Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

Team project ©2017
Dony Pratidana S. Hum | Bima Agus Setyawan S. IIP

BAB II
TINJAUAN PUSTAKA

2.1. Web Semantik

Secara garis besar web semantik adalah informasi dalam jumlah sangat besar di World Wide Web yang terhubung secara global dengan suatu cara tertentu dan dapat dimengerti oleh mesin, sehingga dapat diproses secara langsung oleh mesin menjadi *knowledge* untuk ditampilkan kepada *user*. Web semantik juga dapat dikatakan sebagai sebuah cara yang efisien untuk merepresentasikan data di World Wide Web sebagai sebuah *database* yang terhubung secara global (Siahaan, 2006).

Istilah web semantik digagas oleh Tim Berners-Lee, penemu dari World Wide Web (WWW) dan direktur World Wide Web Consorsium (W3C). Tim Berners-Lee mendefinisikan web semantik sebagai "*Web of data* yang dapat diproses secara langsung maupun tidak langsung oleh mesin". Web semantik merupakan *extension* dari web saat ini, di mana informasi memiliki makna yang telah didefinisikan dengan baik sehingga dapat menampilkan hasil pencarian yang lebih relevan dan dapat mengintegrasikan data dari berbagai sumber yang berbeda (Berners-Lee, 2001).

Rasionalisasi untuk web semantik adalah bahwa data yang umumnya tersembunyi di dalam file HTML terkadang berguna dalam beberapa konteks, tapi tidak dalam konteks yang lain. Masalah umum pada data yang berada di web adalah, dalam bentuknya yang sekarang sulit untuk digunakan dalam skala besar,

Implementasi Ontologi ..., Alexander tyas Aji Wardhana, FTI UMN, 2013

karena tidak terdapat sistem global untuk menyebarluaskan data dalam sebuah cara yang mudah untuk diproses oleh semua orang. Sebagai contoh, sebuah *event* olahraga, informasi cuaca, jadwal pesawat, acara televisi, dan lain-lain, semua informasi ini di hadirkan oleh banyak *website*, namun semua dalam format HTML. Masalahnya adalah, dalam beberapa konteks, sulit untuk menggunakan data yang ada tersebut sebagaimana yang dinginkan orang lain (Siahaan, 2006).

Metode web semantik telah membantu terjadinya revolusi dalam hal penyampaian dan pemanfaatan informasi pada World Wide Web ke tingkat pemanfaatan yang lebih baik. Menurut Tim Berners-Lee (1999), ada dua visi dalam pengembangan web ke depan, yaitu dengan membuat web semakin baik sebagai media kolaborasi dan membuat web semakin mudah dipahami oleh mesin.

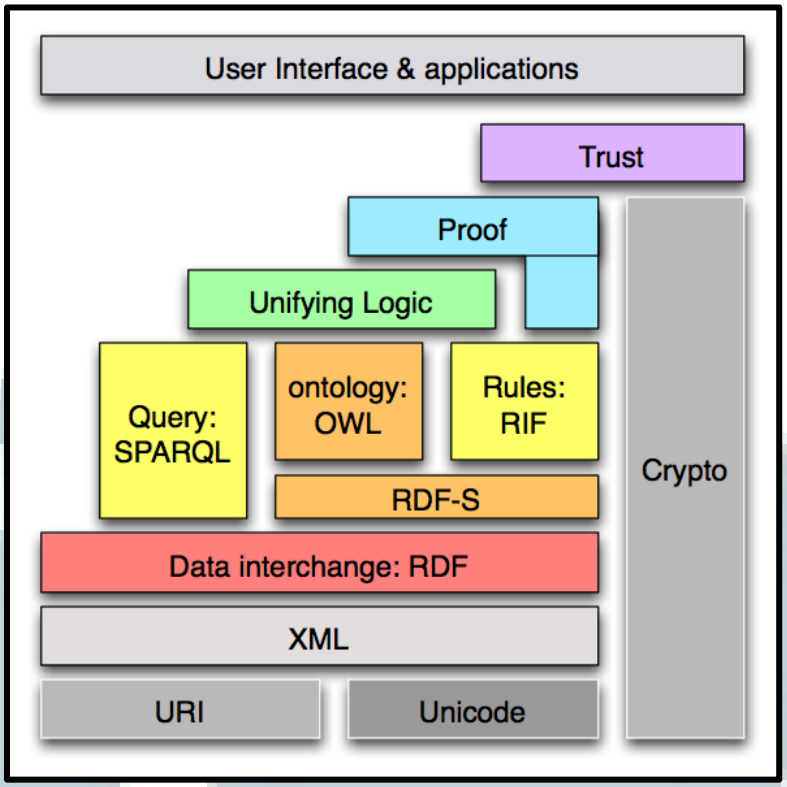
Sebagai contoh, jika seseorang ingin menyusun informasi dari *website* sekaligus, maka dengan teknologi web yang sekarang ada, dia harus mengunjungi situs-situs tersebut satu-persatu dan kemudian melakukan *cut and paste* pada konten dari masing-masing situs untuk menciptakan suatu informasi yang menyeluruh. Hal ini sangatlah membuang waktu dan tenaga, karena dilakukan secara manual oleh manusia, dan dengan berbasis teknologi yang sudah ada tidak akan mungkin dibuat otomatisasinya, mengingat halaman web berbasis HTML hanyalah dirancang untuk dipahami oleh manusia bukan mesin (Muslimin *et al.*, 2006).

