

Pencarian Aturan Asosiasi *Semantic Web* Untuk Obat Tradisional Indonesia

Ridowati Gunawan¹, Khabib Mustofa²

Abstract—Indonesia has more than 2000 types of plants that can be used for medicine. Indonesian traditional medicine called jamu utilizes various of medicinal plants. Since each medicinal plant has different efficacy, jamu also has different efficacy. Jamu can be used to cure certain type of diseases. Jamu that has same efficacy can be produced by many different companies and has different composition. In order to bring benefit for the consumer, knowledge about efficacy of medicinal plants, efficacy of jamu, and composition of jamu is needed. One way to gain knowledge about jamu, along with the entire composition, is to utilize association rule mining technique. If in general the technique only utilizes a single database, in this paper the data source is obtained from semantic web. The data in semantic web is stored in the form of RDF or OWL according to ontology jamu. Data in the form of RDF/OWL is converted into transaction data using library rrdof of R, and its results will be processed using Apriori, which is one of the algorithms in association rule. Results of Apriori algorithm produce association rules on the composition of the jamu along with the value of the support, confidence, and lift ratio. These results indicate the value of lift ratio > 1 which means medicinal plants depend on each other.

Intisari—Indonesia memiliki lebih dari 2000 jenis tanaman yang dapat digunakan untuk pengobatan. Obat tradisional Indonesia yang dikenal dengan nama jamu memanfaatkan berbagai jenis tanaman obat tersebut. Setiap tanaman obat dapat memiliki khasiat yang berbeda-beda, demikian juga ketika tanaman obat tersebut telah menjadi ramuan jamu. Seperti halnya obat yang dapat digunakan untuk menyembuhkan penyakit, demikian juga dengan jamu. Jamu yang memiliki khasiat yang sama dapat diproduksi oleh berbagai perusahaan dan memiliki komposisi yang berbeda. Agar jamu membawa manfaat bagi yang meminumnya, pengetahuan mengenai khasiat dari tanaman obat, khasiat dari ramuan jamu, dan komposisi dari ramuan jamu sangatlah dibutuhkan. Salah satu cara memperoleh pengetahuan tentang ramuan jamu beserta dengan seluruh komposisinya adalah dengan memanfaatkan teknik *association rule mining*. Jika pada umumnya teknik ini hanya memanfaatkan basis data tunggal, maka pada makalah ini sumber data diperoleh dari hasil *semantic web*. Data dalam *semantic web* disimpan dalam bentuk RDF atau OWL sesuai dengan *ontology* mengenai jamu. Data bentuk RDF/OWL diubah ke dalam bentuk data transaksi dengan menggunakan *library rrdof* dari R, dan selanjutnya diolah menggunakan algoritme Apriori, yang merupakan salah satu

algoritme dalam *association rule*. Hasil dari algoritme Apriori ini adalah aturan-aturan asosiasi tentang komposisi tanaman jamu beserta dengan nilai *support*, *confidence*, dan *lift ratio*. Hasil menunjukkan nilai *lift ratio* lebih dari 1 yang artinya antara satu tanaman obat dengan tanaman obat yang lainnya saling bergantung satu sama lain.

Kata Kunci— *Association Rule Mining*, *Semantic Web*, Obat Tradisional Indonesia.

I. PENDAHULUAN

Jamu merupakan obat herbal tradisional Indonesia. Jamu diformulasikan dari berbagai tanaman yang ada di Indonesia. Penduduk Indonesia sangat memercayai khasiat dari jamu. Jamu merupakan obat asli Indonesia, sudah dikenal sejak jaman dahulu. Relief Karmawibhanga Candi Borobudur menunjukkan tradisi minum jamu. Jamu dalam bahasa Jawa sering disebut dengan Jampi, istilah Jampi ditemukan pada naskah kuno seperti Ghatotkacasraya (Mpu Panuluh), juga pada Serat Centhini tahun 1814, dan Serat Kawruh Bab Jampi-Jampi Jawi tahun 1831.

Karena merupakan tradisi sebagian besar masyarakat Indonesia, maka khasiat dari jamu dan kebiasaan meminum jamu harus dipertahankan. Oleh karenanya, untuk tetap melestarikan jamu, Pemerintah RI pada tahun 2010 melalui Menteri Kesehatan membuat kebijakan tentang saintifikasi jamu. Jamu tidak hanya sebagai pengetahuan, tetapi jamu harus dapat dirasakan khasiatnya oleh orang yang meminumnya. Kajian-kajian ilmiah terhadap ramuan jamu harus terus dilakukan dan dilestarikan.

Banyaknya jamu yang beredar di pasaran, baik yang diproduksi oleh industri rumahan maupun pabrik, serta jumlah tanaman yang dapat digunakan sebagai formula jamu, maka perlu dibuat sebuah analisis sains mengenai relasi antara ramuan jamu dan tanaman yang digunakan. Masyarakat perlu mengetahui sebuah ramuan jamu tersusun dari tanaman obat apa saja.

Satu tanaman dapat digunakan oleh berbagai jenis jamu. Setiap tanaman memiliki efek yang unik. Jenis jamu yang sama dan diproduksi oleh perusahaan jamu yang berbeda akan memiliki komposisi tanaman yang berbeda. Hal ini tentunya membingungkan bagi orang yang hendak mengkonsumsi jamu, harus memilih produk jamu dari perusahaan/industri jamu yang mana.

Untuk memperoleh informasi mengenai hubungan antara tanaman yang satu dengan tanaman yang lain dalam membentuk ramuan jamu beserta dengan industri yang memproduksinya dapat digunakan berbagai cara. Teknik *association rule* merupakan salah satu teknik dalam penambangan data yang dapat digunakan untuk mendapatkan

¹Mahasiswa Pascasarjana, Ilmu Komputer F MIPA Universitas Gadjah Mada, Sekip Utara Bulaksumur Yogyakarta 55281 INDONESIA. Dosen, Program Studi Teknik Informatika Universitas Sanata Dharma Yogyakarta, (telp: 0274-555133; e-mail: ridowati.gunawan@mail.ugm.ac.id).

²Dosen, Jurusan Ilmu Komputer dan Elektronika, F MIPA Universitas Gadjah Mada, Sekip Utara Bulaksumur Yogyakarta 55281 INDONESIA (telp: 0274-555133; email:khabib@ugm.ac.id)

aturan asosiasi antara tanaman obat yang satu dengan yang lainnya dalam membuat ramuan jamu.

Untuk mendapatkan aturan asosiasi, sumber data diperoleh dari berbagai sumber dan dapat dalam bentuk yang kompleks dan heterogen. Salah satunya dapat diperoleh dari *semantic web*. Konten yang terdapat dalam *web* haruslah memiliki kesamaan makna (semantik) antara apa yang dituliskan oleh manusia dan yang dibaca oleh mesin. Data dalam *semantic web* disimpan dalam bentuk *triple* (*subject, predicate, object*) dan diekspresikan dalam OWL dan RDF. Data yang disimpan dalam bentuk ini dapat digunakan untuk mendapatkan pengetahuan yang memiliki makna yang sama pada saat menyebutkan objek yang sama. Data yang diperoleh dalam *semantic web* dapat diolah untuk berbagai kepentingan, salah satunya dalam mencari pengetahuan mengenai jamu. Teknologi penambangan data dapat dikombinasikan dengan teknologi *semantic web*.

Penambahan data dalam *semantic web* dilakukan untuk mendapatkan pengetahuan, khususnya untuk mendapatkan aturan asosiasi dari data yang disimpan dalam bentuk OWL dan RDF.

