BAB III
METODE PENELITIAN

3.1. Tahapan dan Prosedur Penelitian

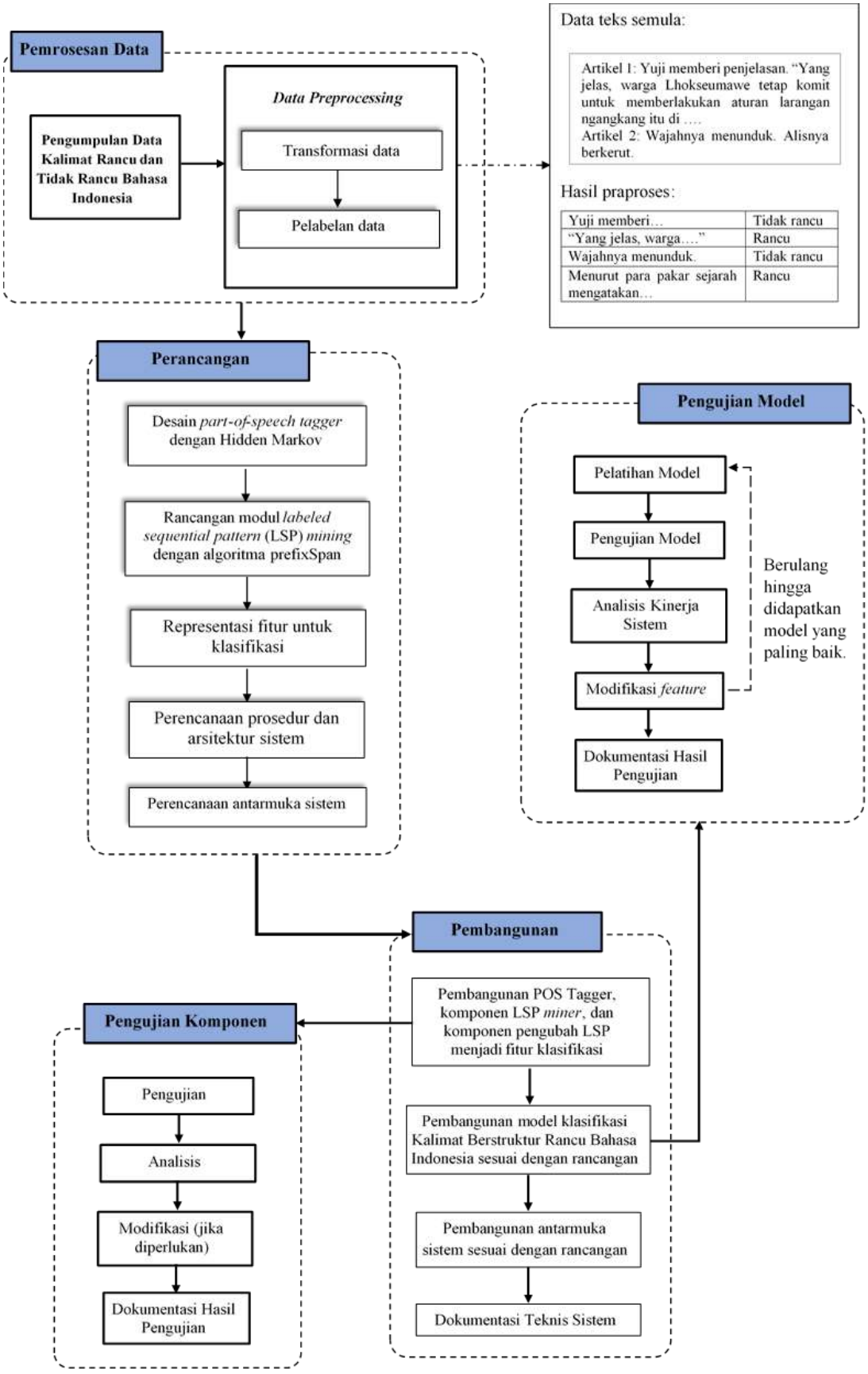
Tujuan akhir penelitian sistem deteksi kalimat berstruktur rancu bahasa Indonesia ini adalah untuk mengembangkan sistem otomatis yang dapat mendeteksi atau mengelompokkan kalimat bahasa Indonesia yang rancu dari segi struktur, menggunakan *Labeled Sequential Patterns* (LSP) dan klasifikasi teks. Gambar 3.1 menunjukkan garis besar tahapan dan prosedur dari penelitian ini.

