BAB IV

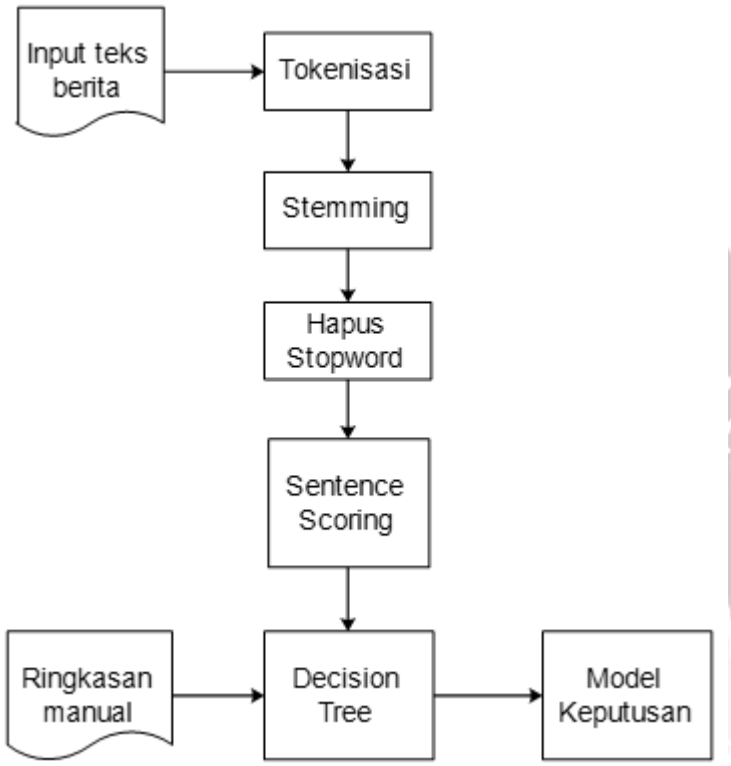
METODOLOGI PENELITIAN

Penelitian ini dilakukan dengan melalui empat tahap utama, dimana tahap pertama adalah proses pengumpulan dokumen teks yang akan digunakan data training dan data testing. Kemudian dilanjutkan pada tahap training untuk menghasilkan model atau *rule* menggunakan metode *decision tree*. Setelah model keputusan dibuat, dilanjutkan pada tahap testing untuk menghasilkan ringkasan sistem. Tahap akhir adalah evaluasi, untuk menguji tingkat akurasi antara hasil ringkasan sistem dengan ringkasan secara manual.

4.1 Pengumpulan Dokumen

Penelitian ini membutuhkan masukan dokumen teks berbahasa Indonesia dengan dokumen berjenis file teks. Pada penelitian ini digunakan sebanyak 100 dokumen teks berita nasional, yang terdiri atas 50 dokumen digunakan sebagai data training, dan 50 dokumen lainnya digunakan sebagai data testing. Dokumen tersebut bersumber dari berita online harian Kompas yang merupakan korpus penelitian (Aristoteles, Herdiyeni, Ridha, & Adisantoso, 2012). Masing-masing dokumen sudah memiliki hasil ringkasan manual. Pada tahap testing ringkasan manual digunakan untuk membandingkan dengan hasil ringkasan sistem, serta menilai seberapa akurat sistem peringkasan yang dibuat.