Makalah ini berfokus pada bagaimana membangun sebuah *ontology* yang menyimpan informasi mengenai jamu dalam bentuk OWL dan RDF. Kemudian *instance* dari *ontology* ditambang dengan teknik *association rule*. Perangkat lunak yang digunakan adalah R yang merupakan perangkat lunak *open source* untuk mengolah data statistik dan grafik. R memiliki *library* yang dapat digunakan untuk membaca data RDF dan selanjutnya data dipilih dengan menggunakan SPARQL. Hasil *query* SPARQL inilah yang dikenakan algoritme Apriori. Algoritme Apriori dipilih karena algoritme ini sesuai dengan kasus dalam pencarian aturan asosiasi dalam jamu karena selain mendapatkan aturan asosiasinya juga dapat dilihat tingkat ketergantungan antara satu tanaman obat dengan tanaman obat yang lainnya, yang diukur dengan *lift ratio*-nya.

II. PENAMBANGAN DATA DALAM SEMANTIC WEB

Pembahasan ini dibagi menjadi tiga bagian. Bagian pertama tentang *semantic web*, OWL dan SPARQL, bagian kedua mengenai penambangan data khususnya *association rule mining*, dan bagian yang ketiga adalah tentang penambangan data pada data dalam *semantic web*.

A. Semantic Web, Ontology Web Language dan SPARQL

Semantic web merupakan perluasan dari *Web* yang ada saat ini, dimana *semantic web* memiliki konten yang dipahami oleh manusia dan mesin. Diperlukan pemahaman yang sama antara mesin dan manusia terhadap sebuah objek tertentu. Sebagai contoh, kata "Apel", apakah apel merupakan konsep buah atau Apel sebagai konsep Merk. Untuk memberikan makna terhadap konsep dan menghilangkan salah arti terhadap konsep maka perlu ditambahkan *metadata* pada setiap konsep yang dibicarakan. Akibat adanya penambahan *metadata* ini, dihasilkan mesin yang cerdas karena tidak memberikan informasi yang salah mengenai suatu konsep

tertentu dan mesin pencarian informasinya pun akan menjadi lebih cepat.

Untuk tetap mempertahankan arti, penyimpanan data disimpan dalam bentuk *triple* yang diekspresikan dalam OWL dan RDF. Pada hasil penyimpanan data dalam bentuk *triple* selanjutnya dapat dilakukan *query* dengan memanfaatkan *query* RDF yang dinamakan dengan SPARQL.

Khusus di bidang pemanfaatan tanaman obat, telah dikembangkan *ontology* dari Indonesia *Ethnomedicine*. Uji coba dilakukan dengan menggunakan dua puluh pertanyaan yang dibangun dengan SPARQL [1]. Selain tanaman dan khasiatnya, kelas yang dibangun melibatkan unsur *etnic*. Pertimbangan memasukan unsur *etnic* adalah keberagaman pulau yang ada di Indonesia dan ketersediaan tanaman obat. Ada tanaman obat yang hanya diperoleh pada daerah tertentu saja. Perbedaan pemberian nama terhadap tanaman obat yang sama dapat pula terjadi. Kelemahan dari *ontology* tersebut adalah tidak dapat menjelaskan jika terdapat ramuan jamu yang terdiri atas beberapa tanaman obat dan memiliki khasiat yang berbeda dengan tanaman obat itu sendiri. Fokus penelitian tersebut hanya pada satu tanaman obat saja beserta dengan khasiat dari tanaman tersebut. Pada bagian taksonomi yang menjelaskan kelas dari tanaman obat, terdapat subkelas yang hanya memiliki satu kelas saja. Hal ini kurang tepat karena pada sebuah kelas yang hanya memiliki sebuah subkelas saja terdapat permasalahan dalam pemodelannya, atau *ontology* yang dibangun belumlah lengkap [2].

Pendekatan *socio-technical* telah dilakukan untuk membangun *ontology* tanaman obat Indonesia [3]. Metodologi yang dikembangkan adalah melakukan diskusi grup (FGD), dan melakukan pembicaraan dengan para pakar di bidang tanaman obat. Sumber data dari *ontology* terdiri atas lima aspek, yaitu dari aspek farmakologi, pelaku kesehatan, taksonomi tanaman, cara penanaman, dan konservasi untuk tanaman obat yang langka. *Ontology* yang dibangun masih berupa tanaman obat yang memiliki khasiat tertentu, belum memperhatikan kumpulan tanaman obat yang digunakan untuk membuat ramuan jamu. Tanaman obat yang digunakan dalam penelitian tersebut merupakan tanaman obat yang telah terdaftar di Badan Pengawas Obat dan Makanan (BPOM).

Manfaat dari pengembangan *ontology* adalah dengan membangun Sistem E-Health. pemakai memasukkan gejala-gejala yang dideritanya dan sistem memberikan rekomendasi tanaman apa yang sesuai. Aturan-aturan dapat dimasukan ke dalam sistem sehingga pertanyaan yang diajukan oleh pemakai dapat dijawab sistem sesuai dengan aturan yang dimiliki oleh sistem. Sistem dibangun untuk tanaman dari Thailand (Thai Herb) [4].

Selain menggunakan *query* SPARQL, pemerolehan informasi berupa aturan-aturan dapat juga diperoleh dengan menggunakan SWRL. Bahan alami, kimia, dan penyakit merupakan dasar dari pembuatan *ontology*. Penyakit yang disebabkan oleh organisme dan aturan mengenai bahan makanan alami yang dapat diberikan untuk mengobati penyakit dapat diperoleh dari *ontology* dan SWRL yang dibangun [5].

Ontology dengan menggunakan OWL dan SWRL dikembangkan juga untuk tanaman obat Indian, yang diberi nama MP-ONTOLOGY. Secara teknis, *ontology* dibangun dengan menggunakan Protege 3.4.4, *query* menggunakan SQWRL, pengecekan inkonsistensi data menggunakan Pellet, dan aturan dibangun dengan SWRL. Tanaman obat merupakan bagian dari kategori obat dan tanaman obat digunakan untuk mengobati penyakit tertentu. Properti dari tanaman itu sendiri adalah warna, rasa, dan ukuran [6].

Penelitian-penelitian terdahulu dalam bidang tanaman obat dengan memanfaatkan teknologi *semantic web* masih berfokus pada satu tanaman obat saja dan belum menggabungkan beberapa tanaman obat. Khasiat tanaman obat yang digunakan untuk membuat ramuan jamu belum dikembangkan dalam bentuk *semantic web*. Yang telah dikembangkan adalah untuk ramuan dari tanaman Thai.

B. Association Rule Mining

Association Rule Mining merupakan salah satu teknik yang ada dalam penambangan data yang bertujuan untuk mendapatkan aturan asosiasi atau relasi antara sekumpulan *item*. Aturan asosiasi dapat diperoleh dari berbagai sumber data, di antaranya berasal dari basisdata transaksional, gudang data, maupun dari tempat penyimpanan informasi yang lainnya. Pada umumnya data yang diolah bersifat homogen.

Studi pertama tentang pencarian aturan asosiasi diperoleh dari *itemset* yang sering muncul secara bersama-sama [7]. Salah satu algoritme yang sering digunakan untuk pencarian aturan asosiasi adalah Apriori [8]. Penting tidaknya sebuah aturan asosiasi dapat diketahui dengan dua parameter, yaitu *support* (nilai penunjang), yaitu persentase kejadian kombinasi *item* atau *support count* jumlah *itemset* yang muncul dalam sekumpulan transaksi, dan *confidence* (nilai kepastian), yaitu kuatnya hubungan antar *item* dalam aturan asosiasi [9]. Analisis asosiasi didefinisikan suatu proses untuk menemukan semua aturan asosiatif yang memenuhi syarat minimum untuk *support* (*minimum support*), dan syarat minimum untuk *confidence* (*minimum confidence*).

