3. ANALISIS DAN DESAIN SISTEM

Pada bab ini akan dibahas analisa data yang dilakukan dan desain sistem yang dibuat beserta dengan gambaran alur sistem.

3.1 Analisa Masalah

Pada subbab ini akan dibahas mengenai analisa masalah mengenai ringkasan ekstraktif dan metode yang digunakan pada penelitian ini. Berita merupakan salah satu sumber informasi yang tersedia secara online, akan tetapi jumlah informasi yang tersedia pada sumber berita online sangat banyak sehingga membuat manusia kerepotan untuk mengikutinya satu persatu. Selain itu, membaca keseluruhan informasi berita terkadang juga memerlukan waktu yang lama sehingga perlu dilakukan pembuatan ringkasan dari berita-berita yang tersedia tersebut. Ringkasan ekstraktif digunakan pada penelitian ini karena akan mengambil dari teks aslinya sehingga diharapkan informasi yang dihasilkan dalam ringkasan tidak berubah terlalu jauh dari topik yang ada dalam berita asli. Kemudian, pada penelitian ini akan menghasilkan ringkasan berita dari berita asli karena untuk proses pembuatan ringkasan dengan mengambil langsung dari berita asli dapat membantu mengatasi beberapa masalah seperti mengurangi waktu baca dengan mengambil kalimat penting dari berita asli dan membantu mengurangi beban kerja manusia dalam membaca informasi berita dengan menghilangkan informasi yang kurang relevan. Penerapan metode dari BERT pada ringkasan ekstraktif dilakukan untuk mengatasi keterbatasan RNN dalam memproses kata demi kata yang dilakukan secara berurutan dimana pada metode BERT akan langsung memproses seluruh input kata secara bersamaan dalam melakukan proses pelatihan model sehingga diharapkan melalui penelitian ini model dapat belajar lebih baik dengan menggunakan metode BERT. Selain itu, pada penelitian yang telah dilakukan oleh Liu dan Lapata, metode BERT yang digunakan dapat memperoleh hasil yang baik saat diterapkan pada berita berbahasa Inggris (Liu & Lapata, 2019).

3.2 Analisa Data

Pada subbab ini akan dibahas mengenai analisa dari dataset yang telah diambil dan pengolahan yang akan direncanakan.

3.2.1 Pengambilan Data

Data yang akan diambil dari dataset terdapat 3 bagian yaitu ‘gold_labels’, ‘paragraphs’, dan ‘summary’. Dataset berita berbahasa Indonesia yang digunakan diambil dari penelitian yang telah disediakan oleh Kurniawan dan Louvan. Dataset memiliki format *json* dan dibagi menjadi 3 bagian yaitu untuk *train*, *test*, dan *dev*. Selain itu, dataset juga dibagi menjadi 5 *fold* dimana tiap *fold* akan memiliki data sebanyak 18.774 berita. Dataset *train* digunakan untuk proses *training*, *test* untuk proses *test*, dan *dev* untuk proses evaluasi *training*.

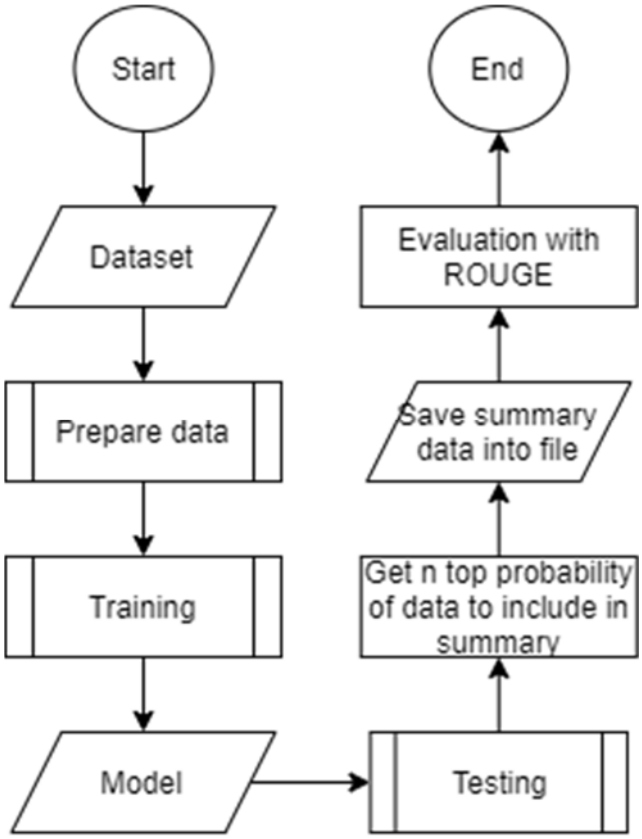
3.2.2 Pengolahan Data

Sebelum data digunakan ke dalam program, maka data akan diolah terlebih dahulu sehingga memudahkan pada implementasi ke dalam program. Beberapa hal yang akan dilakukan dalam pengolahan data diantaranya:

1. Menggunakan data yang terdapat pada bagian ‘gold_labels’, ‘paragraphs’, dan ‘summary’ dari dataset. ‘gold_labels’ digunakan untuk proses *training* dan pengujian dengan referensi ekstraktif, ‘paragraphs’ untuk isi berita yang akan diringkas, dan ‘summary’ untuk melakukan pengujian dengan referensi abstraktif.
2. Membuka array pada bagian ‘paragraphs’ untuk disusun menjadi satu kesatuan dokumen berita dan tiap kalimat dipisahkan dengan *delimiter* ‘<q>’.
3. Membuka array pada bagian ‘gold_labels’ untuk menunjukkan kalimat yang dipilih sebagai ringkasan ekstraktif yang kemudian diubah menjadi ‘labels’ dan tiap label dipisahkan dengan *delimiter* ‘<q>’.
4. Membuka array pada bagian ‘summary’ untuk diubah menjadi ‘target’ yang digunakan sebagai target dari referensi abstraktif yang akan diuji dan tiap kalimat dipisahkan dengan *delimiter* ‘<q>’.

3.3 Desain Sistem

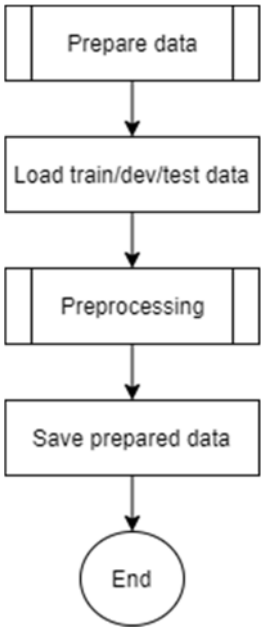
Dalam melakukan peringkasan secara ekstraktif, gambaran keseluruhan sistem yang akan diterapkan pada penelitian ini dapat dilihat pada Gambar 3.1. *Library* yang akan digunakan adalah *pytorch* dan *pytorch-lightning* untuk penerapan *encoder transformer* dan *feed forward neural network*. Kemudian, untuk model BERT akan menggunakan *library huggingface* (Wolf et al., 2020).



Gambar 3.1 Alur sistem secara keseluruhan

3.3.1 BERT Embedding

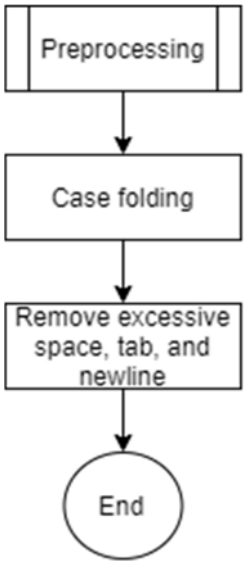
Dalam penerapan dari BERT akan memperhatikan konteks dari kalimat dimana *embedding* yang dihasilkan bisa lebih dari satu tergantung dari konteks dari kalimat (*context dependent*) sedangkan untuk model yang menggunakan *word2vec* akan menghasilkan representasi vektor yang sama untuk kata yang sama atau dengan kata lain sudah ditetapkan pada satu nilai vektor tertentu (*context independent*). Dalam penerapan BERT *embedding* akan menggunakan model indobert-base-uncased yang diambil dari *library huggingface* yang telah dilakukan training sebelumnya pada *wikipedia*, korpus web, dan artikel berita berbahasa Indonesia dengan total sejumlah 220 juta kata (Koto et al., 2020). Selain itu, juga akan digunakan bert-base-multilingual-uncased sebagai perbandingan dengan indobert-base-uncased. Sebelum dilakukan training pada model BERT akan disiapkan terlebih dahulu datanya dengan melakukan persiapan data yang dapat dilihat pada Gambar 3.2.



Gambar 3.2 Alur persiapan data

3.3.2 Preprocessing

Preprocessing yang akan dilakukan pada penelitian ini adalah hanya dengan mengubah data teks menjadi huruf kecil. Kemudian, dilanjutkan dengan merapikan data dari karakter spasi, *tab*, dan *newline* yang berlebihan. *Preprocessing* ini dilakukan untuk membantu model dalam mempelajari data. Selain itu, *preprocessing* seperti *stopword removal* dan *stemming* tidak dilakukan pada penelitian ini dengan alasan supaya model BERT dapat menangkap dengan lengkap hubungan dari tiap kalimat. Alur *preprocessing* dapat dilihat pada Gambar 3.3.