Sebelum memulai penelitian, dilakukan studi literatur yang bertujuan untuk mempelajari dan memahami teori-teori yang berhubungan dengan penelitian. Teori-teori yang dipelajari yaitu kalimat rancu atau *grammatical error* pada Bahasa Indonesia dan juga bahasa lain sebagai referensi dan perbandingan; *Natural Language Processing* (NLP) yang terdiri dari *part-of-speech* menggunakan model *Hidden Markov; sequence pattern mining* beserta jenis dan algoritma-algoritmanya, yang salah satunya adalah algoritma PrefixSpan; hingga *classification* pada *machine learning.* Teori-teori tersebut diperoleh melalui buku, jurnal dalam negeri maupun internasional, artikel, situs internet, dan sumber ilmiah lainnya. Adapun dokumentasi dari studi literatur yang dilakukan telah ditampilkan pada bab sebelumnya.

3.1.1. Tahap Pemrosesan Data

a. Pengumpulan Data

Data kalimat yang tidak rancu dan rancu dari strukturnya dikumpulkan dari beberapa sumber, yaitu media massa *online* yang tidak mengedarkan berita dalam bentuk cetak seperti detik.com (pada tahun 2013 – 2014) dan grid.id. Sebagian data juga diambil dari novel berbahasa Indonesia yang tidak sesuai dengan tata bahasa baku, terlepas dari “unsur kesengajaan” atau “ciri khas” penulis dari novel tersebut. Pemilihan sumber data ini didasari oleh urgensi pembangunan sistem yang ditujukan untuk mendeteksi kalimat berstruktur rancu pada artikel, buku pelajaran, esai, novel berbahasa Indonesia, dan naskah terjemahan. Setelah data terkumpul dalam jumlah tertentu (minimal 1000 data) dengan rasio data kalimat rancu dan data kalimat rancu 1:1, kemudian dilakukan *data preprocessing.*



Gambar 1.1. Desain penelitian

b. *Data Preprocessing*

Proses ini dimaksudkan untuk mempersiapkan data sebelum data diolah oleh algoritma. *Data preprocessing* sendiri terdiri dari beberapa tahap, yaitu:

1) Transformasi data kalimat bahasa Indonesia, dimaksudkan agar data sesuai dengan algoritma yang dipakai. Proses transformasi dilakukan dengan mengubah data yang awalnya berupa paragraf biasa, menjadi bentuk tabel berformat *tab delimited* atau .tsv, dengan memenggal setiap kalimat menjadi baris-baris baru. Hal ini dilakukan untuk memudahkan program dalam membaca teks pada berkas tersebut.
2) Tahap selanjutnya data yang telah diubah menjadi format .tsv satu persatu (setiap baris/kalimat) dilabeli sebagai "*rancu*" atau "*tidak rancu*" menggunakan instrumen daftar cek. Daftar cek tersebut telah disusun sebelumnya dengan cara sebagai berikut.
   i. Melakukan pencarian sumber literatur mengenai tata bahasa baku Indonesia untuk mengetahui cakupan unsur kalimat yang termasuk pada kategori struktural.
   ii. Mengacu pada hasil mempelajari literatur pada poin (1), dilakukan lagi pencarian sumber literatur dan wawancara mengenai struktur kalimat yang rancu dan kesalahan tata bahasa pada sisi sintaksis.
   iii. Dari pencarian yang didapat pada poin (2), dicari pembahasan yang secara eksplisit menyebutkan '*penyebab kerancuan struktur kalimat*' atau *'kesalahan tata bahasa*'. Poin-poin pada pembahasan tersebut kemudian diadopsi dan dicantumkan pada daftar cek.

