

**PERINGKASAN REVIEW PRODUK BERBASIS FITUR DENGAN  
MENGUNAKAN USER SCORE**

**FEATURE BASED REVIEW PRODUCT SUMMARIZATION USING  
USER SCORE**

Keri Wisnu Nurhidayat<sup>1</sup>, Warih Maharani<sup>2</sup>, Mohamad Syahrul Mubarak<sup>3</sup>

<sup>1,2,3</sup>Prodi S1 Teknik Informatika, Fakultas Informatika, Universitas Telkom

<sup>1</sup>[keriwisnu@gmail.com](mailto:keriwisnu@gmail.com), <sup>2</sup>[wmaharani@gmail.com](mailto:wmaharani@gmail.com), <sup>3</sup>[msvahrulmubarak@gmail.com](mailto:msvahrulmubarak@gmail.com)

---

**Abstrak**

Ulasan sebuah produk dari konsumen dapat digunakan oleh perusahaan untuk meningkatkan penjualan dan kualitas produknya sedangkan bagi konsumen dapat memberikan keputusan dalam pembelian. Dengan meningkatnya jumlah ulasan serta *user score* dari konsumen memunculkan masalah baru yaitu ketika ulasan yang diberikan oleh konsumen tidak sesuai dengan *user score* yang diberikan. Untuk itu dibangun sebuah sistem yang dapat melakukan peringkasan untuk memberikan nilai terhadap ulasan konsumen dengan bantuan *user score* dari konsumen. Tahapan pertama untuk membangun sistem ini adalah *preprocessing* dimana *preprocessing* terdiri dari 3 tahapan yaitu *Stopword Removal*, *Lemmatization*, dan *POS Tagging*. Tahap kedua adalah ekstraksi fitur dan opini dengan menggunakan *Type Dependency Parser*. Tahap ketiga adalah penentuan polaritas fitur dengan menggunakan *Semantic Orientation – Pointwise Mutual Information*. Tahap keempat adalah peringkasan dengan menggunakan *feature scoring* dan *Pearson Correlation* untuk mengetahui nilai dari setiap fitur yang terekstrak. Hasil proses ekstraksi dan penentuan polaritas adalah daftar fitur dan opini serta polaritas dari setiap fitur. Peringkasan dokumen dengan menggunakan *feature scoring* menghasilkan skor dari setiap fitur yang berhasil terekstrak pada setiap ulasan yang kemudian dilanjutkan dengan proses *Pearson Correlation* untuk melihat *trend* antara skor dari *dataset* dengan skor yang dihasilkan oleh *feature scoring* dimana nilai korelasi tertinggi adalah pada *dataset* Nook Tablet 16Gb yaitu fitur ke-46 *Seller* dengan nilai korelasi 1.00 atau berkorelasi sangat kuat dan nilai korelasi terendah adalah pada *dataset* Nook Tablet 16Gb yaitu fitur ke-62 *Unit* dengan nilai korelasi -0.94 atau berkorelasi terbalik sangat kuat.

**Kata kunci:** analisis sentimen, ulasan produk, *feature based opinion summarization*, *type dependency parser*, *semantic orientation – pointwise mutual information*, *feature scoring*, *scoring product feature*, *pearson correlation*.

---

**Abstract**

Product reviews from consumers can be used by a company to increase sales and product quality, and for consumers product reviews can provide a purchase decision. The increasing number of reviews and user scores from consumers raises a new problem that is when the reviews provided by consumers are not in accordance with the given user score. So, a system is needed to perform a summary to provide value to consumer reviews with the help of user scores from consumers. The first stage to build this system is preprocessing where preprocessing consists of 3 stages, Stopword Removal, Lemmatization, and POS Tagging. The second stage is the feature and opinion extraction using the type dependency parser. The third stage is the determination of feature polarity by using Semantic Orientation – Pointwise Mutual Information. The fourth stage is a summary using feature scoring and Pearson correlation to find out the value of each extracted feature. The result of the extraction process and the polarity determination is a list of features and opinions as well as the polarity of each feature. Document summary using feature scoring generates scores of each feature successfully extracted in each review, followed by a Pearson Correlation process to see trends between scores of datasets and scores generated by feature scoring, and the highest correlation value is on the 16Gb Nook Tablet dataset is the 46th feature, Seller with a correlation value of 1.00 or very string correlated, while the lowest correlation is on the 16Gb Nook Tablet dataset that is the 62nd feature, Unit with a correlation value of -0.94 or very strong inverse correlation.

**Keywords:** sentiment analysis, product review, feature based opinion summarization, type dependency parser, semantic orientation- pointwise mutual information, feature scoring, pearson correlation.