4.2 Training Data



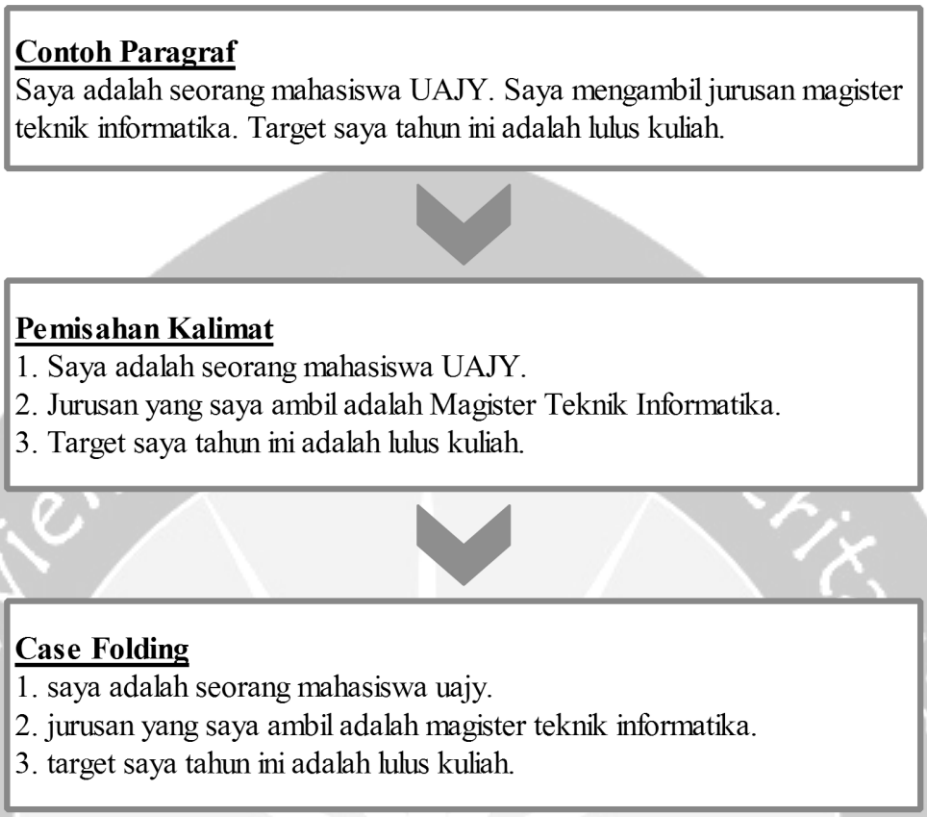
Gambar 1 Tahap Training

Gambar 4 menunjukan tahapan training data dengan dua jenis masukan yaitu teks berita dan ringkasan manual. Tujuan akhir dari tahapan ini adalah untuk menghasilkan model keputusan yang terdiri dari rules atau aturan-aturan.

Pada tahapan training, proses pertama adalah tokenisasi, dimana terdapat input masukan yaitu teks berita. Tokenisasi adalah proses untuk menghilangkan tanda baca dalam teks, kemudian memecah paragraf menjadi kalimat, dan dari kalimat dipecah menjadi kata-kata untuk mempermudah proses pembobotan pada tiap kalimat. Hasil dari tokenisasi ini akan diolah pada tahapan berikutnya yaitu stemming. Pada penelitian ini tokenisasi terdiri atas empat tahap, yaitu pemisahan

kalimat, case folding, dan yang terakhir pemisahan kata. Tokenisasi sangatlah penting dilakukan karena pada proses fitur teks, dokumen yang akan diolah harus sudah terpisah dengan bentuk kalimat dan kata. Proses pertama dalam tokenisasi yaitu memisahkan dokumen menjadi kumpulan kalimat. Menurut Aristoteles (2011), kalimat adalah gabungan dari dua buah kata atau lebih yang menghasilkan suatu arti dan diakhiri dengan suatu tanda berhenti. Tanda berhenti yang dimaksud adalah tanda baca titik. Pemisahan kalimat ini sangat penting, karena masing-masing kalimat ini nantinya akan diberikan skor dan akan diseleksi dengan batasan tertentu. Kalimat yang memiliki skor tertinggi merupakan hasil ekstraksi dokumen.

Proses selanjutnya adalah case folding, case folding merupakan proses mengubah semua huruf dalam dokumen ke dalam bentuk yang seragam, pada kasus ini menjadikan lower case. Proses ini dilakukan dengan tujuan menyeragamkan kata yang sebenarnya sama tetapi berbeda dalam penulisan. Contoh pemisahan kalimat dan case folding bisa dilihat pada gambar berikut:



Gambar 2 Alur pemrosesan tokenisasi

Setelah proses pemisahan kalimat dan kata, tahap selanjutnya dilakukan proses stemming. Stemming adalah proses mentransformasi kata-kata yang terdapat dalam suatu dokumen ke kata-kata dasarnya (*root word*) dengan cara menghilangkan imbuhan baik awalan, akhiran, maupun sisipan.. Sebagai contoh, kata berdiri, pendirian, mendirikan, akan distem ke kata dasarnya yaitu “diri”.

