PENGGALIAN PREFERENSI KEBUTUHAN FUNGSIONAL DENGAN *CONVERSATIONAL RECOMMENDER SYSTEM* BERBASIS ONTOLOGI

Romy Adzani A[1], Z.K Abdurahman Baizal [2], Erliansyah Nasution [3]

[1,2,3]Prodi Ilmu Komputasi Telkom University, Bandung
[1]romyadiputra@windowslive.com, [2]bayzal@gmail.com, [3]rlinst@yahoo.com

Abstrak

Perkembangan teknologi yang semakin cepat menuntut pencarian informasi yang semakin mudah. Banyak sekali forum, blog, toko *online*, dan sebagainya yang menyediakan informasi mengenai suatu produk tertentu. Namun masalah yang sering terjadi saat ini adalah cara untuk mendapatkan suatu produk terbaik yang sesuai dengan kriteria pembeli dengan waktu yang singkat.

***Conversational Recommender System* merupakan salah satu solusi yang bisa digunakan. *Recommender system* yang tersebar di dunia maya pada umumnya hanya berbasis filter spesifikasi tanpa menjelaskan kebutuhan yang akan pengguna butuhkan. Dengan adanya fitur *conversation* pada suatu *recommender system*, pengguna akan dipandu layaknya seorang calon pembeli yang sedang dibantu oleh seorang *expert* atau konsultan suatu produk dalam menemukan produk yang cocok dengan kebutuhan calon pembeli.**

Penggalian preferensi pada sistem akan memperluas kebebasan pengguna dalam menentukan preferensi yang diinginkan, sehingga perekomendasian produk akan semakin akurat.

Dengan menyematkan basis atau model ontologi yang berbentuk jaringan semantik, sistem dapat melakukan *learning* terhadap kebutuhan pengguna dan mengkonversinya menjadi sebuah keputusan dalam merekomendasikan suatu produk layaknya seorang manusia dalam memberikan keputusan. Dengan begitu pengguna akan lebih nyaman dan cepat dalam menemukan produk yang sesuai dengan kebutuhannya.

Kata kunci : ***conversational recommender system, knowledge-based filtering, ontology, preference elicitation***

Abstract

Rapid development of technology nowadays demands informations to be gathered faster and easier. There are lots of forums, blogs, online shops that provide informations about specific products out there. However their problems are the same, it's always about how buyers get the best products that meet their needs instantly.

The Conversational Recommender System is one of the solutions that can be used to face that problem. Most of Recommender Systems on the Internet are generally just a filter-by-specification based, which doesn't specified what users really need. By applying conversation feature into a Recommender System, users will be guided just like they were being helped by an expert or consultant to find products that fit their needs best.

Preference elicitation will expand user's desire to decide their true preferences that make product recommendation more accurate.

By integrating an ontology base or model that have semantic web form, the system will do a learning process of user's requirements and convert them to a conclusion to recommends which products fits them best, just like there was a human being giving advice. That way it will be more convenient and faster for the users finding a product that meets their requirements.

Keywords: conversational recommender system, knowledge-based filtering, ontology, preference elicitation

1. Pendahuluan

Saat ini teknologi bukanlah hal asing lagi bagi masyarakat umum. Seseorang dapat membeli atau menjual suatu produk di mana saja, baik *online* maupun *offline*. Namun karena besarnya tuntutan akan kecepatan dalam membeli atau menjual suatu produk, jual beli *online* merupakan cara yang paling banyak dipilih atau biasa disebut *e-commerce*. Dengan adanya *e-commerce* para calon pengguna suatu produk akan dengan mudah melakukan pencarian dan transaksi dari produk yang diinginkan. Tetapi akan menimbulkan masalah bagi calon pengguna yang masih awam terhadap produk yang akan dibelinya, terlebih lagi jika produk tersebut merupakan produk kompleks seperti kendaraan, produk elektronik, hingga kebutuhan sehari-hari seperti pakaian dan sebagainya.

Alat yang bisa digunakan tidak lain adalah *recommender system*. *Recommender system* merupakan alat yang dapat merekomendasikan

produk yang akan dipilih untuk dibeli sesuai dengan keinginan dan kebutuhan penggunanya masing-masing. Dalam teknik merekomendasikan produk, terdapat 3 pendekatan dasar yang bisa dilakukan *recommender system* dengan penggunanya, yaitu *collaborative-based*, *content-based*, dan *knowledge-based* [2]. *Collaborative-based* merupakan teknik pendekatan dengan menyaring pendapat pengguna sebelumnya mengenai suatu barang. *Content-based* merupakan pendekatan dengan cara menyaring fitur-fitur dan spesifikasi mengenai suatu barang. Dan teknik terakhir, yaitu *knowledge based*, merupakan pendekatan dengan mempelajari *habit* atau kebiasaan dari pengguna suatu barang sebelumnya. Sehingga alat ini dapat memberikan pilihan barang berdasarkan "pengalaman".

Dalam penelitian Tugas Akhir yang akan dibahas di sini, *recommender system* yang akan diimplementasikan menggunakan teknik dasar *knowledge-based* dengan cara *conversation* atau tanya jawab yang akan dilengkapi dengan penggalian preferensi. Pada situs *e-commerce* yang bertebaran saat ini, masih banyak yang mengandalkan *recommender system* dengan model *filtering* spesifikasi suatu produk, namun tidak memperhatikan kebutuhan pengguna. Dengan adanya dialog tanya jawab pada *recommender system*, diharapkan dapat membantu pengguna dapat memahami dan menemukan kebutuhan yang diinginkan layaknya berkonsultasi dengan konsultan suatu domain produk dibandingkan dengan model *filtering* spesifikasi produk.

Pertanyaan-pertanyaan yang diajukan akan didapat dari suatu model ontologi. Menurut [1], ontologi merupakan sebuah model yang merepresentasikan jaringan pengetahuan manusia, namun dalam format yang dapat dibaca oleh sebuah mesin (komputer) yang terdiri dari entitas, atribut, relasi, dan aksioma. Sehingga untuk pertanyaan yang merupakan kebutuhan fungsional suatu produk akan diajukan sesuai hirarki yang terbentuk dalam model ontologi.