Dengan metode web semantik, data berbasis HTML dapat dirubah menjadi format yang dapat dipahami oleh mesin, sehingga mesin dapat melakukan proses pengumpulan informasi dan memahami hubungan antara informasi. Web

semantik bukanlah *Artifitial Intelegent* (kecerdasan buatan), karena mesin tidak dengan sendirinya memahami bahasa manusia secara menyeluruh. Konsep ini hanya menandakan kemampuan mesin untuk memecahkan *well-defined problems* (permasalahan yang telah ditentukan) dengan cara melakukan *well-defined operations* (operasi untuk memecahkan masalah yang juga telah ditentukan) pada *well-defined data* (data yang juga telah ditentukan) yang tersedia. Jadi, untuk bahasa manusia yang berada di luar *well-defined data*, mesin sudah tidak mampu lagi untuk memahami bahasa tersebut (Muslimin *et al*., 2006).

2.1.2. Komponen Web Semantik

Pembuatan web semantik dimungkinkan dengan adanya sekumpulan standar yang dikoordinasi oleh World Wide Web Consortium (W3C). Standar yang paling penting dalam membangun web semantik adalah XML, XML Schema, RDF, OWL, dan SPARQL. Gambar 2.1 adalah layer dari web semantik sebagaimana direkomendasikan oleh W3C (Wicaksana, 2006).



Gambar 2.1. Arsitektur Web Semantik
Sumber : http://www.w3.org/DesignIssues/diagrams/sweb-stack/2006a.png

Gambar 2.1 menjelaskan arsitektur web semantik, di mana tiap layer memiliki fungsinya masing-masing. Extensible Markup Language (XML) berfungsi sebagai *markup language* untuk menyimpan isi halaman web, RDF adalah suatu lapisan yang berfungsi untuk mempresentasikan semantik dari isi halaman tersebut. Ontologi digunakan untuk menjelaskan *vocabulary* dari domain. *Unifying Logic* memungkinkan untuk mengambil data yang diinginkan (Wicaksana, 2006).

2.2. Resource Description Framework

Berdasarkan Web World Web Consurtium, Resource Description Framework (RDF) adalah bahasa untuk merepresentasikan informasi mengenai *resource* yang ada dalam World Wide Web. RDF merupakan suatu metadata yang digunakan untuk mendeskripsikan alamat sumber daya pada web. Metadata ini dapat berupa judul, pengarang, hak cipta, dan lisensi dalam dokumen web. RDF memiliki tripel, biasanya ditulis sebagai subyek, predikat, dan obyek sebagai model data dasar. Sebuah obyek bisa dibalik sehingga menjadi subyek untuk obyek yang lain (Lausen, 2007).

Elemen pernyataan dalam RDF adalah sesuatu yang dideskripsikan dan biasanya berupa alamat URI. Dalam hal ini alamat URI merepresentasikan sumber daya. Predikat merupakan properti dari sumber daya yang menjelaskan hubungan antara subyek dengan obyek. Selain itu obyek merupakan nilai dari sebuah predikat. Obyek mempunyai dua tipe data yaitu obyek yang mempunyai tipe URI misalnya *http://airplane.com/id/102* dan obyek yang bertipe literal misalnya "adam air". Subyek dan predikat berisikan data yang berisikan sumber daya sedangkan obyek dapat bertipe sumber daya maupun literal (Wicaksana, 2006).