Secara umum, aturan asosiasi diperoleh dengan cara sebagai berikut: Misalnya terdapat $I = \{i_1, i_2, i_3, \dots, i_n\}$ yang merupakan sekumpulan *item*, sementara D adalah sekumpulan transaksi, di mana setiap transaksi T memiliki sekumpulan *item* di mana $T \subseteq I$. Setiap transaksi akan memiliki TID (*Transaction Identifier*) yang unik. Setiap transaksi dikatakan mengandung X, kumpulan *item* dalam I, jika $X \subseteq T$. Sebuah aturan asosiasi diformulasi dengan bentuk " $X \rightarrow Y$ ", di mana $X \subseteq I$; $Y \subseteq I$; dan $X \cap Y = \Phi$. Aturan $X \rightarrow Y$ memiliki *support* s dalam transaksi D jika s% atau jumlah s dari transaksi dalam D mengandung $X \cup Y$. Atau dengan kata lain, *support* dari suatu aturan adalah probabilitas kejadian X dan Y secara bersama-sama atau jumlah kejadian X dan Y secara bersama. Aturan $X \rightarrow Y$ memiliki nilai *confidence* c jika c% dari transaksi D mengandung X juga mengandung Y. Atau dengan kata lain, *confidence* dari sebuah aturan adalah probabilitas kondisional dengan konsekuen Y adalah benar, jika X merupakan antisedennya. *Support* adalah probabilitas dari

item atau sekumpulan *item* dalam sebuah basisdata transaksional seperti pada (1).

$$\text{Support}(X) = \frac{n(X)}{n} \quad (1)$$

di mana n adalah jumlah total transaksi dalam basisdata, sedangkan n(x) adalah jumlah transaksi yang mengandung *itemset* X, atau *support count* yaitu jumlah *item* yang terdapat dalam transaksi. *Confidence* adalah probabilitas kondisional, untuk aturan asosiasi $X \rightarrow Y$ didefinisikan seperti pada (2).

$$\text{Confidence}(x \rightarrow y) = \frac{\text{Support}(x \cup y)}{\text{Support}(x)} \quad (2)$$

Untuk mengukur tingkat akurasi dari sebuah aturan yang diperoleh, digunakan rumusan *Lift Ratio*. *Lift Ratio* dari aturan $X \rightarrow Y$ didefinisikan seperti pada (3).

$$\text{Lift Ratio}(x \rightarrow y) = \frac{\text{confidence}(x \rightarrow y)}{\text{expected confidence}(x \rightarrow y)} \quad (3)$$

di mana *Expected Confidence* didefinisikan seperti pada (4).

$$\text{Expected Confidence}(x \rightarrow y) = \frac{\text{Support}(y)}{\text{Support}(x \cup y)} \quad (4)$$

Jika nilai *lift ratio* > 1 , aturan $X \rightarrow Y$ muncul lebih sering dari yang diharapkan, dan X dan Y tidak saling bergantung (*dependent*). Jika *lift ratio* = 1, maka aturan $X \rightarrow Y$ muncul sesuai dengan yang diharapkan. Tetapi jika *lift ratio* < 1 , maka aturan $X \rightarrow Y$ muncul lebih jarang dari yang diharapkan dan X, Y saling bergantung (*independent*).

Terdapat banyak algoritme yang digunakan untuk mendapatkan *itemset* yang sering muncul selain menggunakan algoritme Apriori, salah satunya adalah dengan menggunakan FP-Growth [10]. Variasi dari algoritme Apriori dan FP-Growth juga sudah banyak dikembangkan, akan tetapi hasil dari aturan asosiasi yang dihasilkan tidak jauh berbeda antara algoritme yang satu dengan yang lainnya [11].

Penerapan algoritme Apriori telah banyak dilakukan untuk berbagai bidang, di antaranya untuk melakukan analisis keranjang belanja. Salah satu penerapan dalam bidang *medicine*, algoritme Apriori digunakan dalam mencari aturan asosiasi pada *oriental medicine* yaitu obat tradisional Korea (*Traditional Korean Medicine*) [12], di mana basisdatanya diekstrak dari Bangyakhappyeon. Teknik *association rule* digunakan untuk menunjukkan relasi antara gejala (*symptoms*) dan material herbalnya. Nilai *minimum confidence* yang digunakan adalah 20%. Nilai *confidence* dan *lift ratio* digunakan untuk melakukan justifikasi kuatnya aturan yang terbentuk. *Radar Chart* dan NetMiner digunakan untuk membuat analisis *network* dari aturan asosiasi yang terbentuk.

Analisis terhadap basisdata mengenai jamu untuk mendapatkan korelasi antara tanaman, jamu, dan korelasi antara tanaman, jamu, dan khasiat tentang jamu dilakukan dengan menggunakan model statistik [13][14]. Metode yang digunakan adalah Biplot, *Partial Least Square (PLS)* dan metode *bootstrapping* untuk membuat kesimpulan dan fokus pada prediksi untuk membuat formula jamu.

Selain itu, pengembangan studi berbasis jaringan (*network*) dilakukan untuk memprediksi relasi antara penyakit dan tanaman obat [15]. *Network* jamu dibangun berdasarkan kesamaan dari bahan-bahan yang digunakan dan kemudian dilakukan *clustering* terhadap jamu dengan menggunakan algoritme *clustering network*, yaitu DPCLUSO. Selanjutnya relasi tanaman-penyakit digunakan untuk melakukan prediksi tanaman yang tepat untuk mengobati penyakit sesuai dengan *cluster* jamu yang yang dipilih.

C. Association Rule dalam Semantic Web

Selain digunakan untuk mendapatkan aturan asosiasi langsung dari basisdata, algoritme asosiasi juga dapat diterapkan pada data dalam *semantic web*. Pencarian aturan asosiasi dalam data *semantic web* diperoleh dari data yang disimpan dalam bentuk RDF dan OWL. Pemakai memberikan *query* dalam bentuk SPARQL yang diperluas, kemudian dilakukan proses ekstraksi terhadap hasil yang diperoleh. Data hasil ekstraksi tersebut diproses kembali untuk menghasilkan aturan asosiasi dengan menggunakan teknik *Association Rule Mining* dengan nilai *support* dan *confidence* yang dimasukkan oleh pemakai [16]. Pemanfaatan SPARQL yang diperluas menggunakan *syntax* yang diturunkan dari DMX (*Microsoft Data Mining Extension*). *Grammar* dari SPARQL juga sedikit diperluas yaitu dengan menambahkan kata kunci CREATE MINING MODEL. Contoh yang diberikan adalah menganalisis pasien sebagai target, kemudian *feature* yang dianalisis adalah penyakit dan obat yang diresepkan. Hasil dari *query* menjadi data masukan untuk proses pencarian aturan asosiasi.

