulation_steps * num_train_epochs)

class XLNetBiGRUAttention(nn.Module):
    def __init__(self, xlnet_model_path, num_labels=1):
        super().__init__()
        # Load pre-trained XLNetForSequenceClassification model
        self.xlnet = XLNetForSequenceClassification.from_pretrained(xlnet_model_path, num_labels=num_labels)
        # All XLNet parameters will be updatable
        for param in self.xlnet.parameters():
            param.requires_grad = True
        hidden_size = self.xlnet.config.d_model
        # BiGRU layer
        self.bigru = nn.GRU(hidden_size, hidden_size // 2,
                            bidirectional=True, batch_first=True)
        # Attention mechanism
        self.attention = nn.Linear(hidden_size, 1)
        # Text vector representation (fully connected layer)
        self.fc = nn.Linear(hidden_size, hidden_size)
        # Dropout layer
        self.dropout = nn.Dropout(0.1)
        # Replace the original classifier
        self.classifier = nn.Linear(hidden_size, num_labels)
        # Loss function for binary classification
        self.loss_fct = nn.BCEWithLogitsLoss()
    def forward(self, input_ids, attention_mask, labels=None):
        # XLNet forward pass
        outputs = self.xlnet(input_ids=input_ids,
                             attention_mask=attention_mask,
                             labels=labels,
                             output_hidden_states=True)
        sequence_output = outputs.hidden_states[-1]
        # BiGRU
        gru_output, _ = self.bigru(sequence_output)
        # Attention mechanism
        attention_scores = self.attention(gru_output)
        attention_weights = torch.softmax(attention_scores, dim=1)
        context_vector = torch.sum(attention_weights * gru_output, dim=1)
        # Text vector representation with dropout
        text_vector = self.dropout(torch.relu(self.fc(context_vector)))
        # Output layer
        logits = self.classifier(text_vector)
        # Use our own loss function if labels are provided
        if labels is not None:
            loss = self.loss_fct(logits.view(-1), labels.view(-1))
        else:
            loss = None
        return loss, logits, outputs.hidden_states
# Model initialization
xlnet_model_path = 'M:\Informatika\End Game\Proposal Skripsi\Model\mimiciii_xlnet_5e_128b\mimiciii_xlnet_5e_128b'
model = XLNetBiGRUAttention(xlnet_model_path)
# Send data to the chosen device
model.to(device)
```

Lampiran 5. *Source Code Program: Training Model* XLNet-BiGRU-ATT

```
import torch
import numpy as np
import pandas as pd
from tqdm import tqdm, trange
from torch.utils.data import TensorDataset, DataLoader, RandomSampler, SequentialSampler, DistributedSampler
from transformers import get_linear_schedule_with_warmup
import os
import logging

# Set up logging
logging.basicConfig(level=logging.INFO, format='%(asctime)s - %(levelname)s - %(message)s')
logger = logging.getLogger(__name__)

if do_train:
    no_decay = ['bias', 'LayerNorm.bias', 'LayerNorm.weight']
    optimizer_grouped_parameters = [
        {'params': [p for n, p in model.named_parameters() if not any(nd in n for nd in no_decay)], 'weight_decay': 0.01},
        {'params': [p for n, p in model.named_parameters() if any(nd in n for nd in no_decay)], 'weight_decay': 0.0}
    ]

    optimizer = torch.optim.AdamW(optimizer_grouped_parameters, lr=learning_rate)
    warmup_steps = int(warmup_proportion * num_train_steps)
    scheduler = get_linear_schedule_with_warmup(optimizer,
                                                num_warmup_steps=warmup_steps,
                                                num_training_steps=num_train_steps)

    # Prepare training data
    train_features = convert_examples_to_features(
        train_examples, label_list, max_seq_length, tokenizer)

    logger.info("***** Running training *****")
    logger.info("  Num examples = %d", len(train_examples))
    logger.info("  Batch size = %d", train_batch_size)
    logger.info("  Num steps = %d", num_train_steps)
    all_input_ids = torch.tensor([f.input_ids for f in train_features], dtype=torch.long).to(device)
    all_attention_mask = torch.tensor([f.attention_mask for f in train_features], dtype=torch.long).to(device)
    all_label_ids = torch.tensor([f.label_id for f in train_features], dtype=torch.float).to(device)

    train_data = TensorDataset(all_input_ids, all_attention_mask, all_label_ids)

    if local_rank == -1:
        train_sampler = RandomSampler(train_data)
    else:
        train_sampler = DistributedSampler(train_data)

    train_dataloader = DataLoader(train_data, sampler=train_sampler, batch_size=train_batch_size)

    model.train()
    global_step = 0

    for epoch in range(int(num_train_epochs)):
        logger.info(f"***** Epoch {epoch+1}/{num_train_epochs} *****")
        progress_bar = tqdm(train_dataloader, desc="Training", disable=local_rank not in [-1, 0], miniters=1, ncols=100)
        tr_loss = 0
        nb_tr_examples, nb_tr_steps = 0, 0

        for step, batch in enumerate(progress_bar):
            batch = tuple(t.to(device) for t in batch)
            input_ids, attention_mask, label_ids = batch

            loss, logits, hidden_states = model(input_ids=input_ids,
                                                attention_mask=attention_mask,
                                                labels=label_ids)

            if n_gpu > 1:
                loss = loss.mean()

            if gradient_accumulation_steps > 1:
                loss = loss / gradient_accumulation_steps
```

