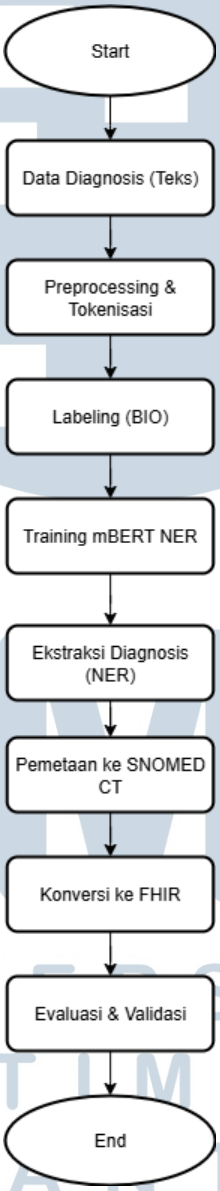
BAB 3
METODOLOGI PENELITIAN

Proses penelitian ini disusun secara sistematis untuk menjelaskan tahapan-tahapan yang dilakukan dalam membangun *pipeline* ekstraksi diagnosis medis berbasis NER dan FHIR. Setiap langkah dalam proses ditampilkan dalam bentuk diagram alur agar memudahkan pemahaman terhadap alur metodologi penelitian ini secara menyeluruh.



Gambar 3.1. Diagram Alur Penelitian

Pada Gambar 3.1 ditampilkan tahapan-tahapan utama penelitian, yang meliputi: pengumpulan data diagnosis medis, *preprocessing* data dan pelabelan BIO, pelatihan model mBERT NER,

pemetaan diagnosis ke SNOMED CT, konversi ke FHIR CONDITION, dan evaluasi hasil ekstraksi. Penjelasan dari setiap tahapan pada diagram tersebut akan dijabarkan lebih lanjut pada subbab berikutnya.

3.1 Sumber Data

Data yang digunakan pada penelitian ini berasal dari dua sumber utama, yaitu:

1. **Dataset Sintetis**: Dibuat berdasarkan pola diagnosis medis asli dari *template* rekam medis rumah sakit Indonesia.
2. **Dataset Validasi**: Dikumpulkan dari sampel diagnosis melalui *website* kesehatan yang sudah divalidasi dengan validator kementerian kesehatan (satu sehat platform).

Dataset dirancang agar merepresentasikan variasi nyata dalam penulisan diagnosis medis. Karakteristik yang diperhatikan dalam penyusunan data meliputi:

1. **Variasi Bahasa**: Penggunaan istilah Indonesia, Latin, dan Inggris.
2. **Variasi Format**: Singkatan, akronim, dan penulisan tidak baku.
3. **Kompleksitas Diagnosis**: Diagnosis tunggal, dengan keterangan tambahan.

Dataset diagnosis tersebut kemudian dibagi menggunakan metode *stratified random* sampling menjadi dua kelompok utama:

1. Data Latih (Training Set): 80% dari total dataset.
2. Data Uji (Test Set): 20% dari total dataset.

Pembagian ini bertujuan untuk memastikan bahwa distribusi diagnosis dan variasi istilah tetap terjaga di setiap kelompok data, dengan mempertimbangkan frekuensi kemunculan setiap jenis diagnosis.

Secara ringkas, Tabel 3.1 berikut menyajikan informasi mengenai asal dan jenis dataset yang digunakan dalam penelitian ini. Dataset training dan validasi dibuat dengan bantuan model bahasa ChatGPT versi 4.1 dan 4.5 Plus, disesuaikan dengan format FHIR dan divariasikan secara manual. Sementara itu, dataset testing diperoleh dari situs kesehatan daring yang kredibel seperti Satu Sehat, Alodokter, dan Halodoc, yang mencerminkan variasi diagnosis medis di dunia nyata.