Langkah berikutnya adalah menghapus stopword. Stopword adalah kata umum yang tidak memiliki arti penting atau tidak relevan. Contohnya adalah kata-kata penghubung seperti dan, yang, atau, di, dari, dan seterusnya. Langkah ini perlu dilakukan agar sistem dapat mengenali kata-kata penting yang memiliki makna dalam sebuah dokumen teks.

4.2.1 Sentence Scoring

Proses berikut adalah sentence scoring, dimana setiap kalimat akan diberi nilai atau bobot berdasarkan 8 fitur pembobotan yaitu TF/IDF, kalimat dengan huruf kapital, kalimat dengan kata benda, frasa penting, data angka, panjang kalimat, posisi kalimat, dan kemiripan kalimat dengan judul. Penjelasan dan rumus tiap-tiap fitur adalah sebagai berikut:

1. **TF/IDF (F1)**

   Pembobotan diperoleh berdasarkan jumlah kemunculan term atau kata dalam kalimat (TF) dan jumlah kemunculan term atau kata pada seluruh kalimat dalam dokumen (IDF).

$$TF(s,t) = \frac{term\ frequency(s,t)}{\max(term\ frequency(s,ti))} \quad .....(15)$$

$$Score\ f1(s,t) = TF(s,t) \times log\left(\frac{N}{sft}\right) \quad .....(16)$$

2. **Kalimat Mengandung Huruf Kapital (F2)**

   Metode ini memberikan bobot yang lebih tinggi untuk kata-kata yang mengandung satu atau lebih huruf kapital, misalnya adalah nama orang, kota, negara, dan singkatan. Berikut rumus yang digunakan:

$$CW(s) = \frac{jumlah\ kata\ huruf\ besar\ dalam\ s}{jumlah\ kata\ dalam\ s} \quad .....(17)$$

$$Score\ f2(s) = \frac{CW(s)}{\max(CW(s))} \quad .....(18)$$

3. **Kalimat Mengandung Kata Benda (F3)**

Kalimat yang mengandung jumlah kata benda yang lebih banyak mendapat bobot lebih tinggi dan cenderung untuk dimasukkan dalam ringkasan dokumen. Kata benda seperti nama orang atau nama tempat.

$$Score\ f3(s) = \frac{jumlah\ kata\ benda\ dalam\ s}{jumlah\ kata\ dalam\ s} \quad \text{.....(19)}$$

4. **Kalimat Mengandung Frasa Penting (f4)**

Secara umum, kalimat dimulai dengan frasa "dengan demikian", "penyelidikan", dan menekankan seperti "yang terbaik", "paling penting", "menurut penelitian", "khususnya", serta frasa lainnya dapat menjadi indikator yang baik dari dokumen teks.

$$Score\ f4(s) = \frac{jumlah\ frasa\ dalam\ s}{jumlah\ frasa\ dalam\ dokumen} \quad \text{.....(20)}$$

5. **Kalimat Mengandung Data Angka (F5)**

Pada peringkasan teks mempertimbangkan data angka dalam dokumen, karena pada kalimat yang berisi data angka biasanya mengandung informasi yang penting.

$$Score\ f5(s) = \frac{data\ angka\ dalam\ s}{panjang\ s} \quad \text{.....(21)}$$

6. Panjang Kalimat (F6)

Kalimat yang panjang memiliki bobot yang lebih tinggi. Panjang kalimat dihitung berdasarkan jumlah kata dalam kalimat dikali dengan rata-rata panjang kalimat dalam dokumen.

$$Score\ f6(s) = panjang\ s * rerata\ panjang\ kalimat \quad .....(22)$$

7. Posisi Kalimat (F7)

Posisi kalimat adalah letak kalimat dalam sebuah paragraf. Asumsinya bahwa kalimat pertama pada tiap paragraf adalah kalimat yang paling penting.

$$Score\ f7(s) = \frac{posisi\ s\ dalam\ paragraf}{jumlah\ total\ kalimat\ dalam\ dokumen} \quad .....(23)$$

8. Kemiripan Kalimat Dengan Judul (F8)

Kalimat yang menyerupai judul dokumen adalah kata yang muncul dalam kalimat sama dengan kata yang ada dalam judul dokumen.