Kebanyakan pengguna belum mendapat preferensi yang benar-benar mereka inginkan meskipun sudah mendapat rekomendasi awal dari sistem [2]. Preferensi di sini bisa berupa properti produk (seperti harga, merk, dan sebagainya), tipe produk (seperti Tablet PC pada domain komputer atau SUV pada domain mobil), kebutuhan fungsional (seperti *gaming*, *photography*, dan sebagainya), atau pun kombinasi dari ketiga elemen tadi. Dengan adanya penggalian preferensi, sistem akan membangkitkan interaksi tanya jawab hingga beberapa kali untuk menggali preferensi lain yang mungkin akan dibutuhkan oleh pengguna. Bedasarkan penjelasan pada [5], penggalian preferensi akan membagi preferensi seorang pengguna ke dalam dua kategori, yaitu *hard constraint* dan *soft constraint*. *Hard constraint* merupakan preferensi yang diset pengguna dengan prioritas mutlak, sehingga sistem akan memastikan produk-produk yang akan ditampilkan nanti mendukung penuh preferensi tersebut. Sedangkan *soft constraint* merupakan kebalikan dari *hard constraint*, yaitu preferensi dengan prioritas yang tidak mutlak, sehingga sistem akan tetap menampilkan produk-produk meskipun tidak mendukung penuh preferensi pengguna.

Menurut [5], teknik "pembelajaran" sistem terhadap jawaban-jawaban yang diberi pengguna terdapat tiga tahap, yaitu *user profile modelling*, *product recommending*, *question generating*. Pada tahap awal, sistem akan menyimpan preferensi yang sudah diset pengguna dalam suatu model (variabel). Setelah *user profile model* terbentuk, maka tahap selanjutnya akan dibangkitkan. Pada tahap ke-2 (*product recommending*), sistem akan mencari relasi yang terkait antara preferensi-preferensi yang terdapat pada *user profile model* dengan model ontologi yang sudah terbentuk untuk mendapatkan produk rekomendasi. Jika produk rekomendasi masih belum didapat, maka tahap selanjutnya (*question generating*) sistem akan mencoba membangkitkan pertanyaan kembali berdasarkan jawaban sebelumnya yang dipilih pengguna dengan melihat hirarki pada ontologi. Misalnya seorang pengguna sistem memasukkan harga Rp1.500.000,00 sebagai preferensi harga, Tablet PC sebagai preferensi tipe produk, dan *Gaming* sebagai preferensi kebutuhan fungsional. Setelah itu sistem akan menyimpan preferensi tersebut dan mencari produk yang memiliki relasi dengan preferensi pengguna. Jika produk rekomendasi masih belum didapatkan, sistem akan melanjutkan pertanyaan berdasarkan hirarki *Gaming* pada model ontologi yang ada.

2. Ontologi

2.1 Pengertian Ontologi

Ontologi adalah sebuah konseptualisasi dari domain ke dalam format yang dapat dimengerti manusia, tapi dapat dibaca oleh mesin yang terdiri dari entitas, atribut, hubungan, dan aksioma [4]. Ontologi dapat memberikan konseptualisasi domain kerja yang banyak dari suatu organisasi, yang merepresentasikan konsep-konsep utama dan hubungan kegiatan kerja. Hubungan ini bisa merepresentasikan informasi terisolasi seperti nomor telepon rumah karyawan, atau mereka bisa mewakili kegiatan seperti perizinan kepemilikan dokumen, atau menghadiri konferensi [3].

2.2 Komponen Ontologi

Ontologi menggunakan berbagai macam variasi struktur, tergantung dari penggunaan bahasa ontologi termasuk sintaksis yang digunakan. Ontologi bergantung kepada aplikasi yang digunakan, tidak hanya dari data yang terdapat dalam ontologi. Ontologi tidak melakukan fungsi

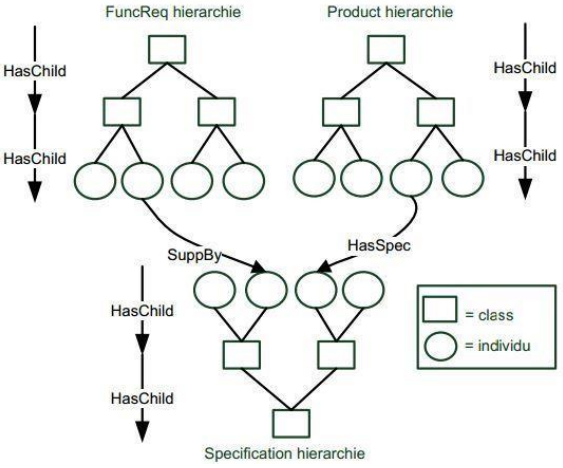
perhitungan atau fungsi lainnya yang memproses ontologi. Ontologi hanya memberikan komponen kepada node dari graf suatu jaringan semantik. Seperti yang sudah disebutkan sebelumnya ontologi memiliki komponen sebagai berikut:

1) Entitas *(Entity)*
   Entity (juga dikenal sebagai *classes*, *objects* dan *categories*) menjelaskan konsep-konsep suatu domain. Sebuah entitas terdiri dari obyek-obyek yang merupakan penjelasan dari tugas, fungsi, aksi, strategi, dan sebagainya. Sebuah kelas juga bisa memiliki subkelas yang akan mempresentasikan konsep yang lebih spesifik daripada superkelasnya.
2) Atribut (*Attribute*)
   Atribut adalah komponen dasar dari suatu obyek ontologi. Atribut menyatakan obyek-obyek dalam suatu domain yang diteliti yang digunakan untuk merepresentasikan elemen nyata seperti hewan, tanaman, dan manusia, maupun elemen abstrak seperti bilangan dan huruf.
3) Hubungan (*Relation*)
   Merupakan representasi sebuah tipe dari interaksi antara konsep dari sebuah domain. Secara formal dapat didefinisikan sebagai subset dari sebuah produk dari n set, R:C1 x C2 X . . . x Cn. Sebagai contoh dari relasi binari termasuk *subclass-of* dan *connected-to*. Relasi harus mampu mendefinisikan hubungan dari entitas yang ada.
4) Aksiom (*Axioms*)
   Digunakan untuk memodelkan sebuah *sentence* yang selalau benar.

