Daftar Istilah

Dataset	Kumpulan data berisi informasi terkait
Fitur	Atribut dominan yang dimiliki produk karena memiliki kelebihan atau daya tarik yang diambil dari kalimat
Polaritas	Orientasi objek berupa nilai positif atau negatif
Cluster	Kelompok data yang memiliki nilai kesamaan
Centroid	Pusat cluster atau nilai tengah

1 Pendahuluan

1.1 Latar Belakang

Perdagangan merupakan pekerjaan yang telah lama dilakukan oleh sebagian besar penduduk dunia. Metode dan teknik perdagangan mulai ditekuni sejak awal abad 18 dengan adanya pasar komoditas di Wall Street [1]. Perkembangan teknologi membantu perdagangan menjadi lebih mudah dan cepat dalam transaksinya. Dewasa ini perdagangan sudah umum dilakukan secara daring, perantara yang biasa digunakan adalah media sosial dan Electronic Commerce (EC). Persaingan dagang yang semakin marak membuat produsen harus terus meningkatkan kualitas produknya, sedangkan konsumen harus lebih teliti dalam memlilih tempat belanja. Adanya ulasan produk yang berisi tanggapan konsumen yang telah menggunakan dapat membantu calon konsumen memilih tempat berbelanja dan produk yang diinginkan, juga membantu produsen untuk meningkatkan kualitas produk sehingga mampu bersaing. Penelitian *Dimensional Search* menyatakan 90% calon konsumen terpengaruh dengan ulasan produk dengan tanggapan positif, dan 86% calon konsumen terpengaruh dengan ulasan produk dengan tanggapan negatif [2]. Sayangnya banyaknya tanggapan yang masuk menyebabkan penumpukan yang menimbulkan ketidakefisienan sehingga membingungkan pembacaannya. *BrightLocal* melakukan penelitian bahwa 85% calon konsumen membaca 10 ulasan dan 7% saja membaca lebih dari 20 ulasan produk [3].

Ringkasan ulasan produk dapat membantu pembaca opini menemukan kesimpulan dari seluruh opini sehingga membantu untuk mengambil keputusan. Proses peringkasan ulasan produk memiliki tiga tahapan yang harus dilakukan. Tahap pertama yaitu ekstraksi fitur produk, ekstraksi yang digunakan dalam penelitian ini adalah *Pattern Knowledge* [4], yaitu melakukan pengekstrasian dengan mengambil kata dari pola tertentu, kemudian hasil tersebut dibandingkan dengan fitur yang ada di *dataset* sehingga didapatkan fitur teridentifikasi.

Tahap kedua yaitu klasifikasi opini dilakukan untuk menentukan orientasi positif dan/atau negatif pada fitur dalam ulasan produk. Klasifikasi dalam penelitian ini menggunakan kamus kata pada SentiWordNet yaitu kamus yang berisikan kata bernilai positif dan negatif cukup lengkap untuk analisis sentimen [5]. Pengklasifikasian pada penelitian ini menggunakan kata sifat yang nantinya dipasangkan dengan fitur yang telah diesktraksi pada tahapan sebelumnya.

Proses terakhir yang dilakukan adalah peringkasan ulasan produk. Peringkasan dilakukan agar penumpukan ulasan produk menjadi lebih mudah dipahami. Berdasarkan penelitian yang ada sebelumnya peringkasan ulasan hanya menggunakan peringkasan ekstrakstif yang sebenarnya masih bisa di spesifikkan lagi dengan pengelompokan kalimat dengan nilai kesamaan tinggi. Penelitian ini menggunakan tf-idf untuk memberikan nilai kepada kalimat terklasifikasi, dilanjutkan dengan pengelompokkan kalimat menggunakan *centroid based clustering* [6] sehingga didapatkan hasil peringkasan yang lebih spesifik dan terkelompok.

1.2 Perumusan Masalah

Peringkasan dengan pendekatan ekstraktif yang telah dilakukan pada penelitian-penelitian sebelumnya melakukan peringkasan dokumen dengan memilih beberapa kalimat sebagai representasinya. Kalimat representasi tersebut masih dapat dikelompokkan berdasarkan kesamaan yang dimiliki dalam dokumen. Oleh karena itu, peringkasan lebih lanjut dibutuhkan dengan menghitung nilai kalimat menggunakan penilaian tf-idf yang dikelompokkan dengan *k-means clustering* dimana semakin rendah jarak antara *centroid cluster* dengan kalimat maka kalimat tersebut memiliki nilai kesamaan.

