

# Pembangunan *Biodiversity Informatics* Genetika Tumbuhan Berbasis Ontologi

Bayu Santoso(G64134017)\*, Yeni Herdiyeni

## Abstrak/Abstract

Indonesia memiliki lebih dari 32.000 spesies tumbuhan. Saat ini hutan Indonesia mengalami kerusakan dan kepunahan. Oleh karena itu, diperlukan upaya untuk melestarikan tumbuhan. Salah satu cara untuk melestarikan tumbuhan adalah dengan cara mengenali spesies tumbuhan tersebut. Berdasarkan hal tersebut maka muncul bidang baru dalam pengumpulan informasi tumbuhan yang bernama *biodiversity informatics*. Metode pemodelan data yang dapat menangani sistem berbasis inferensi adalah ontologi. Ontologi dapat diterapkan pada web semantik. Penelitian ini akan mengembangkan sistem web semantik yang memberikan informasi genetika tumbuhan. Selain itu juga merumuskan masalah bagaimana melakukan inferensi pengetahuan dari sistem web semantik gen tumbuhan dengan sistem web semantik yang lain.

## Kata Kunci/Keywords

*biodiversity informatics*; tumbuhan; ontologi; web semantik

\*Email: bayusantoso.mail@gmail.com

## PENDAHULUAN

### Latar Belakang

Indonesia memiliki lebih dari 32.000 spesies tumbuhan (Bappenas 2003). Saat ini hutan Indonesia mengalami kerusakan dan kepunahan (Zuhud 2008). Oleh karena itu, diperlukan upaya untuk melestarikan tumbuhan. Salah satu cara untuk melestarikan tumbuhan adalah dengan cara mengenali tumbuhan tersebut. *Biodiversity Informatics* merupakan upaya untuk membuat sumber informasi keanekaragaman hayati global tersedia dalam format digital yang efisien, dan untuk mengembangkan alat yang efektif dalam menganalisis dan memahami data tersebut (Gilman et al. 2009). Informasi yang dapat diperoleh dari *biodiversity informatics* adalah informasi mengenai taksonomi, gambar tumbuhan, lingkungan, dan DNA tumbuhan.

Implementasi dari *biodiversity informatics* sudah menghasilkan beberapa sistem yang menyediakan informasi mengenai tumbuhan. Integrated Taxonomic Information System (ITIS) dan Global Biodiversity Information Facility (GBIF) menyediakan informasi yang luas tentang tumbuhan. Proses identifikasi dan pengelolaan informasi keanekaragaman hayati tersebut memerlukan sistem yang terpadu dan holistic dengan menggunakan

IPTEKS komputer yang berkembang pesat saat ini (Herdiyeni et al. 2013). Upaya pemanfaatan IPTEKS yang telah dilakukan seperti diantaranya pembangunan sistem IPB Biodiversity Informatics (IPBiotics) untuk pengelolaan informasi keanekaragaman hayati sumber daya alam Indonesia. Sistem BI tersebut berguna meningkatkan pengelolaan pengetahuan (knowledge management), eksplorasi, analisis, sintesis dan interpretasi data keanekaragaman hayati mulai dari level genomik, level spesies sampai dengan level ekosistem (Herdiyeni et al. 2013). Pada pengembangan selanjutnya, sistem IPBiotics didesain agar dapat melakukan inferensi pengetahuan. Sistem yang ada saat ini masih menggunakan model basis data relasional. Permasalahannya adalah model basis data relasional kurang sesuai diterapkan pada sistem berbasis inferensi (Laallam, Kherfi, and Benslimane 2013).

Ontologi adalah metode yang digunakan untuk merepresentasikan ide, fakta dan lain sebagainya, yang digunakan untuk mendefinisikan hubungan dan klasifikasi dari pengetahuan tertentu (Jepsen 2009). Ontologi dapat menentukan kelas, hubungan, fungsi dan objek lain (DiLecce V 2008). Selain itu, model ontologi lebih sesuai diterapkan pada web semantik dibandingkan dengan model basis data relasional (Laallam, Kherfi, and

Benslimane 2013).

Penelitian dengan menggunakan ontologi mengenai tumbuhan sudah banyak dilakukan, seperti penelitian tentang ontologi yang digunakan untuk menganalisis hubungan tumbuhan obat dengan istilah medis yang standar (Vadivu G 2012). Penelitian yang terkait dengan ontologi gen juga sudah pernah dilakukan untuk menghasilkan data gen yang dinamis dan terkontrol (Ashburner et al. 2000) dan pemodelan ontologi tumbuhan obat menggunakan pengetahuan etnobotani (Sanjaya 2014). Namun penelitian tersebut berfokus pada pemodelan ontologi. Berdasarkan latar belakang di atas penelitian ini akan membangun sistem web semantik yang memanfaatkan ontologi yang sudah ada untuk mengelola informasi tumbuhan.

