

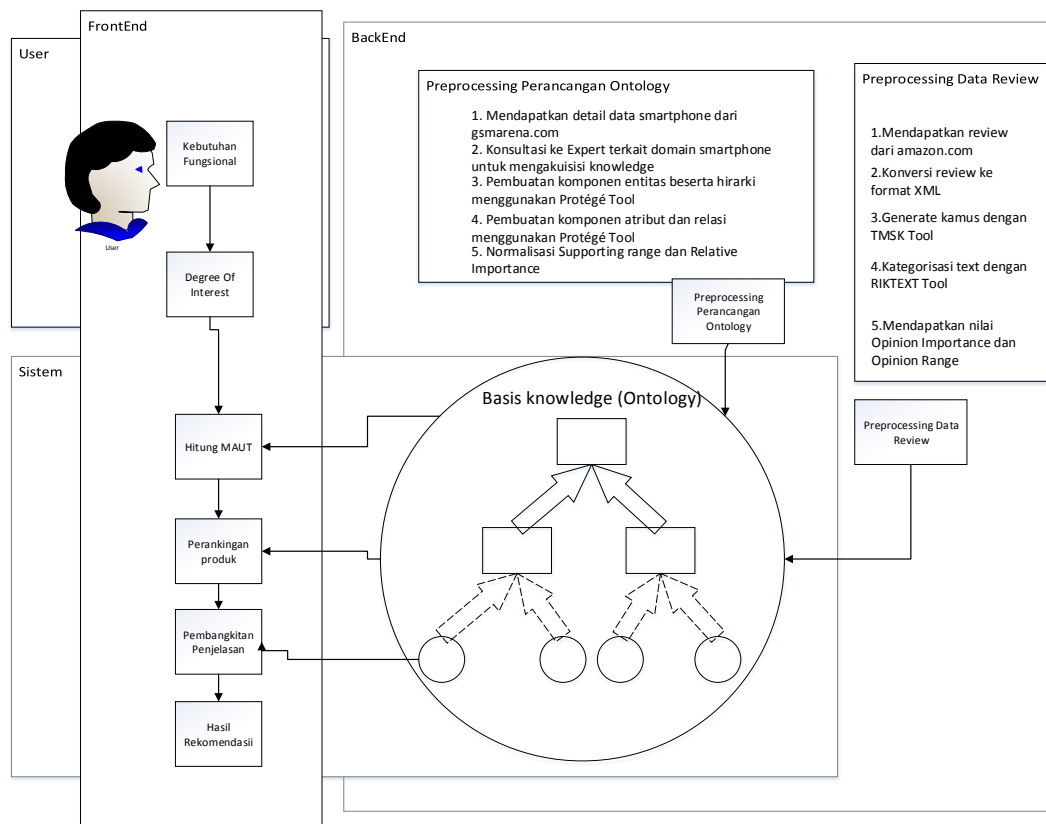
BAB III

Analisis Kebutuhan & Perancangan Sistem

3.1 Gambaran Umum Sistem

Sistem yang akan dibuat berupa sebuah aplikasi berbasis web dimana inputan dari system ini merupakan preferensi pengguna dan outputan berupa rekomendasi dari *smartphone* dan item beserta detail spesifikasi dari produk.

Pada penelitian ini, sistem dibangun dengan menggunakan bahasa pemrograman Java Standard Enterprise Edition. Berikut Gambaran umum sistem.



Gambar 1 .Gambaran Umum Sistem

Adapun penjelasan flowchart pada gambar 5 diatas adalah sebagai berikut :

BackEnd :

1. Preprocessing Data Review

a. Mendapatkan *review* dari amazon.com

Dataset *review* diambil dari amazon.com tiap produk dari setiap pengguna yang memberikan *review*.

b. Konversi *review* ke dalam format XML.

review dikonversi berdasarkan setiap *sentence* dalam satu dokumen tiap produk dari setiap pengguna.

- c. Generate kamus dengan TMSK Tool.

Kamus digenerate berdasarkan inputan *review* berformat XML yang nantinya akan berupa *sparse vector* yang berkorelasi dengan inputan.

- d. Kategorisasi text dengan RIKTEXT Tool.

Hasil outputan TMSK Tool yang berupa kamus *sparse vector* dibagi dalam data training dan testing untuk inputan RIKTEXT Tool. Outputan RIKTEXT Tool yaitu kata yang sudah terkategori ke dalam kategori *good* dan *bad*.

- e. Mendapatkan nilai *Opinion Relative Importance* dan *Opinion Range*.

Hasil dari outputan RIKTEXT Tool berupa kumpulan kata yang sudah terkategori *good* dan *bad*. Kumpulan kata tadi akan dicocokkan dengan kamus yang bersesuaian dengan kata-kata dalam komponen entitas *Ontology* yaitu hirarki *specification*. Nilai *Opinion Importance* didapat dari nilai proporsi. Untuk nilai proporsi *good* didapat dari seberapa besar banyak kata *good* dibanding keseluruhan kata. Sebaliknya nilai proporsi *bad* didapat dari seberapa besar banyak kata *bad* dibanding keseluruhan kata.

2. Preprocessing Perancangan Ontology

- a. Mendapatkan detail data smartphone dari gsmarena.com.

Detail data *smartphone* akan digunakan dalam *Ontology* yang dikonversi ke dalam format owl.

- b. Konsultasi kepada Expert terkait domain smartphone untuk mendapatkan akuisisi *knowledge*.