Gambar 3.3 Alur *preprocessing*

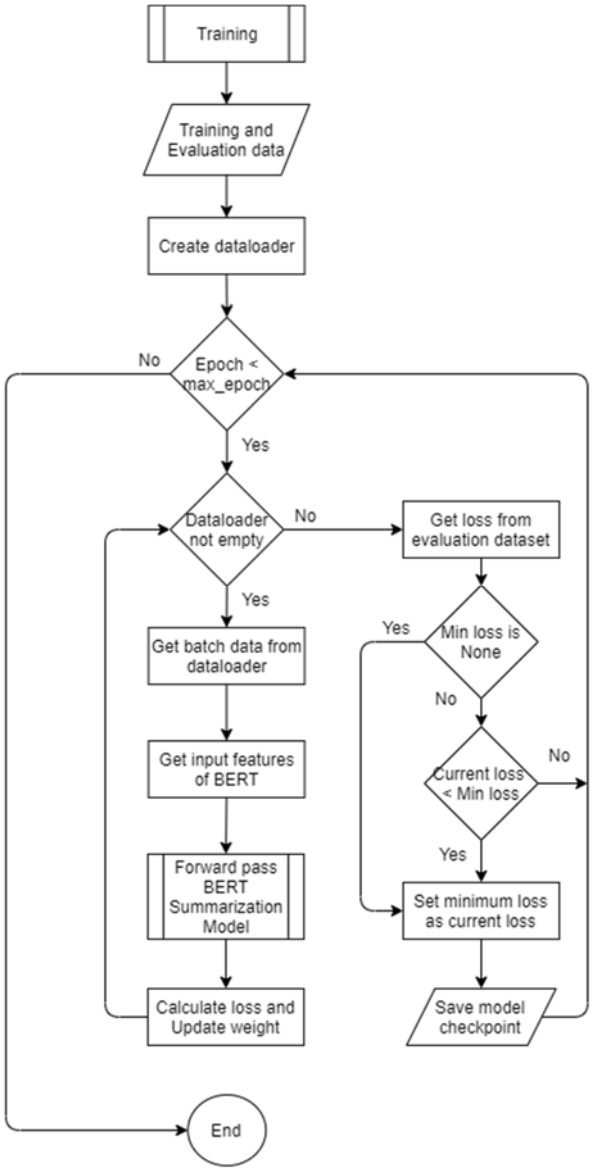
3.3.3 Training

Proses training dimulai dengan input model BERT yang digunakan dan dataset sebagai training dan evaluasi. Kemudian, akan dilakukan inisialisasi parameter *transformer encoder*, *optimizer*, dan lain-lain. Optimizer yang digunakan adalah *AdamW optimizer* dan fungsi *loss* yang akan digunakan adalah *binary cross entropy*. *AdamW optimizer* dipilih karena memiliki performa yang baik dengan melakukan modifikasi terhadap *weight decay*. Setelah itu, *training* akan dilakukan dalam beberapa *epoch*. Dalam satu *epoch*, proses akan dijalankan beberapa kali bergantung pada jumlah *batch*.

Pencarian *loss* menggunakan fungsi *loss* dengan target label sebagai ukuran kebenaran. Selama melakukan proses training, bila suatu *epoch* terpenuhi maka akan dilakukan proses evaluasi dimana model yang telah dilatih pada semakin baik atau tidak. Ketika dilakukan evaluasi model semakin baik maka model akan disimpan. Baik tidaknya suatu ditentukan berdasarkan dari *loss* yang didapat dari proses evaluasi dimana *loss* kecil menunjukkan bahwa model semakin membaik.

Tambahan susunan *transformer encoder* yang akan digunakan akan mengikuti penelitian yang telah dilakukan oleh Liu dan Lapata (Liu & Lapata, 2019), dimana model yang digunakan untuk peringkasan adalah *transformer encoder* untuk mengetahui relasi antar kalimat. Pada model yang digunakan akan mengambil pada level kalimat untuk diteruskan ke dalam layer klasifikasi (Liu, 2019). Layer klasifikasi akan dihitung berdasarkan dari *salience* suatu kalimat yaitu seberapa penting kalimat dari artikel berita yang bersangkutan. Setelah itu, akan dilakukan penghitungan dengan menggunakan fungsi aktivasi *sigmoid* untuk menghasilkan *array* yang menyimpan probabilitas dari tiap kalimat untuk masuk ke dalam ringkasan.