### Tujuan

Tujuan yang ingin dicapai dalam penelitian ini adalah:

1. Membangun sistem *biodiversity informatics* tumbuhan menggunakan ontologi gen (*gene ontology*).
2. Menerapkan sistem inferensi pengetahuan pada ontologi gen untuk mengembalikan informasi berupa *molecular function*, *biological processes* dan *cellular components* yang terdapat pada tumbuhan.

### Ruang Lingkup

Ruang lingkup penelitian adalah:

1. Ontologi yang digunakan dalam penelitian ini berasal dari situs [geneontology.org](http://geneontology.org).
2. Membangun *biodiversity informatics* pada level genetik.

### Manfaat

Manfaat yang diinginkan dari penelitian ini adalah membantu proses dokumentasi data dan pengetahuan keanekaragaman hayati tumbuhan. Dengan memanfaatkan ontologi, memungkinkan sistem untuk melakukan inferensi dan mengembalikan informasi detail tumbuhan mengenai *molecular function*, *biological processes* dan *cellular components* tumbuhan. Dengan informasi tersebut, diharapkan proses dokumentasi keanekaragaman hayati tumbuhan berjalan lebih efektif.

karagaman yang dipelajari meliputi bidang molekuler, organisme dan ekologi. *Biodiversity information* (BI) merupakan upaya untuk membuat sumber informasi keanekaragaman hayati global tersedia dalam format digital yang efisien, dan untuk mengembangkan alat yang efektif dalam menganalisis dan memahami data tersebut (Gilman et al. 2009). Salah satu tantangan utama dalam BI adalah untuk memberikan data pengetahuan mengenai keanekaragaman hayati dengan cepat, dengan pengetahuan tersebut dapat dibangun sistem keanekaragaman hayati yang terpadu (Gilman et al. 2009).

### Ontologi

Ontologi adalah suatu metode yang digunakan untuk merepresentasikan pengetahuan yang mendefinisikan hubungan dan klasifikasi dari pengetahuan tertentu (Jepsen 2009). Seperti pemrograman berorientasi objek, ontologi juga menggunakan kelas dan *instance* dalam merepresentasikan pengetahuan. Namun, ontologi dengan pemrograman berorientasi objek berbeda dalam implementasinya. Pada pemrograman berorientasi objek *instance* merupakan contoh nyata dari kelas tersebut. Pada ontologi, *instance* bermakna anggota suatu domain tertentu (Jepsen 2009).

### Gene Ontology (GO)

*Gene ontology* (GO) dikembangkan oleh Gene Ontology Consortium untuk memberikan informasi mengenai gen yang lebih dinamis, terkontrol dan memiliki aturan yang sama. Dibuat dari tahun 1998 GO pada awalnya mengkolaborasi tiga buah *database* organisme yaitu Fly Base (*Drosophila*), *Saccharomyces Genome Database* (SGD) dan *Mouse Genome Database* (MGD). Semenjak itu GO terus berkembang dengan bergabungnya penelitian-penelitian mengenai ontologi gen dari lembaga lain. Saat ini GO sudah menampung data-data mengenai gen tumbuhan, hewan dan mikroba.

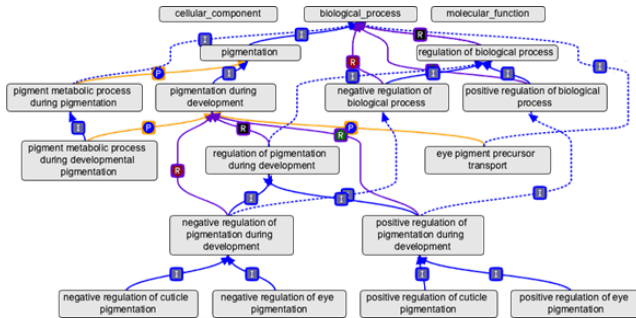
GO memiliki tiga komponen ontology yaitu *Molecular Function*, *Biological Processes* dan *Cellular Components* (Ashburner et al. 2000). *Molecular Function* merupakan aktivitas yang terjadi pada level molekuler seperti aktivitas katalis atau aktivitas binding. *Biological processes* menggambarkan serangkaian kegiatan dengan tujuan tertentu yang melibatkan proses molekuler function. *Cellular components* menggambarkan komponen-komponen sel yang merupakan bagian dari objek yang lebih besar sebagai contoh retikulum endoplasma atau nukleus. Selain itu *cellular components* juga memberikan informasi *gene-product*, sebagai contoh ri-

## TINJAUAN PUSTAKA

### *Biodiversity Information*

*Biodiversity* atau keanekaragaman hayati mengacu pada upaya untuk mempelajari kehidupan di bumi. Keane-

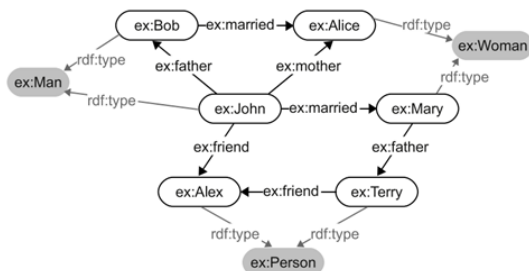
*bosome* dan *proteasome*. Gambar 1 merupakan sebagian kecil dari term yang ada di bawah *biological processes*.