---

## 1. Pendahuluan

Meningkatnya transaksi dalam bentuk *online* baik melalui *e-commerce* maupun toko *online* memberikan kesempatan bagi perusahaan untuk berlomba-lomba meningkatkan penjualan dan kualitas *e-commerce* maupun toko *online* dengan cara memberikan kesempatan bagi konsumen untuk memberikan ulasan dan *rating* atau *user score* terhadap produk yang dijual oleh perusahaan. Ulasan dan *user score* produk tersebut dapat digunakan juga oleh konsumen untuk memberikan keputusan pembelian. Namun seiring meningkatnya jumlah ulasan serta *user score* dari konsumen memunculkan sebuah masalah yaitu ketika ulasan yang diberikan oleh konsumen tidak sesuai dengan *user score* yang diberikan, selain itu dalam sebuah kalimat memungkinkan konsumen memberikan komentar terhadap beberapa fitur sebuah produk dimana seringkali konsumen memberikan komentar baik pada fitur yang satu dan komentar buruk pada fitur yang lain serta memberikan *user score* tinggi [1]. Dengan terjadinya hal seperti itu maka sistem yang dibangun mengkombinasikan antara ulasan dan *user score* untuk mendapatkan nilai dari setiap fitur dari sebuah produk.

Kombinasi antara ulasan dan *user score* untuk mendapatkan nilai dari setiap fitur sebuah produk adalah metode *feature scoring*. Metode *feature scoring* adalah sebuah metode peringkasan dimana metode ini menghasilkan nilai dari setiap fitur dari sebuah produk sehingga calon pembeli dapat mengetahui lebih detail produk yang akan dibeli [1].

Sistem peringkasan ini terdiri dari 4 tahap yaitu, (1) *preprocessing*, (2) ekstraksi fitur dan opini, (3) penentuan polaritas, (4) peringkasan. Tahap pertama yang dilakukan untuk melakukan peringkasan adalah tahap *preprocessing*, tahap ini adalah tahap pembersihan data dimana dalam tahap ini dilakukan 3 kali pemrosesan data yaitu, (1) *Stopword Removal*, (2) *Lemmatization*, (3) *POS Tagging*. Setelah data dibersihkan dengan melalui 3 tahap *preprocessing* kemudian data tersebut dimasukkan kedalam proses ekstraksi fitur dan opini. metode ekstraksi fitur dan opini yang digunakan yaitu *Type Dependency Parser*. Setelah didapatkan daftar fitur dan opini dari tahap Ekstraksi selanjutnya dilakukan tahap penentuan polaritas untuk menentukan polaritas dari ulasan tersebut apakah frasa tersebut termasuk kedalam frasa positif atau negatif dengan menggunakan metode *Semantic Orientation-Pointwise Mutual Information* [1]. Setelah proses penentuan orientasi dilakukan maka dilanjutkan dengan peringkasan ulasan produk dengan melakukan *feature scoring* yang akan di bangun berdasar dari hasil penentuan polaritas yang telah didapat.

## 2. Landasan Teori

### 2.1 Analisis Sentimen

Analisis sentimen mining merupakan sebuah bidang studi yang menganalisis pendapat, sentimen, evaluasi, sikap dan emosi terhadap suatu entitas seperti produk, jasa, organisasi, individu, masalah, topik dan atribut dari entitas tersebut [2]. Ada banyak sebutan dari analisis sentimen ini seperti *opinion mining*, *opinion extraction*, *sentimen mining*, *subjectivity analysis*, *affect analysis*, *emotion analysis*, *review mining* dan lain sebagainya. Analisis sentimen dapat dilakukan pada beberapa level seperti level dokumen, level kalimat, level entitas, dan level aspek dengan setiap level tersebut akan menghasilkan sentimen positif dan sentimen negatif

### 2.2 Lemmatization

*Lemmatization* adalah salah satu teknik normalisasi dimana setiap kata yang berinfleksi akan diubah ke dalam bentuk kata dasar dari kata tersebut [7]. Proses *Lemmatization* sejatinya adalah proses penghapusan infleksi dan mengembalikan kata kedalam bentuk kata dasar tanpa menghilangkan makna awal kata. Contoh pada kalimat “*supposed purchase fast shipping connect camera*” hasil dari proses *Lemmatization*nya adalah “*suppose purchase fast ship connect camera*”.

### 2.3 Dependency Parsing

*Dependency parser* pada sebuah kalimat merupakan sebuah proses penjabaran relasi dari bagian-bagian kalimat (kata-kata) tersebut berdasarkan *Dependency Grammar*. *Dependency Grammar* merupakan sebuah teori sintaksis yang mendasarkan pada *Dependency Relation* atau hubungan kata dalam kalimat.