Konsultasi kepada *expert* dilakukan untuk memastikan bahwa *break down* domain atau pembagian tingkatan High, Medium dan Low terhadap *specification* dari produk adalah sesuai.

- c. Pembuatan komponen entitas berupa hirarkinya menggunakan Protégé Tool.

Komponen entitas memiliki 3 hirarki yaitu *Specification*, *Functional Requirement* dan *Product*. Ketiga hirarki akan diimplementasikan menggunakan Protégé Tool dan menghubungkan relasi beserta *data property*-nya

- d. Pembuatan komponen atribut beserta komponen relasinya menggunakan Protégé Tool.

Komponen atribut merupakan data property dari sebuah entitas seperti data property *Relative Importance* yang berelasi dengan komponen entitas *Specification*.

- e. Normalisasi nilai *Supporting Range* dan *Relative Importance*.

Supporting Range dinormalisasi agar mendapatkan nilai support yang rata dalam pembagian tingkatan sebuah *specification*.

3. Kebutuhan Fungsional

Query dari Ontology untuk filterisasi harga, brand, os beserta tipe produk sesuai preferensi pengguna menggunakan SparQL Query. Hasil query bisa berupa produk yang mempunyai relasi dengan preferensi pengguna.

4. Hitung DOI

Nilai *Degree Of Interest (DOI)* didapat dari kebutuhan fungsional pengguna terhadap produk yang diinginkan. Nilai *Functional requirement* ini memiliki nilai range 1 sampai 100 dan akan dinormalisasi berdasarkan menjadi range 0 sampai 1 serta dibagi berdasarkan jumlah *functional requirement* pengguna.

5. Hitung MAUT

Perhitungan MAUT didapat dari hasil query nilai *supporting range*, *relative importance*, *opinion range*, *relative importance* dari Ontology. Kemudian dikalikan dengan DOI sesuai pada rumus MAUT pada kajian pustaka diatas. Perangkingan produk merupakan hasil dari nilai MAUT dan diambil 5 nilai teratas untuk ditampilkan ke pengguna.

6. Pembangkitan Penjelasan.

Pembangkitan penjelasan merupakan hasil *backtrack* dari penelusuran pengguna. Penelusuran tersebut mempunyai node dan relasi yang saling terkait terhadap produk. Sehingga pengguna dapat lebih jelas terkait deskripsi produk hasil sistem rekomendasi.

3.2 Pengumpulan Data

3.2.1 Dataset Smartphone

Pada penelitian ini, domain yang akan diuji adalah domain *smartphone*. Jenis data yang akan diambil adalah data spesifikasi *smartphone* yang diambil dari <http://www.gsmarena.com>, <http://lazada.co.id>, dan beberapa sumber lainnya. Data yang terkumpul adalah 257 *smartphone*.

berikut gambar 6 adalah contoh data berupa spesifikasi sebagai berikut:

```

<owl:NamedIndividual rdf:about="http://www.risetgroup.org/ontologies/recommendersystem/smartphone#Sony_Xperia_F4">
  <rdf:type rdf:resource="http://www.risetgroup.org/ontologies/recommendersystem/smartphone#Middle_Low_End_Handphone"/>
  <hasRam rdf:datatype="xsd:string">1</hasRam>
  <hasProcessor rdf:datatype="xsd:string">1.3</hasProcessor>
  <hasVideo_Record_Quality rdf:datatype="xsd:string">1080</hasVideo_Record_Quality>
  <hasPrice rdf:datatype="xsd:long">2000000</hasPrice>
  <hasTotalStorage rdf:datatype="xsd:string">40</hasTotalStorage>
  <hasScreen_Size rdf:datatype="xsd:string">5</hasScreen_Size>
  <hasPrimary_Camera rdf:datatype="xsd:string">5</hasPrimary_Camera>
  <hasBattery_Endurance rdf:datatype="xsd:string">56</hasBattery_Endurance>
  <hasMemori_Internal rdf:datatype="xsd:string">8</hasMemori_Internal>
  <hasOS rdf:datatype="xsd:string">Android</hasOS>
  <hasCHIPSET rdf:datatype="xsd:string">Mediatek MT6582</hasCHIPSET>
  <hasBrand rdf:datatype="xsd:string">Sony</hasBrand>
  <hasDetails>* Features Sensors : Accelerometer, proximity</hasDetails>
  <hasDetails>* Connection Data Speed : HSPA 21.1/5.76 Mbps</hasDetails>
  <hasDetails>* Connection WLAN :Wi-Fi 802.11 b/g/n, Wi-Fi Direct, hotspot</hasDetails>
  <hasDetails>* Features GPU : Mali-400MP2</hasDetails>
  <hasDetails>* Battery Music Play : Up to 40 h</hasDetails>
  <hasDetails>* 2G Network : GSM 850/900/1800/1900</hasDetails>
  <hasDetails>* Display Multitouch : Yes, up to 4 fingers</hasDetails>
  <hasDetails>* Memory Card Slot : microSD, up to 32 GB</hasDetails>
  <hasDetails>* Sound 3.5mm Jack : Yes</hasDetails>
  <hasDetails>* Features Colors : Black, White, Purple</hasDetails>
  <hasDetails>* Sound Loudspeaker : Yes</hasDetails>
  <hasDetails>* Battery Talk Time : Up to 12 h (2G) / Up to 12 h 40 min (3G)</hasDetails>
  <hasDetails>* Camera Primary : 5 MP, 2592 x 1944 pixels, autofocus, LED flash</hasDetails>
  <hasDetails>* Battery Type : Non-removable Li-Ion 2300 mAh battery</hasDetails>
  <hasDetails>* Sound Alert Types : Vibration; MP3 ringtones</hasDetails>
  <hasDetails>* Connection USB : Yes, microUSB v2.0</hasDetails>

```

Gambar 2 . Contoh data smartphone dalam ontology

Data tersebut akan dimasukkan dalam model *ontology*, kemudian dicocokkan dengan hasil akuisisi oleh seorang *expert* di bidang smartphone.