3.1.2. Tahap Perancangan

Pada tahap perancangan, dilakukan perencanaan atau perancangan dari kerangka sistem yang akan dibangun, mulai dari menyusun desain modul-modul yang akan digunakan pada sistem, merancang arsitektur sistem, aliran data sistem, hingga antar muka sistem. Secara garis besar, rancangan aliran sistem yang akan dibangun ditunjukkan oleh Gambar 3.2.

b. Modul *part-of-speech* (POS) *tagger* akan digunakan untuk untuk memberi *tag* kelas kata pada setiap token yang ada dalam data. Algoritma yang akan

digunakan pada modul ini adalah *Hidden Markov Model* (HMM) Bigram. Perancangan modul POS *tagger* ini sendiri dapat dijabarkan seperti berikut.

(1) Mengumpulkan data latih berupa kalimat-kalimat bahasa Indonesia yang sudah diberi *tag* kelas kata secara manual. Data latih ini disebut dengan *training corpus* dan isi serta penggunannya berbeda dengan data yang disebutkan pada tahap analisis di poin 3.1.1. Untuk sistem pendeteksi kalimat rancu bahasa Indonesia ini, *training corpus* yang digunakan didapat dari penelitian Dinakaramani, Rashel, Luthfi, & Manurung (2014) yang tersedia di Github.

(2) Menentukan bentuk data yang akan diterima, diproses dan dikeluarkan oleh modul. Masukan untuk modul POS *tagger* ini berupa satu kalimat yang sudah ditokenisasi atau dipisah per kata dan tanda bacanya. Modul akan memproses setiap kata dan tanda baca tersebut sampai diperoleh kelas kata yang sesuai. Keluaran dari modul ini merupakan sekuen kelas kata yang menjadi penyusun kalimat tersebut, beserta id data asalnya dan labelnya (rancu/tidak rancu).

(3) Menentukan arsitektur modul, dalam hal ini menyesuaikan dengan masukan yang diterima dan keluaran yang diinginkan, bahasa dan teknik pemrograman yang digunakan, yaitu Java dan *Object Oriented Programming* (OOP), serta mengacu pada algoritma HMM yang telah disampaikan pada Bab II poin 2.5.4.1.

c. Selanjutnya, dilakukan perancangan modul untuk penambangan *labeled sequential pattern* (LSP) pada data dengan algoritma PrefixSpan. Tahap ini menuntut POS *tagger* telah siap digunakan, karena melibatkan sekuens kelas kata (POS) pada kalimat. Masukan untuk modul ini berupa sekuens kelas kata yang menyusun suatu kalimat, sementara keluarannya merupakan pola bigram atau trigram yang terdapat pada kalimat tersebut, beserta id data asalnya, dan label kelasnya.

d. Pada tahap berikutnya, dibuat kerangka modul yang akan mentransformasi hasil keluaran dari tahap sebelumnya, yaitu LSP, ke dalam bentuk vektor/matriks untuk digunakan sebagai *feature* bagi model klasifikasi. Keluaran dari tahap ini adalah data .arff yang nantinya akan digunakan pada klasifikasi. Adapun modul

classifier tidak dibangun oleh peneliti, melainkan menggunakan *library machine learning* yang sudah tersedia, yaitu Weka.

e. Tahap selanjutnya adalah merancang prosedur dan arsitektur sistem secara keseluruhan, berorientasi pada *user* dan *performance* sistem itu sendiri, mulai dari cara sistem memproses masukan, hingga menampilkan keluaran akhir. Setelah sistem membaca dokumen, data yang menyimpan isi dokumen tersebut diproses dalam modul POS *tagger,* kemudian diproses kembali pada modul *labeled sequential pattern* (LSP) dan secara otomatis membuat rangkaian *feature* sebelum masuk pada model klasifikasi. *User* akan melihat keluaran yang menunjukkan hasil prediksi dari proses klasifikasi tersebut.

f. Tahap desain selanjutnya adalah desain antar muka. Meskipun penelitian Deteksi Kalimat Berstruktur Rancu Bahasa Indonesia ini berbasis eksperimen, namun antar muka dirasa tetap diperlukan agar pengguna lebih mudah memahami cara kerja dan hasil yang diberikan oleh sistem.

