**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**INSTITUT TEKNLOGI SEPULUH NOPEMBER**

**USULAN TUGAS AKHIR**

# IDENTITAS PENGUSUL

**NAMA : CHRISTIAN CANDRABIANTARA**

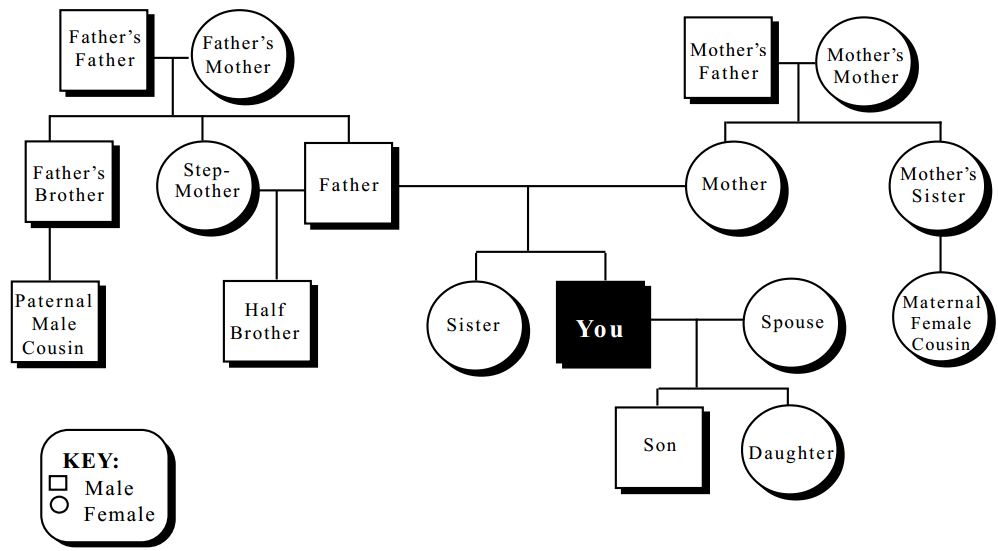
**NRP : 5109100132**

**DOSEN WALI : Diana Purwitasari, S.Kom, M.Sc.**

# JUDUL TUGAS AKHIR

“Rancang Bangun Aplikasi Visualisasi Silsilah Keluarga Berbasis Ontologi”

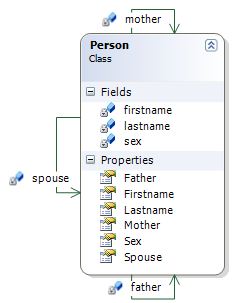
# LATAR BELAKANG



Gambar 3. Silsilah Pohon Keluarga

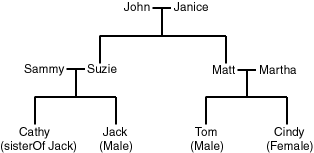
Silsilah keluarga merupakan suatu bagan yang menampilkan susunan keluarga dalam suatu pohon. Silsilah keluarga memiliki informasi yang menyatakan hubungan antar anggota. Hubungan kunci dalam silsilah keluarga adalah perkawinan, dan keturunan. Ikatan ini yang menghubungkan individu dalam pohon silsilah keluarga. Setiap perkawinan akan menggabungkan dua keluarga sehingga akan menyebabkan peningkatan kompleksitas secara ekponential seperti contoh yang ditunjukkan oleh gambar 3.1.

Dalam bidang kedokteran, silsilah keluarga digunakan untuk membantu diagnosa suatu penyakit seperti penyakit turunan. Silsilah keluarga juga bermanfaat bagi genealog untuk menggambarkan grafis pohon keluarga kompleks yang menunjukkan perkawinan dan perceraian, keluarga dibentuk kembali, adopsi, ikatan keluarga, dan lain-lain. Para ahli *genealog* dapat menggunakan silsilah keluarga untuk menemukan dan menganalisis fakta-fakta menarik tentang sejarah keluarga mereka, seperti pola penamaan, persaingan antar saudara, atau peristiwa penting seperti imigrasi.