2.3 Struktur Model Ontologi

Menurut [5], suatu model ontologi terdiri dari tiga kelas hirarki, yaitu kelas kebutuhan fungsional, produk, dan spesifikasi. Ketiga kelas tersebut masing-masing memiliki *subclasses* dan individu yang merepresentasikan sebuah hirarki.

Setiap kelas hirarki tersebut memiliki struktur *rooted tree* yang *node*-nya merupakan *subclasses* dari masing-masing kelas hirarki dan daunnya merupakan individu. Individu di sini berperan sebagai titik relasi antara satu kelas hirarki dengan kelas hirarki lainnya. Dalam struktur model ontologi yang dijabarkan dalam [5], terdapat dua relasi, yaitu relasi antar kelas hirarki dan relasi dalam kelas hirarki. Relasi antar kelas hirarki menghubungkan dua individu antar kelas hiraki yang terdiri atas relasi *SuppBy* dan *HasSpec*. Sedangkan relasi dalam kelas hirarki menghubungkan dua node yang merupakan subclasses dalam masing-masing kelas hirarki yang terdiri atas relasi *HasChild*.



Gambar 2-1 Struktur model ontologi [5]

3. Penggalian Preferensi

Pengguna pada awalnya tidak dapat mengidentifikasi dan menyatakan preferensi mereka secara jelas dalam membeli sebuah barang. Pengambilan keputusan lebih ditandai dengan proses penggalian preferensi. Cara *recommender system* berinteraksi kepada pengguna memiliki dampak yang besar pada hasil dari proses pengambilan keputusan [2]. Di sini lah peran *preference elicitation* atau penggalian preferensi sangat dibutuhkan.

Bisa kita analogikan penggalian preferensi dengan orang yang ingin membeli sebuah komputer dan ingin berkonsultasi dahulu dengan ahli komputer yang dikenalnya.

Menurut [5], preferensi seorang pengguna dapat dibagi ke dalam dua kategori, yaitu hard constraint dan soft constraint.

3.1 *Hard Constraint*

Hard constraint merupakan batasan suatu preferensi yang diset pengguna sebagai kebutuhan mutlak. Sehingga produk-produk yang tidak mendukung penuh terhadap suatu preferensi akan dikeluarkan dari daftar rekomendasi produk.

Dalam penelitian ini beberapa preferensi yang bisa diset sebagai *hard constraint* adalah properti produk (harga, merk, dan sebagainya), tipe produk, dan kebutuhan fungsional.

3.2 *Soft Constraint*

Soft constraint merupakan batasan suatu preferensi yang diset pengguna sebagai kebutuhan yang tidak mutlak. Sehingga produk-produk yang tidak mendukung penuh terhadap suatu preferensi akan tetap dimasukkan ke daftar rekomendasi produk.

Dalam penelitian ini preferensi yang bisa diset sebagai *soft constraint* hanya kebutuhan fungsional.

4. Rekomendasi Produk

Untuk merekomendasikan suatu produk berdasarkan [5], dibutuhkan beberapa tahap untuk melakukan rekomendasi produk sesuai *constraint* yang ditetapkan pengguna.

Hard constraint akan diperlukan untuk menyaring produk yang mendukung penuh *constraint* tersebut dengan menggunakan *recommendation function*. *Soft constraint* tidak akan menyaring produk, tetapi akan memberikan *utility score* terhadap produk yang mendukung penuh *constraint* tersebut. Sehingga untuk produk yang tidak mendukung penuh suatu *soft constraint* akan memiliki *utility score* yang tidak bertambah.

4.1 *Recommendation Function*

Pada [5] dijelaskan bahwa untuk menetapkan suatu produk layak atau tidaknya untuk direkomendasikan terhadap suatu individu kebutuhan fungsional dengan mengacu persamaan (1).

[illegible] ..... (1)

Di mana :

[illegible] = individu dari *functional requirement*

[illegible] = individu dari produk

[illegible] = himpunan dari komponen yang mendukung individu *functional requirement* [illegible]

[illegible] = himpunan dari komponen yang dimiliki individu produk [illegible]

[illegible] = himpunan dari level atas dari hirarki suatu himpunan komponen pada ontologi

Yang kemudian akan dinotasikan dengan [illegible]

→ [illegible] Kemudian untuk perekomendasian pada functional requirement yang bertipe class pada ontologi yang bernotasi [illegible] dikatakan memenuhi jika dan hanya jika [illegible], $i > 0$.

4.2 *Utility Function*

Sesuai yang dijelaskan pada [5], *utility function* digunakan untuk memberikan *score* kepada produk sesuai tingkat dukungannya terhadap suatu *functional requirement*. Adapun parameter yang digunakan untuk memberikan suatu *utility score* terhadap suatu produk adalah *supporting range*, *relative importance*, dan DOI.

requirement. Jika suatu *functional requirement HD Gaming* membutuhkan komponen GPU dan RAM, bisa kita pastikan bahwa keterkaitan GPU akan lebih besar pengaruhnya daripada RAM untuk menjalankan *functional requirement HD Gaming*. Maka nilai *relative importance* GPU akan lebih tinggi dari RAM.