3.2.1 Dataset Reviews

Dataset *reviews* diambil dari website amazon.com. dataset *reviews* diambil beberapa produk untuk sampel. Sampel yang diambil 5 produk *smartphone* dari setiap Brand dan Satu produk mempunyai sekitar 1000 line text dalam format XML.

berikut contoh data review *smartphone* Samsung Galaxy S5 adalah sebagai berikut:

```

<DOC>
<TITLE> A Samsung Evolution</TITLE>
<AUTHORS>Johnny Saigon</AUTHORS>
<BODY>
First off, I am not a professional reviewer, nor am I employed or compensated by Samsung or any other company.
</BODY>
<TOPICS>good</TOPICS>
</DOC>
<DOC>
<TITLE> A Samsung Evolution</TITLE>
<AUTHORS>Johnny Saigon</AUTHORS>
<BODY>
Instead of boring you with facts - which you can find anywhere on the Net - I will just give you some real-world impressions on how it
looks, feels, and runs.
</BODY>
<TOPICS>good</TOPICS>
</DOC>
<DOC>
<TITLE> A Samsung Evolution</TITLE>
<AUTHORS>Johnny Saigon</AUTHORS>
<BODY>
With that out of the way, let's get to the point and the nitty gritty, shall we?
</BODY>
<TOPICS>good</TOPICS>
</DOC>
<DOC>
<TITLE> A Samsung Evolution</TITLE>
<AUTHORS>Johnny Saigon</AUTHORS>
<BODY>
* THE SCREEN - that is the very first thing you will notice when you look at the S5.

```

Gambar 3 . Contoh data review dalam XML

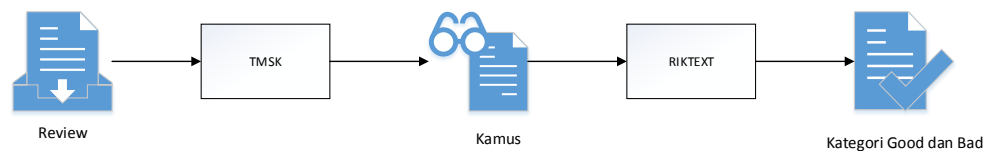
Contoh data diatas sudah dalam format XML, dipenggal satu kalimat dari setiap paragraf *review*. Kemudian setiap kalimat harus dibagi manual mana yang termasuk kalimat yang *good* atau *bad* . Selanjutnya data tersebut akan diolah pada *text mining toolkit*.

3.3 Pengolahan Data

3.3.1 Pengolahan Data Reviews

Pengolahan data *review* menggunakan Text Miner Software Kit (TMSK) dan The Rule Induction Kit for Text (RIKTEXT).

Pada gambar 8 dibawah ini, proses pengolahan data *review* dapat diilustrasikan sebagai berikut:



Gambar 4 Alur Tool TMSK dan RIKTEXT

Berdasarkan gambar 8 diatas, langkah pertama menggunakan TMSK, data input dari TMSK adalah data *review smartphone* berupa dokumen berbasis .XML. Sedangkan output yang dihasilkan adalah kamus berupa vektor matriks hasil indeks dari dokumen input berbasis .XML.

Selanjutnya menggunakan RIKTEXT, data input dari RIKTEXT berupa dokumen vektor matriks yang harus dibagi dua dokumen. Dokumen tersebut dibagi manual untuk data *training* dan data *testing*. Sedangkan output yang dihasilkan berupa *rule* kategori *good* dan *bad*.

Berikut pada gambar 9 contoh hasil ekstraksi fitur yang dihasilkan:

```

(*) = minimum error; ** = within 1-SE of minimum error>
RSet   Rules   Vars   Train Err   Test Err   Test SD   MeanVar   Err/Var
1**    44      48      0.2161     0.0000     0.0000     0.0       0.00
Training cases only

*****
Selected rule set

1. s3 --> GOOD
2. ad --> GOOD
3. card --> GOOD
4. while --> GOOD
5. amount --> GOOD
6. thing --> GOOD
7. cell --> GOOD
8. learn --> GOOD
9. rear --> GOOD
10. tmobile --> GOOD
11. install --> GOOD
12. place --> GOOD
13. additional --> GOOD
14. smartphone --> GOOD
15. configure --> GOOD
16. highly --> GOOD
17. performance --> GOOD
18. fast --> GOOD
19. iphone --> GOOD
20. live --> GOOD
21. interest --> GOOD
22. s4 & 16GB --> GOOD
23. image --> GOOD
24. dim --> GOOD
25. size --> GOOD
26. spee --> GOOD
27. android --> GOOD
28. resolution --> GOOD
29. pack --> GOOD
30. case --> GOOD
31. extend --> GOOD
32. glance --> GOOD
33. consumer --> GOOD
34. require --> GOOD
35. useful --> GOOD
36. complaint --> GOOD
37. consideration --> GOOD
38. around --> GOOD
39. heavy --> GOOD
40. communication --> GOOD
41. s4 & screen --> GOOD
42. device & s4 --> GOOD
43. everyth & more --> GOOD
44. [TRUE] --> ~GOOD

Additional Statistics <Training Cases>:
precision: 100.0000    recall: 92.2414    f-measure: 95.9641