2.3. Resources Description Framework Schema

Sebuah Resources Description Framework (RDF) Schema adalah dokumen sederhana atau bagian kode yang mengendalikan sekumpulan terminologi pada sebuah dokumen atau bagian kode yang lain. *Schema* seperti

sebuah master *checklist* atau definisi tata bahasa. RDF Schema didesain untuk menjadi pemodelan data sederhana dalam RDF (Bekke, 1992).

Web World Wide Consurtium menyatakan, RDF Schema sebagai bahasa untuk mendeskripsikan RDF *vocabulary* dan merupakan *semantic extension* dari RDF. RDF Schema menyediakan mekanisme untuk mendeskripsikan sekelompok *resources* yang berkaitan dan hubungan antar *resources* tersebut.

2.4. Ontology Web Language

Ontology Web Language (OWL) adalah suatu bahasa untuk memproses informasi yang terdapat dalam *website* (Raimond, 2008). Sebuah Ontologi adalah sebuah struktur hirarki dari istilah untuk menjelaskan sebuah domain yang dapat digunakan sebagai landasan untuk sebuah *knowledge base* (Sachs, 2006).

Ontologi sendiri dapat didefinisikan sebagai suatu cara untuk mendeskripsikan arti dan relasi dari istilah-istilah. Deskripsi tersebut berisi *classes*, *properties*, dan *instances*. Deskripsi ini dapat membantu sistem komputer dalam menggunakan istilah-istilah tersebut dengan cara yang lebih mudah (Bekke, 1992).

Dengan menggunakan OWL, kita dapat menambah *vocabulary* tambahan di samping semantik formal yang telah dibuat sebelumnya menggunakan XML, RDF, dan RDF Schema. Hal ini sangat membantu penginterpretasian mesin yang lebih baik terhadap isi Web (Wicaksana, 2006).

2.5. Simple Protocol and RDF Query Language (SPARQL)

Simple Protocol and RDF Query Language (SPARQL) adalah sebuah bahasa untuk mengekspresikan *query* untuk mengakses data pada RDF atau OWL. Hasil query dari SPARQL dapat mengembalikan data dalam format XML, JSON, RDF, dan HTML.

Berikut adalah contoh sederhana dari implementasi SPARQL.

Data :

```
@prefix  foaf:  <http://xmlns.com/foaf/0.1/> .

_:a    foaf:name   "Alice" .
_:a    foaf:knows  _:b .
_:a    foaf:knows  _:c .

_:b    foaf:name   "Bob" .

_:c    foaf:name   "Clare" .
_:c    foaf:nick   "CT" .
```

Gambar 2.2. Data Awal
Sumber : http://www.w3.org/TR/rdf-sparql-query/#select

Query :

```
PREFIX foaf:    <http://xmlns.com/foaf/0.1/>
SELECT ?nameX ?nameY ?nickY
WHERE
  { ?x foaf:knows ?y ;
       foaf:name ?nameX .
    ?y foaf:name ?nameY .
    OPTIONAL { ?y foaf:nick ?nickY }
  }
```

Gambar 2.3. Contoh SPARQL *Select Query*
Sumber : http://www.w3.org/TR/rdf-sparql-query/#select

Hasil :

nameX	nameY	nickY
"Alice"	"Bob"	
"Alice"	"Clare"	"CT"

Gambar 2.4. Hasil SPARQL S*elect Query*
Sumber : http://www.w3.org/TR/rdf-sparql-query/#select

2.6. Alasan Penggunaan Ontologi

Terdapat tiga alasan untuk menggunakan ontologi (Sachs, 2006), yaitu:

1. Menjelaskan suatu domain secara eksplisit

   Memberikan struktur hierarki dari istilah untuk menjelaskan sebuah domain dan bagaimana mereka berhubungan.

2. Berbagi pemahaman dari informasi yang terstruktur

   Perangkat lunak dapat menggunakan kumpulan informasi untuk menjawab permintaan *user* atau sebagai data *input* untuk aplikasi yang lain.

3. Penggunaan ulang domain pengetahuan

   Dengan digunakannya ontologi, suatu sistem dapat mengintegrasikan dengan beberapa ontologi yang sudah ada.

Selain itu, menurut Jasper (1999), dengan digunakannya ontologi sebagai bagian dari aplikasi, pemeliharaan menjadi lebih mudah. Secara eksplisit penggunaan ontolgi dapat meningkatkan dokumentasi sistem yang tentunya dapat mengurangi biaya perawatan.