1.3 Tujuan

Berdasarkan uraian rumusan masalah di atas disimpulkan bahwa tujuan dalam penelitian tugas akhir ini memiliki tahapan yang dilakukan untuk peringkasan ulasan produk, yaitu:

1. Mengekstraksi dan menganalisis daftar fitur dari data ulasan produk menggunakan *Pattern Knowledge*.
2. Mengklasifikasi dan menganalisis hasil orientasi kalimat positis dan/atau negatif untuk dikelompokkan menggunakan SentiWordNet.
3. Menganalisis dan meringkas ulasan dengan penilaian menggunakan tf-idf dan *k-means clustering*.

1.4 Batasan Masalah

Tugas akhir ini memiliki batasan masalah sebagai berikut:

1. *Dataset* yang digunakan berupa ulasan produk berbahasa inggris dengan format .txt.
2. *Dataset* yang digunakan berasal dari sembilan produk yang digunakan pada paper Minqing Hu dan Bing Liu [7].
3. Ekstraksi fitur yang dilakukan terhadap fitur eksplisit, tidak termasuk fitur implisit pada ulasan produk.
4. Penentuan orientasi opini dilakukan pada level fitur.
5. Bahasa pemrograman yang digunakan adalah bahasa java.

1.5 Metode Penyelesaian Masalah

Tahapan metode penyelesaian masalah yang dilakukan pada penelitian ini, antara lain:

1. Studi literatur

Tahap untuk mengumpulkan informasi, mempelajari, dan mendalami konsep mengenai penambangan data teks khususnya analisis sentimen menggunakan *clustering* kalimat

yang diterapkan dalam proposal. Studi literatur dilakukan pada kajian seperti jurnal maupun buku terkait

2. Pengumpulan dan pengolahan data

Bahan penelitian tugas akhir yang disiapkan merupakan kumpulan dokumen berisi opini mengenai ulasan produk dalam bentuk file .txt.

3. Pembangunan model

Tahap ini dilakukan untuk mengetahui lebih dalam mengenai pembelajaran dari studi literatur baik studi kasus yang diangkat berikut penyelesaian masalah menggunakan metode terkait dengan.

4. Implementasi model

Merancang sistem berdasarkan hasil pembangunan model untuk mengolah data yang telah dipersiapkan. Pengimplementasian rancangan yang digambarkan menggunakan bahasa pemrograman java.

5. Analisis dan Pengujian

Berdasarkan implementasi yang telah dilakukan pada tahap sebelumnya dilakukan analisis dan pengujian hasil review berupa ringkasan produk. Hal ini bertujuan untuk meningkatkan kinerja sistem.

6. Pembuatan laporan tugas akhir

Hasil penelitian yang dilakukan dijadikan ke dalam buku tugas akhir sebagai dokumentasi lengkap.

2 Landasan Teori

2.1 Data Mining

Data mining atau dikenal juga dengan penambangan data merupakan proses ekstraksi informasi dari *dataset* bertujuan untuk mendapatkan pola data baru. *Data mining* merupakan bagian dari *Knowledge Discovery in Database* (KDD) yang merujuk pada ekstraksi implisit nontrivial yang sebelumnya tidak diketahui dan berpotensi memiliki bobot informasi dari data dalam database. Tahapan KDD dimulai dari data mentah hingga menjadi informasi baru, adalah sebagai berikut [8]:

1. *Data cleaning*: fase dimana *noise* data dan data yang tidak relevan dihapus dari kumpulan data.
2. *Data Integration*: tahap ini mengintegrasikan data dari data berulang, data berbeda, dapat digabungkan menjadi satu.
3. *Data selection*: memilih data yang relevan untuk analisis dan dikembalikan ke kumpulan data.
4. *Data transformation*: atau *consolidation* yaitu fase dimana data terpilih ditransformasikan menjadi data relevan yang dibutuhkan.
5. *Data mining*: fase dimana menggunakan beberapa teknik dalam pengaplikasian untuk mengekstraksi pola yang berpotensi.
6. *Pattern evaluation*: menggambarkan dan mengidentifikasi pola yang telah didapatkan sebelumnya.
7. *Knowledge representation*: seluruh *knowledge* atau informasi sudah terepresentasikan kepada pengguna.