Nilai *supporting range* maupun *relative importance* akan berlaku :

$$\sum_{i=1}^{n} [illegible] = 1 \text{ dan } \sum_{i=1}^{n} [illegible] = 1$$

Dengan :

[illegible] = jumlah tingkatan suatu komponen (*Low End Component* hingga *High End Component*)

[illegible] = *supporting range* untuk komponen [illegible]

[illegible] = jumlah komponen yang dibutuhkan suatu *functional requirement* [illegible]

[illegible] = *relative importance* dari komponen [illegible] terhadap *functional requirement* [illegible]

Setelah definisi parameter telah diketahui, maka untuk *utility function* yang digunakan pada penelitian ini ada pada persamaan (2).

$$[illegible] = \sum_{i=1}^{[illegible]} (\sum_{i=1}^{[illegible]} [illegible] \cdot [illegible] \cdot [illegible] \quad \text{..... (2)}$$

Dengan :

[illegible] = jumlah *functional requirement* yang dipilih pengguna

[illegible] = jumlah komponen yang mendukung *functional requirement* [illegible]

5. Pembangkitan Pertanyaan

Dalam pembangkitan pertanyaan terdapat

Supporting range adalah nilai *support level* terhadap komponen yang dimiliki suatu produk. Jika produk A memiliki komponen *High End Processor*, sedangkan produk B memiliki *Low End Processor*, bisa dipastikan *supporting range* milik produk A akan lebih tinggi dari produk B setelah dilakukan normalisasi.

Relative importance adalah nilai keterkaitan suatu komponen terhadap suatu *functional*

lima macam kasus yang dapat ditemukan saat sistem sedang digunakan, yaitu kasus U1, U2, U3, U4, dan U5 [5].

5.1 Kasus U1 - *Empty User Profile* [5]

Pada kasus ini sistem masih belum mendapatkan *user profile model* yang berisi preferensi properti produk, tipe produk, dan kebutuhan fungsional. Karena *user profile model* masih kosong, maka semua *functional requirement nodes* pada level paling atas (level 1) potensial untuk ditanyakan.

Pada penelitian ini, urutan pertanyaan yang akan ditampilkan untuk kasus U1 adalah secara acak. Sehingga akan mengikuti persamaan (3).

$$\text{[illegible]} = \{\text{[illegible]}\ \mathcal{N}_{\text{[illegible]}(1)})\} \cup \text{[illegible]} \cup \text{[illegible]} \quad \text{..... (3)}$$

Di mana :

[illegible] = himpunan pertanyaan

$\mathcal{N}_{\text{[illegible]}(1)}$ = himpunan hirarki *functional requirement* level 1

[illegible] = himpunan tipe produk

[illegible] = himpunan properti produk

5.2 Kasus U2 - Terdapat Lebih dari Satu Produk yang Dipilih Pengguna [5]

Kasus ini dapat terjadi jika pengguna memiih lebih dari satu produk saat sistem menampilkan hasil rekomendasi produk. Ini menandakan pengguna masih ragu terhadap produk yang dipilihnya. Sehingga dibutuhkan pembangkitan pertanyaan yang berbeda.

Strategi pembangkitan pertanyaan yang akan digunakan dalam penelitian ini terdapat 3 tahap.

1. Mencari elemen yang akan membedakan produk satu dengan produk lainnya. Elemen tersebut bisa berupa komponen, segmen produk, *functional requirement*, atau kombinasi ketiganya. Untuk elemen *functional requirement* akan diambil berdasarkan preferensi akhir yang terdapat pada *user profile model* yang berhubungan ontologi dengan notasi $\mathcal{N}_{ontologi}$.
2. Untuk menentukan elemen pembeda (*distinguishing element*) setiap produk yang akan dinotasikan dengan DE_p, maka persamaan (4) akan digunakan.

$$DE_p = P_{produk} - \left(\bigcup_{p' \in (P - \{p\})} P_{produk}(p')\right) \quad \text{... (4)}$$

Di mana :

P

$P_{produk} = \{x \in X_{ontologi} \mid x \rightarrow \ \cap \ \mathcal{N}_{ontologi}\}$

P_{produk} = himpunan *functional requirement* yang didukung produk

3. Setelah *distinguishing element* setiap produk telah didapat, maka langkah selanjutnya adalah memilih satu dari *distinguishing element* $DE(p)$ sebagai pertanyaan yang berasosiasi dengan produk Kemudian menggabungkan ke dalam sebuah himpunan pertanyaan yang merupakan himpunan asosiasi produk.

5.3 Kasus U3 - Tidak Ada Produk yang Dipilih Pengguna [5]

Pada kasus ini, pengguna tidak memilih satu pun prroduk hasil rekomendasi yang ditawarkan sistem. Ini menandakan bahwa pengguna masih merasa belum menemukan produk yang sesuai.

Terdapat dua langkah yang akan dilakukan sistem pada penelitian ini. Pertama sistem akan memunculkan produk-produk selain produk yang tidak dipilih pengguna berurut dari *utility score* tertinggi ke terendah. Langkah pertama ini akan berulang berdasarkan *threshold* tertentu. Jika pengguna masih belum memilih satu pun produk dari langkah pertama, langkah kedua adalah dalam ontologi dengan *functional requirement* pada pertanyaan sebelum hasil rekomendasi ditampilkan. Jika pada level yang sama tersebut tidak ditemukan lagi *functional requirement* yang potensial, sistem akan melakukan *backtrack* hingga terdapat *functional requirement* potensial lainnya yang kemudian akan dibangkitkan sebagai pertanyaan.

Untuk melakukan langkah kedua, maka persamaan (5) akan digunakan.

$$= \begin{cases} , \neq \{\} \\ , = \{\} \end{cases} \quad \text{..... (5)}$$

Di mana :
= himpunan pertanyaan
= batas jumlah pertanyaan

= himpunan *functional requirement* potensial pada level dalam ontologi

5.4 Kasus U4 - Definisi Kebutuhan Masih Kurang Spesifik untuk Membangkitkan Rekomendasi [5]

Pada kasus ini, kebutuhan yang kurang spesifik dikarenakan jumlah produk rekomendasi masih di atas *threshold* tertentu. Sehingga strategi pembangkitan pertanyaan pada kasus ini adalah membangkitkan *functional requirement* level bawah dalam ontologi dari preferensi yang dipilih pengguna.