2.7. Ontologi Friend of a Friend

Friend of a Friend (FOAF) adalah ontologi yang ditujukan untuk menghubungkan orang dan informasi dengan menggunakan web, tanpa memperhatikan apakah informasi berada dalam pikiran manusia, dalam dokumen fisik atau digital, atau dalam bentuk data faktual yang dapat dihubungkan. Ontologi FOAF menggambarkan pandangan sederhana dari alam semesta dan informasi jaringan yang berbasis web untuk menghubungkan dan berbagi deskripsi secara independen yang ada di dunia dan saling berhubungan. FOAF tidak bersaing dengan situs jejaring sosial atau situs yang berorientasi hubungan sosial manusia, melainkan memberikan pendekatan di mana berbagai web yang berbeda dapat saling berbagi informasi satu sama lain (Dan & Miller, 2010).

FOAF menyediakan *framework* untuk merepresentasikan informasi mengenai informasi karakteristik seseorang, minat, dan relasi seseorang dengan hubungan sosialnya. FOAF diidentifikasikan dengan *namespace* URI 'http://xmlns.com/foaf/0.1/'. Beberapa *class* yang didefinisikan oleh ontologi FOAF (http://xmlns.com/foaf/spec/) :

1. **foaf:Agent** : *Class* yang merepresentasikan *legal agent*, orang, grup, software, atau artifak fisik.
2. **foaf:Person** : *Class* yang merepresentasikan orang. Merupakan *sub-class* dari foaf:Agent.
3. **foaf:Organization** : *Class* yang merepresentasikan suatu organisasi, institusi sosial seperti perusahaan, organisasi sosial, dan lain-lain.

4. **foaf:OnlineAccount** : *Class* yang merepresentasikan penyedia layanan online.

2.8. Ontologi Musik

Ontologi Musik menyediakan *vocabularies* untuk menghubungkan berbagai informasi yang berhubungan dengan musik, untuk mempublikasikan data mengenai musik dan menghubungkan dengan data yang sudah tersedia (Raimond *et al*., 2006). Ontologi Musik merupakan ontologi berbasis FOAF yang mempertimbangkan pengguna ontologi tersebut untuk mengekspresikan selera konsumen musik dan hubungan antara pengguna tersebut. Menurut Yves Raimond, untuk mendeskripsikan genre musik, digunakan repositori DBpedia yang secara otomatis diekstrak dari Wikipedia (Raimond *et al*., 2006).

Ontologi musik diidentifikasi dengan *namespace* URI 'http://purl.org/ontology/mo/'. Beberapa *class* yang didefinisikan Ontologi Musik.

1. **mo:Performance** : *Class* yang merepresentasikan sebuah acara pertunjukan musik.
2. **Mo:Genre** : *Class* yang merepresentasikan *expressive style* suatu musik

2.9. Sistem Rekomendasi

Sistem rekomendasi adalah *software* aplikasi yang menyediakan informasi untuk seseorang yang membutuhkan informasi yang sesuai dengan kebutuhan tersebut. Istilah "sistem rekomendasi" muncul sebagai evolusi dari logika penelitian dalam sistem *information retrieval* (IR). Evolusi tersebut

memiliki fitur utama pada penekanan pada konsep definisi dan representasi *query*. Sistem rekomendasi dianggap sebagai *information filtering system* (Raimond, 2008).

Terdapat dua pendekatan utama utuk merekomendasikan item kepada *user*, yaitu *collaborative filtering* dan *content-based filtering*.

1. **Collaborative Filtering**

   Menggunakan timbal balik dari *user* untuk meningkatkan kualitas materi yang dipresentasikan kepada *user*. Untuk memperoleh timbal balik terdapat dua cara, eksplisit atau implisit. Timbal balik secara eksplisit didapat dari *rating* yang diberikan oleh *user* maupun berupa anotasi dari *user*. Sedangkan timbal balik secara implisit didapat dari kebiasaan *user* tersebut. Yang perlu diperhatikan dari pendekatan ini adalah satu-satunya cara untuk merekomendasikan *item* yang baru, *user* tersebut harus terlebih dahulu melakukan rating atau melakukan *review* suatu *item* (Linden *et al*., 2003).