Tahapan pertama hingga ke empat merupakan tahapan *preprocessing*, yaitu pengolahan data sebelum diekstrak di *data mining.*

2.2 Text Mining

Text mining atau dikenal dengan penambangan teks merupakan pengembangan *data mining.* Elemen kuncinya adalah megnhubungkan informasi yang telah diekstrak menjadi satu untuk membentuk pola baru untuk dieksplorasi lebih lanjut. Tujuan utama dari penggalian teks adalah menyingkirkan informasi tidak relevan untuk menemukan informasi berguna yang belum diketahui atau belum ditulis. Metodologi ini digunakan dalam kehidupan sehari-hari misal untuk analisis garansi, analisis rekam kesehatan [9] [10].

2.3 Analisis Sentimen

Analisis sentimen adalah *Opinion Mining* untuk mengetahui suasana hati khalayak tentang suatu topik atau produk. Analisis sentimen melibatkan pembangunan sistem untuk mengumpulkan dan menguji opini tentang produk berdasarkan tulisan blog,

komen, *review*, atau *tweets*. Perdagangan membantu menilai kesuksesan sebuah iklan bergantung dengan versi produk atau layanan yang terkenal dan bahkan diidetifikasi berdasarkan demografi disukai atau tidaknya fitur tersebut. Secara umum, analisis sentimen dibagi menjadi beberapa level yaitu level dokumen, kalimat, aspek, dan entitas.

2.3.1 Level Dokumen

Level dokumen biasa disebut *document-level sentiment classification* memiliki tugas mengklasifikasikan suatu dokumen termasuk dalam sentimen positif, negatif, atau netral. Analisis pada level ini mengasumsikan setiap dokumen menunjukkan opini pada satu entitas. Maka dari itu, level ini tidak dapat dipakai pada dokumen yang mengevaluasi atau membandingkan banyak entitas.

2.3.2 Level Kalimat

Level kalimat memiliki tugas mengklasifikasikan masing-masing kalimat opini bernilai positif, negatif, atau netral. Analisis level ini mengasumsikan bahwa setiap kalimat mendeskripsikan sentimen yang sama atau mayoritas dari opini yang dikandung dalam suatu kalimat.

2.3.3 Level Aspek dan Entitas

Level ini memiliki performa yang lebih baik disbanding level dokumen dan kalimat karena mampu menentukan apa yang disuka dan tidak disuka dari suatu produk. Hal ini disebabkan karena level aspek langsung merujuk pada opini tersebut disbanding mencari stuktur menyeluruh [11].

2.4 Tokenisasi

Tokenisasi merupakan hal dasar dalam *preprocessing*, yang berarti pemecahan suatu teks menjadi masing-masing unit kata, tanda baca, nomor, dll. Identifikasi unit perlu dilakukan untuk pemrosesan selanjutnya, kesalahan pada proses ini memungkinkan untuk mendorong lebih banyak kesalahan di proses selanjutnya pemrosesan teks [12].

Kalimat *Input*:

```
We really enjoyed shooting with the Canon PowerShot SD500
```

Hasil *Lemmatization*:

```
We | really | enjoy | shooting | with | the | canon | powershot |
sd500
```

2.5 Case Folding

Cara paling umum untuk melakukan *case folding* adalah dengan mengubah semua huruf menjadi *lower case* atau huruf kecil. Proses ini dibutuhkan memudahkan perbandingan token dalam *dataset* dengan suatu *query* [13].

Kalimat *Input*:

```
We really enjoyed shooting with the Canon PowerShot SD500
```

Hasil *Lemmatization*:

```
we really enjoy shooting with the canon powershot sd500
```

2.6 Stopword Removal

Stopword Removal merupakan bagian dari *preprocessing* dimana proses ini mengidentifikasi dan menghapus *stop words* dari teks. *Stop word removal* sebagai kata yang tidak memiliki makna dan tidak berpengaruh pada pemrosesan selanjutnya. Adanya *stop word* menambah jumlah *noise* sehingga menyulitkan di pengerjaan proses selanjutnya. Contoh dalam Bahasa Inggris, *stop word* antara lain *auxiliary* seperti '*have*', '*be*', *pronouns* seperti '*I*', '*it*', *presposition* seperti *'to'*, *'for'* [14]. Berikut contoh penggunaan *stop word removal*:

Kalimat *Input*:

```
We really enjoyed shooting with the Canon PowerShot SD500
```

Hasil *Stop word removal*:

```
enjoyed shooting canon powershot sd500
```

2.7 Lemmatization

Lemmatization merupakan proses pembentukan sebuah kata ke dalam bentuk dasar yang memiliki arti di kamus. Contoh perubahan kata kerja dalam Bahasa Inggris '*to be*', '*is*', '*am*', '*was*', '*were*' menjadi kata dasarnya kembali yaitu '*be*'. *Stemming* memiliki tujuan yang sama dengan *lemmatization,* perbedaannya adalah *stemming* dilakukan dengan memotongn langsung pada imbuhan katanya. Sedangkan *lemmatization* memperhatikan kosa kata dan morfologi kalimat sehingga merubah bentuk kata dan mengembalikan pada bentuk kamus [15].