Kalimat akan dibagi kedalam beberapa kata pada *dependency tree*, dimana kata akan saling berhubungan atau berinteraksi antara *head* dengan *dependant*. Kemudian akan ada dua kata yang dihubungkan oleh *dependency relation*, dimana diantara keduanya merupakan *head* yang satunya lagi merupakan *dependant* yang kemudian ada *link* yang menghubungkan antara kedua kata tersebut. Secara umum, *dependent* merupakan *modifier*, *object*, atau *complement*. *Head* memiliki peran yang lebih besar dalam menentukan perilaku pasangannya dan *dependent* memerlukan adanya *head* [10].

## 2.4 Semantic Orientation-Pointwise Mutual Information (SO-PMI) Algorithm

Algoritma *SO-PMI* (*Semantic orientation – pointwise mutual information*) adalah algoritma untuk menentukan orientasi positif dan negatif dari frasa dalam kalimat. Proses penentuan dilakukan dengan cara menghitung kedekatan kata dalam setiap frasa dalam kalimat dengan referensi kata yang ditentukan sebelumnya dengan bantuan *library* BING API [12]. Pada BING API kata yang ditentukan sebagai referensi positif adalah “*excellent*” dan kata dengan referensi negatif adalah “*poor*” sesuai dengan *five star rating system*. Nilai SO-PMI dapat dilihat pada persamaan (1).

$$SO - PMI = \log_2 \left( \frac{hits(\text{phrase NEAR} "Excellent") hits ("Poor")}{hits(\text{phrase NEAR} "Poor") hits ("Excellent")} \right) \quad (1)$$

$hits(\text{phrase NEAR} "excellent") hits("poor")$  adalah nilai *hits* dari frasa yang dilihat kemunculannya dengan “*excellent*” yang kemudian akan dikalikan dengan kemunculan dari kata “*poor*” sedangkan  $hits(\text{phrase NEAR} "poor") hits("excellent")$  adalah nilai *hits* dari frasa yang dilihat kemunculannya dengan “*poor*” yang kemudian akan dikalikan dengan kemunculan dari kata “*excellent*”. Hasil yang didapat dari perhitungan pada persamaan 2.1 akan menentukan orientasi dari fitur, ketika fitur bernilai lebih dari 0 maka fitur berorientasi positif sedangkan ketika fitur bernilai kurang dari 0 maka fitur berorientasi negatif.

## 2.5 Feature Scoring

*Feature scoring* adalah sebuah metode untuk mencari nilai dari fitur yang ada didalam sebuah ulasan dengan syarat fiturnya sudah diketahui arah orientasi agar dapat ditentukan rumus manakah yang akan digunakan [1]. Persamaan *feature scoring* yang digunakan untuk mendapatkan skor pada setiap fiturnya dapat dilihat pada persamaan (2).

$$e_{ij} = \begin{cases} s_j + \frac{D}{2} \cdot \frac{N_{negative}(R_j)}{N_{total}(R_j)} \cdot w_{ij}, & \text{if } P_y \text{ is a positive number} \\ s_j - \frac{D}{2} \cdot \frac{N_{positive}(R_j)}{N_{total}(R_j)} \cdot w_{ij}, & \text{if } P_y \text{ is a negative number} \end{cases} \quad (2)$$

$S_j$  adalah skor dari dataset dari ulasan ke-j.  $D$  adalah perbedaan nilai maksimal dan minimal dalam ulasan.  $N_{total}(R_j)$  adalah jumlah seluruh fitur didalam komentar ke-j.  $N_{negative}(R_j)$  adalah jumlah fitur negatif yang terdapat pada komentar ke-j.  $N_{positive}(R_j)$  adalah jumlah fitur positif yang terdapat pada komentar ke-j [1].  $W_{ij}$  adalah sebuah bobot yang digunakan untuk menyesuaikan *feature scoring* berdasarkan *user score* sesuai dengan persamaan (3).

$$w_{ij} = \frac{F(f_{ij}, R_j) \cdot N_f(p_{ij}, R_j)}{f_{sum}(p_{ij}, R_j)} \quad (3)$$

$F(f_{ij}, R_j)$  adalah frekuensi fitur ke-i disebutkan didalam komentar j.  $N_f(p_{ij}, R_j)$  adalah frekuensi fitur yang memiliki polaritas sama dengan  $p_{ij}$  yang terdapat pada komentar j namun selain fitur itu sendiri ( $F_{sum}(p_{ij}, R_j) - 1$ ).  $F_{sum}(p_{ij}, R_j)$  adalah frekuensi yang memiliki polaritas yang sama dengan  $p_{ij}$  [1].