3.1.3. Tahap Pembangunan

Untuk dapat dimengerti oleh mesin, dalam hal ini komputer, maka rancangan yang telah dibuat harus diubah ke dalam bentuk yang dapat dimengerti oleh komputer, yaitu ke dalam bahasa pemrograman melalui proses *coding.* Bahasa yang dipakai untuk membangun sistem pada penelitian adalah bahasa pemrograman Java, sehingga arsitektur sistem nantinya akan mengikuti prinsip Object Oriented Programming (OOP).

3.1.4. Tahap Pengujian (Komponen & Sistem)

Sistem yang telah dibangun dalam bentuk bahasa pemrograman haruslah diujikan untuk memverifikasi apakah hasil dari sistem sudah sesuai dengan kebutuhan yang telah didefinisikan pada rancangannya. Selain menguji algoritma yang dipakai pada modul POS *tagger* dan *sequential pattern,* pengujian pada hal ini juga mencakup eksperimen pada model *classifier* yang digunakan untuk mendeteksi, atau lebih tepatnya mengklasifikasikan, kalimat Bahasa Indonesia. Metode *classifier* ini sendiri menggunakan *library* Weka.

Sistem melakukan eksperimen model yang dimaksud dalam beberapa tahap yang dapat diuraikan sebagai berikut.

a. Jika sistem telah menerima data, tepatnya data latih yang telah dikumpulkan pada tahap analisis, proses pertama yang dilakukannya adalah melatih model *classifier* menggunakan data latih tersebut. Tahap ini bertujuan untuk memberitahu model mengenai mana LSP yang merepresentasikan kalimat rancu, dan LSP mana yang merepresentasikan kalimat tidak rancu.
b. Berikutnya hasil pelatihan diuji dengan data uji. Dengan data ini, sistem diuji untuk mengevaluasi kemampuan *classifier* dalam mengenali pola kalimat yang diberikan, dan akurasi model *classifier* dalam memprediksi kerancuan kalimat. Adapun metode yang digunakan untuk menguji model *classifier* pada sistem ini adalah *10 Cross Validation.*
c. Analisis akurasi dan presisi dilakukan dan didapatkan dengan metode *confusion matrix* (CM) ketika melakukan *cross validation*. Dari sekian data yang diuji pada model *classifier*, hasil prediksinya kemudian dibandingkan dengan hasil yang sebenarnya, sehingga *confusion matrix* dapat mencatat:
    1. jumlah data kalimat rancu yang diprediksi rancu oleh model (*true positive*)
    2. jumlah data kalimat rancu yang diprediksi tidak rancu oleh model (*false positive*)
    3. jumlah data kalimat tidak rancu yang diprediksi tidak rancu oleh model (*true negative*)
    4. jumlah data kalimat tidak rancu yang diprediksi rancu oleh model (*false negative)*

    Dari keempat komponen tersebut, melalui proses perhitungan tertentu, bisa didapatkan nilai akurasi, presisi, *f-measure* hingga *recall* dari model yang telah dibangun. Pada penelitian ini, karena evaluasi model menggunakan metode 10 *cross validation,* maka angka perhitungan akurasi hingga *recall* yang diambil adalah rerata dari nilai perhitungan ke-10 *fold* yang telah dilakukan.

    1. Akurasi akan menjawab pertanyaan ‘berapa kali model benar dalam memprediksi apakah suatu data kalimat termasuk rancu atau tidak rancu?’
    2. Presisi akan menjawab ‘berapa kalimat yang rancu merupakan kalimat yang memang rancu?’
    3. *Recall* akan menjawab ‘berapa kalimat rancu yang diprediksi rancu oleh model?’
    4. *F-score* menjawab pertanyaan, ‘berapa rerata presisi dan *recall* dari model yang dibangun?’