Kalimat *Input*:

```
We really enjoyed shooting with the Canon PowerShot SD500
```

Hasil *Lemmatization*:

```
we really enjoy shooting with the canon powershot sd500
```

2.8 Part-of-Speech Tagging

Part-of-Speech Tagging (*POS Tagging*) merupakan bagian perangkat lunak yang menerima *input*-an data dalam suatu bahasa berupa teks yang kemudian memberikan *tag* pada setiap kata, seperti kata benda, kata kerja, kata sifat, dll [11]. Stanford *POS Tagging* merupakan *POS Tagging* yang dikembangkan Standford dalam Bahasa Inggris. Berikut contoh penggunaan POS *Tagging*:

Kalimat *Input*:

```
We really enjoyed shooting with the Canon PowerShot SD500
```

Hasil *POS Tagging*:

```
we_PRP really_RB enjoyed_VBD shooting_NN with_IN the_DT canon_NN
powershot_NN sd500_CD
```

Hasil *POS Tagging* di atas merupakan kata yang telah diberikan *tag* sesuai dengan tata Bahasa Inggris. Berikut keterangan *tag* pada Stanford *POS Tagging* [16]*:*

Tabel 2.1 Deskripsi *tag* pada *POS Tagging*

Tag	Deksripsi	Contoh
CC	Coordinated conjunction	and, or, for, so, but
CD	Cardinal number	one, two
DT	Determinier	a, the, every
EX	Existensial 'there'	There
FW	Foreign word	Nasi
IN	Preposition/ sub-ordinating conj.	on, in, by
JJ	Adjective	Good, Tangled
JJR	Adjective, comparative	Better, More Tangled
JJS	Adjective, superlative	Best, Most Tangled
LS	List item marker	1, 2
MD	Modal	can, shall
NN	Noun, singular or mass	Cat
NNS	Noun, plural	Cats
NNP	Proper noun, singular	Google
NNPS	Proper noun, plural	Carolinas
PDT	Predeterminer	a lot of, both
POS	Possessive ending	s
PRP	Personal pronoun	I, you

Tag	Desksripsi	Contoh
PRP$	Possessive pronoun	my, your
RB	Adverb	Fast, Slowly
RBR	Adverb, comperative	Faster, More slowly
RBS	Adverb, superlative	Fastest, Most Slowly
RP	Particle	up, off
SYM	Symbol	!, ?
TO	"to"	To
UH	Interjection	ah, oops, wow
VB	Verb, base form	Go
VBD	Verb, past tense	Went
VBG	Verb, gerund or present participle	Going
VBN	Verb, past participle	Gone
VBP	Verb, non-3rd person singular	Go
VBZ	Verb, 3rd person singular	Goes
WDT	Wh-determiner	which, that
WP	Wh-pronoun	what, who
WP$	Possessive wh-pronoun	Whose
WRB	wh-adverb	how, where

2.9 Pattern Knowledge

Pattern Knowledge merupakan salah satu metode untuk mengekstraksi fitur yang ada pada kata benda, kata sifat, atau frasa kata benda dengan menggunakan pola tertentu. Berikut merupakan pola atau *pattern knowledge* untuk menentukan fitur produk [4]:

Tabel 2.2 *Rule* pada *Pattern Knowledge*

Pola	Kata Pertama	Kata Kedua	Kata Ketiga
Pola 1	JJ	NN/NNS	-
Pola 2	JJ	NN/NNS	NN/NNS
Pola 3	RR/RBR/RBS	JJ	-
Pola 4	RR/RBR/RBS	JJ/ RR/RBR/RBS	NN/NNS
Pola 5	RR/RBR/RBS	VBN/VBD	-
Pola 6	RR/RBR/RBS	RR/RBR/RBS	JJ
Pola 7	VBN/VBD	NN/NNS	-
Pola 8	VBN/VBD	RB/RBR/RBS	-

2.10 WordNet

WordNet merupakan kamus leksikal menggunakan Bahasa Inggris kata benda, kata sifat, kata kerja, dan keterangan digabungkan dalam sebuah *synsets* yang dihubungkan oleh *conceptual-semantic* dan hubungan leksikal. Struktur yang dimiliki WordNet membantu dalam *computational linguistic* dan *Natural Language Project* (NLP) [5] [17]. Kamus ini nantinya digunakan pada klasifikasi untuk menentukan positif atau negatif suatu fitur.