Gambar 1. Term di bawah *biological processes*

### Resource Description Framework (RDF)

eXtensible Markup Language (XML) adalah suatu bahasa yang direpresentasikan dengan *metadata*. XML merepresentasikan informasi agar dapat dengan mudah diakses oleh mesin. XML tidak menyediakan tag-tag tertentu, tetapi penggunaannya dapat mendefinisikan tag secara mandiri. Namun, XML tidak dapat menunjukkan arti dari suatu data. RDF merupakan suatu bentuk dari data model. RDF mengadopsi sintaks-sintaks yang dimiliki oleh XML. Menurut Antoniou dan Hermalen (2008), RDF memiliki konsep dasar RDF yaitu *resource*, *properties*, dan *statement*. *Resource* adalah objek atau tentang sesuatu yang ingin diungkapkan. Beberapa contoh *resource* seperti judul buku, pengarang, penerbit, orang dan sebagainya. Setiap *resource* memiliki *Universal Resource Identifier* (URI). *Properties* adalah deskripsi hubungan antar *resource*, misalnya properti yang menghubungkan antara buku dengan penulisnya adalah “ditulis oleh”. *Statement* adalah bentuk literal dari *resource*, *properties* dan nilainya. Menurut Raimbault (2010), statement disebut juga dengan triple model.



Gambar 2. Contoh Model RDF

Gambar 2 menunjukkan contoh model RDF yang direpresentasikan dalam bentuk *graph*. Contoh *statement* yang dapat dilihat dari Gambar 2 adalah John menikah dengan Mary.

### Web Ontology Language (OWL)

*Web Ontology Language* (OWL) adalah bahasa yang digunakan untuk membaca dan mendefinisikan kelas dan properti model RDF (Segaran, C, and J 2009). OWL merupakan bahasa yang kompleks. Oleh karena itu, OWL dibagi menjadi tiga sub bahasa berdasarkan tingkat kompleksitasnya, yaitu OWL-Lite, OWL DL dan OWL Full. OWL-Lite merupakan bentuk sub bahasa OWL yang paling sederhana. OWL DL memiliki fitur yang dapat digunakan untuk memaksimalkan penggunaan sistem *reasoning* dan dapat menjamin kelengkapan informasi yang ingin didapat. OWL Full memberikan fitur yang membebaskan dalam mendeskripsikan sintaks RDF, tetapi tidak menjamin kelengkapan informasi yang didapat (He dan An 2011). OWL memiliki struktur data yang terdiri dari *namespace* dan *ontology head*. *Namespace* digunakan untuk mengelompokkan *identifier*. *Ontology head* digunakan untuk mengumpulkan informasi *metadata* OWL yang digunakan (He and An 2011).

### Simple Protocol and RDF Query Language (SPARQL)

*Simple Protocol and RDF Query Language* (SPARQL) adalah bahasa *query* yang digunakan untuk melakukan query data terhadap model RDF. Sama halnya dengan query pada SQL, SPARQL juga digunakan untuk melakukan query data-data yang diinginkan. Pembedanya, SQL melakukan query pada *database* yang terdiri dari satu atau beberapa tabel, sedangkan SPARQL melakukan query pada data RDF yang berupa triple model (Segaran et al. Segaran, C, and J 2009). Berikut contoh sintaks SPARQL:

```
PREFIX fb:<http://rdf.freebase.com/ns/>
SELECT ?who ?film
WHERE {
  ?film fb:film.film.directed_by ?who
  ?film fb:film.film.starring ?who
}
```

Query dari SPARQL hampir sama seperti pada bahasa *query* yang lain. Seperti sintaks SELECT, WHERE dan lainnya. Sintaks PREFIX digunakan untuk menyinkronkan alamat sumber dari data ontologi.

### Semantic Web Object Oriented Design Methodology (SW-ODM)

*Object Oriented Design Methodology* (OOMD) adalah metode yang digunakan untuk mengembangkan aplikasi web. Metode ini berguna untuk menangkap kebutuhan

pengguna. Dengan adanya metode ini aktifitas pengembangan dapat dikontrol dengan baik (Farooq, Ahsan, and Sha 2010). SW-OODM merupakan metode pengembangan aplikasi web semantik yang merupakan pengembangan dari OODM. SW-OODM meliputi aktivitas yang berguna untuk memformulasikan dan mendeskripsikan pengetahuan, sehingga aplikasi yang dibuat dapat dimengerti baik oleh mesin maupun oleh manusia (Farooq, Ahsan, and Sha 2010). Aktivitas yang terdapat di metode SW-OODM yaitu meliputi analysis phase dan design phase. Tahapan-tahapan yang berada dalam *analysis phase* dikelompokkan ke dalam empat model: *preliminary web-ontology model*, *information model*, *user model*, dan *operation model*. Sedangkan pada design phase dikelompokkan menjadi enam unit: *namely, building component model, building navigation model, building operation partitioning model, web ontology model* dan *semantic web-page design* (Farooq et al. 2010).