## 2.6 Pearson Correlation

Pearson’s correlation adalah sebuah metode untuk mengukur derajat asosiasi atau hubungan linier antara dua variabel yang didefinisikan sebagai kovarians dari variabel dibagi dengan standar deviasinya. Perhitungan korelasi menggunakan *Pearson’s Correlation* umumnya digunakan pada data yang bersifat kuantitatif dan kedua variabel merupakan bivariat yang memiliki distribusi normal [13]. Perhitungan *Pearson’s Correlation* dapat dilihat pada persamaan (3) dan tingkat kekuatan korelasi dapat dilihat pada tabel 2-1.

$$r(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (3)$$

Dengan  $r$  adalah koefisien korelasi pearson. Total pasangan kata adalah  $n$ . Variabel  $x$  ke-i (skor dari dataset) adalah  $x_i$ . Variabel  $y$  ke-i (skor hasil keluaran *feature scoring*)  $y_i$ .  $\bar{x}$  adalah rata – rata dari variabel  $x$ .  $\bar{y}$  adalah rata – rata dari variabel  $y$ .

Table 1-1 Tingkat Kekuatan Korelasi

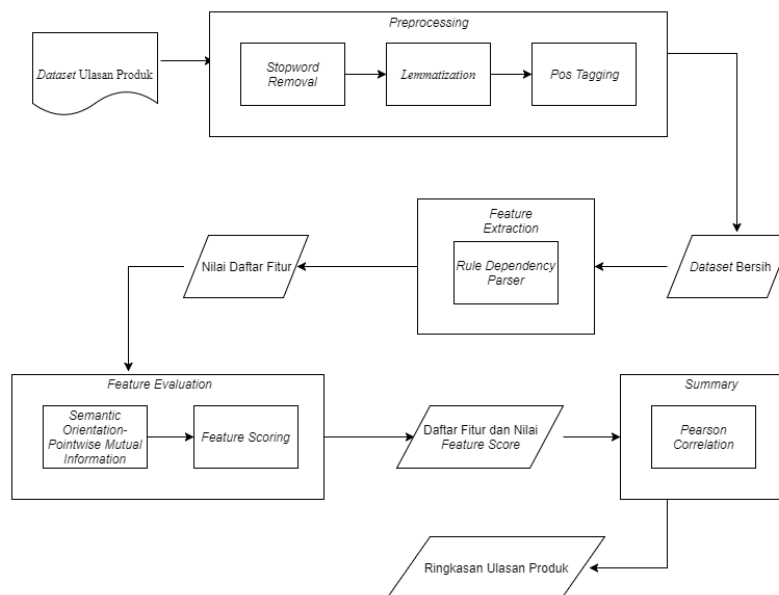
| Rentang   | Tingkat Kekuatan Korelasi |
|-----------|---------------------------|
| 0,00-0,19 | Sangat Lemah              |
| 0,20-0,39 | Lemah                     |
| 0,40-0,59 | Sedang                    |
| 0,60-0,79 | Kuat                      |

|           |             |
|-----------|-------------|
| 0,80-1,00 | Sangat Kuat |
|-----------|-------------|

### 3. Pembahasan

#### 3.1 Gambaran Umum Sistem

Tahapan dalam penentuan opini ini terdiri dari empat bagian utama yaitu: (1) Ekstraksi fitur yang berdasar dari data *review*, (2) Identifikasi orientasi fitur dan opini dari sebuah kalimat, (3) Penghitungan nilai dari setiap fitur yang telah dideteksi (4) Pembangkitan ringkasan. Gambaran sistem yang dibangun ditunjukkan pada Gambar 3-1.



Gambar 3-1 Gambaran Umum Sistem

#### 3.2 Ekstraksi Fitur dan Opini

Ringkasan ulasan membutuhkan tahapan ekstraksi fitur dan opini yang bertujuan untuk menemukan fitur dan opini berdasarkan tanggapan yang diberikan *user*. Kata benda yang sering muncul di tanggapan dapat dijadikan kandidat potensial untuk fitur [14]. Sistem ini menggunakan *Dependency Parser* yang pengambilan hasilnya disesuaikan dengan aturan dari *Rules Dependency Relation*.

*Type dependency parser* merupakan sebuah metode ekstraksi yang mengambil *type dependency* kalimat yang kemudian disesuaikan dengan *rules dependency relation*. Proses ekstraksi dimulai dengan membaca hasil data *preprocessing* kemudian dilakukan proses *parsing* menggunakan *library Stanford Core NLP*. Proses yg dilakukan oleh *dependency parser* ialah mengeluarkan hubungan pada struktur kata dalam kalimat. Berikut adalah contoh kalimat yang akan diproses dengan parser.