$$Score\ f8(s) = \frac{keyword\ dalam\ s \cap keyword\ dalam\ judul}{keyword\ dalam\ s \cup keyword\ dalam\ judul} \quad .....(24)$$

4.2.2 Decision Tree

Setiap bobot yang dimiliki masing-masing kalimat akan digunakan sebagai data training pada algoritma decision tree untuk menghasilkan model keputusan. Sebelum melakukan ekstraksi data ke dalam bentuk model *tree*, tentumya ada beberapa proses yang harus diperhatikan dalam pembentukan struktur pohon ini, yaitu:

a. Pilih *root* berdasarkan *gain ratio* terbesar
b. Pilih internal *root* atau cabang root berdasar gain ratio terbesar setelah menghapus atribut yang telah terpilih sebagai *root*
c. Ulangi sampai semua atribut terhitung nilai *gain ratio* pada masing-masing atribut.

Parameter yang tepat digunakan untuk mengukur efektifitas suatu atribut dalam melakukan teknik pengklasifikasian sampel data, salah satunya adalah dengan menggunakan *information gain*. Sebelum mencari nilai gain, terlebih dahulu mencari peluang kemunculan suatu record dalam atribut (*entropy*). Secara matematis nilai *entropy* dapat dihitung dengan menggunakan formula sebagai berikut :

$$Entropi(S) = \sum_{i=1}^{n} - pi * log_2\ pi \qquad .....(25)$$

Dimana S adalah himpunan kasus, n adalah jumlah nilai yang ada pada atribut target (jumlah kelas), sedangkan pi adalah jumlah sampel pada kelas i. Ketika kita sudah mendapatkan nilai *entropy*, maka langkah selanjutnya adalah melakukan perhitungan terhadap *information gain* dengan formula sebagai berikut:

$$Gain(S, A) = Entropi(S) \sum_{i=1}^{n} \frac{|S_i|}{|S|} * Entropi(S_i) \qquad .....(25)$$

Dimana S adalah himpunan kasus, A adalah atribut, n adalah jumlah partisi atribut A, sedangkan i menyatakan suatu nilai yang mungkin untuk atribut

A dan Si adalah jumlah kasus partisi ke i. Selanjutnya untuk menghitung rasio perolehan perlu diketahui suatu term baru yang disebut pemisahan informasi (*SplitInformation*). Pemisahan informasi dihitung dengan cara :

$$SplitInformation(S,A) = -\sum_{t=1}^{c} \frac{S_i}{S} log_2 \frac{S_i}{S} \quad .....(26)$$

Dimana S 1 sampai S c adalah c subset yang dihasilkan dari pemecahan S dengan menggunakan atribut A yang mempunyai banyak C nilai. Selanjutnya rasio perolehan (*gain ratio*) dihitung dengan cara :