Tabel 3.1. Sumber Dataset Diagnosis Medis

Jenis Dataset	**Sumber Data**
Dataset Training dan Validasi	ChatGPT 4.1 dan 4.5 Plus (OpenAI), disesuaikan sesuai format FHIR dan divariasikan secara manual
Dataset Testing	Website kesehatan: Satu Sehat, Alodokter, Halodoc

3.2 Preprocessing Data

Preprocessing data diagnosis medis merupakan tahapan fundamental untuk memastikan bahwa data yang digunakan dalam pelatihan model benar-benar bersih, terstruktur, dan konsisten. Proses ini dilakukan secara bertahap dan sistematis sebagai berikut:

1. **Normalisasi Teks**
   Normalisasi bertujuan untuk menyamakan format penulisan diagnosis agar seragam dan mudah diproses oleh algoritma. Seluruh huruf diubah menjadi huruf kecil (*lowercase*) agar perbedaan kapitalisasi tidak mempengaruhi makna klinis. Pada tahap ini juga dilakukan:
   a. Penghapusan tanggal yang tidak relevan terhadap konteks diagnosis.
   b. Penghapusan karakter atau simbol yang tidak sesuai `ASCII`, seperti "Ä" atau karakter asing lainnya.
   c. Penghapusan punctuation atau tanda baca seperti koma, titik, kutip, dan simbol-simbol khusus, kecuali yang kontekstual seperti garis miring pada "tbc/paru".
   d. Perbaikan ejaan istilah diagnosis, misalnya dari "hipertensi essensial" menjadi "hipertensi esensial".
   e. Penyamaan bentuk singkatan, seperti mengubah "mg" menjadi "miligram".
2. **Tokenisasi**
   Tokenisasi merupakan proses memecah diagnosis menjadi unit-unit token (kata atau sub-kata) yang dapat diproses oleh model BERT. Proses ini menggunakan *BertTokenizer* atau *BertTokenizerFast* untuk menghasilkan token sesuai format input model, termasuk menangani kata yang tidak dikenal dengan metode *subword tokenization*.
3. **Penerapan BIO Tagging**
   Setelah proses tokenisasi, setiap token diberi label menggunakan skema BIO (*Begin, Inside, Outside*) untuk keperluan pelatihan *Named Entity Recognition* (NER). Contohnya, pada diagnosis "asma bronkial berat", kata "asma" diberi label `B-CONDITION`, sedangkan "bronkial" dan "berat" masing-masing diberi label `I-CONDITION`. Token lain di luar diagnosis diberi label `O`. Format akhir disusun dalam bentuk tabel atau file berstruktur seperti `.conll`.

Seluruh proses *preprocessing* ini sangat krusial untuk menghasilkan dataset yang bersih, seragam, dan berkualitas tinggi. Kesalahan pada tahap ini—seperti tokenisasi yang salah, ejaan tidak baku, atau label tidak konsisten—dapat menurunkan akurasi model secara signifikan dan menggagalkan proses ekstraksi diagnosis dari teks medis.

3.2.1 Isi Data Diagnosis dan Contoh Format BIO

Dataset diagnosis medis yang digunakan dalam penelitian ini dikumpulkan dari berbagai sumber terpercaya, seperti artikel kesehatan daring, publikasi medis, dan data terbuka dari platform kesehatan Indonesia. Seluruh data telah melalui proses anonimisasi dan diseleksi hanya bagian diagnosisnya untuk menjaga privasi serta fokus penelitian.

Setiap entri data terdiri dari kalimat diagnosis yang utuh, yang kemudian diproses menjadi pasangan token dan label BIO. Contoh isi data diagnosis asli dan hasil pelabelan BIO disajikan pada Tabel 3.2 berikut:

Tabel 3.2. Contoh Kalimat Diagnosis dan Label BIO Tagging

Token	Label BIO
pasien	O
mengalami	O
asma	B-Condition
bronkial	I-Condition
berat	I-Condition
sejak	O
lama	O

Tabel 3.2 menunjukkan bagaimana sistem melabeli token berdasarkan posisi dan fungsi dalam frasa diagnosis medis. Token "asma" diberi label B-CONDITION karena menjadi awal entitas diagnosis, diikuti oleh token "bronkial" dan "berat" yang merupakan bagian lanjutan dari diagnosis dan diberi label I-CONDITION. Token lainnya, seperti "pasien" atau "sejak", bukan bagian dari diagnosis sehingga dilabeli O (*Outside*).