**Kalimat:**

Works great Instructions were straightforward and easy to follow buying a second one just in case I need it in the future

**Hasil Dependency Parser:**

[root(ROOT-0, work-1), amod(instruction-3, great-2), nsubj(follow-6, instruction-3), amod(easy-5, straightforward-4), amod(instruction-3, easy-5), ccomp(work-1, follow-6), xcomp(follow-6, buy-7), nn(future-9, case-8), dobj(buy-7, future-9)]

Hasil dari proses *parser* yang telah dilakukan kemudian akan dimasukkan ke dalam *rules dependency relation*. *Rules dependency relation* merupakan aturan untuk mengecek hasil *parser* dan mengambil pasangan fitur dan opini hasil ekstraksi dari *parser* tersebut. Aturan pengambilan kalimat yang digunakan oleh *rules dependency relation* dapat dilihat pada Tabel 3-1.

| <i>Rules</i>                | <b>Fitur</b> | <b>Opini</b> |
|-----------------------------|--------------|--------------|
| NN – amod – JJ              | NN           | JJ           |
| NN – nsubj – JJ             | NN           | JJ           |
| NN – nsubj – VB – dobj – NN | NN pertama   | NN Terakhir  |
| VB – avmod - RB             | VB           | RB           |

Tabel 3- 1 Rule Dependency Relation [14]

Hasil dari *parser* kemudian dilakukan POS *tag* menggunakan library POS *tagger* untuk mengetahui *tag* setiap kata. Hasil *parser* dengan *tag* kemudian akan dicocokkan dengan *rules dependency relation*. Output dari pencocokan *rules* akan disimpan sebagai kandidat fitur dan opini. Pasangan fitur dan opini dari proses *parser* dapat dilihat pada tabel 3-2.

| Pasangan sesuai <i>rules</i>     | Fitur       | Opini |
|----------------------------------|-------------|-------|
| instruction_NN - amod – great_JJ | Instruction | Great |
| instruction_NN – amod – easy_JJ  | Instruction | Easy  |

Tabel 3- 2 Pasangan Fitur dan Opini yang Terekstrak

Pada contoh pertama di atas proses ekstraksi fitur dan opini berdasarkan relasi **amod** dengan *headnya* adalah **instruction** (*noun*) dan *dependencynya* adalah **great** (*adjective*). Pasangan fitur dan opini ini sesuai dengan *rules dependency* pertama NN-**amod**-JJ sehingga akan disimpan sebagai pasangan fitur dan opini hasil ekstraksi.

### 3.3 Semantic Orientation – Pointwise Mutual Information

*Semantic orientation* bertujuan untuk melakukan klasifikasi atas frasa pasangan fitur dan opini yang telah di ekstraksi. Pada tugas akhir ini penentuan polaritasnya diproses dengan menggunakan algoritma SO-PMI. Algoritma SO-PMI bekerja dengan menghitung rata-rata orientasi dari suatu opini. Opini positif ditentukan dengan menghitung kedekatan suatu frasa pasangan fitur dan opini dengan kata yang mempunyai referensi positif seperti “*Excellent*” dan opini negatif ditentukan dengan menghitung kedekatan frasa pasangan fitur dan opini dengan kata yang mempunyai referensi negatif seperti “*poor*” [12].

Kata “*excellent*” dan “*poor*” dipilih karena dalam *five stars rating system* kata yg termasuk one star adalah “*poor*” dan kata yang termasuk *five star* yaitu “*excellent*”. jadi orientasi positif akan lebih kuat jika semakin dekat dengan kata “*excellent*” sebaliknya kata dengan orientasi negatif yang kuat jika semakin dekat dengan kata “*poor*” [8]. Pada tugas akhir ini perhitungan kedekatan kata dilakukan dengan menggunakan BING API. Mesin pencari BING mempunyai *NEAR* operator yang dapat menghitung kedekatan maksimal 10 kata antara satu dan lainnya. Perhitungan *semantic orientation* dengan operator *NEAR* yang memanfaatkan mesin pencari BING diinterpretasikan dengan persamaan (2).

Persamaan (2) dilakukan setelah pencarian kedekatan frasa dengan menghitung hits kedekatan kata yang muncul melalui operator *NEAR* pada search engine *BING*. Berikut adalah contoh hasil perhitungan dengan menggunakan persamaan (2) yang dapat dilihat pada Tabel 3-3 dengan hasil pada Tabel 3-4.

| <i>Hits</i> SO-PMI   | Nilai               |
|--|---------------------|
| <i>Hits</i> “ <i>excellent</i> ”   | 10762.0             |
| <i>Hits</i> “ <i>poor</i> ”  | 10929.0             |
| <i>Hits</i> ( <i>great instruction NEAR</i> “ <i>excellent</i> ”)  | 8127.0              |
| <i>Hits</i> ( <i>great instruction NEAR</i> “ <i>poor</i> ”)   | 7560.0              |
| SO – PMI = $\log_2 \left( \frac{\text{hits}(\text{phrase NEAR} \text{ "Excellent"}) \text{hits} (\text{ "Poor"})}{\text{hits}(\text{phrase NEAR} \text{ "Poor"}) \text{hits} (\text{ "Excellent"})} \right)$ | 0.12655184882389983 |