2.11 Nearest Opinion Word

Nearest Opinion Word merupakan cara dalam proses *Feature-Opinion Association* (FOA) *Problem*, yaitu dengan memasangkan kata opini berorientasi positif atau negatif dengan fitur terdekat. Metode ini digunakan untuk mencari kedekatan antara kata opini dan fitur, apabila kata opini memiliki jarak terdekat dengan fitur dan jaraknya memenuhi, maka menjadi kata opini dari fitur terdekat tersebut. Kedekatan dinyatakan dengan nilai *rel(f,w)* terbesar, dengan rumus penghitungan:

$$rel(f,w) = \frac{1}{dist(f,w)} \tag{2.1}$$

Keterangan dari rumus di atas adalah sebagai berikut: *rel(f.w)* merupakan nilai invers dari jarak antara kata opini (*w*) dengan kata fitur produk (*f*) dinyatakan dengan *dist(f,w)* [18]. Semakin besar jarak kata opini dan fitur maka nilai kedekatannya semakin kecil, dan sebaliknya semakin pendek jarak kata dan fitur makan nilai kedekatannya semakin besar. Metode ini juga memberikan batasan atau *threshold* yang menjadi batas maksimum kedekatan suatu kata opini dengan kata fitur.

2.12 Peringkasan

Peringkasan adalah pemrosesan teks yang akan diproduksi menjadi satu atau lebih kalimat yang mengandung informasi penting berasal dari teks aslinya. Tujuannya adalah untuk mendapatkan informasi sebanyak-banyaknya dengan waktu yang singkat. Peringkasan teks dapat diklasifikasikan menjadi dua yaitu abstraktif dan ekstraktif. Abstraktif adalah peringkasan yang menunjukkan inti dari teks yang diungkapkan dengan bahasa. Ekstraktif adalah peringkasan yang memilih atau mengklasifikasikan berdasarkan info penting [19].

2.13 TF-IDF Scoring

Term Frequency (TF) adalah banyaknya *term* atau kata yang muncul dalam satu dokumen. Penilaian pada tf dilakukan dengan menghitung jumlah kemunculan *term* dalam dokumen.

Inverse Document Frequency (IDF) adalah banyaknya dokumen yang memunculkan suatu *term*. Semakin jarang suatu *term* muncul maka semakin tinggi nilai idf-nya [13]. Berikut rumus penghitungan idf:

$$idf_{\mathrm{t,d}} = \log(\frac{N}{df_t}) \tag{2.2}$$

Frekuensi kemunculan kata di dalam dokumen menunjukkan seberapa penting kata tersebut dalam dokumen. Frekuensi dokumen yang mengandung kata tersebut menunjukkan seberapa umum kata tersebut. Sehingga bobot hubungan antara sebuah kata dan sebuah dokumen akan tinggi apabila frekuensi kata tersebut tinggi di dalam dokumen dan frekuensi keseluruhan dokumen yang mengandung kata tersebut rendah pada kumpulan dokumen [20]. Rumus umum tf-idf:

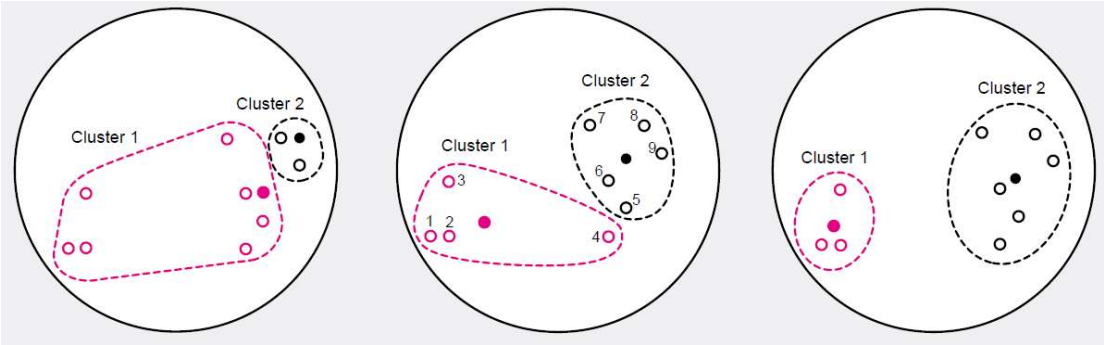
$$w_{t,d} = tf \text{ x } idf$$
$$w_{t,d} = tf_{t,d} \text{ x } \log(\frac{N}{df_t})$$