2. **Content-Based Filtering**

   Menggunakan ekstraksi informasi yang berguna dari suatu item yang telah dimiliki oleh *user* yang bisa merepresentasikan kebutuhan *user*. Pendekatan ini

menyelesaikan keterbatasan pada *collaborative filtering* di mana pendekatan ini dapat merekomendasikan *item* baru walaupun belum ada timbal balik dari *user* dengan cara membandingkan setiap *item* yang dimiliki oleh *user* dan melakukan kalkulasi kemiripan antar *item*. Dalam sistem rekomendasi musik, aplikasi akan melakukan ekstraksi file mentah audio dan melakukan komputasi diantara musik tersebut (Celma *et al*., 2007).

2.10. DBpedia

DBpedia merupakan *open source data set* berbasis web dalam bentuk RDF triples berbasis data dari Wikipedia. Tujuan utama dari *knowledge* tersebut adalah memungkinkan pengguna untuk menggunakan data lengkap Wikipedia dan menggunakan data antar *website*. DBpedia mendeskripsikan lebih dari 3,5 juta benda yang dibagi ke dalam beberapa kategori, misalnya *people, places, music albums, films, video games, organization, species,* dan *disease* (Harbers, 2011).

Perbedaan penggunaan SPARQL *query* dan SQL adalah pada SPARQL *query* dimungkinkan untuk mendefinisikan "*type*" *of* "*thing*" yang akan menghasilkan data yang sesuai dengan "*type*" *of* "*thing*" tersebut disebabkan tidak ada metode standar untuk mendefinisikan sesuatu, perbedaan *template* mempunyai kemungkinan menggunakan nama yang berbeda untuk satu atribut, contohnya adalah "*birthplace*" dan "*placeofbirth*". Untuk mengakses data dalam DBpedia, digunakan *query* SPARQL yang akan mengembalikan data dalam bentuk XML atau bentuk lainnya. Salah satu media untuk melakukan akses data

yaitu melalui OpenLink Virtuoso *browser*, yang memungkinkan pencarian data pada DBpedia dengan menggunakan teks singkat (Harbers, 2011).

2.11. Protégé

Protégé merupakan *open-sorce platform* yang menyediakan fungsi sebagai ontologi editor untuk membangun domain model dan aplikasi *knowledge based*. Protégé mengimplementasikan struktur pemodelan *knowledge* dan dapat memvisualisasikan hasil ontologi dalam berbagai format, yaitu OWL, RDF(s), dan XML. Sebuah ontologi digambarkan sebagai konsep dan hubungan-hubungan yang penting dalam domain yang khusus, yang menyediakan kosakata dalam domain tersebut (Nuriana, 2009).

Pada Protégé terdapat dua cara untuk melakukan pemodelan ontologi yaitu:

1. **Protégé Frame Editor**

   User dapat membangun ontologi dalam *frame-based* sesuai dengan Ontologi Knowledge Base Connectivity Protocol (OKBC). Pada pemodelan Frame Editor, sebuah ontologi terdiri dari seperangkat *class* yang terorganisir pada suatu hierarki yang merepresentasikan sebuah domain, seperangkat slot yang berhubungan dengan *class* serta *instance* dari tiap *class* tersebut.

2. **Protégé OWL**

   Protégé-OWL *editor* memungkinkan pengguna untuk membangun sebuah ontologi untuk web semantik, khususnya

berdasarkan standar bahasa yang telah disahkan oleh World Wide Web Consurtium (W3C). Protégé-OWL editor memungkinkan *user* untuk:

1. Mengambil dan menyimpan OWL dan ontologi RDF
2. Mengubah dan memvisualisasikan *class*, *properties*, dan Semantic Web Rule Language (SWRL)
3. Menjabarkan karakteristik *class* secara logis sebagai ekspresi OWL
4. Mengeksekusi penalaran seperti *description logic classifier*
5. Mengubah OWL *indivisual* untuk web semantik

Protégé-OWL berhubungan erat dengan Apache Jena. Berdasarkan keterangan dalam *website* resmi Apache Jena, pada Agustus 2005, Protégé-OWL resmi terintegrasi dengan Jena.

2.12. ARC2 RDF Store

ARC adalah PHP *library* yang menyediakan fungsi untuk mengolah RDF juga menyediakan *triplestore* berbasis MySQL. Berdasarkan keterangan yang diperoleh dari halaman resmi pada Github, ARC dibangun pada tahun 2004 sebagai sistem untuk melakukan *parsing* dan *serializing* file RDF/XML. ARC merupakan open source PHP *script* ringan untuk optimisasi pengembangan RDF pada *hosted web service* (Nowack, 2006).