Tabel 3- 3 Hasil Perhitungan SO-PMI

| Fitur dan Opini yang Terekstrak | Tag dari POS Tagging | Orientasi Fitur dan Opini |
|---------------------------------|----------------------|---------------------------|
| Great Instruction               | JJ NN                | 0.1266                    |
| Good Price                      | JJ NN                | 0.9266                    |
| Easy Instruction                | JJ NN                | -0.0070                   |
| Steady Tv                       | JJ NN                | -0.0124                   |

Tabel 3- 4 Hasil Orientasi Fitur dan Opini

Dari Tabel 3-4 dapat dilihat hasil *semantic orientation* dari sebuah frasa Great Instruction adalah lebih dari 0 yang berarti frasa tersebut berorientasi positif. Sebaliknya jika hasil rata-rata *semantic orientation* dari suatu ulasan adalah negatif maka ulasan tersebut diklasifikasikan sebagai opini negatif.

### 3.4 Feature Scoring

Pada tahap ini dilakukan pemberian nilai pada setiap fitur yang telah di ekstraksi dan sudah memiliki nilai *semantic orientation*nya seperti pada Tabel 3-5.

|                 | S   | f <sub>1</sub> | f <sub>2</sub> | f <sub>3</sub> | f <sub>4</sub> | f <sub>5</sub> | f <sub>6</sub> | f <sub>7</sub> | f <sub>8</sub> | f <sub>9</sub> | f <sub>10</sub> | f <sub>11</sub> |
|-----------------|-----|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|-----------------|-----------------|
| R <sub>1</sub>  | 5.0 |                |                |                |                | P              |                |                |                |                |                 |                 |
| R <sub>2</sub>  | 5.0 |                |                |                | P              |                |                |                |                |                |                 |                 |
| R <sub>3</sub>  | 5.0 | P              |                |                |                |                |                |                |                |                |                 |                 |
| R <sub>4</sub>  | 4.0 |                |                | N              | P              |                |                |                |                |                |                 |                 |
| R <sub>5</sub>  | 5.0 |                |                |                |                |                |                |                |                |                | P               |                 |
| R <sub>6</sub>  | 5.0 |                | P              |                |                |                |                |                |                | N              |                 |                 |
| R <sub>7</sub>  | 4.0 |                |                |                |                |                |                |                | P              |                |                 |                 |
| R <sub>8</sub>  | 4.0 |                |                |                | P              |                |                | P              |                |                |                 |                 |
| R <sub>9</sub>  | 5.0 |                |                |                | P              |                | P              |                |                |                |                 |                 |
| R <sub>10</sub> | 4.0 |                |                |                |                |                |                |                |                |                |                 | N               |

Tabel 3- 5 Tabel Hasil Semantic Orientation

Data pada Tabel 3-5 mengatakan bahwa S adalah skor yang diberikan oleh *user*. Nilai fitur ke-1, fitur ke-2 dan seterusnya didapatkan dari perhitungan dengan hasil SO-PMI. Perhitungan dengan menggunakan persamaan (3) dengan ulasan ke-4, fitur ke-3 sebagai contoh maka dihasilkan nilai seperti pada Tabel 3-6 Tabel 3-7.

| Variabel                                | Nilai | Variabel   | Nilai |
|---|-------|--|-------|
| S <sub>j</sub>                          | 4     | N <sub>negative</sub> (R <sub>j</sub> )              | 1     |
| D / 2                                   | 2     | F(f <sub>ij</sub> , R <sub>j</sub> )                 | 1     |
| N <sub>total</sub> (R <sub>j</sub> )    | 2     | N <sub>f</sub> (p <sub>ij</sub> , R <sub>j</sub> )   | 0     |
| N <sub>positive</sub> (R <sub>j</sub> ) | 1     | F <sub>sum</sub> (p <sub>ij</sub> , R <sub>j</sub> ) | 1     |

Tabel 3- 6 Perhitungan Feature Score ke-3 pada Ulasan ke-4

| Variabel  | Nilai |
|---|-------|
| $W_{ij}$  | 2 / 3 |
| $s_j + \frac{D}{2} \cdot \frac{N_{negative}(R_j)}{N_{total}(R_j)} \cdot w_{ij}$ | 4,26  |

Tabel 3- 7 Hasil Bobot dan Nilai Feature Score fitur ke-3 pada Ulasan ke-4

Dari hasil perhitungan dengan menggunakan persamaan (3) didapatkan hasil nilai setiap fitur seperti pada Tabel 3-9.