Selain menghitung nilai akurasi, presisi, *f-measure* dan *recall* dari setiap percobaan yang dilakukan, analisis juga mencakup observasi secara ilmiah tentang ‘mengapa percobaan A mendapatkan akurasi sekian’, LSP mana yang paling berpengaruh bagi *classifier* dalam membedakan kalimat rancu dan kalimat tidak rancu, serta membandingkan antara hasil percobaan A dengan hasil percobaan lain.

d. Untuk memperbaiki model (jika akurasi model dirasa belum cukup baik), dilakukan modifikasi *feature* dengan memodifikasi modul *labeled sequential pattern* (LSP), menguji dengan metode *classifier* yang berbeda dari *library*, atau memodifikasi *training corpus* untuk *tagger*. Kemudian model dilatih dan diuji kembali, dan hasilnya dicatat dalam bentuk *log*.
e. Hasil pelatihan dan pengujian dengan berbagai kasus, kondisi dan modifikasi yang berbeda-beda, kemudian dicatat untuk dibuat laporan penelitian yang mencakup skripsi, jurnal dan dokumen teknis.

3.2. Alat dan Bahan Penelitian

3.2.1. Alat Penelitian

Proses pengumpulan data dan pembangunan sistem dalam penelitian ini dilaksanakan dengan menggunakan beberapa perangkat, yakni sebagai berikut..

1. Instrument daftar cek sebagai pedoman pengumpulan data, yang secara umum diadopsi dari :
    a. 1001 Kesalahan Berbahasa oleh Zaenal Arifin dan Farid Hadi (2001)
    b. Sintaksis Bahasa Indonesia oleh Supriyadi (2014)
    c. Tata Kalimat Bahasa Indonesia oleh Ida Bagus Putrayasa (2006)
    d. Analisis Kalimat (Fungsi, Kategori, dan Peran) oleh Ida Bagus Putrayasa (2008)
    e. Kalimat Efektif (Diksi, Struktur, dan Logika) oleh Ida Bagus Putrayasa (2009)
    f. Bahasa Indonesia untuk Perguruan Tinggi oleh R. K. Rahadi (2009)
    g. Morfologi (Sebuah Tinjauan Deskriptif) oleh M. Ramlan (1987)
    h. Garis-Garis Besar Tatabahasa Baku Bahasa Indonesia oleh Mansur Muslich (2010).

i. Situs Badan Pengembangan dan Pembinaan Bahasa di bawah Kementerian Pendidikan dan Kebudayaan (http://badanbahasa.kemendikbud.go.id/lamanbahasa/petunjuk_praktis/473)
j. Hasil wawancara dengan Nia Kurniasih, S.Pd.
k. Hasil wawancara melalui surat elektronik dengan Ivan Razela Lanin, S.Si., M.T.

2. Perangkat keras (*hardware*) yang digunakan pada pengembangan sistem:
   a. Prosessor AMD Quad Core A8-7410 2.5 GHz
   b. RAM 8 GB
   c. Harddisk 500GB
   d. Mouse
   e. Keyboard
3. Perangkat lunak (*software*) untuk mengembangkan sistem:
   a. Java Development Kit (JDK) 1.8.0_162
   b. Eclipse Java Oxygen
   c. *Text editor* (Notepad, Sublime Text versi 3)
   d. Weka Java Library versi 3.8.2
   e. Microsoft Office Excel 2016
4. Sistem Operasi (SO)

   Sistem operasi yang digunakan adalah Microsoft Windows 10 Pro 64 bit.

3.2.2. Bahan Penelitian

Data yang menjadi input sistem adalah dokumen berisi kalimat-kalimat bahasa Indonesia yang telah ditransformasi ke dalam format berkas *tab separated value* (.tsv) atau *comma separated value* (.csv) dengan bentuk satu kalimat per baris. Sementara *output* sistem adalah prediksi apakah kalimat-kalimat pada berkas tersebut merupakan kalimat yang strukturnya rancu, atau kalimat yang benar.