Format BIO tagging ini menjadi standar penting untuk pelatihan model NER, karena membantu model belajar membedakan entitas diagnosis medis dari teks umum. Hasil akhir preprocessing ini disimpan dalam format berstruktur seperti `.conll` agar kompatibel dengan pipeline pelatihan model.

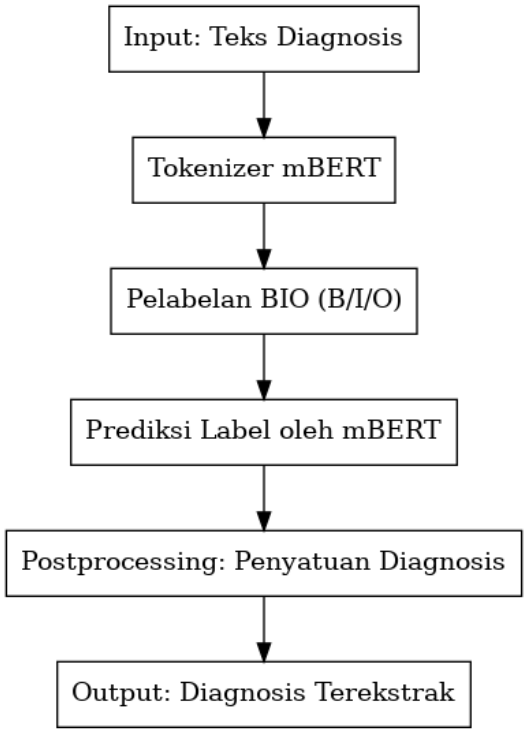
3.3 Proses Named Entity Recognition (NER)

Proses Named Entity Recognition (NER) dalam penelitian ini bertujuan untuk mengidentifikasi dan menandai entitas diagnosis medis (CONDITION) dalam teks diagnosis berbahasa Indonesia. Metode ini dilakukan secara otomatis menggunakan pendekatan *deep learning* berbasis *Multilingual* BERT (mBERT) yang telah dilatih sebelumnya.

Proses NER dilakukan dalam beberapa tahap utama sebagai berikut:

1. **Tokenisasi**: Kalimat diagnosis dipecah menjadi token-token menggunakan *tokenizer* dari mBERT, dengan tetap mempertahankan keterhubungan antar kata.
2. **Pelabelan BIO**: Setiap token diberikan label B-CONDITION, I-CONDITION, atau O berdasarkan posisi dan fungsi token tersebut dalam entitas diagnosis.
3. **Prediksi Label**: Model mBERT melakukan klasifikasi token berdasarkan konteks sekitarnya untuk memprediksi label yang sesuai.
4. **Postprocessing**: Token-token dengan label diagnosis disatukan kembali menjadi frasa diagnosis yang utuh sebagai hasil akhir ekstraksi entitas.

Diagram alur pada Gambar 3.2 berikut ini memperjelas tahapan proses NER dari input teks mentah hingga hasil prediksi diagnosis.



Gambar 3.2. Diagram Alur Proses Named Entity Recognition (NER)

Diagram Gambar 3.2 menunjukkan bagaimana sistem mengolah input teks diagnosis melalui proses tokenisasi, pelabelan BIO, prediksi oleh model mBERT, hingga menghasilkan entitas diagnosis yang siap dipetakan ke standar terminologi medis.

3.4 Pengembangan Model NER Berbasis `Multilingual` BERT

Tahap utama dalam penelitian ini adalah pelatihan model ekstraksi diagnosis berbasis *Multilingual* BERT (mBERT). Model ini dipilih karena kemampuannya menangani bahasa Indonesia sekaligus istilah Latin/Inggris yang sering muncul dalam diagnosis medis. Implementasi dilakukan menggunakan *library Huggingface Transformers* dan *PyTorch*, yang mendukung *pipeline* NER secara efisien dan modular.

Secara umum, proses pengembangan model terdiri atas beberapa langkah utama berikut:

1. Persiapan Dataset untuk *Training*
    a. Dataset diagnosis yang telah melalui tahap *preprocessing* dan pelabelan BIO *tagging* dikonversi ke format yang dapat diproses oleh model BERT.
    b. Setiap kalimat diagnosis dipetakan menjadi dua vektor: satu untuk token input dan satu untuk label BIO yang sesuai.

c. Dataset dibagi menjadi dua bagian utama: data latih (*training*) dan data uji (*testing*), dengan proporsi umum 80%:20%.