Pembangkitan *functional requirement* level bawah bertujuan untuk mempersempit lagi preferensi pengguna, sehingga produk rekomendasi akan semakin sedikit hingga di bawah *threshold* tertentu. Untuk melakukan penelusuran level bawah suatu *functional requirement*, maka persamaan (6) akan digunakan.

$$= \begin{cases} , \neq \{\} \\ , = \{\} \end{cases} \quad \text{..... (6)}$$

membangkitkan kembali pertanyaan untuk menggali lagi preferensi pengguna.

Pada langkah kedua, pertanyaan yang dibangkitkan adalah *functional requirement* potensial lainnya yang masih dalam level yang sama

Di mana :
= himpunan pertanyaan
= batas jumlah pertanyaan
= = himpunan *functional requirement* pada level bawah dalam ontologi dari *functional requirement*

5.5 Kasus U5 - Tidak Ada Produk yang Sesuai dengan *User Profile Model* [5]

Tidak adanya produk yang sesuai dengan *user profile model* menandakan bahwa sistem telah melakukan penyaringan terhadap produk hingga produk habis. Penyebab produk ini habis bisa karena preferensi tipe produk, properti produk, kebutuhan fungsional, ataupun kombinasinya. Untuk itu dalam kasus ini, sistem akan melakukan *constraint relaxation*.

Dalam proses *constraint relaxation*, sistem akan menawarkan constraint atau preferensi yang akan direlaksasi kepada pengguna. Pengguna bisa

mengubah tipe produk, properti produk, ataupun kebutuhan fungsional untuk direlaksasi, sehingga sistem akan melakukan proses rekomendasi ulang terhadap pengubahan *constraint* yang dilakukan oleh pengguna.

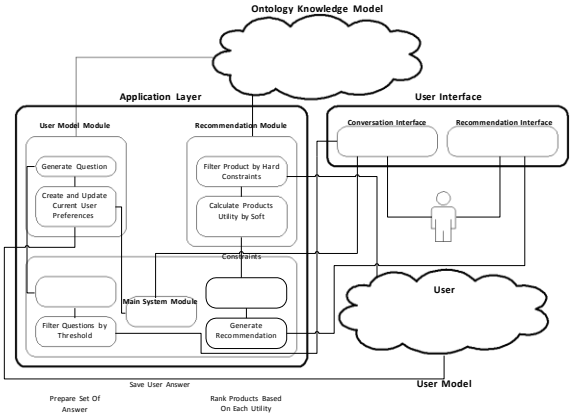
Untuk pada bagian preferensi kebutuhan fungsional, yang akan direlaksasi hanyalah preferensi yang diset sebagai *hard constraint*. Di sini sistem akan mencoba mencari himpunan kebutuhan fungsional yang tidak didukung penuh oleh produk terakhir yang menyebabkan himpunan produk rekomendasi menjadi kosong. Persamaan (7) akan digunakan untuk melakukan *constraint relaxation* tersebut.

$$\text{[illegible]} = \{\{\text{[illegible]} \in h_i\} \cup \text{[illegible]} \cup \text{[illegible]} \mid \text{[illegible]} = \{\}\} \quad (7)$$

Di mana :
[illegible] = himpunan pertanyaan
h = himpunan *hard constraint functional requirement*
[illegible] = himpunan produk hasil rekomendasi terhadap *user profile model* [illegible]

6. Gambaran Umum Sistem

Pada penelitian ini, sistem dibagun dengan menggunakan bahasa pemrograman Java Standard Edition. Di dalam sistem terdapat tiga modul utama, yaitu *User Model Module*, *Recommendation Module*, dan *Main System Module*. Selanjutnya untuk model pengetahuannya digunakan model ontologi yang disimpan dalam file berformat .owl (*Ontology Web Language*). Lalu sebuah *user model* yang berupa file berformat .java akan disediakan untuk menyimpan preferensi pengguna yang didapat dari hasil tanya jawab dengan sistem. *User model* akan menyimpan preferensi properti produk, tipe produk, dan kebutuhan fungsional (*hard constraint* dan *soft constraint*).



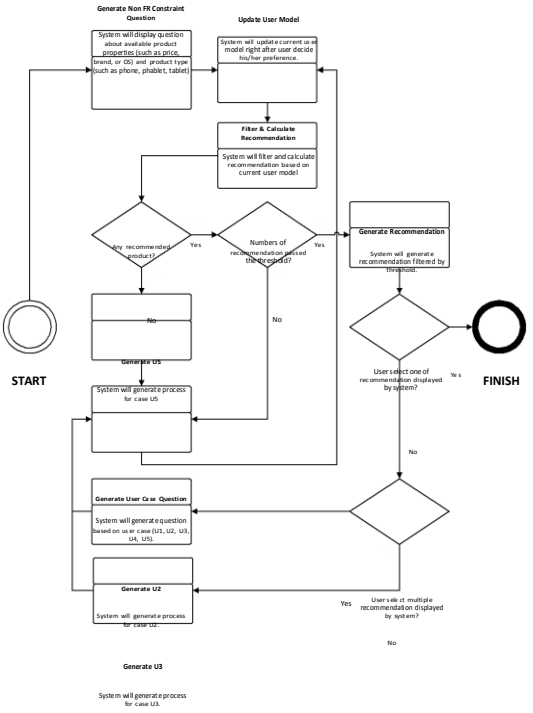
Gambar 6-1 Gambaran umum arsitektur sistem

Dari Gambar 6-1, setiap modul dari sistem memiliki fungsi masing-masing. Berikut penjabaran fungsi dari setiap modul.