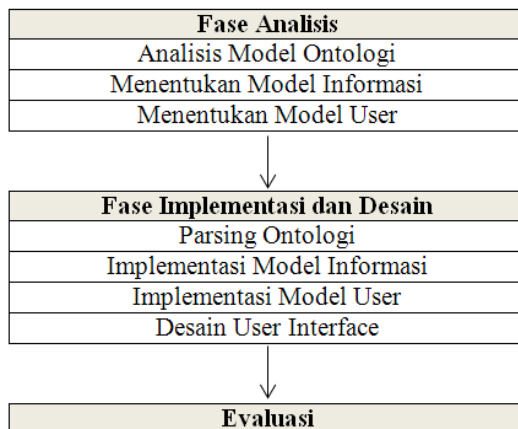
## METODE PENELITIAN

### Data Penelitian

Data yang digunakan pada penelitian ini adalah data ontologi gen yang berasal dari situs [geneontology.org](http://geneontology.org).

### Tahapan Penelitian

Tahapan-tahapan yang dilakukan pada penelitian ini mengacu pada metode pengembangan aplikasi SW-OODM. Tahapan pengembangan aplikasi SW-OODM dapat dilihat pada Gambar 3.



Gambar 3. Tahapan Penelitian

#### Fase Analisa

Pada fase analisa terdapat 3 aktivitas yang akan dilakukan, yaitu analisis model ontologi, menentukan model informasi, menentukan model user. Aktivitas analisis model ontologi akan diidentifikasi domain yang

terdapat pada ontologi gen. Hasil identifikasi model ontologi digambarkan dengan bentuk graph, sehingga bentuk dari ontologi gen akan dapat lebih mudah dipelajari.

Aktivitas menentukan model informasi akan memanfaatkan hasil dari analisis model ontologi untuk menentukan kelas, atribut dan keterkaitan antar objek yang ada di dalam ontologi. Hal ini perlu dilakukan untuk menjadi acuan dalam pembuatan query dan membuat query yang tepat dengan SPARQL untuk mengambil informasi yang terdapat di dalam ontologi dan juga melakukan inferensi pengetahuan tumbuhan yang ada di dalam ontologi.

Pada aktivitas menentukan model user akan dianalisis kebutuhan dari user yang akan menggunakan sistem yaitu berupa kelompok user yang mengakses sistem, rancangan kelas dari sistem, alur aktivitas yang dilakukan user dan skenario alur akses sistem dari user. Hasil dari tahapan ini berupa definisi diagram use case, class diagram, activity diagram dan sequence diagram.

#### Fase Implementasi dan Desain

Pada fase implementasi dan desain akan diawali dengan parsing ontologi. Parsing ontologi memetakan ontologi gen menjadi triple yang berupa subject, predicate dan object. Setelah ontologi dilakukan parsing dan menghasilkan triple, hasil ini yang akan dilakukan query dengan menggunakan SPARQL.

Pada tahapan implementasi model informasi mencakup pembuatan query SPARQL untuk mengembalikan informasi yang terdapat dalam ontologi, yaitu informasi gene-product, cellular location dan sequence. Kemudian pada tahapan implementasi model akan dibuat fungsi-fungsi yang akan memanfaatkan query yang telah dibuat pada tahap implementasi model sehingga informasi dari sistem ontologi gen dapat diakses dengan menggunakan web.

Pada tahapan implementasi user akan menghasilkan fungsi-fungsi yang akan menangani input yang diberikan oleh user, melakukan pengambilan informasi berdasarkan input yang diterima, dan memberikan output yang sesuai dengan input yang sudah diberikan. Tahap desain user interface akan menghasilkan halaman yang akan diakses oleh user. Halaman yang dibuat berupa halaman input dan halaman output yang dapat dilihat oleh user.

#### Evaluasi

Fase evaluasi akan dilakukan pengujian dari sistem yang sudah dibuat. Pengujian yang dilakukan menggunakan metode black box. Metode ini akan memberikan kasus untuk dilakukan oleh sistem dengan memberi input dan menguji kesesuaian hasil yang diberikan oleh



sistem.