2. Tokenisasi dengan *Tokenizer* mBERT

   a. Tokenisasi dilakukan menggunakan *Tokenizer* mBERT (*BertTokenizerFast*) yang mendukung tokenisasi berbasis *subword*.

   b. *Alignment* antara token hasil tokenisasi dan label BIO dijaga agar setiap *subword* tetap mendapatkan label yang benar (menggunakan parameter *is_split_into_words=True*).

   c. *Token padding* dan *truncation* diterapkan untuk menyeragamkan panjang input pada model (misal, maksimum 64 token per kalimat).

3. Inisialisasi dan Konfigurasi Model

   a. Model mBERT diinisialisasi untuk tugas *token classification* dengan jumlah label yang sesuai (misal, 3 label: B-CONDITION, I-CONDITION, O).

   b. Konfigurasi lain seperti *mapping label2id* dan *id2label* diterapkan untuk menjaga konsistensi output model.

4. Pelatihan Model (*Fine-tuning*)

   a. Parameter utama pelatihan yang digunakan antara lain:

      i. **Batch size**: biasanya 8 atau 16, menyesuaikan kapasitas GPU.

      ii. **Learning rate**: pada kisaran 3e-5 hingga 5e-5.

      iii. **Epoch**: antara 3 hingga 10, disesuaikan dengan kestabilan *loss* dan *overfitting*.

   b. Proses pelatihan menggunakan *Trainer* dari *Huggingface*, dengan evaluasi berkala di data validasi pada setiap *epoch*.

   c. Untuk menghindari *overfitting*, bisa diterapkan teknik *early stopping* atau pengaturan *weight decay*.

5. Evaluasi Model

   a. Selama pelatihan, model dievaluasi menggunakan metrik:

      i. **Akurasi** (accuracy) untuk keseluruhan prediksi label token.

      ii. **Precision**, **Recall**, dan **F1-score** untuk label diagnosis (B/I-CONDITION).

   b. Hasil evaluasi divisualisasikan dalam bentuk *learning curve* dan *confusion matrix* untuk melihat distribusi kesalahan prediksi.

   c. Model terbaik (berdasarkan metrik validasi tertinggi, misal *F1-score*) dipilih untuk diuji pada data uji.

3.5 Pemetaan ke Kode SNOMED CT

Setiap entitas CONDITION yang berhasil diekstraksi oleh model NER BERT akan dipetakan ke dalam kode standar SNOMED CT. Tahap pemetaan ini dirancang untuk memastikan bahwa diagnosis medis yang teridentifikasi dapat terintegrasi secara terstandarisasi dan interoperabel ke dalam sistem informasi kesehatan berbasis FHIR.

Proses pemetaan dilakukan menggunakan pendekatan *exact match*, di mana istilah diagnosis yang diekstraksi dari teks medis akan dicocokkan secara langsung dengan entri yang tersedia dalam kamus diagnosis berbasis SNOMED CT. Jika ditemukan kecocokan secara persis, maka sistem akan secara otomatis mengambil kode SNOMED CT beserta *display name*-nya, dan memasukkannya ke dalam struktur *resource* FHIR CONDITION.

Sebaliknya, apabila entitas diagnosis tidak ditemukan dalam kamus (tidak memiliki *exact match*), maka diagnosis tersebut akan ditandai sebagai *unknown condition*. Hal ini berguna untuk proses evaluasi atau kurasi manual selanjutnya. Pendekatan *exact match* dipilih untuk menjaga konsistensi serta meminimalkan kesalahan dalam proses konversi diagnosis ke terminologi standar.

Pemetaan ke SNOMED CT merupakan tahap krusial dalam *pipeline* yang dikembangkan, karena menjamin hasil ekstraksi diagnosis dapat diubah menjadi data yang dapat diproses lebih lanjut, divalidasi, dan diintegrasikan ke dalam berbagai sistem rekam medis elektronik yang menggunakan standar FHIR di Indonesia maupun secara global.