1. *User Model Module*
   Modul ini berfungsi untuk membangkitkan pertanyaan yang akan disalurkan kepada pengguna melalui *Main System Module*. Kemudian modul ini akan membuat *user model* (jika *user model* masih kosong) dan memperbarui *user model* (jika *user model* sudah terisi) dengan preferensi pengguna dari hasil tanya jawab sistem dengan pengguna yang terjadi pada *Conversation Interface*.
2. *Recommendation Module*
   Modul ini berfungsi untuk menyaring produk berdasarkan preferensi *hard constraint* pengguna yang didapat dari *user model*. Setelah disaring, modul ini akan memberikan *utility score* kepada setiap produk berdasarkan preferensi *soft constraint* pengguna yang didapat dari *user model*. Lalu produk hasil rekomendasi akan disalurkan ke *Main System Module* untuk kemudian ditampilkan di *Recommendation Interface*.
3. *Main System Module*
   Modul ini berfungsi sebagai perantara dua modul sebelumnya untuk disalurkan ke *User Interface*. Ditambah juga dengan fungsi penyaringan jumlah pertanyaan dan produk hasil rekomendasi sesuai dengan *threshold* yang sudah ditentukan, penyimpanan jawaban pengguna, dan pengurutan produk hasil rekomendasi.

7. Alur Interaksi Sistem

Pada penelitian ini, pengguna diasumsikan sudah mengetahui preferensi kebutuhan fungsionalnya. Tugas sistem di sini adalah untuk mengolah preferensi tersebut menjadi sebuah hasil rekomendasi. Untuk alur proses interaksi sistem hingga ditemukan hasil rekomendasi secara umum dapat dilihat pada Gambar 7-1.



Gambar 7-1 Diagram *state* sistem

Berdasarkan diagram *state* diatas, penjabaran proses dalam sistem adalah sebagai berikut :

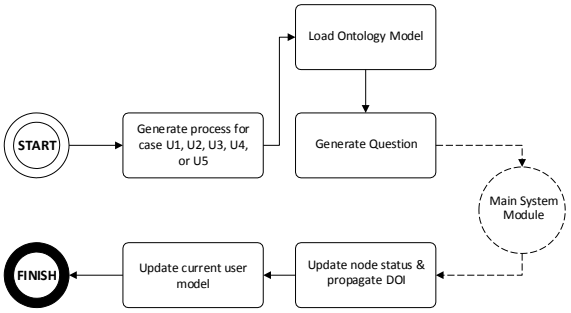
1. Sistem memulai dengan mengajukan pertanyaan mengenai Non *Functional Requirement* (FR) *Constraint* yang meliputi properti produk (harga, merk, dan sistem operasi) dan tipe produk (phone, phablet, dan tablet).
2. Setelah pengguna memilih preferensi yang diinginkan, sistem akan menyimpan preferensi tersebut ke dalam *user model*. Semua preferensi pada proses ini akan dibagi menjadi *hard constraint* dan *soft constraint*.
3. Setelah *user model* diperbarui, sistem akan menyaring produk dari preferensi *hard constraint* dan menghitung *utility score* setiap produk dari preferensi *soft constraint* yang terdapat pada *user model*.
4. Kemudian sistem akan melakukan pengecekan bahwa terdapat atau tidaknya produk rekomendasi, jika tidak sistem akan masuk ke proses Generate U5, namun jika ya sistem akan melakukan pengecekan jumlah produk rekomendasi dengan *threshold* yang sudah ditentukan. Jika belum memenuhi persyaratan *threshold*, sistem akan membangkitkan pertanyaan selanjutnya (proses Generate U1 atau Generate U4), namun jika ya sistem akan membangkitkan hasil rekomendasi.
5. Jika hasil rekomendasi telah dibangkitkan dan pengguna memilih satu produk rekomendasi, sistem akan menyudahi interaksi dengan pengguna. Jika pengguna memilih lebih dari satu produk, maka sistem akan masuk ke proses Generate U2. Jika pengguna tidak memilih satu pun produk hasil rekomendasi, sistem akan masuk ke proses Generate U3.
6. Sistem hanya akan berhenti jika pengguna hanya memilih satu produk dari hasil rekomendasi yang ditawarkan oleh sistem.

8. Perancangan Sistem

Dari Gambar 6-1, sistem terdiri dari tiga modul utama, yaitu *User Model Module*, *Recommendation Module*, dan *Main System Module*.

8.1 *User Model Module*

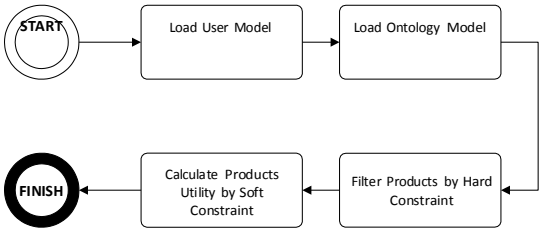
Tugas utama modul ini adalah membangkitkan pertanyaan dan memperbarui *user model* yang ada. Untuk pembangkitan pertanyaan, terdapat lima kasus yang dapat terjadi selama sistem berinteraksi dengan pengguna, yaitu kasus U1, U2, U3, U4, U5.



Gambar 8-1 Diagram aktivitas *User Model Module*

8.2 *Recommendation Module*

Modul ini berfungsi untuk menyaring produk dari *hard constraint* yang terdapat pada *user model* dan memberikan *utility score* ke setiap produk berdasarkan *soft constraint* yang terdapat pada *user model*. Proses penyaringan dan penghitungan *utility score* produk dilakukan melalui *reasoning* dengan model ontologi.