Pada tahun 2011 diperkenalkan ARC2. ARC2 berkembang menjadi salah satu *framework* yang paling banyak digunakan. ARC2 berada bawah lisensi World Wide Web Consurtium (W3C) dan GNU General Software License.

Penelitian yang dilakukan oleh Benjamin Nowack (2006), menunjukkan bahwa ARC merupakan *framework* yang memiliki *performance* paling cepat dibandingkan *framework* RAP dan SimpleRdfParser.

	RAP	SimpleRdfParser	ARC
Processing (total)	61 s	12 s	6 s
Triples/s (total)	~141	~717	~1434

Gambar 2.2. Tabel Perbandingan *Performance*

Gambar 2.2 menunjukkan tabel hasil pengujian ketika melakukan *parsing* FOAF yang mengandung 8601 *triples*. Dari pengujian tersebut didapatkan fakta bahwa ARC memiliki *performance* terbaik dibandingkan RAP maupun SimpleRdfParser.

Gambar 2.3, Gambar 2.4, dan Gambar 2.5 menggambarkan contoh tampilan SPARQL *query* untuk melakukan *insert select*, dan *delete*.

```
INSERT INTO <http://example.com/inferred> CONSTRUCT {
  ?s foaf:knows ?o .
}
WHERE {
  ?s xfn:contact ?o .
}
```

Gambar 2.3. *Insert Query* pada ARC
Sumber : https://github.com/semsol/arc2/wiki/SPARQL-

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT * WHERE {
  ?x a foaf:Person ;
     foaf:name ?name ;
     foaf:knows ?y .
  FILTER regex(?name, "^A")
}
ORDER BY ASC(?name)
LIMIT 10
```

Gambar 2.4. *Select Query* pada ARC
Sumber : https://github.com/semsol/arc2/wiki/Stand-alone-SPARQL-Parsing