```

Gambar 5 . Output feature text RIKTEXT

Selanjutnya RIKTEXT akan menghasilkan data seperti gambar diatas. Dimana kata - kata yang cocok terkategoriisasi langsung ke *good* atau *bad*. Hasil datanya juga sudah mempunyai akurasi berupa *precision*, *recall* dan *m-measure*.

3.4 Perancangan Ontology

Terdapat empat komponen pada suatu model ontologi, yaitu entitas, atribut, hubungan, dan aksioma. Namun pada penelitian ini yang diambil hanya tiga komponen, yaitu entitas (kelas atau individu), atribut (*data property*), dan hubungan (relasi).

3.4.1 Komponen Entitas (Kelas Hirarki)

Pada komponen entitas akan dibentuk tiga kelas hirarki, yaitu kelas spesifikasi, kelas kebutuhan fungsional (*functional requirement*), dan kelas produk.

3.4.1.1 Hirarki Specifications

Pada kelas ini, spesifikasi yang dimaksud adalah komponen suatu produk. Kedalaman kelas hirarki spesifikasi tidak selalu tetap. Kedalaman tergantung pada klasifikasi dari komponen domain suatu produk. Pada penelitian kali ini kedalaman kelas hirarki spesifikasi adalah empat level.

Jumlah individu pada kelas ini akan bergantung pada tingkatan suatu *subclass* suatu komponen. Misal, suatu spesifikasi Kamera memiliki beberapa ukuran pixel diantaranya 2Mpix, 5Mpix, dan 8Mpix. Dari beberapa ukuran pixel tersebut dapat kita klasifikasikan ke dalam suatu tingkatan menjadi *Low* Kamera (2Mpix), *Medium* Kamera (5Mpix), dan *High* Kamera (8Mpix). Dari tingkatan tersebut dapat kita bentuk bahwa jumlah individu dari *subclass* Kamera adalah tiga, yaitu *Low* Kamera, *Medium* Kamera, dan *High* Kamera. Pengklasifikasian didapat dari *expert* untuk domain *smartphone*. Klasifikasi lengkap setiap komponen dapat dilihat pada bagian lampiran buku laporan ini.

Setiap individu pada kelas ini memuat *data property* berupa nilai *supporting range*. Dapat diketahui bahwa nilai *supporting range* didapat dari *support level* suatu komponen yang memenuhi $\sum S(ci) = 1 \text{ } i=1 \dots n$. Misal, untuk mendapatkan nilai *supporting range* dari masing-masing individu pada *subclass* spesifikasi Kamera, seperti yang sudah disebutkan sebelumnya, maka teknik normalisasi dapat dilakukan. Contoh Normalisasi :

$$\text{Supporting Raange (High Kamera)} = 3-1+1/(1+2+3) = 0.5$$

$$\text{Supporting Raange (High Kamera)} = 2-1+1/(1+2+3) = 0.333$$

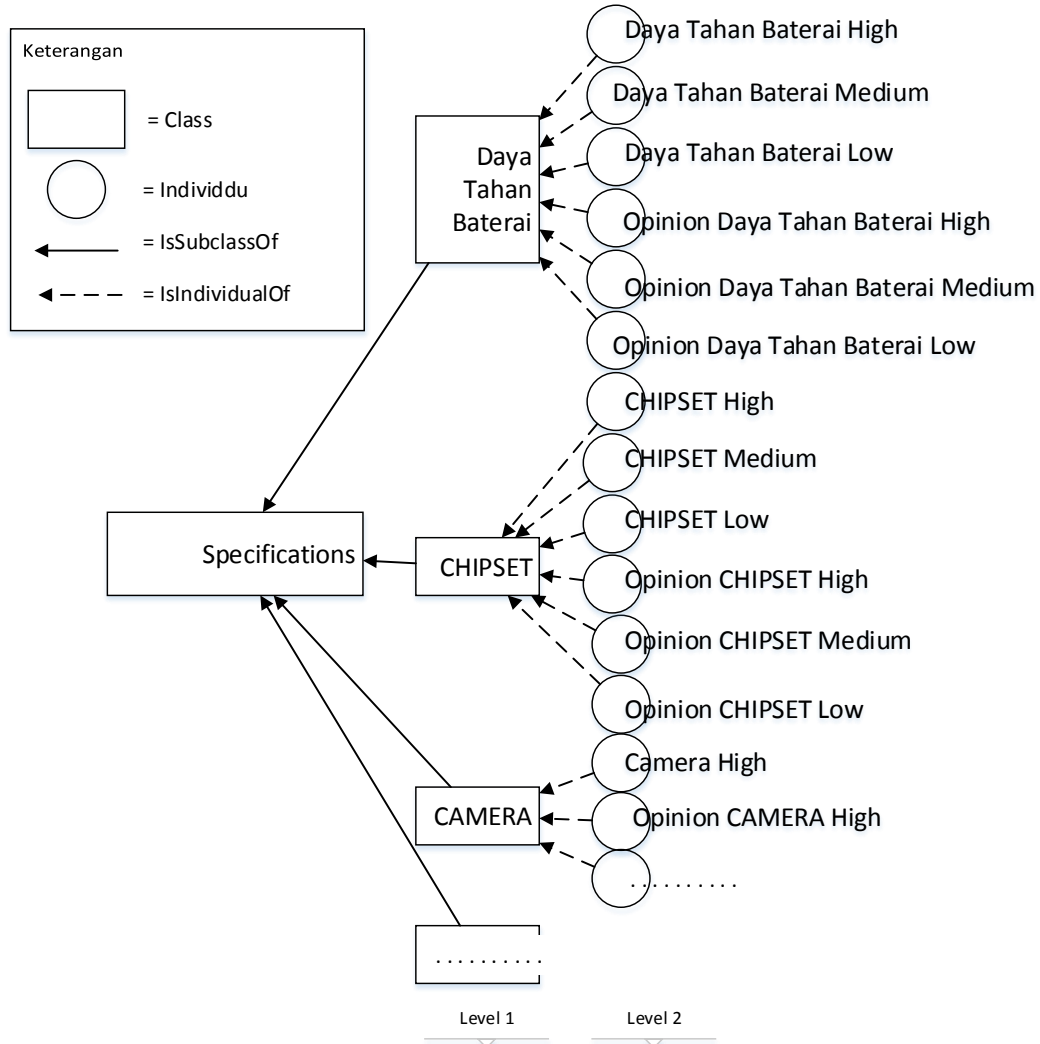
$$\text{Supporting Raange (High Kamera)} = 1-1+1/(1+2+3) = 0.167$$