## Lingkungan Pengembangan

Pembangunan sistem ontologi gen tanaman berbasis web ini dilakukan dengan menggunakan perangkat keras dan perangkat lunak sebagai berikut:

- Prosesor Intel Core i7 4500U 1,8 GHz
- Memori 12 GB
- Hard disk 1 TB
- Sistem operasi Windows 7 Ultimate
- Bahasa pemrograman Python dengan Flask sebagai web framework
- RDFLib sebagai library yang digunakan untuk penggunaan RDF pada Phyton
- Lingkungan pengembangan (IDE) Light Table
- Apache Jena Fuseki

## HASIL DAN PEMBAHASAN

### Fase Analisis

#### Model Ontologi

Pada penelitian ini ontologi yang digunakan adalah Gene Ontology (GO). Struktur ontologi dari GO berupa struktur *graph*. Pada struktur ontologi ini *terms* merupakan *node*-nya dan relasi antar *terms* tersebut merupakan *edge*-nya. Setiap *terms* atau *node* yang terdapat pada GO memiliki definisi masing-masing, begitu juga dengan relasi atau *edge*. Pada GO terdapat beberapa jenis relasi yaitu *is a*, *part of*, *has part* dan *regulates*.

#### Ruang Lingkup

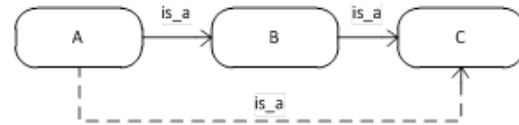
GO memiliki tiga domain yaitu *cellular component*, *molecular function* dan *biological process*. Pada penelitian ini akan dibuat sistem yang dapat mencari dan merepresentasikan *gene product*.

#### Relasi

##### 1. *is\_a*

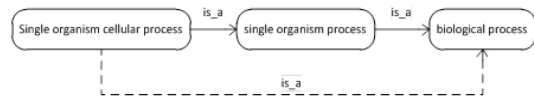
Relasi *is a* adalah relasi paling sederhana yang terdapat pada GO. Relasi *is a* dapat dikatakan seperti *A is a B*, artinya adalah setiap kelas yang ada pada A merupakan bagian / *sub class* dari B. Relasi ini digunakan untuk

menunjukkan keterkaitan antara kelas yang lebih spesifik ke kelas yang lebih umum. Relasi *is a* juga dapat menunjukkan inferensi antar kelas, sebagai contoh pada Gambar 4 apabila *A is a B* dan *B is a C* maka dapat dikatakan bahwa *A is a C*.



Gambar 4. Relasi *is\_a*

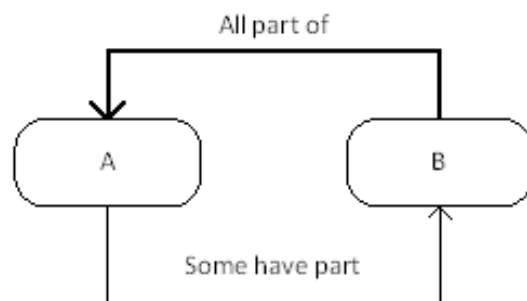
Pada GO contoh dari relasi ini di tunjukkan seperti pada Gambar 5. Terlihat bahwa *single organism cellular process is a single organism process* dan *single organism process is a biological process*, maka dari itu *single organism cellular process is a biological process*.



Gambar 5. Contoh relasi *is\_a*

##### 2. *part\_of*

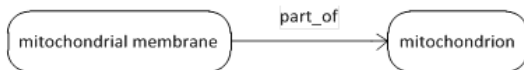
Relasi *part of* ini menjelaskan bahwa apabila A *part of* B, maka A merupakan salah satu kelas yang ada di dalam B. Relasi ini biasanya dipakai untuk menjelaskan suatu proses yang memiliki beberapa proses di dalamnya. Pada Gambar 6 menunjukkan hubungan *part of* antara A dan B, sehingga apabila proses B tidak ada maka proses A pun tidak ada, tetapi apabila A tidak ada belum tentu B tidak ada.



Gambar 6. Relasi *part\_of*

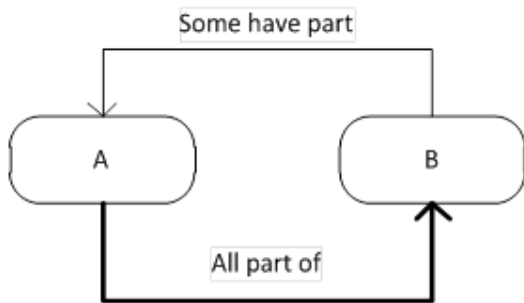
Pada GO contoh relasi ini ditunjukkan seperti pada Gambar 7 yang menunjukkan bahwa *mitochondrial membrane* merupakan *part of* *mitochondrion*.

##### 3. *has part*



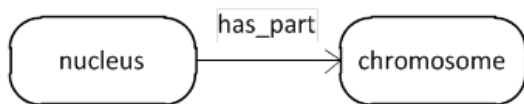
Gambar 7. Contoh relasi *part\_of*

Relasi *has part* merupakan kebalikan dari *part of*. Apabila *part of* menunjukkan dari bagian kecil ke bagian yang lebih besar, maka *has part* menunjukkan bagian besar memiliki bagian-bagian yang lebih kecil. Pada Gambar 8 menunjukkan skema relasi *has part*, B memiliki bagian berupa A.



Gambar 8. Relasi *has\_of*

Di dalam GO relasi ini ditunjukkan dengan hubungan antara *nucleus* dengan *chromosome* seperti terlihat pada Gambar 9.