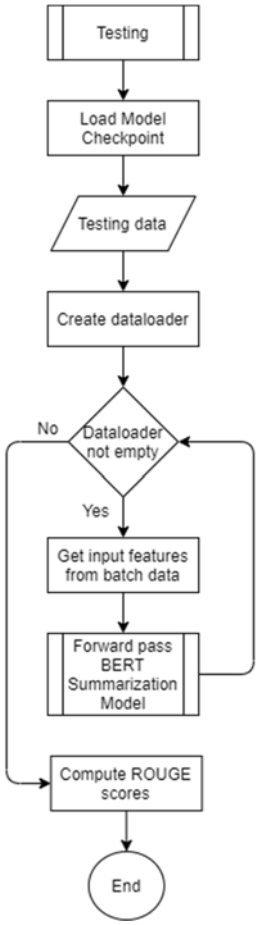
Proses evaluasi akan mirip dengan training hanya saja model tidak dilakukan training, namun langsung digunakan untuk mencari *loss* dari dataset evaluasi. Semua *loss* dari dataset akan digunakan untuk menentukan apakah model semakin baik atau tidak. Proses evaluasi bertujuan untuk menghindari terjadinya *overfitting* pada dataset *training*.



Gambar 3.4 Alur *training* secara garis besar

3.3.4 Testing

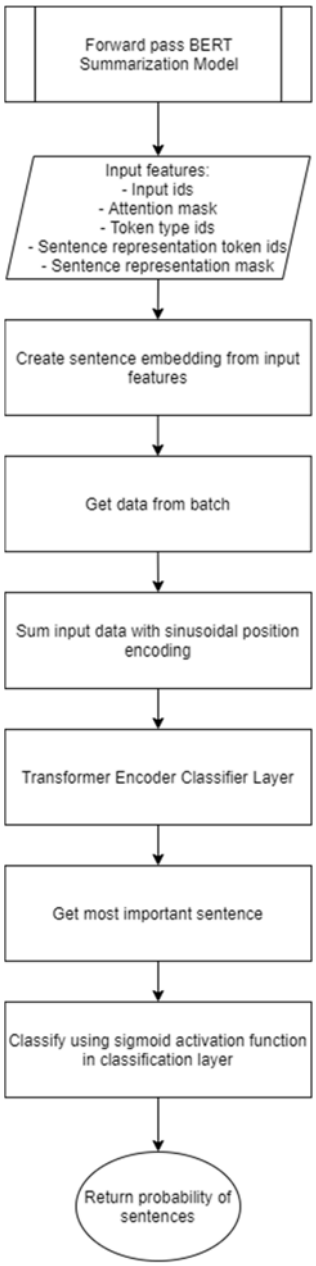
Pada awal proses testing akan dilakukan *load* dari *dataset test*. Lalu, akan dilanjutkan dengan melakukan *load* dari model BERT. Setelah itu, tiap data berita akan diproses oleh model untuk menghasilkan *array* berisi probabilitas dari tiap kalimat. Kemudian, data berita yang telah didapatkan probabilitasnya akan diambil sejumlah 'n' terbaik untuk dijadikan sebagai ringkasan. Hasil ringkasan akan disimpan ke dalam file teks yang selanjutnya dapat dilakukan penilaian dengan *ROUGE*.



Gambar 3.5 Alur *testing* secara garis besar

3.3.5 Forward Pass BERT Summarization Model

Pada penerapan dari *forward pass*, akan dilakukan pemilihan kalimat dari data teks berita pada *batch* tertentu dengan mempertimbangkan seberapa penting suatu kalimat berdasarkan artikel berita bersangkutan. Parameter tersebut kemudian akan dihitung dengan menggunakan fungsi *sigmoid* yang akan menghasilkan suatu array probabilitas dari tiap kalimat yang dapat masuk ke dalam ringkasan.



Gambar 3.6 Alur proses *forward pass BERT Summarization Model*

3.3.6 Evaluasi ROUGE

Dalam melakukan evaluasi terhadap ringkasan akan menggunakan ROUGE score. Penerapan dari ROUGE akan dibantu dengan menggunakan *library pyrouge*. Mengenai penjelasan dari ROUGE dapat dilihat pada subbab 2.1.

3.4 Desain Program

Pada subbab ini dibahas mengenai desain program yang akan dibuat berupa *website* secara *local*. Desain dari tampilan *website* dapat dilihat pada Gambar 3.7.

Gambar 3.7 Desain tampilan *website*

Untuk membuat ringkasan, maka *user* dapat menginputkan *url* dari sebuah portal berita online atau memilih sebuah file berita yang ingin diringkas. Kemudian, *user* dapat menginputkan persentase atau jumlah kalimat yang ingin diringkas. Persentase kalimat yang diisikan akan melakukan peringkasan berita sesuai dengan jumlah persentase tersebut.