$$GainRatio(S,A) = \frac{Gain(S,A)}{SplitInformation(S,A)} \quad .....(27)$$

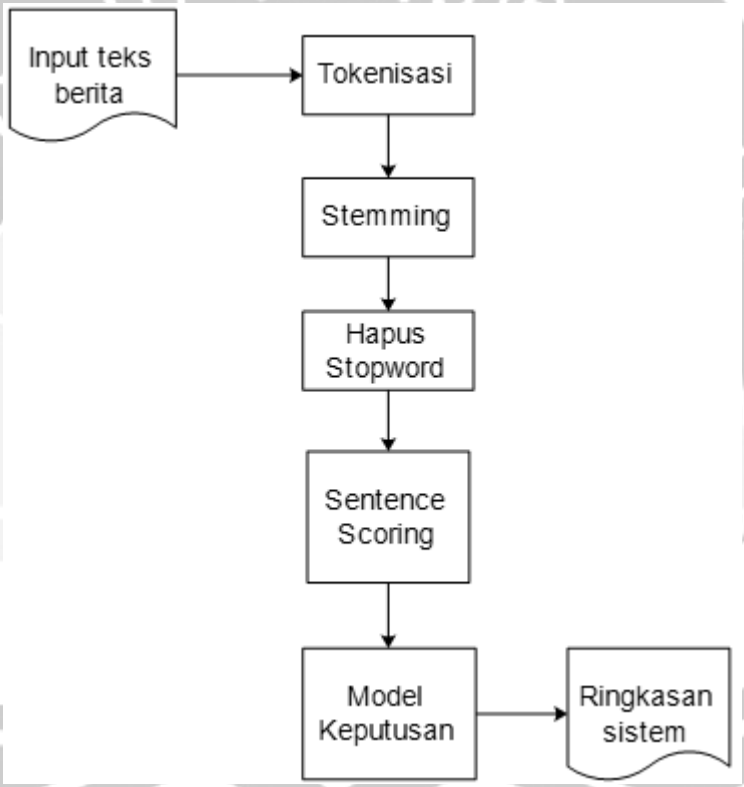
Tabel 1 Contoh data training

Kalimat	f1	f2	f3	f4	f5	f6	f7	f8	Ringkasan
1	0.1	0.2	0.0	0.0	0.1	0.3	0.5	0.1	YA
2	0.3	0.1	0.2	0.0	0.1	0.4	0.2	0.0	YA
3	0.4	0.2	0.1	0.2	0.2	0.2	0.3	0.3	TIDAK
4	0.2	0.3	0.3	0.1	0.4	0.1	0.5	0.4	YA
dst	0.1	0.4	0.0	0.0	0.1	0.4	0.1	0.0	TIDAK

Dari tabel 1 diatas dapat dijelaskan bahwa setiap kalimat adalah sebagai data sample, f1 sampai f8 adalah atribut, dan pada kolom ringkasan adalah target atribut dimana terdapat dua atribut YA dan TIDAK. Atribut YA berarti kalimat yang muncul dalam ringkasan, sedangkan atribut TIDAK menunjukan kalimat yang tidak masuk dalam ringkasan.

4.3 Testing Data

Tahapan selanjutnya adalah testing data. Pada tahapan ini, model keputusan yang sudah dihasilkan pada proses training akan digunakan untuk menghasilkan ringkasan sistem. Dokumen teks yang digunakan berjumlah 50 data. Adapun beberapa tahap testing dapat digambarkan sebagai berikut:



Gambar 3 Tahap Testing

Bagan diatas menunjukan tahapan testing, dimana alur prosesnya hampir sama dengan tahapan training. Dimulai dengan masukan berupa dokumen teks berita dan menghasilkan keluaran berupa ringkasan sistem.

4.4 Evaluasi

Dalam penelitian ini, evaluasi dilakukan dengan mengukur keakuratan antara hasil ringkasan sistem dengan ringkasan manual. Pengujian terhadap ringkasan sistem ini akan menggunakan metode precision, recall, dan f-measure. Precision mengevaluasi proporsi ketepatan untuk kalimat dalam ringkasan sedangkan recall digunakan untuk mengevaluasi proporsi kalimat yang relevan termasuk dalam ringkasan (Prasad & Kulkarni, 2010).

$$Precision = \frac{tp}{tp + fp} \quad \text{.....(28)}$$

$$Recall = \frac{tp}{tp + fn} \quad \text{.....(29)}$$

$$F - measure = \frac{2 * Precision \times Recall}{Precision + Recall} \quad \text{.....(30)}$$

Dimana:

- **tp** (true positif) adalah kalimat yang ada dalam ringkasan manual dan muncul dalam ringkasan sistem.
- **fp** (false positif) adalah kalimat yang tidak ada dalam ringkasan manual tetapi kalimat tersebut muncul dalam ringkasan sistem.
- **fn** (false negatif) adalah kalimat yang ada dalam ringkasan manual tetapi tidak muncul dalam ringkasan sistem.
- **tn** (true negatif) adalah kalimat yang tidak ada dalam ringkasan manual maupun dalam ringkasan sistem.

4.5 Lingkungan Pengembangan

Lingkungan pengembangan sistem peringkasan teks yang akan digunakan dalam penelitian ini sebagai berikut:

- Perangkat lunak: Windows 10, XAMPP, Rapid PHP 2015, Firefox developer edition, Microsoft Office 2016
- Perangkat keras: Intel(R) Core(TM) i3-2356 CPU @ 1.40 GHz, 2037 MB RAM
- Bahasa Pemrograman PHP dan database MySQL.