Berdasarkan rumus di atas, berapapun nilai $tf_{t,d}$, apabila N = df_t maka akan didapatkan hasil 0 untuk perhitungan idf. Untuk itu ditambahkan nilai 1 pada sisi idf, sehingga perhitungan bobotnya menjadi:

$$w_{t,d} = tf_{t,d} \text{ x } (1 + \log\left(\frac{N}{df_t}\right)) \tag{2.4}$$

2.14 Centroid Based Clustering

K-means adalah salah satu algoritma *unsupervised clustering* untuk pengelompokkan sesuai kesamaannya. Berikut ilustrasi k-means:



Gambar 2.1 Ilustrasi perubahan *centroid* pada *k-means* [21]

Ide utamanya adalah menentukan banyaknya *k-centroid*, satu untuk masing-masing *cluster. Centroid-centroid* diletakkan sejauh mungkin antara satu dan lainnya. Selanjutkan memetakan masing-masing titik yang berupa data (*d*) masuk ke dalam salah satu *cluster* bergantung dari nilai *centroid* (*c*) terdekatnya. Rumus yang digunakan untuk menghitung kedekatan keduanya menggunakan *cosine similarity* [13]:

$$sim(c,d) = \frac{\sum_{i=1}^{n} c_i d_i}{\sqrt{\sum_{i=1}^{n} c_i}\sqrt{\sum_{i=1}^{n} d_i}} \tag{2.5}$$

Saat seluruh titik telah memiliki *cluster*, hitung nilai *centroid* baru dari nilai rata-rata seluruh data di *cluster* tersebut. Setelah itu kembali pertakan masing-masing titik dengan *centroid* terdekatnya seperti cara sebelumnya. Pengulangan ini terus di lakukan hingga letak seluruh centroid tidak berubah lagi atau tetap pada posisinya [6].

2.15 Evaluasi

Evaluasi digunakan untuk melakukan perhitungan ketepatan atau akurasi. Perhitungan *Precision, Recall*, dan *F-Score* merupakan cara yang digunakan untuk perhitungan akurasi dokumen apakah sesuai dengan informasi yang diinginkan. Sebelum dilakukannya penghitungan digunakan *confusion matrix* yang berisi nilai hasil prediksi.

Tabel 2.3 *Confusion Matrix*

	Benar	**Salah**
Diidentifikasi benar	TP	FP
Diidentifikasi salah	FN	TN

Berikut empat kategori dokumen dalam proses pencarian [12], yaitu:

1. *True Positives* (TP) adalah item yang diidentifikasi benar dan faktanya item tersebut bernilai benar.
2. *True Negatives* (TN) adalah item yang diidentifikasi salah dan faktanya item tersebut bernilai salah.

3. *False Positives* (FP) adalah item yang diidentifikasi benar, namun faktanya item tersebut bernilai salah.
4. *False Negatives* (FN) adalah item yang diidentifikasi salah, namun faktanya item tersebut bernilai benar.

Precision digunakan untuk menghitung banyaknya item yang teridentifikasi relevan. *Precision* memiliki rumus:

$$Precision = \frac{TP}{TP+FP} \tag{2.6}$$

Recall digunakan untuk menghitung banyaknya item relevan yang berhasil diidentifikasi. *Recall* memiliki rumus*:*

$$Recall = \frac{TP}{TP+FN} \tag{2.7}$$

F-Score mengnhitung akurasi keseluruhan yang merupakan penggabungan *precision* dan *recall*. *F-Score* memiliki rumus:

$$F - Score = \frac{2\ x\ Prec\ x\ Rec}{Prec + Rec} \tag{2.8}$$

Evaluasi ekstraksi dilakukan menggunakan perhitungan level kalimat dengan contoh perhitungan sebagai berikut:

Tabel 2.4 Contoh Evaluasi Ekstraksi

No	Fitur Pada Corpus	Fitur Teridentifikasi	Precision	Recall
1.	-	-	1.0	1.0
2.	-	camera	0.0	0.0
3.	picture	-	0.0	0.0
4	battery canon	battery	1.0	0.5
5.	canon	canon memory	0.5	1.0
6	image	memory	0.0	0.0
Rata-rata			0.42	0.42

Evaluasi dilakukan pada level kalimat. Dapat dilihat dari contoh kasus diatas perhitungan *precision* dan *recall* didapat dari kesesuaian antara fitur pada dokumen dengan fitur yang teridentifikasi.