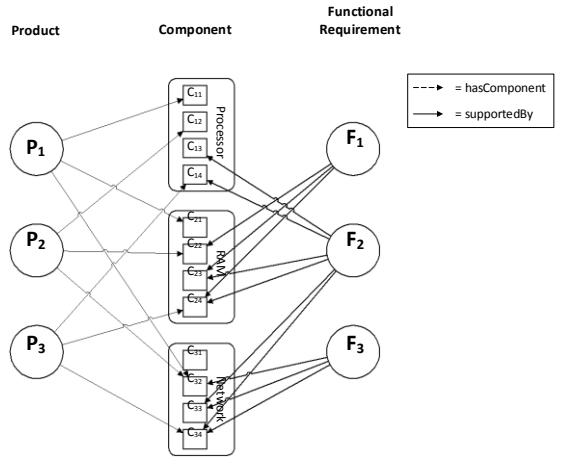


Gambar 8-2 Diagram aktivitas *Recommendation Module*

8.2.1 *Filter Products by Hard Constraint*

Hard constraint pada penelitian ini adalah properti produk, tipe produk, dan kebutuhan fungsional. Sehingga sebelum sistem melakukan penyaringan produk terhadap kebutuhan fungsional, sistem melakukan penyaringan terlebih dahulu terhadap preferensi properti produk dan tipe produk yang terdapat pada *user model*.

Pada proses ini terdapat dua skenario, yaitu skenario jika *hard constraint functional requirement* pada *user model* $\neq$ *null* dan sebaliknya. Pada skenario pertama, proses penyaringan produk dilakukan melalui persamaan (1). Untuk skenario kedua, sistem akan menyaring produk berdasarkan *utility threshold*.



Gambar 8-3 Ilustrasi model ontologi beserta relasinya

Dari Gambar 8-3, misal, terdapat himpunan *hard constraint functional requirement* F1, F2, dan F3 pada suatu *user model*. Untuk mendapatkan produk yang mendukung penuh *hard constraint* tersebut, persamaan (1) akan digunakan.

$$\text{[illegible}(\text{[illegible}(\text{[illegible}(

utility score ≥ 0.188533 yang akan dimasukkan ke

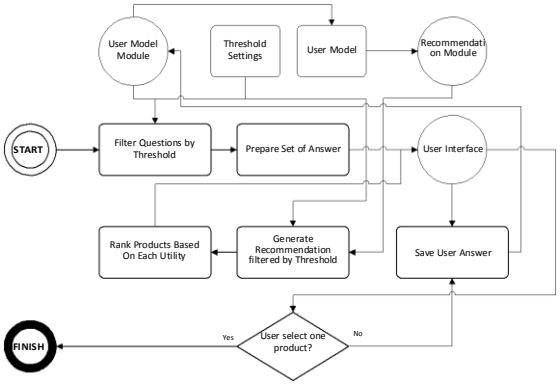
dalam hasil rekomendasi. Pada kasus di atas, hanya produk P3 yang akan masuk hasil rekomendasi.

Namun jika *hard constraint functional requirement* ≠ *null*, penyaringan tetap akan dilakukan berdasarkan *hard constraint*. Dan proses ini hanya berfungsi untuk menghitung dan memperbarui *utility score* produk.

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

8.3 *Main System Module*

Modul ini berperan untuk mengolah pertanyaan yang akan diberikan pengguna dari *User Model Module* dan menyimpan jawaban pengguna yang kemudian akan diolah oleh *Recommendation Module*. Modul ini juga berfungsi untuk melakukan pengecekan terhadap *threshold* yang ada, mulai dari *question threshold* hingga *recommendation threshold*.



Gambar 8-4 Diagram aktivitas *Main System Module*

Dari Gambar 8-4, dapat disimpulkan bahwa fungsi utama *Main System Module* adalah sebagai pengendali semua modul yang ada, mulai dari *User Model Module*, *Recommendation Module*, hingga *User Interface* untuk menjalankan fungsinya masing-masing. Modul ini juga berfungsi untuk memberikan *state* berhenti pada sistem saat pengguna telah menentukan satu produk dari hasil rekomendasi yang ditampilkan sistem.

9. Pengujian dan Analisis

9.1 Skenario Pengujian

Pada penelitian ini terdapat dua skenario pengujian, yaitu pengujian akurasi *ranking* oleh *expert* dan pengujian melalui kepuasan pengguna.

9.1.1 Pengujian Akurasi *Ranking* oleh *Expert*

Pengujian akurasi ranking dilakukan oleh *expert* dari domain yang diteliti, yaitu domain *smartphone*. Pengujian akurasi dilakukan oleh *expert* yang memiliki kompetensi di bidang *mobile application development*.

Berikut tahapan skenario pengujian akurasi *ranking* oleh *expert*.

- Pengujian akan dilakukan sebanyak tiga kali. Pengujian pertama, sistem akan menampilkan hasil rekomendasi sebanyak 5 produk. Pengujian kedua, sistem akan menampilkan 8 produk. Dan pengujian ketiga, sistem akan menampilkan 10 produk.
- Untuk setiap pengujian, *expert* diperkenankan hanya untuk memilih preferensi, namun tidak diperkenankan untuk melihat hasil rekomendasi yang ditampilkan sistem (*blackbox*).
- Jika sistem sudah menampilkan hasil rekomendasi, penulis mencatat hasil rekomendasi dari sistem dengan urutan acak dan memberikan catatan tersebut kepada *expert*.
- Setelah itu, *expert* diperkenankan untuk mengurutkan produk berdasarkan kemampuan obyektifnya dalam merekomendasikan produk.
- Setelah *expert* mengurutkan produk secara *blackbox*, penulis akan melakukan penghitungan akurasi menggunakan MAE (*Mean Absolute Error*).

9.1.2 Pengujian Kepuasan Pengguna

Pada penelitian ini, berikut skenario yang dilakukan dalam menguji kepuasan pengguna.

- Pengguna dapat mencoba sistem melalui perangkat keras milik penulis ataupun mengunduh sistem dari *link* internet yang disediakan penulis.
- Setelah pengguna selesai menggunakan sistem, penulis menyediakan form *feedback* untuk diisi pengguna.
- Berikut beberapa faktor yang akan diajukan pada form *feedback* yang diisi pengguna.
  - Metode tanya jawab yang digunakan memudahkan dalam menentukan preferensi pengguna.
  - Sistem dapat memandu pengguna dengan baik.
  - Sistem tanya jawab dapat mempercepat pengguna dalam menentukan preferensi dibandingkan sistem *specification filtering*.
  - Penggunaan fitur *hard constraint* dan skala prioritas (*soft constraint*) membantu pengguna.
- Pengujian dilakukan kepada pengguna dengan rentang umur (14-50 tahun).
- Setelah pengujian selesai, penulis akan menghitung persentasi kepuasan dari setiap faktor yang diajukan kepada pengguna.