SWApriori merupakan salah satu pendekatan yang berbeda untuk mendapatkan aturan asosiasi dari data *semantic web* [17]. Proses pencarian aturan asosiasi langsung dilakukan dari *data set* dalam *semantic web* pada level *instance*, dengan persyaratan data telah disimpan dalam bentuk *triple*. Kelebihan dari metode ini adalah tidak perlu melibatkan pemakai akhir, serta tidak perlu dilakukan proses konversi ke dalam bentuk yang tradisional, yaitu dipindahkan ke dalam penyimpanan tunggal. Selain itu, dengan metode ini tidak perlu lagi diterapkan algoritme pencarian aturan asosiasi yang tradisional seperti Apriori. Uji yang dilakukan adalah pada *data set* obat-obatan. Walaupun tidak melibatkan pemakai secara langsung, kelemahan dari metode ini adalah perlunya ketelitian dalam membangun *ontology* untuk mendapatkan aturan-aturan yang diinginkan. Karena tidak adanya campur tangan pemakai, jumlah data yang banyak juga menjadi kendala tersendiri karena aturan asosiasi menjadi tidak terbentuk atau dengan kata lain tidak ada yang menarik dari data dalam data *semantic web*. Hal ini terlihat dari nilai *support* yang dihasilkan dalam melakukan uji coba metode ini.

Penggabungan antara kekuatan dari *association rule* dan *semantic web* dilakukan dengan cara menghasilkan terlebih dahulu semantik transaksi dengan menggunakan SPARQL yang diperluas. Kemudian tahap selanjutnya adalah mengimplementasikan algoritme Apriori dari hasil transaksi semantik yang diperoleh sebelumnya. Fokus dari *semantic association rule* ini adalah untuk melakukan transformasi dan membuat semantik data yang kompleks menjadi lebih

seederhana dengan membuat hirarki kelas sesuai dengan aturan asosiasi yang diperoleh [18]. Akan tetapi karena sangat tergantung pada aturan yang diperoleh, maka model kelas hirarkinya sangatlah kaku dan sulit untuk dikembangkan.

Extended SPARQL digunakan untuk mencari asosiasi semantik, diberi nama dengan SPARQLeR [19]. SPARQLeR menambahkan kemampuan SPARQL sehingga dapat digunakan untuk mencari asosiasi semantik antara entitas yang ada dalam basis pengetahuan RDF. Penambahan kemampuan SPARQL dilakukan seminimal mungkin, dengan membuat *pattern path* yang menghubungkan *path* secara langsung maupun tidak langsung. Dibuat tipe *path* yang merupakan urutan dari data RDF yang dibangun. Uji coba berhasil dilakukan pada DBLP menggunakan domain *Ontology Glycomics* dengan nama *ontology GlyCO*.

Semantic data mining secara garis besar dilakukan dengan dua tahapan, yaitu bagaimana menghasilkan transaksi semantik dan melakukan algoritme semantik asosiasi ke dalamnya. Tahapan kedua adalah mengimplementasikannya menggunakan algoritme Apriori. Uji coba dilakukan terhadap *domain ontology*, dari FUM-LD, untuk mendapatkan relasi antara profesor beserta grupnya serta fasilitas yang digunakannya [20].

Integrasi untuk mengekstrak aturan asosiasi berdasarkan batasan tertentu dari data multilevel dengan dukungan *ontology* telah dilakukan [21]. Skema *ontology* dibuat berdasarkan pemetaan tabel dalam struktur basisdata. Pemetaan *multilevel* dari basisdata transaksi dipetakan dalam bentuk *ontology* dengan konsep subkelas. Percobaan dilakukan dengan menggunakan SeRQL (*Sesame RDF Query Language*), yaitu sebuah bahasa untuk melakukan *query* dalam *ontology* di mana hasil dari *query* dapat disimpan dalam penyimpanan sesuai dengan struktur *ontology*-nya, dan selanjutnya menggunakan algoritme Apriori untuk mendapatkan aturan asosiasinya.

Format data RDF/OWL diimpor dengan menggunakan *library rrd* yang ada dalam bahasa pemrograman R dan selanjutnya proses penambangan data dapat dilakukan. Teknik penambangan data yang diterapkan adalah menggunakan klasifikasi, dan selanjutnya dilakukan proses prediksi terhadap model klasifikasi yang dilakukan [22]. Kelemahan dari penelitian ini adalah penggunaan *dataset Iris* yang langsung diubah ke dalam bentuk RDF/OWL, tidak membangun sebuah *ontology* yang sesuai dengan *dataset Iris*.

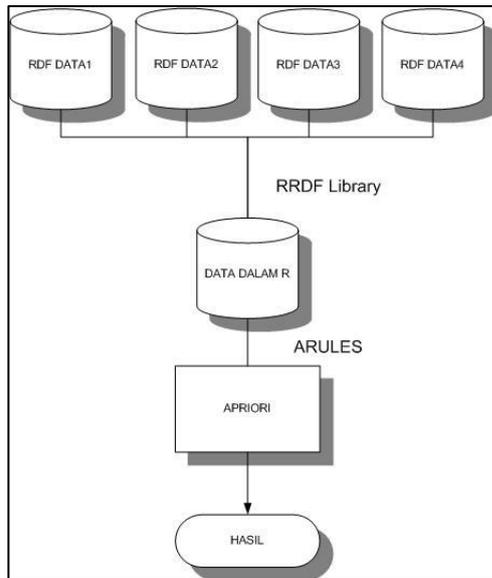
III. METODOLOGI

Metode yang diusulkan untuk melakukan pencarian aturan asosiasi dari *semantic web* adalah dengan membangun terlebih dahulu sebuah *ontology* Jamu. Domain jamu dipilih karena dari beberapa penelitian terdahulu, domain ini sangatlah jarang ditemui dan penelitian terdahulu fokus pada khasiat tanaman obat, bukan pada ramuan jamu.

Ontology dengan serialisasi RDF/OWL diimpor dan diambil data transaksinya dengan menggunakan SPARQL. Selanjutnya dilakukan proses penambangan data. Proses pembuatan *ontology* dilakukan dengan menggunakan Protege 5.00 beta 20, sedangkan proses konversi data dari RDF ke

dalam data transaksi yang siap ditambang dan algoritme Apriori yang dikenakan pada transaksi dilakukan menggunakan perangkat lunak R, dengan *library* utama *rrdf* [23] dan *arules* [24].

Aliran data dari *ontology* jamu dapat dilihat pada Gbr. 1.



Gbr. 1 Aliran data dan proses sistem.

Tahapan detail yang dilakukan untuk proses pencarian aturan asosiasi dari *Ontology* jamu adalah sebagai berikut:

1. Membangun *ontology* jamu.
2. Mengimpor *data set* dari *ontology* Jamu yang dibuat dari bentuk RDF sebagai data *frame* dengan menggunakan *library* *rrdf* yang ada dalam bahasa R.
3. Mengubah bentuk format data ke dalam bentuk *data set* yang dapat dimengerti oleh algoritme asosiasi dalam bahasa R.
4. Membaca data transaksi yang telah diubah dan menggunakan algoritme Apriori dengan menggunakan *library* *arules* yang terdapat dalam bahasa R.
5. Melakukan percobaan dengan mengubah parameter-parameter masukan, yaitu nilai *minimum support* dan *minimum confidence*, yang sesuai.
6. Melakukan proses *prunning* agar aturan yang diperoleh lebih ringkas.
7. Melakukan analisis terhadap aturan asosiasi yang diperoleh.

IV. HASIL DAN PEMBAHASAN

Pada bagian ini akan dibahas mengenai cara membangun *ontology jamu* dan mengekstrak data sampai diperoleh aturan asosiasi mengenai jamu.