Evaluasi klasifikasi dilakukan pada level kalimat. Penghitungan akurasi dilakukan dengan membandingkan nilai polaritas fitur yang ada pada dokumen dengan nilai polaritas fitur teridentifikasi pada sistem. Berikut rumus evaluasi klasifikasi:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.9}$$

Akurasi hanya dihitung pada kalimat yang memiliki fitur teridentifikasi dan berpolaritas. Hasil akurasi dari masing-masing kalimat dijumlahkan dan dihitung rata-ratanya untuk mendapatkan akurasi klasifikasi. Penghitungan hanya melibatkan fitur teridentifikasi yang telah dibandingkan dengan *dataset* karena klasifikasi hanya memasangkan fitur produk teridentifikasi dengan opini. Berikut contoh evaluasi klasifikasi pada level kalimat:

Tabel 2.5 Contoh Evaluasi Klasifikasi

No	Polaritas Fitur pada Dokumen	Polaritas Fitur Teridentifikasi	Akurasi
1	-	-	0.0
2	picture[+]	picture[+]	1.0
3	picture[-]	picture[-]	1.0
4	picture[+]	picture[-]	0.0
5	picture[+]	picture[+] picture[+]	1.0
6	picture[+]	picture[+] picture[-]	0.5
7	picture[+] camera[+]	picture[+] picture[-] picture[-] camera[+]	0.5
8	picture[+] picture[-]	picture[+]	0.5
Rata-Rata			4.5/7 = 64.3 %

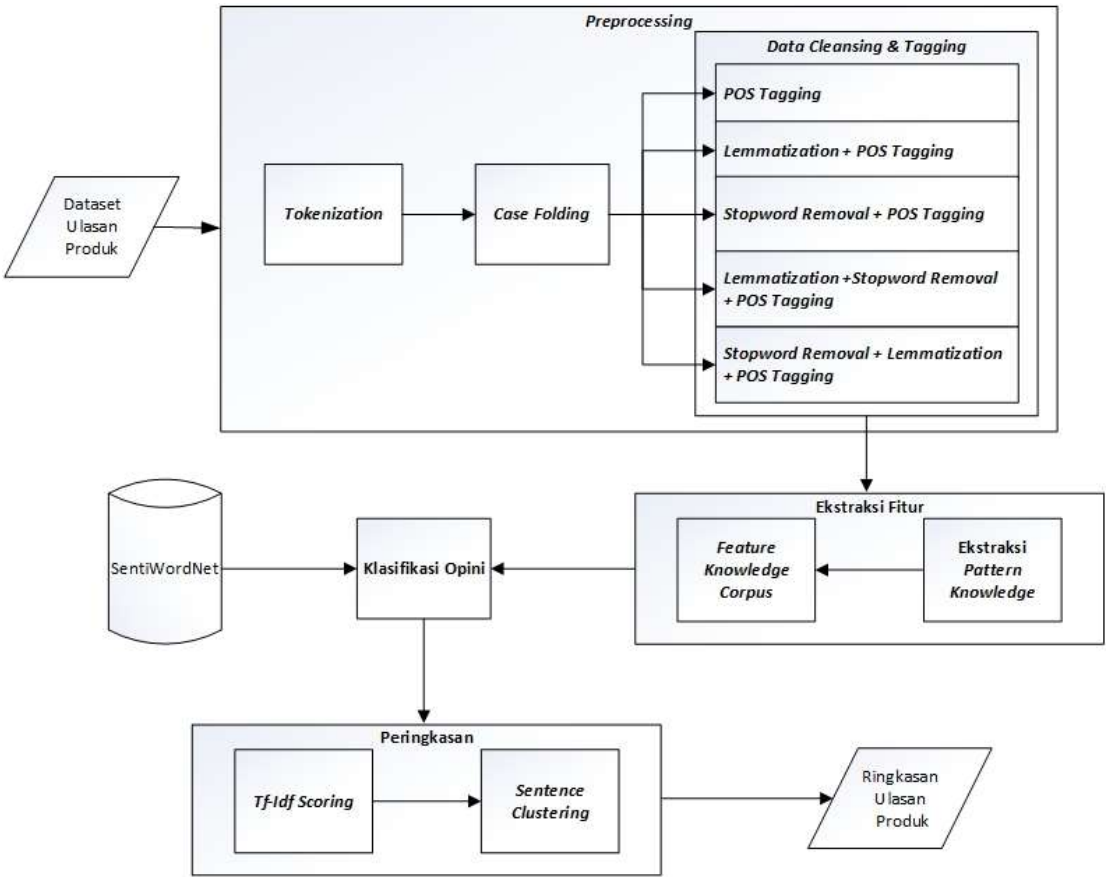
Evaluasi *cluster* dilakukan dengan memberikan nilai kualitas pada *cluster*. Penghitungan kualitas *cluster* dilakukan dengan menghitung rata-rata jarak antar-*cluster* dibagi dengan rata-rata jarak data dengan *centroid* dalam *cluster* [21]