Setiap individu pada kelas ini memuat *data property* berupa nilai *opinion range*. Dapat diketahui bahwa nilai *opinion range* didapat dari nilai *proporsi* hasil ekstraksi *feature* suatu spesifikasi dari sebuah produk. Misalnya hasil dari ekstraksi *feature* pada produk A yang terkategori *good* adalah “camera”, “pixel”, “flash” dan “battery”. Kata tersebut akan dicari pada kamus *specification camera* dan kamus *specification battery*. Jika kata “camera”, “pixel”, “flash” ada dalam kamus *specification battery* maka akan dihitung tiga kata yang terkategori *good*. Sebaliknya apabila pada produk A yang terkategori *bad* adalah “charge”, maka akan dihitung satu kata yang terkategori *bad*. Menurut persamaan nomor (2), nilai proporsi keseluruhan dari *specification battery* = $3/4 = 0.75$. Nilai tersebut setelah dinormalisasikan menyesuaikan normalisasi dari nilai *supporting range* didapat kategori *opinion range* adalah *high opinion battery*. Berikut normalisasi dari *opinion range* dibagi menjadi 3 level yaitu High, Medium, dan Low :

- a. Jika nilai proporsi kurang dari 0.333 maka level *Opinion Range Low*

- b. Jika nilai proporsi berkisar diantara $0.333 \leq x \leq 0.67$ maka level *Opinion Range Medium*.
- c. Jika nilai proporsi lebih dari 0.67 maka level *Opinion Range High*.

Pada gambar 10 dibawah ini diilustrasikan hirarki kelas *specifications* dalam *ontology*:



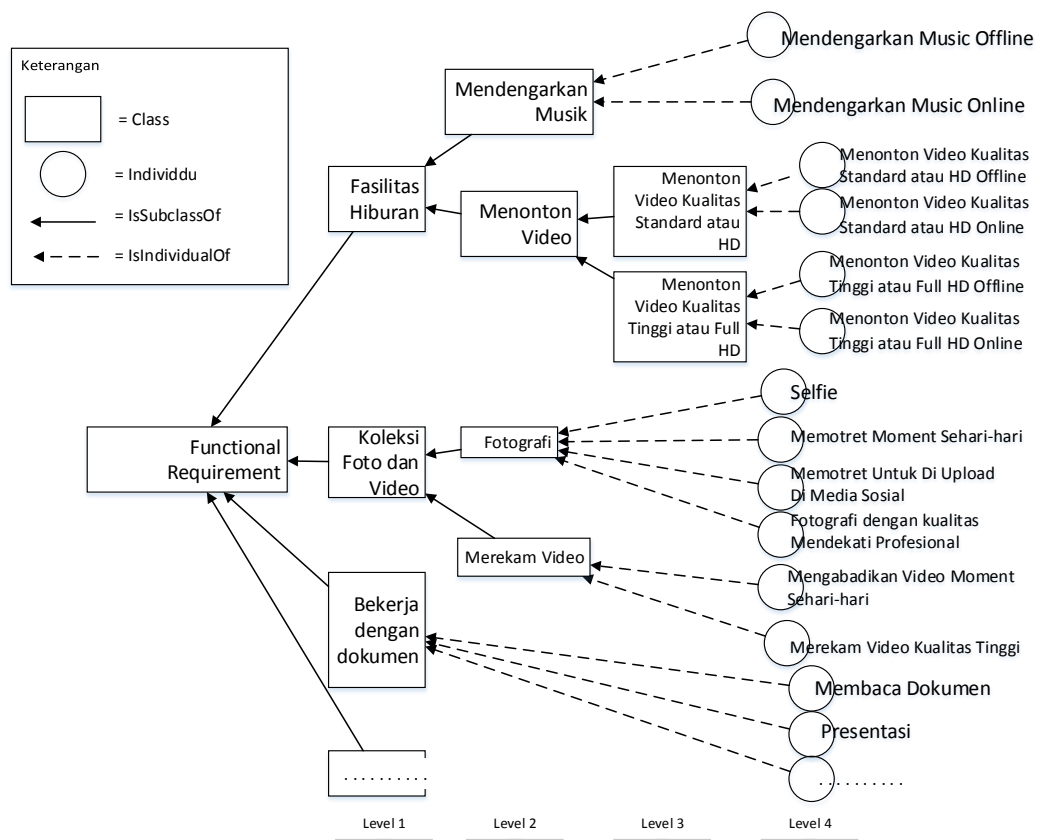
Gambar 6 . Hirarki kelas spesifikasi dalam *ontology*