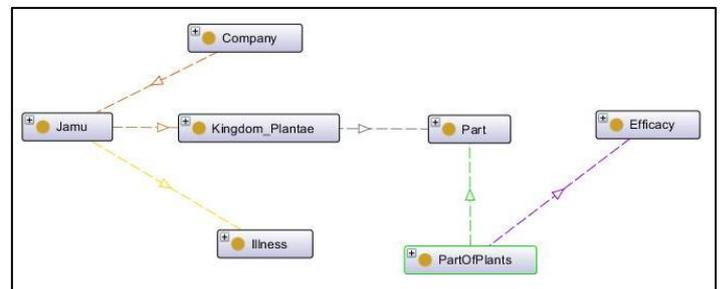
A. *Ontology* Jamu

Ontology yang dibangun mencakup pengetahuan tentang jamu yang beredar di Indonesia. Formula jamu yang digunakan dikumpulkan dari KNApSack Family *Databases* khusus pada bagian Jamu yang sampai dengan *update* terakhir 30 Maret 2015 memiliki 1133 jenis tanaman dan 5310

formula jamu [25]. Data juga dikombinasikan dengan informasi mengenai produk obat tradisional yang telah mendapatkan nomor registrasi dari BPOM Indonesia yang jumlahnya telah mencapai 9145 produk [26], dan informasi mengenai obat bahan alami Indonesia yang diinformasikan oleh BPOM Indonesia [27]. Sedangkan taksonomi tanaman obat yang digunakan merupakan spesies dari Kingdom *Plantae*. Taksonomi diperoleh pada Maret 2016 dari basisdata secara *online* dari Integrated Taxonomic Information System (ITIS) pada alamat <http://www.itis.gov>. *Ontology* dibangun berdasarkan beberapa konsep yaitu:

1. Taksonomi dari spesies tanaman obat.
2. Bagian yang digunakan dalam tanaman obat untuk membuat jamu..
3. Perusahaan pembuat jamu.
4. Khasiat dari tanaman obat.
5. Penyakit yang dapat disembuhkan oleh jamu.
6. Jamu sebagai obat tradisional yang dibuat oleh perusahaan jamu memiliki khasiat untuk mengobati penyakit atau gejala tertentu dan terdiri atas beberapa tanaman obat.

Dari keenam konsep tersebut, diturunkan beberapa buah kelas yaitu kelas *Kingdom_Plantae*, kelas *Company*, kelas *TypeOfJamu*, kelas *Efficacy*, kelas *PartOfPlants* dan kelas *Illness*. *Kingdom_Plantae* memiliki subkelas seperti pada taksonomi *Plantae* pada basisdata ITIS. Karena kompleks dan luas, maka hirarki subkelas untuk *Kingdom_Plantae* tidak digambarkan secara detail. Hirarki kelas dari *ontology* yang dibangun dapat dilihat pada Gbr. 2.



Gbr. 2 Hirarki kelas jamu.

1) *Kelas Kingdom_Plantae*: Memiliki subkelas seperti taksonomi pada basisdata ITIS. Salah satu subkelas dari *Kingdom_Plantae* adalah *Subkingdom_viridiplantae* atau dapat ditulis dengan notasi $\text{Subkingdom_viridiplantae} \subset \text{Kingdom_Plantae}$. Setiap *species* atau tanaman obat yang digunakan untuk membuat jamu akan menjadi *instance* dari sebuah Genus tertentu. Misalnya untuk *Oryza Sativa Amylum* (Padi) merupakan *instance* dari Genus *Oryza*. Kelas beserta dengan subkelas untuk mendapatkan *instance* dari *Oryza Sativa Amylum* adalah: (\subset menunjukkan subkelas) $\text{Genus_Oryza} \subset \text{Family_Poacea} \subset \text{Order_Poales} \subset \text{Superorder_Lilianae} \subset \text{Class_Magnoliopsida} \subset$

Subdivision_Spermatophytina ⊂
 Division_Tracheophyta ⊂
 Subkingdom_viridiplantae ⊂ Kingdom_Plantae.

Kelas Kingdom_Plantae dapat dikembangkan untuk Kingdom yang lain seperti Kingdom Animale, Kingdom Fungi, dan yang lainnya. Taksonomi yang dibangun sangat fleksibel dan dapat digunakan untuk berbagai kepentingan. Selain itu, model *ontology* memenuhi ketentuan dalam pembuatan kelas *ontology*, di mana jumlah dari subkelas dari setiap kelas harus lebih dari satu. Jika tidak, maka *ontology*-nya kurang lengkap atau terdapat permasalahan dalam mengembangkannya.

2) *Kelas Company*: Merupakan kelas yang menyimpan informasi mengenai perusahaan yang memproduksi jamu.

3) *Kelas Illness*: Merupakan kelas yang menyimpan informasi mengenai jenis-jenis penyakit yang dapat diobati dengan menggunakan jamu.

4) *Kelas Part*: Kelas ini berisi informasi mengenai bagian-bagian dari Kingdom_Plantae. Yang termasuk di dalamnya antara lain root, leaf, rhizoma.

5) *Kelas PartOfPlants*: Merupakan kelas gabungan antara kelas Part dan kelas Kingdom_Plantae. Kelas partOfPlants akan memiliki relasi dengan kelas Efficacy.

6) *Kelas Efficacy*: Merupakan kelas yang berisi informasi mengenai khasiat yang dimiliki oleh *instance* dari kelas PartOfPlants.

Object property dari *ontology* jamu ini dapat dilihat pada Tabel I, sedangkan *data property* dapat dilihat pada Tabel II.

TABEL I
 OBJECT PROPERTY ONTOLOGY JAMU

Object Property	Domain Type	Range Type	Keterangan
composedOf	Jamu	Kingdom_Plantae	
efficacyOf	Efficacy	PartOfPlants	Khasiat dari bagian tanaman Inverse dari : hasEfficacy (RDF property menggunakan Inverse of
fromPlant	PartOfPlants	Part	
hasPart	Kingdom_Plantae	PartOfPlants	
hasEfficacy	PartOfPlants	Efficacy	Inverse dari : efficacyOf
produce	Company	Jamu	Inverse dari : producedBy
producedBy	Jamu	Company	Inverse dari : produce
usedFor	Jamu	Illness	Jamu digunakan untuk mengobati penyakit tertentu

TABEL II
 DATA PROPERTY JAMU

Data Property	Domain Type	Data Type	Keterangan
plantPopularName	Kingdom_Plantae	String	Nama tanaman obat dengan berbagai bahasa (en,id,cn)
plantScientificName	Kingdom_Plantae	String	Nama ilmiah dari tanaman obat

Sebuah objek jamu terdiri atas beberapa Kingdom_Plantae (composedOf), diproduksi oleh sebuah perusahaan (producedBy), dan dapat digunakan untuk mengobati penyakit tertentu (usedFor). Sifat dari kelas Kingdom_Plantae adalah transitif, sehingga dengan menyebutkan *instance* dari setiap Genus akan diperoleh *superclassnya*. Jamu tidak dapat hanya terdiri atas beberapa Kingdom_Plantae saja, tetapi harus memasukan nama Company dan Illness nya. Yang unik dari jamu ini adalah setiap Company akan memproduksi jamu dengan komposisi yang berbeda-beda walaupun memiliki kesamaan fungsi dalam mengobati penyakit tertentu.

Yang menarik dari *ontology* yang dibangun adalah yang berkaitan dengan khasiat dari tanaman obat (Kingdom_Plantae). Setiap bagian dari tanaman obat akan memiliki khasiat yang berbeda-beda. Bagian tanaman rhizome untuk Ginger dan untuk TemuLawak akan memiliki khasiat yang berbeda. Demikian juga untuk leafOfWaron memiliki khasiat yang berbeda dengan seadOfWaron.