|                 | S   | f <sub>1</sub> | f <sub>2</sub> | f <sub>3</sub> | f <sub>4</sub> | f <sub>5</sub> | f <sub>6</sub> | f <sub>7</sub> | f <sub>8</sub> | f <sub>9</sub> | f <sub>10</sub> | f <sub>11</sub> |
|-----------------|-----|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|-----------------|-----------------|
| R <sub>1</sub>  | 5.0 |                |                |                |                | 5.00           |                |                |                |                |                 |                 |
| R <sub>2</sub>  | 5.0 |                |                |                | 5.00           |                |                |                |                |                |                 |                 |
| R <sub>3</sub>  | 5.0 | 5.00           |                |                |                |                |                |                |                |                |                 |                 |
| R <sub>4</sub>  | 4.0 |                |                | 3.00           | 5.00           |                |                |                |                |                |                 |                 |
| R <sub>5</sub>  | 5.0 |                |                |                |                |                |                |                |                |                | 5.00            |                 |
| R <sub>6</sub>  | 5.0 |                | 6.00           |                |                |                |                |                |                | 4.00           |                 |                 |
| R <sub>7</sub>  | 4.0 |                |                |                |                |                |                |                | 4.00           |                |                 |                 |
| R <sub>8</sub>  | 4.0 |                |                |                | 4.00           |                |                | 4.00           |                |                |                 |                 |
| R <sub>9</sub>  | 5.0 |                |                |                | 5.00           |                | 5.00           |                |                |                |                 |                 |
| R <sub>10</sub> | 4.0 |                |                |                |                |                |                |                |                |                |                 | 4.00            |

Tabel 3- 8 Hasil Perhitungan Feature Scoring Pada Setiap Fitur

### 3.5 Pearson Correlation

Pada tahap ini dilakukan peringkasan dengan menggunakan *Pearson's Correlation* yang persamaannya dapat dilihat pada persamaan (4). Nilai yang telah didapat dari tahap *feature scoring* akan dibandingkan dengan nilai skor dari dataset yang terdapat pada *corpus*. Pada Tabel 3-9 dapat dilihat contoh data yang akan dibandingkan antara hasil *feature scoring* dari Tabel 3-8 fitur ke-4.

| Skor dari dataset ( <i>User Score</i> ) | Hasil <i>Feature Scoring</i> |
|---|------------------------------|
| 5.00                                    | 5.00                         |
| 4.00                                    | 5.00                         |
| 4.00                                    | 4.00                         |
| 5.00                                    | 5.00                         |

Tabel 3- 9 Hasil Perhitungan Korelasi Fitur ke-4

Hasil perhitungan korelasi dari Tabel 3-9 dapat dilihat pada Tabel 3-10, dimana hasil korelasi dari fitur ke-4 adalah 0.58 yang mana berarti antara skor dari dataset dengan hasil *feature scoring* pada fitur ke-4 adalah berkorelasi sedang.

| Nilai Korelasi | Tingkatan Korelasi |
|----------------|--------------------|
| 0,58           | Sedang             |

Tabel 3- 10 Hasil Korelasi Fitur ke-4

#### 4. Kesimpulan

Berdasarkan hasil pengujian dan analisis yang telah dilakukan pada bab 4, telah didapat beberapa kesimpulan adalah sebagai berikut:

1. Ekstraksi fitur dengan menggunakan metode *type dependency parser* hanya menghasilkan fitur terekstrak yang sesuai dengan *rules*, ini menurunkan tingkat kesalahan dalam identifikasi fitur.
2. Hasil penentuan polaritas dengan menggunakan SO-PMI menghasilkan nilai yang berbeda-beda setiap waktunya karena memiliki proses update yang bergantung dengan penggunaan fitur dan opini dalam dunia maya.
3. Berdasarkan hasil SO-PMI dapat dilihat bahwa pada persebaran fitur pada *dataset* Nook Tablet 16Gb memiliki jumlah fitur positif dan negatif terbanyak dengan jumlah 123 fitur positif dan 128 fitur negatif sedangkan pada *dataset* Cannon Mini Hdmi memiliki jumlah fitur positif dan negatif terkecil yaitu dengan jumlah 70 fitur positif dan 105 fitur negatif.
4. Berdasarkan hasil *feature scoring* dapat dilihat bahwa pada *dataset* Cannon Mini Hdmi memiliki jumlah fitur positif terbanyak yaitu 151 fitur positif dengan persebaran 18 fitur bernilai 6, 96 fitur bernilai 5, 37 fitur bernilai 4 sedangkan *dataset* Nook Tablet 16Gb memiliki jumlah fitur negatif tertinggi yaitu 84 fitur negatif dengan persebaran yaitu 36 fitur bernilai 2 dan 48 fitur bernilai 1. Kemudian *dataset* yang memiliki fitur positif terendah adalah Nook Ereader dengan jumlah 121 fitur positif dengan persebaran 1 fitur bernilai 7, 16 fitur bernilai 6, 45 fitur bernilai 5, 59 fitur bernilai 4 sedangkan *dataset* yang memiliki jumlah fitur negatif terendah adalah *dataset* Cannon Mini Hdmi dengan jumlah 15 fitur negatif dengan persebaran 7 fitur bernilai 2, 8 fitur bernilai 1.
5. Berdasarkan hasil *pearson correlation* fitur yang memiliki *trend* terbaik pada *dataset* Cannon Mini Hdmi adalah fitur ke-18 yaitu *Length* dengan nilai korelasi 0.99 atau berkorelasi sangat kuat dan fitur yang memiliki *trend* terburuk adalah fitur ke-79 yaitu *Quality* dengan nilai korelasi 0.34 atau berkorelasi lemah. Pada *dataset* Nook Ereader yang memiliki *trend* terbaik adalah fitur ke-129 yaitu *Life* dengan nilai korelasi 0.99 atau berkorelasi sangat kuat dan fitur yang memiliki *trend* terburuk adalah fitur ke-37 yaitu *Reason* dengan nilai korelasi 0.00 atau tidak memiliki korelasi. Pada *dataset* Nook Tablet 16Gb yang memiliki *trend* terbaik adalah fitur ke-46 yaitu *Seller* dengan nilai korelasi 1.00 atau berkorelasi sangat kuat dan fitur yang memiliki *trend* terburuk adalah fitur ke-62 yaitu *Unit* dengan nilai korelasi -0.94 atau berkorelasi terbalik sangat kuat. Pada *dataset* Usb External yang memiliki *trend* terbaik adalah fitur ke-18 yaitu *Order* dengan nilai korelasi 1.00 atau berkorelasi sangat kuat dan fitur yang memiliki *trend* terburuk adalah fitur ke-13 yaitu *Netbook* dengan nilai korelasi 0.72 atau berkorelasi kuat.