Gambar 3. Kelas Person

Kita dapat membuat suatu perangkat lunak yang dapat menyatakan silsilah keluarga dan mencari relasi antar anggota keluarga dengan menggunakan database relational dimana setiap orang memiliki seorang ayah dan seorang ibu seperti pada Gambar 3.2. Dengan pemodelan kelas diatas kita dapat mencari relasi antar anggota keluarga. Seperti contoh berikut:



Gambar 3. Keluarga John

Contoh pada Gambar 3.3 :

1. Pengguna ingin mencari tahu siapa ayah dari Cathy:

Query database:

Select firstname from{

Select father as idF from family\_tree where firstname = “Cathy”

}as t1

where id\_anggota = t1.idF

1. Pengguna ingin mencari tahu siapa kakek dari pihak ibu Jack

SELECT firstname FROM{

SELECT father AS idGF FROM{

SELECT mother AS idM FROM family\_tree WHERE firstname = “Jack”

}AS t1, family\_tree

WHERE t1.idM = family\_tree.id\_anggota

}AS t2,family\_tree

WHERE t2.idGF = family\_tree.id\_anggota

1. Pengguna ingin mengetahui siapa saudara sepupu perempuan dari pihak ibu Jack

SELECT firstname FROM{

SELECT id\_anggota AS idChildGF FROM{anak-anak kakek kecuali ibu

SELECT father AS idGF,idM FROM{kakek

SELECT mother AS idM FROM family\_tree WHERE firstname = “Jack”

}AS t1, family\_tree

WHERE t1.idM = family\_tree.id\_anggota

} AS t2,family\_tree

WHERE family\_tree.father = t2.idGF AND family\_tree.id\_anggota <> t2.idM

} AS t3, family\_tree

WHERE ((t3.idChildGF = family\_tree.father) OR (t3.idChildGF = family\_tree.mother)) AND t3.sex = "woman"

Dari tiga contoh diatas dapat disimpulkan pencarian relasi antar anggota keluarga dapat dilakukan menggunakan database relational. Akan tetapi, membutuhkan *nested query* yang kompleks dan panjang sehingga pencarian relasi menggunakan database relational sangat sulit dilakukan seiring perkembangan silisilah keluarga yang berkembang semakin besar secara eksponential setiap terjadi perkawinan.

Oleh karena itu, dibutuhkan pemodelan data yang lebih mudah yaitu menggunakan ontologi. Ontologi merupakan suatu teori tentang makna dari suatu obyek, properti dari suatu obyek, serta relasi obyek tersebut yang mungkin terjadi pada suatu domain pengetahuan. Ontologi ialah sebuah spesifikasi dari sebuah konseptual, dengan kata lain ontologi adalah penjelasan dari sebuah konsep dan keterhubungannya dari sebuah ilmu tertentu. Ontologi sebagai katalog dalam Semantik Web.

Perangkat lunak untuk membuat silisilah keluarga sudah ada, akan tetapi tidak memiliki fitur mencari relasi anggota keluarga seperti perangkat lunak-perangkat lunak *desktop* yang tidak berbayar berikut ini:

* Ancestries
* Ahnenblatt
* Gramps

Perangkat lunak diatas memiliki fitur seperti:

* Menginput data diri
* Membuat relasi antar anggota keluarga (kebanyakan hanya relasi yang wajar saja)
* Visualisasi *family tree*
* Menyimpan *family tree* dengan beberapa tipe file
* Pencarian data individual, hanya tersedia data ayah,ibu, saudara dan anak dari individu yang dicari

Kekurangan dari fitur diatas ialah tidak adanya fitur pencarian relasi antar anggota keluarga dan hubungan keluarga tersedia hanya sebatas relasi yang wajar.

Perangkat lunak yang akan dibangun dalam tugas akhir ini dapat mencari anggota keluarga berdasarkan query dari pengguna. Perangkat lunak ini juga memiliki sistem input dan query yang tidak dimiliki oleh perangkat lunak-perangkat lunak yang sudah ada yaitu dengan menggunakan entri diagram. Selain itu, perangkat lunak ini juga dapat memetakan hubugan silsilah keluarga yang tidak wajar di Indonesia. Hubungan keluarga yang tidak wajar meliputi pernikahan poligami dan adopsi/anak angkat, seseorang yang menikah lagi karena pasangannya telah meninggal.