Fokus pencarian aturan asosiasi adalah untuk mendapatkan komposisi tanaman obat yang digunakan untuk membuat jamu. Sebagai contoh, jamu yang dicari aturan asosiasinya adalah jamu yang memiliki khasiat mengurangi kelelahan. SPARQL yang dibuat untuk menghasilkan data transaksi sesuai yang diinginkan sebagai berikut:

```

PREFIX jamu:
<http://localhost:8080.owl/herbs.owl#>
SELECT ?Jamu ?Kingdom_Plantae ?x
WHERE {?Jamu
jamu:composedOf ?Kingdom_Plantae.
?Plants jamu:plantPopularName ?x
filter(lang(?x)='id') .
?Jamu jamu:usedFor jamu:Fatigue.
} orderBy ?Jamu
    
```

SPARQL tersebut menampilkan kelas Jamu, kelas Kingdom_Plantae, dan variabel x yang mewakili plantPopularName dengan language 'id'. Jamu yang ditampilkan adalah jamu untuk mengobati (usedFor) kelelahan (fatigue), dan ditampilkan berdasarkan urutan dari nama jamu. Filter dapat dilakukan juga untuk language yang lain seperti 'en', ataupun 'cn'.

B. Proses Transfer Data RDF ke Data Transaksi

Langkah-langkah yang dilakukan untuk mentransfer data RDF dengan menggunakan R adalah sebagai berikut.

1. Memanggil *library rrd* untuk melakukan konversi dari data RDF ke dalam data Frame, dengan menggunakan perintah:

```
library(rrdf)
```

2. Menggunakan *library rrd* untuk memanggil *file* RDF yang akan ditransfer datanya, dengan perintah:

```
rdfdata ←
load.rdf("indonesiaHerbs.owl", format=
"RDF/XML")
```

Fungsi *load.rdf* digunakan untuk memanggil *file* *rdf* yang digunakan, dengan parameter nama *file* *rdf* dan format dari *ontology* yang dibangun.

3. Mengkonversi data *file* RDF ke dalam data frame. Yang dilakukan adalah memanggil terlebih dahulu data dengan menggunakan fungsi *sparql*, selanjutnya dilakukan penyimpanan ke dalam data frame:

```
dataJamuIndonesia ←
data.frame(sparql.rdf(rdfdata,
"PREFIX jamu:
<http://localhost:8080/herbs.owl#>
SELECT ?Jamu ?x
WHERE{?Jamu
jamu:composedOf ?Kingdom_Plantae.
?Jamu jamu:usedFor jamu:Fatigue.
?Kingdom_Plantae
jamu:plantPopularName ?x
filter(lang(?x)='id')
}
orderBy ?Jamu"))
```

Sparql.rdf digunakan untuk menjalankan *query* dalam bentuk *sparql*. Hasil dari *query* disimpan dalam *data frame*. *Query* *sparql* tersebut digunakan untuk mendapatkan data kelas *Kingdom_Plantae* dengan kriteria data setiap jamu yang *usedFor* *Fatigue*. *Data property* yang ditampilkan dari kelas *Kingdom_Plantae* adalah *plantPopularName*.

4. Mengubah *data frame* ke dalam data transaksi. Hal ini dilakukan karena format data yang dibaca oleh *library* *arules* adalah data transaksi. Untuk itu dilakukan perubahan format *data frame* ke dalam data transaksi.

```
dataJamuTransaksi ←
as(split(dataJamuIndonesia[, "x"],
dataJamuIndonesia[, "Jamu"]),
"transactions")
```

Terdapat 89 data untuk hasil proses 3. Dari langkah 4, diperoleh 20 data transaksi dengan *transactionID* adalah jamu beserta dengan *Kingdom_Plantae* yang digunakan untuk membuat jamu. Lima *transactionID* pertama dapat dilihat pada Gbr. 3.

C. Pembuatan Aturan Asosiasi

Setelah proses transformasi data dari RDF menjadi data transaksi dilakukan, langkah selanjutnya adalah melakukan

proses penambangan data dengan menggunakan algoritme asosiasi. Dalam *library* *arules* terdapat dua algoritme yang disediakan, yaitu *Apriori* dan *Eclat*. Berikut ini adalah skenario dan hasil pencarian aturan asosiasi.

1) *Skenario 1*: Mendapatkan seluruh aturan asosiasi, dengan nilai *minimum support* dan *minimum confidence* mengikuti nilai *default* untuk *minimum support* adalah 2 transaksi dari seluruh data transaksi yang ada atau sama dengan 0.1. Perintah yang diberikan adalah:

```
hasilapriori ← apriori(dataJamuTransaksi)
```

Dari data yang dimiliki, dengan menggunakan nilai *default* dihasilkan 84 aturan. Sepuluh aturan yang diperoleh dapat dilihat pada Gbr. 4. Aturan ke-10 dari Gbr. 4 menyatakan bahwa komposisi {kencur, lempuyang wangi} → {Cabe Jawa} memiliki *support* 0.1, *confidence* 1, dan *lift ratio* 2.5. Aturan ini berarti terdapat 10% dari seluruh komposisi jamu menggunakan {kencur, lempuyang wangi} dan {Cabe Jawa} secara bersama-sama.

```
> inspect(dataJamuTransaksi[1:5])
items transactionID
1 {Cabe Jawa,
Jahe,
Lada Hitam,
Lempuyang Wangi,
Lengkuas} jamu:Dami_Jamu_Pegel_Linu
2 {Jahe,
Jungrahab,
Kayu Putih,
Kedawung,
Pulasari} jamu:Idhi_Jamu_Pegel_Linu
3 {Jahe,
Lengkuas,
Pala,
Pulasari} jamu:IKOT_Jamu_Pegel_Linu
4 {Ginseng Asia,
Ginseng Putri,
Mengkudu} jamu:Indorica_Jamu_Pegel_Linu
5 {Alang-Alang,
Cabe merah,
Greges Otot,
Jahe,
Kayu putih,
Kemukus,
Lempuyang,
Lengkuas,
Pulasari} jamu:Kates_Jamu_Pegel_Linu
```

Gbr. 3 Hasil data transactions.

```
> inspect(hasilapriori[1:10])
lhs rhs support confidence lift
1 {Alang-alang} => {Jahe} 0.10 1 2.000000
2 {Kapulaga} => {Cabe Jawa} 0.15 1 2.500000
3 {Lempuyang Wangi} => {Cabe Jawa} 0.20 1 2.500000
4 {Kemukus} => {Lengkuas} 0.10 1 3.333333
5 {Kemukus} => {Pulasari} 0.10 1 2.857143
6 {Kemukus} => {Jahe} 0.10 1 2.000000
7 {Ketumbar} => {Cabe Jawa} 0.15 1 2.500000
8 {Ketumbar} => {Temu Lawak} 0.15 1 2.500000
9 {Kencur} => {Temu Lawak} 0.20 1 2.500000
10 {Kencur, Lempuyang Wangi} => {Cabe Jawa} 0.10 1 2.500000
```

Gbr. 4 Hasil aturan asosiasi.