```
DELETE {
 <#foo> <bar> "baz" .
 <#foo2> <bar2> ?any .
}
```

Gambar 2.5. *Delete Query* pada ARC
(Sumber : https://github.com/semsol/arc2/wiki/SPARQL-)

2.13. Aplikasi Facebook

Pada tahun 2007, Facebook meluncurkan sebuah *platform* untuk pengembangan aplikasi Facebook. Dengan diluncurkan platform tersebut, Facebook telah membangun yang memungkinkan *developer* untuk membangun aplikasi eksternal yang membuat seluruh pengguna Facebook bisa berinteraksi dengan sesama pengguna dengan berbagai cara baru. Dengan adanya platform untuk aplikasi *web-based* maupun aplikasi *desktop*, Facebook berhasil memposisikan dirinya sebagai "pemain utama" dalam bidang *socio-technical development* (Wayne, 2008).

Untuk membuat aplikasi yang baik, *developer* harus mengikuti aturan-aturan yang telah ditentukan oleh Facebook. Ketika bekerja dengan API sosial,

developer aplikasi harus mempertimbangkan bahwa kode program akan secara otomatis mempromosikan diri sendiri. Kode program yang ditulis memungkinkan semua orang untuk berbagi informasi secara cepat dan efektif. Esensi dari membangun aplikasi sosial adalah akan menjadikan pengembang menjadi *programmer* sekaligus *marketer*, karena aplikasi yang dibangun akan "menjual" produk itu sendiri (Jesse, 2011).

Facebook menyediakan beberapa cara yang memungkinkan *developer* aplikasi untuk membangun aplikasi pada platform Facebook. *Developer* dapat mengakses *method* dan mengimplementasikan pada *website*. Beberapa *tools* Facebook, yaitu (Jesse, 2011):

1. **Facebook Graph API.**

Facebook menyediakan sampel Javascript API. *Developer* dapat mengakses komentar, teman, *likes*, dan *streaming* berita dari *website*.

2. **Social Plugins**

Potongan kode HTML yang dapat diimplementasikan pada *website* yang dapat berfungsi sebagai Like *buttons*, *widgets*, dan streaming aktivitas terakhir yang dilakukan oleh *user* Facebook pada *website*.

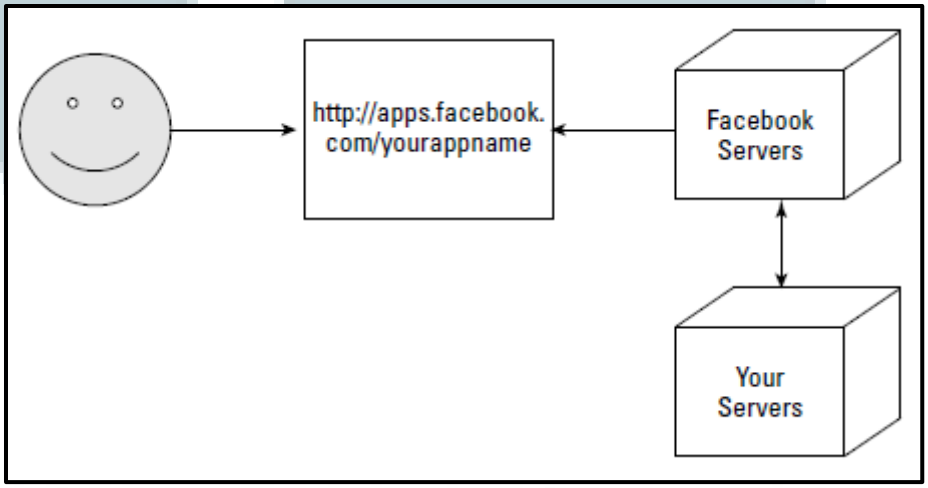
3. **Open Graph Protocol,**

Simpel *meta tags* yang berfungsi untuk mengatur *website* kita agar Facebook bisa *recognize website* sebagai bagian dari halaman Facebook.

4. Facebook Credits,

Sistem kredit yang disediakan oleh Facebook yang memungkinkan pengguna untuk membeli apapun dalam web. Sebuah *tool* yang sangat *powerfull* karena *customer* dapat membeli apapun yang kita jual dengan beberapa langkah singkat.

Terdapat dua macam *environment* dalam pengembangan aplikasi pada platform Facebook.

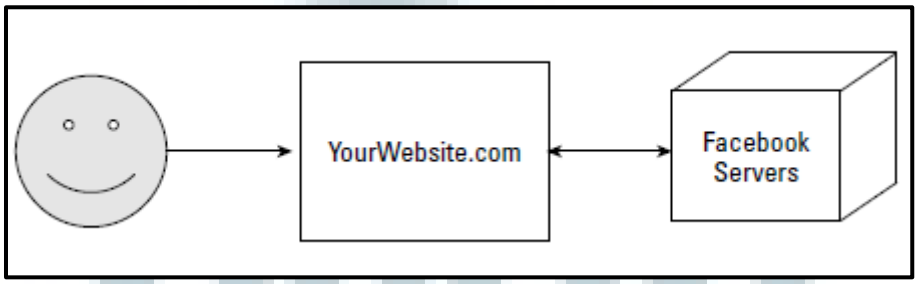


Gambar 2.6. Tipikal Data Flow Aplikasi Dalam Facebook.com
(Sumber : Facebook Application Development for Dummies)

Gambar 2.6 menjelaskan model *environment* pertama. Ketika web aplikasi kita di *host* di Facebook, sebenarnya aplikasi tersebut tidak benar-benar di-*hosting*, karena pada kenyataannya Facebook akan memanggil data ke server kita dan server akan memberikan isi data melalui iFrame HTML.

Gambar 2.7. Aplikasi Facebook Menggunakan Kanvas
(Sumber : http://developers.facebook.com/docs/guides/canvas/)

Aplikasi Facebook model *environment* pertama diunggah ke dalam sebuah halaman kanvas kosong yang akan diisi oleh aplikasi Facebook. Kanvas tersebut akan diisi dengan tampilan aplikasi dengan memanggil URL aplikasi yang telah didaftarkan. Ketika *user* mengakses halaman kanvas tersebut, kanvas akan memuat aplikasi Facebook melalui sebuah iFrame. Gambar 2.7 merupakan contoh tampilan aplikasi Facebook yang menggunakan *environment* pertama.



Gambar 2.8. Arsitektur Aplikasi Facebook yang Di-*hosting* Di Website
Sumber : Facebook Application Development for Dummies

Gambar 2.8 Menjelaskan model *environment* kedua, *user* langsung mengunjungi *website* dan secara otomatis *website* akan melakukan render simpel

JavaScript untuk melakukan *call back* kepada Facebook, dan Facebook akan mengembalikan data otentikasi informasi *user*.

UMN