#### 5. Saran

Saran yang diperlukan dari penelitian tugas akhir ini adalah untuk pembangunan penelitian selanjutnya adalah dengan menambahkan sebagai berikut:

1. Penambahan *rule* lain pada ekstraksi fitur dan opini untuk memperbanyak fitur lain yang terekstrak.
2. Melakukan pengujian SO-PMI dengan jumlah yang cukup banyak agar didapatkan mendapatkan nilai terbaik. Melakukan pengujian SO-PMI dengan API dari google yang tersedia pada bahasa pemrograman python .
3. Penggantian dataset yang memiliki *expert judgement* agar lebih mudah dalam pengujiannya.

#### 6. Daftar Pustaka

- |     |   |
|-----|---|
| [1] | Yang, J. Y., Kim, H. J., & Lee, S. G. (2010). Feature-based Product Review Summarization Utilizing User Score. <i>J. Inf. Sci. Eng.</i> , 26(6), 1973-1990. |
| [2] | Han, J., Pei, J., & Kamber, M. (2011). <i>Data mining: concepts and techniques</i> . Elsevier.  |
| [3] | Karl, A., Rushing, H. (2013), <i>Text Mining in JMP with R</i> .  |



- [4] Weiss, S. M., Indurkha, N., & Zhang, T. (2010). *Fundamentals of predictive text mining* (Vol. 41). London: Springer.
- [5] Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1), 1-167.
- [6] Blanchard, A. (2007). Understanding and customizing stopword lists for enhanced patent mapping. *World Patent Information*, 29(4), 308-316.
- [7] Dave, R., Balani, P. (2015). Survey paper of Different Lemmatization Approaches. *International Journal of Research in Advent Technology*, p. 366
- [8] Stanford Log-linear Part-Of-Speech Tagger," The Stanford Natural Language Preprocessing Group, [Online]. Available: <http://nlp.stanford.edu/software/tagger.shtml>. [Diakses 20 12 2016]
- [9] Santorini, B. (1990). Part-of-speech tagging guidelines for the Penn Treebank Project (3rd revision).
- [10] Covington, M. A. (2001). A fundamental algorithm for dependency parsing. In *Proceedings of the 39th annual ACM southeast conference* (pp. 95-102).
- [11] De Marneffe, M. C., & Manning, C. D. (2008). *Stanford typed dependencies manual* (pp. 338-345). Technical report, Stanford University.
- [12] Turney, P. D. (2002, July). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 417-424). Association for Computational Linguistics.
- [13] S. "Pearson's correlation," Creative Commons Licence , [Online]. Available: <http://www.statstutor.ac.uk/>. [Diakses 20 7 2017]
- [14] Zhuang, L., Jing, F., & Zhu, X. Y. (2006, November). Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management* (pp. 43-50). ACM.
- [15] Yang, J. Y., Myung, J., & Lee, S. G. (2009, February). The method for a summarization of product reviews using the user's opinion. In *Information, Process, and Knowledge Management, 2009. eKNOW'09. International Conference on* (pp. 84-89). IEEE.
- [16] Yang, J. Y., Myung, J., & Lee, S. G. (2009, March). A holistic approach to product review summarization. In *Future Dependable Distributed Systems, 2009 Software Technologies for* (pp. 150-154). IEEE.