2) *Skenario 2*: Dari aturan yang diperoleh pada skenario 1, dilakukan proses *prunning* agar tidak terjadi reduksi aturan yang diperoleh. Setelah proses *prunning* dilakukan, diperoleh dari 84 aturan menjadi 22 aturan. Proses *prunning* beserta hasil dapat dilihat pada Gbr. 5. Aturan dengan *support*

tertinggi 0.20 adalah Lempuyang Wangi → Cabe Jawa dan Kencur → Temu Lawak.

```
> subset.matrix <- is.subset(hasilapriori1, hasilapriori1)
> subset.matrix[lower.tri(subset.matrix, diag=T)] <- NA
> redundant <- colSums(subset.matrix, na.rm=T) >= 1
> hasilapriori1.pruned <- hasilapriori1[!redundant]
> hasilapriori1 <- hasilapriori1.pruned
> inspect(hasilapriori1)
  lhs                rhs      support confidence lift
1 {Alang-alang}      => {Jahe} 0.10      1      2.000000
2 {Kapulaga}         => {Cabe Jawa} 0.15      1      2.500000
3 {Lempuyang Wangi} => {Cabe Jawa} 0.20      1      2.500000
4 {Kemukus}          => {Lengkuas} 0.10      1      3.333333
5 {Kemukus}          => {Pulasari} 0.10      1      2.857143
6 {Kemukus}          => {Jahe} 0.10      1      2.000000
7 {Ketumbar}         => {Cabe Jawa} 0.15      1      2.500000
8 {Ketumbar}         => {Temu Lawak} 0.15      1      2.500000
9 {Kencur}           => {Temu Lawak} 0.20      1      2.500000
25 {Adas, Lengkuas} => {Kencur} 0.10      1      5.000000
27 {Adas, Cabe Jawa} => {Kencur} 0.10      1      5.000000
30 {Adas, Jahe}      => {Kencur} 0.10      1      5.000000
32 {Adas, Kayu Manis} => {Pulasari} 0.10      1      2.857143
34 {Adas, Lengkuas} => {Temu Lawak} 0.10      1      2.500000
35 {Adas, Cabe Jawa} => {Temu Lawak} 0.10      1      2.500000
36 {Adas, Cabe Jawa} => {Jahe} 0.10      1      2.000000
38 {Adas, Jahe}      => {Temu Lawak} 0.10      1      2.500000
41 {Jahe, Kencur}    => {Cabe Jawa} 0.10      1      2.500000
43 {Cabe Jawa, Kayu Manis} => {Pulasari} 0.10      1      2.857143
45 {Pulasari, Temu Lawak} => {Kayu Manis} 0.10      1      3.333333
46 {Lengkuas, Pulasari} => {Jahe} 0.15      1      2.000000
47 {Pulasari, Temu Lawak} => {Jahe} 0.10      1      2.000000
```

Gbr. 5 Aturan asosiasi hasil *prunning*.

3) *Skenario 3*: Nilai *minimum support* dan nilai *confidence* dapat ditentukan oleh pemakai. Sebagai contoh, nilai *minimum support* yang ditetapkan adalah 0.15. Perintahnya adalah:

```
Hasilapriori3 <- apriori(dataJamuTransaksi,
parameter=list(supp=0.15, conf=1))
```

Hasil yang diperoleh dapat dilihat pada Gbr. 6.

```
> inspect(hasilapriori3)
  lhs                rhs      support confidence lift
1 {Kapulaga}         => {Cabe Jawa} 0.15      1      2.5
2 {Lempuyang Wangi} => {Cabe Jawa} 0.20      1      2.5
3 {Ketumbar}         => {Cabe Jawa} 0.15      1      2.5
4 {Ketumbar}         => {Temu Lawak} 0.15      1      2.5
5 {Kencur}           => {Temu Lawak} 0.20      1      2.5
6 {Cabe Jawa, Ketumbar} => {Temu Lawak} 0.15      1      2.5
7 {Ketumbar, Temu Lawak} => {Cabe Jawa} 0.15      1      2.5
8 {Adas, Kencur}     => {Temu Lawak} 0.15      1      2.5
9 {Adas, Temu Lawak} => {Kencur} 0.15      1      5.0
10 {Cabe Jawa, Kencur} => {Temu Lawak} 0.15      1      2.5
11 {Lengkuas, Pulasari} => {Jahe} 0.15      1      2.0
```

Gbr. 6 Aturan asosiasi dengan *Minimum Support* = 0.15.

4) *Skenario 4*: Mencari aturan di mana konsekuennya (rhs = *right hand side*) adalah Jahe. Skenario ini dapat digunakan jika diketahui komposisi tanaman obat yang menggunakan Jahe. Nilai *minimum support* yang ditetapkan adalah 0.10.

```
Hasilapriori3 <-
apriori(dataJamuTransaksi,
parameter=list(supp=0.10, conf=1),
appearance=list(default="lhs", rhs="Jahe"))
```

Hasil yang diperoleh dapat dilihat pada Gbr. 7.

Keempat skenario menunjukkan bahwa dari *ontology* dapat diperoleh aturan-aturan sesuai dengan skenario yang diinginkan, mulai dari hasil aturan asosiasi secara keseluruhan, aturan asosiasi hasil *prunning*, aturan asosiasi dengan nilai *support* tertentu, dan juga dapat diperoleh aturan asosiasi menggunakan tanaman obat tertentu.

```
> inspect(hasilapriori4)
  lhs                rhs      support confidence lift
1 {Alang-alang}      => {Jahe} 0.10      1      2
2 {Kemukus}          => {Jahe} 0.10      1      2
3 {Kemukus, Lengkuas} => {Jahe} 0.10      1      2
4 {Kemukus, Pulasari} => {Jahe} 0.10      1      2
5 {Adas, Cabe Jawa} => {Jahe} 0.10      1      2
6 {Lengkuas, Pulasari} => {Jahe} 0.15      1      2
7 {Pulasari, Temu Lawak} => {Jahe} 0.10      1      2
8 {Kemukus, Lengkuas, Pulasari} => {Jahe} 0.10      1      2
9 {Adas, Cabe Jawa, Kencur} => {Jahe} 0.10      1      2
10 {Adas, Cabe Jawa, Temu Lawak} => {Jahe} 0.10      1      2
11 {Kayu Manis, Pulasari, Temu Lawak} => {Jahe} 0.10      1      2
12 {Adas, Cabe Jawa, Kencur, Temu Lawak} => {Jahe} 0.10      1      2
```

Gbr. 7 Aturan asosiasi dengan konsekuensi Jahe.

Gbr 4 sampai dengan Gbr. 7 menunjukkan hasil asosiasi dengan nilai *support*, *confidence*, dan *lift ratio*. Dari keempat skenario, nilai *support* yang dihasilkan masih sangat rendah, yaitu tidak lebih dari 0.2 atau 20% saja. Artinya tanaman obat yang digunakan secara bersama dalam membuat jamu dengan manfaat yang sama masih sangat bervariasi. Tidak ada tanaman obat yang dominan untuk membuat ramuan jamu.

Walaupun nilai *support* rendah, nilai *confidence* dan *lift* cukup tinggi. Nilai *confidence* yang diperoleh adalah 1, artinya aturan memiliki tingkat kebenaran yang cukup tinggi. Sedangkan *lift ratio* sebagai ukuran akurasi menunjukkan nilai lebih dari 1, yang artinya aturan lebih sering muncul dan menunjukkan hubungan antar satu tanaman obat dengan lainnya tidak saling bergantung.

V. KESIMPULAN

Pencarian aturan asosiasi dengan sumber data berasal dari *semantic web* dengan serialisasi RDF/OWL berhasil diperoleh. Tanaman obat yang digunakan untuk membuat jamu yang bermanfaat untuk mengobati kelelahan (*Fatigue*) telah diperoleh. Aturan asosiasi diperoleh dengan menggabungkan kemampuan dari SPARQL pada *ontology* yang dibangun dengan algoritme dari R.