```
            loss.backward()
            tr_loss += loss.item()
            nb_tr_examples += input_ids.size(0)
            nb_tr_steps += 1

            if (step + 1) % gradient_accumulation_steps == 0:
                optimizer.step()
                scheduler.step()
                model.zero_grad()
                global_step += 1

            if (step + 1) % 200 == 0:
                progress_bar.set_postfix({'loss': f'{loss.item():.3f}'})

        avg_train_loss = tr_loss / len(train_dataloader)
        logger.info(f'Epoch {epoch+1}/{num_train_epochs} - Average Loss: {avg_train_loss:.4f}')

        # Save model after each epoch
        output_model_file = f'/content/drive/MyDrive/TUGAS AKHIR/Output6/xlnet_bigru_attention_model_{readmission_mode}_epoch{epo
        torch.save(model.state_dict(), output_model_file)
        logger.info(f"Model for epoch {epoch+1} saved to {output_model_file}")

    logger.info("Training completed")

    # Save the final model (which is the same as the last epoch's model)
    final_output_model_file = f'/content/drive/MyDrive/TUGAS AKHIR/Output6/xlnet_bigru_attention_model_{readmission_mode}_final.b
    torch.save(model.state_dict(), final_output_model_file)
    logger.info(f"Final model saved to {final_output_model_file}")
```

Lampiran 6. *Source Code Program*: Evaluasi Model XLNet-BiGRU-ATT

```
import torch
import numpy as np
import pandas as pd
from tqdm import tqdm
from torch.utils.data import TensorDataset, DataLoader, SequentialSampler, DistributedSampler
from sklearn.metrics import roc_curve, auc, precision_recall_curve
import os

if do_eval:
    eval_examples = processor.get_test_examples(data_dir)
    eval_features = convert_examples_to_features(
        eval_examples, label_list, max_seq_length, tokenizer)
    logger.info("***** Running evaluation *****")
    logger.info("  Num examples = %d", len(eval_examples))
    logger.info("  Batch size = %d", eval_batch_size)

    all_input_ids = torch.tensor([f.input_ids for f in eval_features], dtype=torch.long).to(device)
    all_attention_mask = torch.tensor([f.attention_mask for f in eval_features], dtype=torch.long).to(device)
    all_label_ids = torch.tensor([f.label_id for f in eval_features], dtype=torch.float).to(device)

    eval_data = TensorDataset(all_input_ids, all_attention_mask, all_label_ids)

    if local_rank == -1:
        eval_sampler = SequentialSampler(eval_data)
    else:
        eval_sampler = DistributedSampler(eval_data)

    eval_dataloader = DataLoader(eval_data, sampler=eval_sampler, batch_size=eval_batch_size)

    model.eval()
    eval_loss, eval_accuracy = 0, 0
    nb_eval_steps, nb_eval_examples = 0, 0
    true_labels = []
    pred_labels = []
    logits_history = []

    for batch in tqdm(eval_dataloader, desc="Evaluating"):
        batch = tuple(t.to(device) for t in batch)
        input_ids, attention_mask, label_ids = batch

        with torch.no_grad():
            loss, logits, hidden_states = model(input_ids=input_ids,
                                                attention_mask=attention_mask,
                                                labels=label_ids)

        logits = torch.sigmoid(logits).squeeze().detach().cpu().numpy()
        label_ids = label_ids.to('cpu').numpy()

        eval_loss += loss.mean().item()
        outputs = np.asarray([1 if i else 0 for i in (logits >= 0.5)])
        tmp_eval_accuracy = np.sum(outputs == label_ids)

        true_labels.extend(label_ids.flatten().tolist())
        pred_labels.extend(outputs.flatten().tolist())
        logits_history.extend(logits.flatten().tolist())

        eval_accuracy += tmp_eval_accuracy
        nb_eval_examples += input_ids.size(0)
        nb_eval_steps += 1

    eval_loss = eval_loss / nb_eval_steps
    eval_accuracy = eval_accuracy / nb_eval_examples

    df = pd.DataFrame({'logits': logits_history, 'pred_label': pred_labels, 'label': true_labels})
    string = f'logits_xlnet_bigru_attention_chunks.csv'
    df.to_csv(os.path.join(output_dir, string))

    df_test = pd.read_csv(os.path.join(data_dir, "test.csv"))
    fpr, tpr, df_out = vote_score(df_test, logits_history, readmission_mode, output_dir)
```