Gambar 9. Contoh relasi *has\_of*

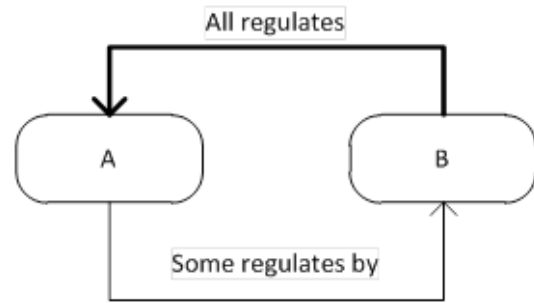
#### 4. *regulates*

Relasi *regulates* merupakan relasi yang umum terdapat pada GO. Relasi ini menunjukkan bahwa node yang satu mempengaruhi node yang lain. Pada Gambar 10 menunjukkan bahwa B *regulates* A, yang artinya apabila B merupakan suatu proses maka proses yang terjadi pada B akan mempengaruhi proses pada A, tetapi proses B tidak selalu terpengaruh dengan proses A.

Relasi ini pada GO dicontohkan hubungan antara *activation of reciprocal meiotic recombination* mempengaruhi proses dari *meiotic cell cycle* yang ditunjukkan pada Gambar 11.

#### Model Informasi

Setelah mendapatkan model ontologi dari GO maka tahap selanjutnya adalah merancang model informasi.



Gambar 10. Relasi *regulates*



Gambar 11. Contoh relasi *regulates*

Informasi yang akan ditampilkan kepada pengguna sistem adalah informasi yang terdapat pada GO akan ditampilkan dengan cara pengguna memasukkan kata kunci yang akan di cari. Kemudian setelah ditemukan untuk mengetahui lebih lanjut mengenai informasi dari hasil yang ditemukan maka pengguna dapat masuk ke halaman detail dengan menggunakan link yang ada pada hasil tersebut. Tabel 1 menunjukkan kelas yang ada di dalam sistem ini.

Tabel 1. Kelas pada sistem GO

Kelas	Atribut	Proses
SearchForm	Keyword, Ontology	ViewForm() submitKeyword(string)
Result	QueryResult	viewQResult() viewOntology() viewDetail(string)

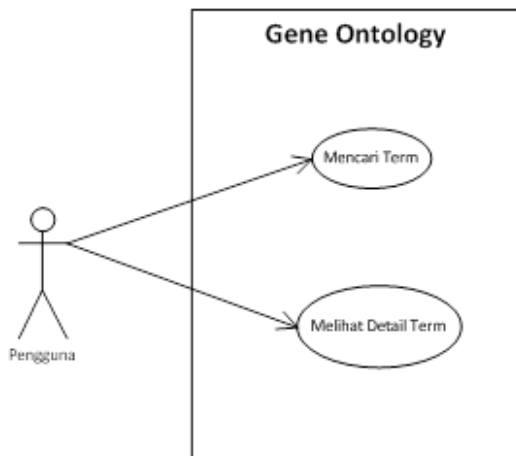
#### Model User

Use Case sistem yang akan dibuat dapat dilihat pada Gambar 12.

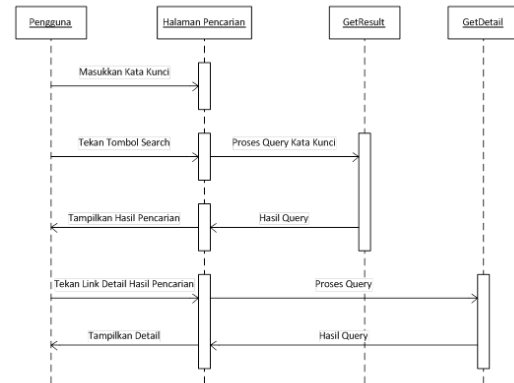
Pada use case ini menunjukkan bahwa pengguna dapat melakukan pencarian *term* berdasarkan kata kunci yang ingin dicari. Kemudian setelah menemukan kata kunci yang dicari pengguna dapat melihat detail dari kata kunci tersebut.

Berdasarkan *use case* pada Gambar 13 maka dibuat *activity diagram* untuk *use case* mencari *term* dan Gambar 14 untuk *use case* melihat *detail term*.