3.4.1.2 Hirarki Functional Requirement

Pada kelas ini, kebutuhan fungsional untuk suatu domain produk akan dibentuk beserta komponennya menjadi sebuah hirarki. Kedalaman level pada kelas hirarki *functional requirement* tidak selalu tetap. Kedalaman tergantung terhadap lingkup pengetahuan domain suatu produk.

Untuk setiap individu pada kelas ini memiliki *data property* berupa nilai *Specification Relative Importance* dan *Specification Opinion Relative Importance*. Nilai *Specification Relative Importance* ini berfungsi untuk menyimpan nilai kedekatan atau kepentingan suatu spesifikasi terhadap suatu kebutuhan fungsional. Sementara nilai *Specification Opinion*

Relative Importance kedekatan nilai proporsi spesifikasi terhadap suatu kebutuhan fungsional. Misal, untuk kebutuhan fungsional *HD Online Gaming* dibutuhkan spesifikasi *Chipset*, dan *network speed*. Kedua komponen tersebut akan memiliki nilai kepentingan terhadap kebutuhan fungsional *HD Online Gaming* yang berbeda-beda. Nilai kepentingan masing-masing spesifikasi atau *Spesification Relative Importance* terhadap suatu kebutuhan fungsional didapat dari *expert* untuk domain produk *smartphone* dan nilai kepentingan dari proporsi opini terhadap spesifikasi atau *Spesification Relative Importance* yang dapat dilihat pada bagian lampiran buku laporan ini. Contoh struktur *ontology* untuk kelas hirarki *functional requirement* dapat dilihat pada Gambar 11 dibawah ini.

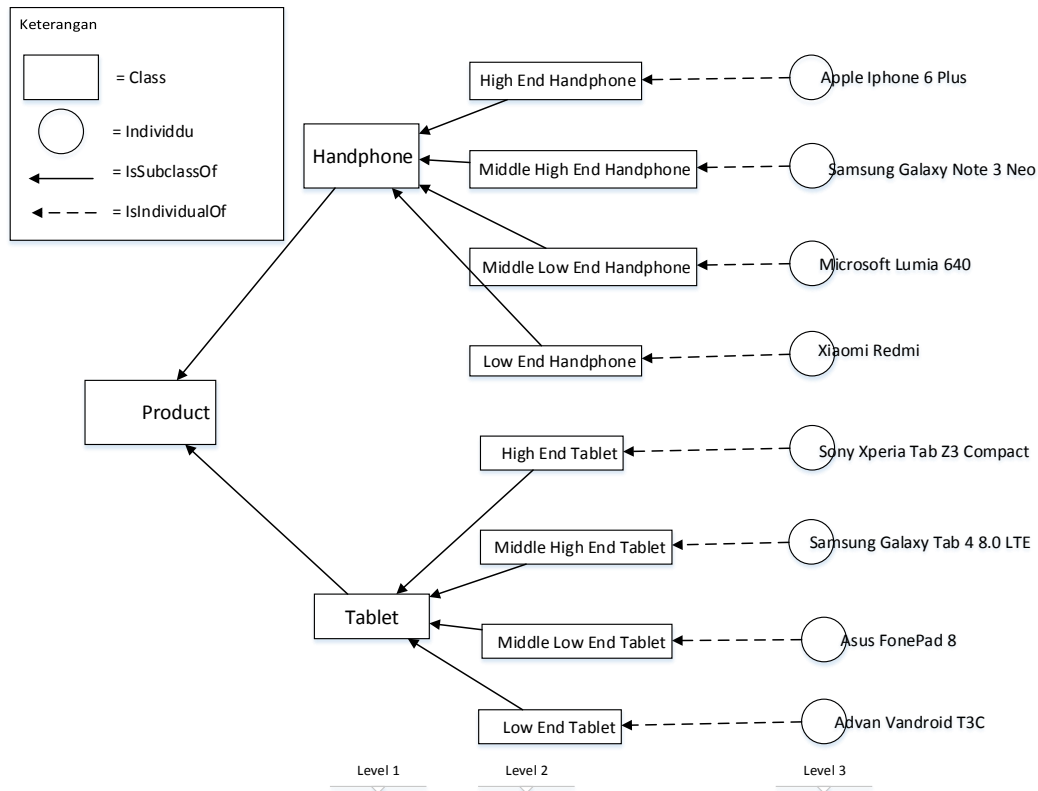


Gambar 7 . Hirarki kelas *functional requirement* dalam *ontology*

3.4.1.3 Hirarki Product

Pada kelas ini, produk akan diklasifikasikan berdasarkan tipe dan segmennya. Untuk level 1 produk akan diklasifikasikan berdasarkan tipenya dan level 2 produk diklasifikasikan berdasarkan segmennya. Individu terdapat pada level 3 dari kelas hirarki ini. Untuk setiap individu dari kelas hirarki produk memiliki *data property* berupa *spec details*, *price*, dan *brand*.