# RUMUSAN MASALAH

Detail permasalahan yang diangkat dalam tugas akhir ini adalah sebagai berikut:

* 1. Bagaimana menampilkan visualisasi input dan output silsilah keluarga dengan menggunakan *drag and drop* diagram silsilah keluarga beserta tampilan pohon silsilah keluarga?
  2. Bagaimana memberikan kesimpulan yang dapat memberikan informasi mengenai silsilah keluarga sesuai inputan pertanyaan pengguna?

# BATASAN MASALAH

Masalah yang dibahas pada tugas akhir ini dibatasi lingkupnya pada:

1. Aplikasi akan dibuat untuk aplikasi *desktop* menggunakan bahasa pemrograman Java.
2. Input dan query dilakukan dengan melakukan *drag and drop* diagram yang telah disediakan pada toolbox.
3. Kata-kata yang digunakan pengguna untuk menyatakan relasi seperti yang dijelaskan pada bagian ringkasan tugas akhir.
4. Jawaban sistem terhadap pertanyaan query pengguna berupa daftar nama anggota keluarga hasil dari pencarian

# TUJUAN PEMBUATAN TUGAS AKHIR

Tujuan dari penyusunan tugas akhir ini adalah:

* Merancang bangun aplikasi *desktop* untuk mencari hubungan kekeluargaan dalam suatu silsilah keluarga.
* Dengan adanya aplikasi ini pengguna tidak perlu pergi jauh kebelakang untuk melihat sejarah keluarga untuk menemukan seseorang yang hilang (dan mungkin memiliki keluarga lain tidak diketahui kedua belah pihak), tidak diakui atau hanya lupa

# MANFAAT TUGAS AKHIR

Manfaat dari tugas akhir ini adalah sebagai aplikasi yang memudahkan pengguna untuk mengetahui hubungan kekerabatan antar anggota dalam suatu silsilah keluarga sehingga dapat mempererat tali kekerbatan.

# TINJAUAN PUSTAKA

* 1. **Silsilah Keluarga**

Silsilah keluarga adalah suatu [bagan](http://id.wikipedia.org/wiki/Bagan) yang menampilkan hubungan [keluarga](http://id.wikipedia.org/wiki/Keluarga) (silsilah) dalam suatu [struktur pohon](http://id.wikipedia.org/wiki/Struktur_pohon). Data [genealogi](http://id.wikipedia.org/wiki/Genealogi) ini dapat ditampilkan dalam berbagai format. Salah satu format yang sering digunakan dalam menampilkan [silsilah](http://id.wikipedia.org/wiki/Silsilah) adalah bagan dengan [generasi](http://id.wikipedia.org/w/index.php?title=Generasi&action=edit&redlink=1) yang lebih tua di bagian atas dan generasi yang lebih muda di bagian bawah. Bagan keturunan yang menampilkan semua keturunan dari satu individu memiliki bagian yang paling sempit di bagian atas.

* 1. **Resource Description Framework (RDF)**

RDF merupakan sebuah framework untuk membuat pernyataan dalam sebuah form yang disebut sebagai triple. RDF memungkinkan untuk merepresentasikan informasi dari sebuah source dalam form dari sebuah graph-semantik web, yang disebut sebagai *Giant Global Graph*.

* 1. **RDF Schema (RDFS)**

menyediakan dasar-dasar istilah untuk RDF. Dengan RDFs, memungkinkan untuk membuat hirarki class dan propertinya.

* 1. **Web Ontologi Language (OWL)**

OWL memperluas RDFs dengan menambahkan konsep yang lebih canggih untuk mendeskripsikan semantik dari pernyataan RDFs. OWL memungkinkan untuk menambahkan sebuah konstrain, seperti kardinalitas, batasan nilai, karakteristik dari properti seperti transitif. OWL didasarkan pada logika sehingga memberikan kekuatan penalaran pada semantik web.

* 1. **SPARQL**

SPARQL adalah sebuah bahasa query RDF, ini dapat digunakan untuk query banyak data RDF(termasuk pernyataan RDF dan OWL). Bahasa query diperlukan untuk merujuk untuk informasi dari aplikasi web semantik.

## Ontologi

Ontologi adalah definisi dari pengertian dasar dan relasi kosa kata dari sebuah wilayah sebagaimana aturan dari kombinasi istilah dan relasi untuk mendefinisikan kosa kata. Buku *Semantic Web Technology* tahun 2006 menyebutkan bahwa Gruber mendefinisikan ontologi sebagai sebuah spesifikasi eksplisit dari konseptualisme. Definisi ini banyak digunakan oleh beberapa orang sebagai landasan dalam mendefinisikan ontologi.