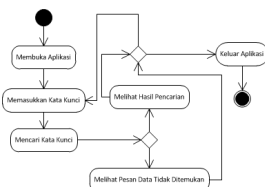
Dari *activity diagram* yang dibuat maka dapat dibuat *sequence diagram*. *Sequence diagram* ini menunjukkan langkah detail yang akan dilakukan pengguna pada saat menggunakan sistem ini. Pada Gambar 15 menunjukkan



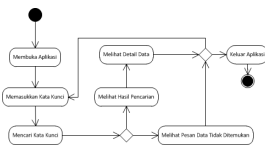
Gambar 12. Use case diagram sistem GO



Gambar 16. Sequence diagram melihat detail term

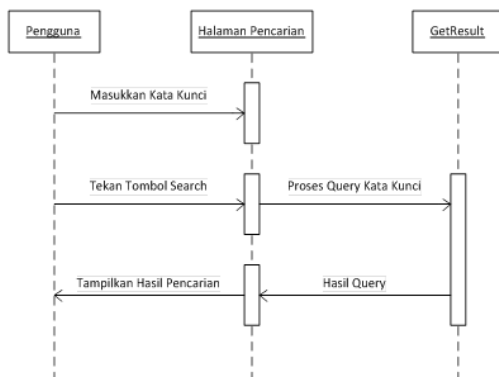


Gambar 13. Activity diagram mencari term



Gambar 14. Activity diagram melihat detail term

sequence diagram untuk use case mencari term dan Gambar 16 sequence diagram untuk use case melihat detail term.



Gambar 15. Sequence diagram mencari term

## Fase Implementasi dan Desain Parsing Ontologi

Ontologi merupakan salah satu cara yang digunakan untuk merepresentasikan pengetahuan. Ontologi menggunakan Uniform Resource Identifier (URI) untuk membentuk struktur dokumennya. URI ini digunakan untuk memberikan keterangan pada suatu objek. Objek dari ontologi terdiri dari tiga bagian yang disebut *triples*. *Triples* ini terdiri dari *subject*, *predicate* dan *object*. Tujuan dari dilakukan *parsing* ontologi adalah untuk membuat ontologi dapat dibaca dan dipahami oleh pembuat sistem.

Model data ontologi yang digunakan pada penelitian ini menggunakan RDF. RDF berisi kumpulan kumpulan *triples* yang disimpan dalam format XML. Model data ini dipilih karena merupakan model data yang menjadi standar pada ontologi, sehingga terdapat banyak *parser* yang dapat melakukan parsing terhadap data RDF.

Parser yang digunakan pada penelitian ini adalah RDLib. RDLib merupakan package dari bahasa pemrograman python yang dapat melakukan *parsing* terhadap data ontologi. RDLib dipilih karena memiliki beberapa kelebihan, seperti dapat mem-parsing berbagai jenis model data, dapat menyimpan graph dari suatu ontologi, dapat menyimpan data pada *memory* maupun secara *presistent*, dan dapat menggunakan SPARQL untuk melakukan operasi pada data ontologi.

Untuk penggunaan RDLib pada python maka sistem harus memanggil package RDLib tersebut dengan syntax seperti pada Gambar 17.

```
import rdflib
```

Gambar 17. Pemanggilan package RDLib

Untuk melakukan parsing terhadap data ontologi maka syntax yang diperlukan seperti pada Gambar 18. Pada Gambar 18 terlihat bahwa RDLib melakukan parsing terhadap data ontology GO.

```
graph = rdflib.Graph()
query_result = graph.parse("go.owl")
```

**Gambar 18.** Proses *parsing* ontologi

## Implementasi Model Informasi

Pada proses implementasi model informasi maka dibuat *user interface* yang akan digunakan oleh pengguna. Penelitian ini akan menghasilkan sistem yang berbasis web dengan menggunakan *web framework* Flask yang merupakan *web framework* python yang ringan. Pembuatan halaman web menggunakan HTML. Dengan dibuat sistem yang berbasis web maka sistem ini dapat dibuka di komputer manapun tanpa melakukan proses instalasi. Halaman yang dibuat pada sistem ini ditunjukkan pada Tabel 2.

**Tabel 2.** Halaman yang dibuat pada sistem GO

Halaman	URL
Search	url/index
Result	url/result
Detail	url/detail

Gambar 19 merupakan halaman Search yang digunakan pengguna untuk melakukan pencarian terhadap kata kunci yang ingin dicari.

**Gambar 19.** Halaman *search*

## Implementasi Model User

Pengguna akan menuliskan kata kunci pada search form yang sudah disediakan. Kemudian apabila tombol Search ditekan maka sistem akan melakukan pencarian terhadap kata kunci yang diberikan. Proses pencarian tersebut dilakukan pada data GO dengan menggunakan SPARQL seperti tampak pada Gambar 20.