9.2 Analisis Pengujian

9.2.1 Analisis Pengujian Akurasi *Ranking* oleh *Expert*

Pengujian Expert I - Toufan D Tambunan											
Jumlah Produk	Rank										MAE
	1	2	3	4	5	6	7	8	9	10	
5	1	3	4	2	5						0.8
8	2	1	6	4	5	3	7	8			1
10	3	2	1	8	7	6	5	4	9	10	1.6

Tabel 8-1 Hasil pengujian oleh *expert* pertama

Pengujian Expert II - Pramuko Aji											
Jumlah Produk	Rank										MAE
	1	2	3	4	5	6	7	8	9	10	
5	3	2	4	1	5						1.2
8	3	4	5	2	6	1	7	8			1.75
10	5	4	1	10	2	8	3	9	7	6	3

Tabel 8-2 Hasil pengujian oleh *expert* kedua

Jumlah Produk	Rata-rata MAE
5	1
8	1.375
10	2.3

Tabel 8-3 Hasil rata-rata akurasi *ranking* produk oleh *expert*

Dari Tabel 8-3, dapat dilihat bahwa akurasi *ranking* produk yang diperoleh sangat baik terdapat pada jumlah 5 produk. Sedangkan nilai rata-rata MAE pada jumlah 8 dan 10 produk masih cukup besar. Artinya kemungkinan terjadinya kesalahan perangkingan pada jumlah produk 8 dan 10 akan lebih besar dari jumlah produk 5. Semakin rendah nilai MAE, maka semakin baik sistem dalam bekerja.

9.2.2 Analisis Pengujian Kepuasan Pengguna

Faktor	Tingkat Kepuasan				
	Sangat Kurang	Kurang	Cukup	Baik	Sangat Baik
Metode tanya jawab memudahkan	0.00%	2.38%	21.43%	64.29%	11.90%
Sistem dapat memandu	0.00%	2.38%	19.05%	50.00%	28.57%
Sistem tanya jawab mempercepat penentuan preferensi	0.00%	2.38%	21.43%	50.00%	26.19%
Hard constraint dan soft constraint membantu	0.00%	2.38%	9.52%	66.67%	21.43%

Tabel 8-4 Persentasi hasil pengujian kepuasan pengguna

Dari hasil pengujian yang dilakukan dengan total responden kurang lebih 50 orang, dapat disimpulkan bahwa :

- 11.90% pengguna menyatakan bahwa metode tanya jawab memudahkan pengguna dengan sangat baik, 64.29% pengguna menyatakan baik, 21.43% pengguna menyatakan cukup, dan sisanya menyatakan kurang.
- 28.57% pengguna menyatakan bahwa sistem dapat memandu pengguna dengan sangat baik, 50% pengguna menyatakan baik, 19.05% pengguna menyatakan cukup, dan sisanya menyatakan kurang.
- 26.19% pengguna menyatakan bahwa sistem tanya jawab mempercepat penentuan preferensi pengguna dengan sangat baik, 50% pengguna menyatakan baik, 21.43% pengguna menyatakan cukup, dan sisanya menyatakan kurang.
- 21.43% pengguna menyatakan bahwa metode penentuan *hard constraint* dan *soft constraint* membantu pengguna dengan sangat baik, 66.67% pengguna menyatakan baik, 9.52% pengguna menyatakan cukup, dan sisanya menyatakan kurang.

10. Kesimpulan

Berdasarkan hasil pengujian dan analisis pada bab IV, dapat disimpulkan bahwa :

- Lebih dari 50% pengguna menyatakan baik dari empat faktor pertanyaan yang diajukan. Sehingga *conversational recommender system* sangat membantu pengguna dalam menemukan produk yang diinginkan.
- Lebih dari 50% pengguna menyatakan metode tanya jawab pada sistem dapat membantu dengan baik. Sehingga dengan basis ontologi, metode tanya jawab dapat membantu pengguna.
- Lebih dari 50% pengguna menyatakan fitur *preference elicitation* (pilihan *hard* dan *soft constraint*) dapat membantu dengan baik. Sehingga dengan adanya fitur *preference elicitation*, pengguna dapat "memberitahu" sistem bahwa produk yang diinginkan harus memenuhi kedua *constraint* tersebut.
- Jumlah produk yang paling efektif untuk ditampilkan adalah 5. Karena jumlah tersebut memiliki kemungkinan paling kecil (nilai rata-rata MAE = 1) untuk salah dalam menampilkan urutan ranking produk hasil rekomendasi dibandingkan jumlah produk 8 (nilai rata-rata MAE = 1.375) dan 10 (nilai rata-rata MAE = 2.3).

Daftar Pustaka

[1] S. E. MIDDLETON, N. R. SHADBOLT and D. C. D. ROURE, "Ontological User Profiling in Recommender Systems," 2001.

[2] Jannach, Zanker, Felfernig and Friedrich, "Recommender Systems An Introduction," *IEEE Intelligent Systems 14*, pp. 44-54, 2011.

[3] I. M. Candiasa, Jaringan Semantik Untuk Memfasilitasi Pembelajaran Matematika Berbantuan Komputer, 2004, pp. 25-44.

[4] L. Y. Banowosari and I. W. S. Wicaksana, "PEMELIHARAAN ONTOLOGI PADA PEER-TO-PEER(P2P) BERBASIS VOTING DAN SIMILARITAS," *1st Seminar on Application and Research in Industrial Technology, SMART* , 2006.

[5] D. &. B. Z. Widyantoro, "A Framework of Conversational Recommender System Based on User Functional Requirement," 2014.