Tabel 3.3. Contoh Pemetaan Diagnosis ke Kode SNOMED CT

Diagnosis (Input)	**Kode SNOMED CT**	**Deskripsi SNOMED (ID)**	**Deskripsi SNOMED (EN)**
gagal ginjal kronis	19944008	Gagal ginjal kronis	Chronic renal failure
asma bronkial	195967001	Asma bronkial	Bronchial asthma
diabetes melitus tipe 2	44054006	Diabetes melitus tipe 2	Type 2 diabetes mellitus
hipertensi esensial	59621000	Hipertensi esensial	Essential hypertension

Setelah proses identifikasi diagnosis selesai, sistem mencocokkan entitas yang berhasil diekstraksi dengan kamus SNOMED CT untuk memperoleh kode diagnosis yang sesuai dengan terminologi internasional. Tabel 3.3 berikut menunjukkan contoh pemetaan beberapa diagnosis ke kode SNOMED CT beserta deskripsi dalam dua bahasa.

3.6 Konversi Diagnosis ke Format FHIR Condition

Tahap akhir dari *pipeline* ini adalah mengubah diagnosis medis yang telah dikenali dan dipetakan ke terminologi standar menjadi struktur data yang sesuai dengan format FHIR CONDITION. Tujuan

dari proses ini adalah untuk memastikan bahwa hasil ekstraksi entitas diagnosis dapat langsung digunakan dalam sistem informasi kesehatan modern yang mendukung interoperabilitas data klinis.

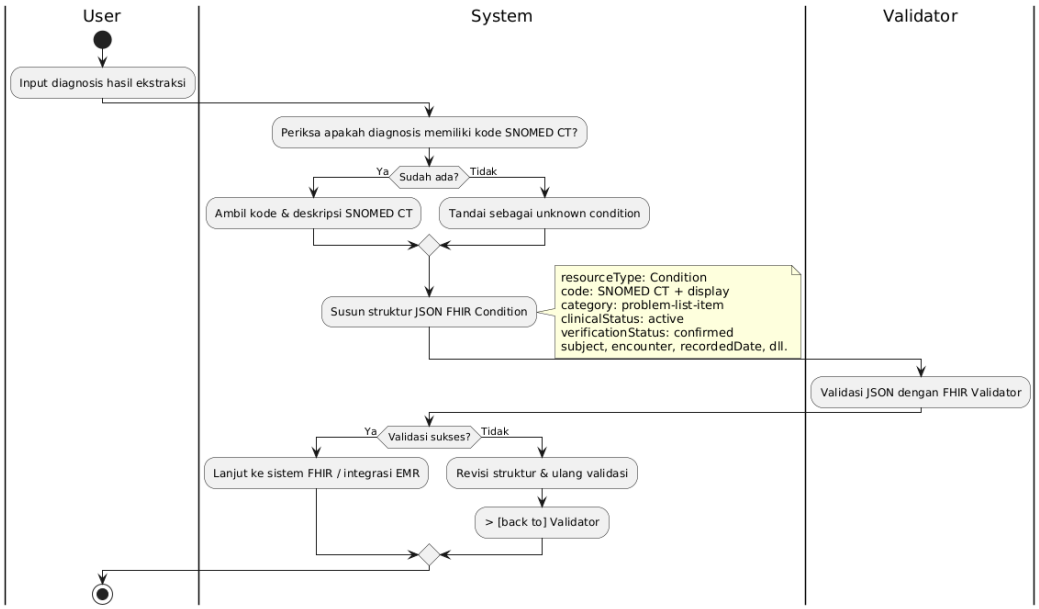
Setiap entitas CONDITION yang berhasil diekstraksi oleh model NER BERT terlebih dahulu diperiksa apakah memiliki padanan kode dalam terminologi SNOMED CT. Pemetaan diagnosis dilakukan secara *exact match* terhadap kamus diagnosis SNOMED CT yang telah disiapkan. Jika diagnosis ditemukan, maka sistem akan mengambil kode dan *display name* yang sesuai. Sebaliknya, jika diagnosis tidak ditemukan dalam kamus, maka entitas tersebut ditandai sebagai *unknown condition* untuk selanjutnya dapat dilakukan kurasi manual.