Pada Gambar 20 terdapat syntax SPARQL, dengan struktur PREFIX dan perintah SELECT. PREFIX digunakan untuk mempersingkat URI yang digunakan pada perintah SELECT. Pada perintah SELECT terdapat bagian yang akan ditampilkan pada halaman web yaitu *?id*, *?aspects*, *?labels* dan *?synonyms*. Kemudian terdapat bagian WHERE menunjukkan variabel apa saja yang akan diambil dari data ontologi. Pada bagian WHERE ini merupakan *triple* yang berupa *subject*, *predicate* dan

```
@app.route('/result', methods=['GET', 'POST'])
def result():
    form = SearchForm()

    graph = rdflib.Graph()
    graph.parse("go.owl")

    query_result = graph.query(
        """
        PREFIX rdfa: <http://www.w3.org/2000/01/rdf-schema#>
        PREFIX owl: <http://www.w3.org/2002/07/owl#>
        PREFIX oboInOwl: <http://www.geneontology.org/formats/oboInOwl#>

        SELECT
        ?id
        (group_concat(distinct ?aspect) AS ?aspects)
        (group_concat(distinct ?label) AS ?labels)
        (group_concat(distinct ?synonym) AS ?synonyms)
        WHERE {
        ?data oboInOwl:id ?id ;
        oboInOwl:hasOBONamespace ?aspect ;
        rdfa:label ?label ;
        oboInOwl:hasExactSynonym ?synonym .
        FILTER (REGEX(?aspect, '.*' + form.query.data + '.*'))
        || REGEX(?label, '.*' + form.query.data + '.*'))
        || REGEX(?synonym, '.*' + form.query.data + '.*'))
        }
        GROUP BY ?id
        """)

    return render_template('result.html', form = form, model = query_result)
```

**Gambar 20.** Syntax proses pencarian *term*

*object*. *Subject* pada query ini disimbolkan dengan *?data*. *Predicate* berupa URI dan atribut yang akan diambil, URI diganti dengan PREFIX agar lebih ringkas dalam penulisannya sebagai contoh *oboInOwl:id*. *Object* digunakan untuk menyimpan data yang didapat dari hasil query, sebagai contoh *object* *?id*. Gambar 21 merupakan hasil dari query yang ditampilkan pada halaman result.

No	Id	Name	Ontology
1	GO:0033027	positive regulation of mast cell apoptotic process	biological_process
2	GO:0034665	integrin alpha1-beta1 complex	cellular_component
3	GO:0005788	endoplasmic reticulum lumen	cellular_component
4	GO:0035891	exit from host cell	biological_process
5	GO:0097271	protein localization to bud neck	biological_process

**Gambar 21.** Halaman hasil pencarian *term*

## DAFTAR PUSTAKA

- Antoniou, G and F Hermalen (2008). *A Semantic Web Primer, Second edition*. Cambridge (GB): The MIT Pr.
- Ashburner, M et al. (2000). *Gene ontology: tool for the unification of biology*. California (US): Stanford University School of Medicine.
- Bappenas (2003). *Indonesia Biodiversity and Action Plan 2003-2020*. Jakarta (ID): Bappenas.
- DiLecce V, Calabrese M, ed. (2008). *LOF: Identifying Density-Based Local Outliers*. Taxonomies and Ontologies in Web Semantic Applications: the New Emerging Semantic Lexicon-Based Model. Vol. CIMCA International Conference, IEEE. 277-283. Viena (AUT).
- Ding, L et al., eds. (2005). *Search on The Semantic Web*. Computer Society, IEEE. 38(10):62-69.



- Farooq, A, S Ahsan, and A Sha, eds. (2010). *Engineering Semantic Web Applications by Using Object-Oriented Paradigm*. Journal of Computer Science. 2(6):156-165.
- Gilman, E et al. (2009). *Building The Biodiversity Data Commons The Global Biodiversity Information Facility*. Graz (AT): Graz University of Technology.
- Guralnick, R and A Hill (2008). *Biodiversity Informatics: Automated Approaches for Documenting Global Biodiversity Patterns and Processes*. Boulder (CO): Department of Ecology and Evolutionary Biology University of Colorado.
- Hamzari, ed. (2008). *Identifikasi tanaman obat-obatan yang dimanfaatkan oleh masyarakat sekitar hutan Tabo-tabo*. Jurnal Hutan dan Masyarakat. 3(2):159-167.
- He, G and L An, eds. (2011). *Ontology Language OWL Research Study*. MASS International Conference, IEEE. 1-4.
- Herdiyeni, Y et al., eds. (2013). *IPB Biodiversity Informatics (IPBiotics) Untuk Pembangunan Berkelanjutan*. Prosiding Seminar Hasil-Hasil PPM IPB 2013. 2: 389–402.
- Jepsen, TC, ed. (2009). *Just What Is an Ontology, Anyway?* IT professional, IEEE. 11(5):22-27.
- Laallam, FZ, ML Kherfi, and SM Benslimane, eds. (2013). *Using ontologies to overcoming draw-back of database and vice versa: a survey*. CSEIJ. 3(2):1-21.
- Raimbault, T, ed. (2010). *Overviewing the RDF(S) semantic web*. CiSE International Conference, IEEE. 1-4.
- Segaran, T, Evans C, and Taylor J (2009). *Programming the semantic web*. New York(US): O'Reilly.
- Vadivu G, Hopper SW, ed. (2012). *Ontology mapping of indian medicinal plants with standardized medical terms*. Journal of Computer Science. 8(9):1576-1584.
- Zuhud, EAM (2008). *Potensi hutan tropika indonesia sebagai penyangga bahan obat alam untuk kesehatan bangsa*. Bogor(ID). Fakultas Kehutanan IPB.