Contoh struktur *ontology* untuk kelas hirarki produk dapat dilihat pada Gambar 12 dibawah ini. Untuk struktur lengkapnya terdapat pada bagian lampiran buku laporan ini.



Gambar 8 . Hirarki kelas product dalam *ontology*

3.4.2 Komponen Atribut (Data Property)

Seperti yang sudah dijelaskan pada subbab sebelumnya, untuk komponen atribut dari ontologi pada penelitian ini adalah *data property* untuk setiap individu pada setiap kelas hirarki. Untuk jenis data property setiap individu pada setiap kelas hirarki bisa dilihat pada Tabel 2 dibawah ini.

Table 1 . Kelas dan data properti *ontology*

Kelas	Data Properti
Kelas <i>Functional Requirement</i>	<ul style="list-style-type: none"> - <i>Specification Relative Importance</i> - <i>Specification Opinion Relative Importance</i>
Kelas <i>Spesifications</i>	<ul style="list-style-type: none"> - <i>Supporting Range</i> - <i>Opinion Range</i>
Kelas Produk	<ul style="list-style-type: none"> - <i>Spec Details</i> - <i>Price</i> - <i>Brand</i>

3.4.3 Komponen Hubungan (Relasi)

Komponen relasi dari ontologi pada penelitian ini terdapat empat jenis, yaitu *suppBy*, *supportOf*, *hasSpec*, dan *isSpecOf*. Keempat komponen tersebut berfungsi untuk menghubungkan satu individu pada suatu kelas hirarki kepada individu lainnya pada suatu kelas hirarki. Misal, Sony Xperia E4 Dual merupakan individu dari kelas hirarki produk dan *Medium Chipset* merupakan individu dari kelas hirarki spesifikasi. Untuk menghubungkan individu Sony Xperia E4 Dual dengan individu *Medium Chipset*, diperlukan sebuah komponen relasi *hasSpec* Sehingga akan terbentuk *knowledge* dalam ontologi bahwa individu Sony Xperia E4 Dual memiliki relasi *hasSpec* dengan individu *Medium Chipset*.

Pada Tabel dibawah ini, dijabarkan domain (kelas hirarki asal) dan *range* (kelas hirarki tujuan) untuk masing-masing komponen relasi dari *ontology* pada penelitian ini.

Tabel 2 . Domain relasi dalam *ontology*

Relasi	Domain	Range	Keterangan
<i>suppBy</i>	Functional Requirement	Spesifikasi	Relasi dari individu pada kelas hirarki <i>functional requirement</i> terhadap individu pada kelas hirarki spesifikasi.
<i>supportOf</i>	Spesifikasi	Functional Requirement	Kebalikan (<i>inverse</i>) dari komponen relasi <i>suppBy</i> .
<i>hasSpec</i>	Produk	Spesifikasi	Relasi dari individu pada kelas hirarki produk terhadap individu pada kelas hirarki spesifikasi.
<i>isSpecOf</i>	Spesifikasi	Produk	Kebalikan (<i>inverse</i>) dari komponen relasi <i>hasSpec</i> .

3.5 Rekomendasi Produk

Pencarian produk dapat difilter dari *hard constraint* melalui harga, brand, tipe produk dan sistem operasi. Sementara perankingan produk dapat menggunakan nilai *utility* dari *soft constraint* ke setiap produk. Proses pencarian produk dan perhitungan nilai *utility* produk dilakukan melalui *semantic reasoning*, yaitu menelusuri node – node model *ontology*.

1.5.1 Filter Product by Hard Constraint

Dalam penelitian kali ini, *hard constraint* yang digunakan adalah tipe produk dan properti produk yaitu harga, *brand* dan sistem operasi. Sistem melakukan pencarian produk terlebih dahulu berdasarkan properti produk yang diinginkan user. Pada proses ini terdapat dua skenario, yaitu skenario pertama user harus memberikan preferensinya sesuai preferensi produk yang disediakan yaitu tipe produk, harga, *brand* dan sistem operasi.