Diagnosis yang telah berhasil dipetakan kemudian disusun ke dalam format JSON FHIR CONDITION. Struktur data ini mencakup informasi-informasi penting seperti:

a. **resourceType**: Condition
b. **code**: SNOMED CT beserta display name
c. **category**: problem-list-item
d. **clinicalStatus**: active
e. **verificationStatus**: confirmed
f. **subject**, **encounter**, **recordedDate**, dan atribut lainnya

Struktur FHIR CONDITION yang telah dibentuk kemudian divalidasi menggunakan FHIR Validator. Jika validasi berhasil, maka data dapat langsung dikirim ke server FHIR (misalnya HAPI Server) atau diintegrasikan dengan sistem rekam medis elektronik (*Electronic Medical Record*, EMR). Apabila validasi gagal, sistem akan melakukan revisi terhadap struktur JSON dan mengulangi proses validasi hingga berhasil.

Untuk menggambarkan keseluruhan proses ini secara visual, Gambar 3.3 memperlihatkan diagram aktivitas yang menjelaskan alur konversi diagnosis ke dalam format FHIR CONDITION, mulai dari input diagnosis hasil ekstraksi hingga proses validasi.



Gambar 3.3. *activity diagram* konversi diagnosis ke FHIR CONDITION

Konversi ke dalam FHIR CONDITION berperan penting dalam memastikan bahwa hasil ekstraksi diagnosis tidak hanya valid secara klinis, tetapi juga memenuhi standar interoperabilitas. Dengan demikian, *pipeline* ini mendukung integrasi data diagnosis medis secara otomatis ke dalam ekosistem kesehatan digital nasional yang mengadopsi standar `FHIR`.

3.6.1 Evaluasi Pipeline dan Analisis Akurasi

Evaluasi *pipeline* bertujuan mengukur seberapa baik sistem yang dibangun dalam mengekstraksi diagnosis, memetakan ke kode SNOMED CT, serta menghasilkan *resource* FHIR yang valid. Penilaian dilakukan menggunakan data uji yang tidak digunakan selama pelatihan model.

Langkah-langkah evaluasi:

1. Evaluasi Ekstraksi Diagnosis:
    a. Menghitung akurasi, *precision*, *recall*, dan *F1-score* untuk label diagnosis hasil ekstraksi model NER B-CONDITION, I-CONDITION).
    b. Menggunakan *confusion matrix* untuk melihat jenis kesalahan prediksi (misal, entitas diagnosis yang terlewat atau salah klasifikasi).
2. Evaluasi Pemetaan SNOMED CT secara Deskriptif:
    a. Mendeskripsikan keberhasilan dan tantangan pada proses pemetaan diagnosis ke kode SNOMED CT berdasarkan hasil observasi manual atau studi kasus yang diujikan pada data hasil ekstraksi.
    b. Memberikan contoh kasus diagnosis yang berhasil maupun yang gagal dipetakan, serta alasan kegagalannya (misal, istilah lokal, singkatan, atau kode tidak ditemukan).

3. Evaluasi Validitas Resource FHIR secara Kualitatif:

    a. Melakukan pengecekan struktur dan kelengkapan *resource* FHIR hasil konversi secara manual menggunakan validator FHIR, serta mendokumentasikan apakah *resource* sudah memenuhi standar minimal HL7 FHIR.

    b. Menyajikan contoh *resource* FHIR yang telah tervalidasi, beserta catatan *error/warning* jika ditemukan selama proses validasi.

4. Evaluasi Efisiensi *Pipeline*:

    a. Mencatat waktu pemrosesan *pipeline* dari input diagnosis hingga *output resource* FHIR.

    b. Menilai kemudahan integrasi *pipeline* ke sistem eksternal, seperti aplikasi rumah sakit atau *platform* kesehatan digital.

Semua hasil evaluasi ini dianalisis dan dibahas secara mendalam pada bab hasil dan pembahasan, untuk menilai keunggulan dan keterbatasan *pipeline* yang dikembangkan.