$$Cluster\ quality = \frac{Distance\ between\ cluster}{Distance\ within\ cluster} = \frac{\frac{1}{n}\sum_{c=1}^{n} Distance_{c,c}}{\frac{1}{d}\sum_{i=1}^{d} Distance_{c,d}} \quad (2.10)$$

3 Perancangan Sistem

3.1 Gambaran Umum Sistem

Sistem yang dibangun dalam peneliatian ini adalah sebuah sistem yang mampu menentukan opini dan melakukan peringkasan ulasan produk. Peringkasan ulasan produk memiliki tahapan umum, yaitu *preprocessing* data, ekstraksi fitur, klasifikasi orientasi opini, dan pembangkitan ringkasan. Berikut gambaran umum sistem:



Gambar 3.1 Gambaran Umum Sistem

3.2 Rancangan Sistem

Berikut merupakan penjelasan rancangan sistem di tiap tahapan.

3.2.1 Data

Data berisi dokumen yang menyimpan ulasan produk yang digunakan dalam analisis. Data yang digunakan adalah ulasan produk dari jurnal Minqing Hu dan Bing Liu [7]. Berikut merupakan contoh salah satu data:

Kalimat:

```
LCD[-2]##The LCD screen is too small since there are so many cameras
that have larger ones.
```

Keterangan:

```
Fitur Produk: LCD

Polaritas: [+2] Positif 2

Kalimat opini: The LCD screen is too small since there are so many
cameras that have larger ones.
```

Data tersebut dilengkapi berberapa informasi, antara lain:

Tabel 3.1 Makna Simbol dalam *Dataset*

Simbol	Keterangan
[t]	Judul dari ulasan produk, merupakan awal dari komentar
xxx[+n|-n]	xxx merupakan *knowledge* fitur produk
[+n]	Opini berpolaritas positif dengan nilai n kekuatan opini. Nilai n terbesarnya adalah 3 dan nilai terkecilnya adalah 1
[-n]	Opini berpolaritas negatif dengan nilai n kekuatan opini. Nilai n terbesarnya adalah 1 dan nilai terkecilnya adalah 3
##	Tanda awal kalimat komentar
[u]	Fitur produk yang tidak muncul pada kalimat komentar
[p]	Fitur produk yang tidak muncul pada kalimat komentar, dibutuhkan adanya *pronoun resolution*
[s]	Rekomendasi
[cc]	Perbandingan dengan produk lain dari merek dagang berbeda
[cs]	Perbandingan dengan produk lain dari merek dagang sama

3.2.2 Preprocessing

Proses analisis sentimen yang pertama adalah *preprocessing*. Tujuan proses ini adalah untuk membersihkan data mentah untuk mendapatkan data baru yang memiliki potensi untuk dapat dikembangkan menjadi informasi baru.

3.2.2.1 Tokenisasi

Tokenisasi adalah proses memecah teks menjadi masing-masing unit seperti kata, tanda baca, nomor, dll. Berikut contoh tokenisasi:

Kalimat *Input*:

```
The LCD screen is too small since there are so many cameras that have
larger ones
```

Hasil Tokenisasi:

```
The | LCD | screen | is | too | small | since | there | are | so |
many | cameras | that | have | larger | ones
```