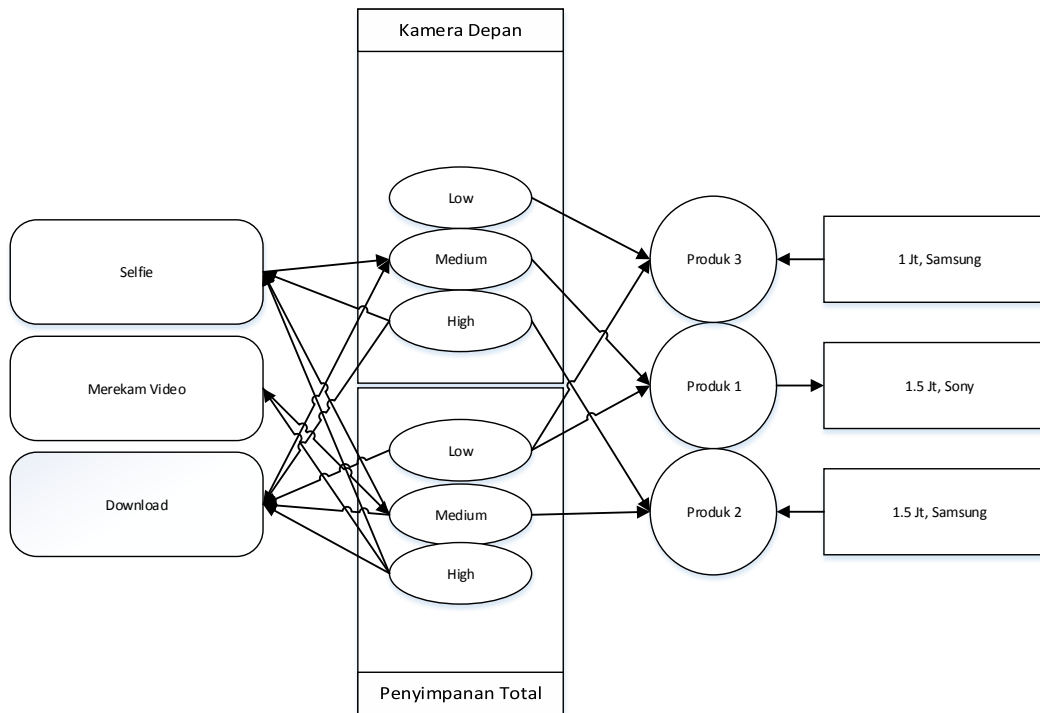
Dalam penyaringan produk berdasarkan *hard constraint* terdapat beberapa kondisi yaitu :

1. Harga diset maksimal yang berarti penyaringan akan menelusuri seluruh produk dibawah harga yang diberikan *user*.
2. Ketika *user* memilih brand dan sistem operasi yang tidak cocok , misalnya memilih brand Apple tapi dengan sistem operasi android. Hal tersebut berarti menandakan bahwa sistem akan merekomendasikan seluruh brand apple dan seluruh sistem android.
3. Semua *hard constraint* tidak boleh dikosongkan.

Skenario kedua, sistem akan menyaring produk berdasarkan *utility threshold* yang akan dibahas pada subbab selanjutnya.

3.5.2 Hitung Product Utility by Soft Constraint

Perhitungan nilai *utility* produk berdasarkan *soft constraint* pada penelitian kali ini hanya kebutuhan fungsional. Untuk menghitung nilai *utility* produk terhadap *soft constraint*, persamaan (3) akan digunakan .



Gambar 13 . Contoh relasi produk, kebutuhan fungsional dan spesifikasi

Misalnya, dari gambar 13 terdapat himpunan *soft constraint functional requirement* F1 berupa selfie , F2 berupa merekam video, dan F3 download yang dipilih oleh *user*. Himpunan produk yang memenuhi *soft constraint* melalui persamaan (1) adalah sebagai berikut :

$$\{P2, P3\}_{ind} \rightarrow F1 \text{ (Selfie)}$$

$$\{P3\}_{ind} \rightarrow F2 \text{ (Merekam Video)}$$

$$\{P1, P2, P3\}_{ind} \rightarrow F3 \text{ (Download)}$$

Spesifikasi relative importance, spesifikasi opinion relative importance, opinion range beserta *supporting range* untuk setiap *functional requirement* seperti pada Tabel dibawah ini.

Tabel 3 . Ilustrasi himpunan *functional requirement* dengan SR, SRI,OR dan OSRI.

Functional Requirment	Spesifikasi Relative Importance	Supporting Range	Spesifikasi Opinion Relative Importance	Opinion Range
F1	Kamera Depan = 0.5 Penyimpanan Total = 0.5	Medium Kamera Depan = 0.333 High Penyimpanan Total = 0.5	Opinion Kamera Depan = 0.5 Opinion Penyimpanan Total = 0.5	Opinion Medium Kamera Depan = 0.4

				Opinion High Penyimpanan Total = 0.61
F2	Kamera Depan = 1.0	Low Kamera Depan = 0.167	Opinion Kamera Depan = 1.0	Opinion Low Kamera Depan = 0.25
F3	Penyimpanan Total = 1.0	High Penyimpanan Total = 1.0	Opinion Penyimpanan Total = 1.0	Opinion High Penyimpanan Total = 0.67

Misalnya diketahui DOI F1 (Selfie) = 0.88, DOI F2 (Merekam Video) = 0.43, dan DOI F3 (Download) = 0.65 dari DOI masing – masing *functional requirment* selanjutnya dihitung rata-rata menggunakan persamaan (5), setelah itu hitung nilai *utility* setiap produk dengan MAUT melalui persamaan (6). Perhitungan dapat dilihat pada lampiran . Berikut Tabel 5 nilai *utility* hasil perhitungan

Table 4 . contoh hasil perhitungan dengan MAUT

Produk	Utility
P1	0.2755
P2	0.4962
P3	0.55032

Hasil perhitungan diatas membuktikan bahwa urutan dari produk yang direkomendasikan adalah P3 ,P2 ,P1. Dengan P3 = 0.55032 , P2 = 0.4962 , P1 = 0.2755.

Untuk skenario kedua pada proses ini, jika *soft constraint* = null, maka nilai *utility* tidak akan dihitung. Sehingga nilai *utility* pada setiap produk memiliki nilai yang sama dan penyaringan produk dilakukan hanya berdasarkan *hard constraint*.