Akurasi dan tingkat kebenaran yang diperoleh dari hasil percobaan cukup tinggi, akan tetapi nilai *support*-nya masih rendah. Hal ini menunjukkan bahwa *ontology* yang dibangun telah dapat menghasilkan aturan-aturan asosiasi yang dapat dipercaya kebenarannya. Untuk tahap selanjutnya dapat dicoba dengan algoritme yang lain untuk menghasilkan nilai *support* yang lebih tinggi.

Penelitian selanjutnya dapat dilakukan dengan menambah *instance* dari setiap kelas yang telah dibangun sehingga menjadi pengetahuan tentang jamu yang lengkap. *Ontology* yang dibangun dapat dikembangkan lagi dengan menambahkan kelas baru, seperti menambahkan jumlah komposisi dan unsur-unsur yang ada dalam setiap *Kingdom Plantae*. Penambahan pengetahuan baru memiliki peluang untuk mendapatkan tanaman obat baru yang dapat digunakan untuk membuat jamu. Untuk proses penambahan data dapat pula dikombinasikan dengan menggunakan teknik klasifikasi dan prediksi.

REFERENSI

[1] D. W. Wardani, S. H. Yustianti, U. Salamah, and O. P. Astirin, "An Ontology of Indonesian Ethnomedicine," in *International Conference on Information, Communication Technology and System*, 2014, pp. 47–52.

- [2] N. F. Noy and D. L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology," Stanford Knowl. Syst. Lab., p. 25, 2001.
- [3] M. Silalahi, D. E. Cahyani, D. I. Sensuse, and I. Budi, "Developing Indonesian Medicinal Plant Ontology Using Socio-Technical Approach," 2015, no. 14ct, pp. 39–43.
- [4] T. Kato, N. Maneerat, R. Varakulsiripunth, F. Engineering, and K. Mongkut, "Ontology-based E-health System with Thai Herb Recommendation 1", Sendai National College of Technology, Sendai, Japan, vol. 1, 2009.
- [5] V. Ganesan, S. Waheeta Hopper, and G. BharatRam, "Semantic Data Integration and Querying Using SWRL," in *Trends in Network and Communications*, vol. 197, D. Wyld, DavidC. and Wozniak, Michal and Chaki, Nabendu and Meghanathan, Natarajan and Nagamalai, Ed. Springer Berlin Heidelberg, 2011, pp. 567–574.
- [6] R. Mohan. and G. Arumugam, "Developing Indian medicinal plant ontology using OWL and SWRL," in *Second International Conference, ICDEM 2010*, 2012, vol. 6411 LNCS, pp. 131–138.
- [7] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," *ACM SIGMOD Rec.*, vol. 22, no. 2, pp. 207–216, 1993.
- [8] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proceeding VLDB '94 Proceedings of the 20th International Conference on Very Large Data Bases*, 1994, vol. 1215, pp. 487–499.
- [9] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, Third edit. Waltham, MA, USA: Morgan Kaufmann, 2012.
- [10] J. Han, J. Pei, and Y. Yin, "Mining Frequent Pattern without Candidate Generation," in *In: Proc. Conf. on the Management of Data (SIGMOD'00, Dallas, TX)*, 2010, pp. 1–12.
- [11] D. Hunyadi, "Performance comparison of apriori and FP-growth algorithms in generating association rules," in *ECC'11 Proceedings of the 5th European conference on European computing conference*, 2011, pp. 376–381.
- [12] D. H. Yang, J. H. Kang, Y. B. Park, Y. J. Park, H. S. Oh, and S. B. Kim, "Association Rule Mining and Network Analysis in Oriental Medicine," *PLoS One*, vol. 8, no. 3, pp. 1–9, 2013.
- [13] F. M. Afendi, N. Ono, Y. Nakamura, K. Nakamura, L. K. Darusman, N. Kibinge, A. H. Morita, K. Tanaka, H. Horai, M. Altaf-Ul-Amin, and S. Kanaya, "Data Mining Methods for Omics and Knowledge of Crude Medicinal Plants toward Big Data Biology.," *Comput. Struct. Biotechnol. J.*, vol. 4, no. 5, p. e201301010, 2013.
- [14] F. M. Afendi, N. Ono, Y. Nakamura, K. Nakamura, L. K. Darusman, N. Kibinge, A. H. Morita, K. Tanaka, H. Horai, M. Altaf-Ul-Amin, and S. Kanaya, "Data Mining Methods for Omics and Knowledge of Crude Medicinal Plants Toward Big Data Biology.," *Comput. Struct. Biotechnol. J.*, vol. 4, no. 5, pp. 1–14, 2013.
- [15] S. H. Wijaya, H. Husnawati, F. M. Afendi, I. Batubara, L. K. Darusman, M. Altaf-Ul-Amin, T. Sato, N. Ono, T. Sugiura, and S. Kanaya, "Supervised clustering based on DPPlusO: Prediction of plant-disease relations using Jamu formulas of KNAPSAcK database.," *Biomed Res. Int.*, vol. 2014, 2014.
- [16] V. Nebot and R. Berlanga, "Finding association rules in semantic web data," *Knowledge-Based Syst.*, vol. 25, no. 1, pp. 51–62, 2012.
- [17] R. Ramezani and C. Engineering, "SWApriori: A New Approach to Mining Association Rules from Semantic Web Data."
- [18] T. Anbutamilazhagan and M. . Selvaraj, "A Novel Model for Mining Association Rules from Semantic Web Data," *Elysium J.*, vol. 1, no. 2, pp. 1–5, 2014.
- [19] K. J. Kochut and M. Janik, "SPARQLer: Extended Sparql for Semantic Association Discovery," in *Proceedings of the 4th European conference on The Semantic Web: Research and Applications - ESWC '07*, 2007, pp. 145–159.
- [20] A. S. H. Yazdi and M. Kahani, "A Novel Model for Mining Association Rules from Semantic Web Data," in *Intelligent Systems (ICIS), 2014 Iranian*, 2014, pp. 1–4.
- [21] A. Bellandi, B. Furlletti, V. Grossi, and A. Romei, "Ontology-Driven Association Rule Extraction: A Case Study," *Context. Ontol. Represent. Reason.*, 2007.
- [22] K. Chomboon, N. Kaoungku, K. Kerdprasop, and N. Kerdprasop, "Data Mining in Semantic Web Data," *Int. J. Comput. Theory Eng.*, vol. 6, no. 6, pp. 6–9, 2014.
- [23] E. L. Willighagen, "A short tutorial on rrdf," 2012. [Online]. Available: <http://cran.r-project.org/web/packages/rrdf/rrdf.pdf>.
- [24] D. Y. Zhao, "RDATA MINING," 2011. [Online]. Available: <http://www.rdatamining.com/docs/association-rule-mining-with-r>.
- [25] Y. Nakamura, H. Asahi, M. Altaf-Ul-Amin, K. Kurokawa, and S. Kanaya, "KNAPSAcK: A Comprehensive Species-Metabolite Relationship Database." [Online]. Available: <http://kanaya.naist.jp/jamu/top.jsp>, tanggal akses: 30-Mar-2015.
- [26] Badan Pengawas Obat dan Makanan Indonesia, "Produk Obat Tradisional." [Online]. Available: <http://ceknie.pom.go.id/>, tanggal akses: 24-Jan-2016.
- [27] Badan Pengawas Obat dan Makanan Indonesia, "Obat Bahan Alamai Indonesia." [Online]. Available: <http://www.pom.go.id/index.php/oai/>, tanggal akses: 24-Jan-2016.