```
string = f'logits_xlnet_bigru_attention_readmissions.csv'
df_out.to_csv(os.path.join(output_dir, string))

rp80 = vote_pr_curve(df_test, logits_history, readmission_mode, output_dir)

result = {
    'eval_loss': eval_loss,
    'eval_accuracy': eval_accuracy,
    'global_step': global_step,
    'training loss': tr_loss / nb_tr_steps,
    'RP80': rp80
}

output_eval_file = os.path.join(output_dir, "eval_results.txt")
with open(output_eval_file, "w") as writer:
    logger.info("***** Eval results *****")
    for key in sorted(result.keys()):
        logger.info("  %s = %s", key, str(result[key]))
        writer.write("%s = %s\n" % (key, str(result[key])))
```

Lampiran 7. Permohonan akses data pada *PhysioNet*

- Profile
- Password
- Emails
- Username
- Cloud
- ORCID
- Credentialing
- Training
- Certification
- Agreements

Data Use Agreements

You have signed the following Data Use Agreements:

Date of Agreement	Agreement	Project	View Agreement
March 6, 2024	PhysioNet Credentialed Health Data Use Agreement 1.5.0	MIMIC-IV (v2.2)	View
March 6, 2024	PhysioNet Credentialed Health Data Use Agreement 1.5.0	Chest X-ray Dataset with Lung Segmentation (v1.0.0)	View
March 6, 2024	PhysioNet Credentialed Health Data Use Agreement 1.5.0	Learning to Ask Like a Physician: a Discharge Summary Clinical Questions (DiSCQ) Dataset (v1.0)	View
March 6, 2024	PhysioNet Credentialed Health Data Use Agreement 1.5.0	Annotated Question-Answer Pairs for Clinical Notes in the MIMIC-III Database (v1.0.0)	View
March 6, 2024	Korea Credentialed Health Data Agreement-1-0-0	INSPIRE, a publicly available research dataset for perioperative medicine (v1.2)	View
Oct. 18, 2023	PhysioNet Credentialed Health Data Use Agreement 1.5.0	MIMIC-III Clinical Database (v1.4)	View

Code of Conduct

You have signed the following Code of Conduct:

Date of Agreement	Name	Version
Sept. 27, 2023	Code of Conduct	1.0

Lampiran 8. *PhysioNet Credentialed Health: Signed Data Use Agreement*

PhysioNet Credentialed Health Data Use Agreement 1.5.0

Data Use Agreement for the MIMIC-III Clinical Database (v1.4)

If I am granted access to the database:

1. I will not attempt to identify any individual or institution referenced in PhysioNet restricted data.
2. I will exercise all reasonable and prudent care to avoid disclosure of the identity of any individual or institution referenced in PhysioNet restricted data in any publication or other communication.
3. I will not share access to PhysioNet restricted data with anyone else.
4. I will exercise all reasonable and prudent care to maintain the physical and electronic security of PhysioNet restricted data.
5. If I find information within PhysioNet restricted data that I believe might permit identification of any individual or institution, I will report the location of this information promptly by email to PHI-report@physionet.org, citing the location of the specific information in question.
6. I have requested access to PhysioNet restricted data for the sole purpose of lawful use in scientific research, and I will use my privilege of access, if it is granted, for this purpose and no other.
7. I have completed a training program in human research subject protections and HIPAA regulations, and I am submitting proof of having done so.
8. I will indicate the general purpose for which I intend to use the database in my application.
9. If I openly disseminate my results, I will also contribute the code used to produce those results to a repository that is open to the research community.
10. This agreement may be terminated by either party at any time, but my obligations with respect to PhysioNet data shall continue after termination.

SIGNED: **Annisa Darwis**

DATED: **Oct. 18, 2023**

LEMBAR PERBAIKAN SKRIPSI

"PREDIKSI KEBERLANJUTAN RAWAT INAP PASIEN RUMAH SAKIT MENGGUNAKAN XLNET-BiGRU-ATT PADA CATATAN KHUSUS KLINIS PASIEN"

OLEH:

ANNISA DARWIS
D121 20 1032

Skripsi ini telah dipertahankan pada Ujian Akhir Sarjana tanggal 04 November 2024. Telah dilakukan perbaikan penulisan dan isi skripsi berdasarkan usulan dari penguji dan pembimbing skripsi.

Persetujuan perbaikan oleh tim penguji:

	Nama	Tanda Tangan
Ketua/Sekretaris	Ir. Anugrayani Bustami, S.T., M.T	
Anggota	Dr. Ir. Ingrid Nurtanio, M.T.	
Anggota	Ir. Tyanita Puti Marindah Wardhani.,S.T.M.Inf.	

Persetujuan Perbaikan oleh pembimbing:

Pembimbing	Nama	Tanda Tangan
I	Ir. Anugrayani Bustami, S.T., M.T	