Ontologi digunakan pada tugas akhir ini untuk memodelkan Hirarki Silsilah Keluarga yang akan diberikan oleh pengguna.

Menurut Gruber ontologi adalah spesifikasi eksplisit dari konseptualisasi. Ontologi dibangun dari *class, individual* dan *properties.*

***Class*** adalah sekumpulan sumber daya yang miliki karakter sama. *Axioms* mendeskripsikan *class.*

***Individuals*** adalah *instance* dari *class* yang spesifik.

***Properties*** adalah menghubungkan antar *individuals* atau *individuals* ke nilai dari data

* 1. **Ontologi Web Language (OWL)**

OWL adalah salah satu bentuk ontologi yang memang dirancang dengan tujuan untuk digunakan oleh aplikasi yang perlu memproses isi informasi ketimbang menampilkan informasi untuk konsumsi manusia. OWL merupakan rekomendasi W3C (World Wide Web Consortium) dalam penulisan ontologi untuk web untuk web semantik. OWL dituliskan dalam syntax XML (eXtended Markup Language). Selain XML dalam OWL digunakan juga bahasa XMLS (XML Schema), RDF (Resource Description Framework) dan RDFS (RDF Schema) dan OWL itu sendiri.

OWL digunakan awalnya karena adanya ide dan visi dari web semantik. Visi untuk masa depan yang mana informasi diberikan secara eksplisit yang membuat mesin atau komputer bisa mengerti dan mampu memproses informasi itu secara otomatis dan mampu mengintegrasikan informasi yang tersedia di web. Komputer diharapkan mampu melakukan proses penalaran sebagaimana yang dilakukan manusia. Ontologi merupakan bahasa  untuk mesin yang digunakan untuk merepresentasikan suatu informasi secara eksplisit. Ontologi juga mendukung adanya penalaran. Itulah mengapa OWL penting dalam mewujudkan visi web semantik.

* 1. **Penalaran**

Penalaran adalah sebuah proses kognitif dalam pencarian sebuah kepercayaan, kesimpulan, tindakan, atau perasaan. Kita menggunakan *rule* untuk melakukan proses penalaran dengan mencocokan kenyataan dalam ontologi. *Rule* adalah bentuk sebuah implikasi antara *body* dan *head*. *Body* adalah pernyataan/*pattern* sebelum, kondisi disesuaikan dengan *rule* yang spesifik. *Head* adalah pernyataan atau *pattern*, pernyataan yang mungkin diperkirakan, dinamakan sisi tangan kanan *rule* atau *Right Hand Side*(RHS).

Dalam bentuk lain, *rule* dari sebuah *rule-based sistem* disebut sebagai pernyataan “IF-THEN”. Jika semua kondisi sesuai, selanjutnya RHS akan dijalankan. Ada dua jenis tipe penalaran, yaitu ontologi penalaran dan *rule-based* penalaran. OWL penalaran mempunyai kemampuan menduga melalui karakteristik dari properti seperti simetri, invers, dan transitif.

Ada beberapa bahasa *rule* untuk membuat sebuah rule yang menyimpulkan sebuah pengetahuan, antara lain Semantik Web Rule Language(SWRL), JESS Rule, Jena Rule, dan Semantik Query-enhanced Web Rule Language(SQWRL).

* 1. **SWRL**

SWRL adalah anjuran bahasa *rule* untuk semantik web, mengkombinasikan OWL DL dan OWL Lite dengan *rule* markup language(RuleML). RuleML adalah markup language yang dikembangkan untuk mempercepat kedua *rule* *forward*(*bottom-up*) dan *backward*(*top-down*) dalam XML untuk pengambilan kesimpulan, penulisan ulang dan kegiatan inferential-transformational lebih jauh.

* 1. **JENA RULE**

Jena[5] adalah API Java untuk menangani model ontologi(OWL) dan penalaran. Jena menggunakan model RDF dan N3 untuk merepresentasikan pengetahuan dalam model. Jena dapat digunakan untuk membuat dan memanipulasi graphs RDF.

* 1. **SQWRL**

SQWRL adalah SWRL berbasis *query language* yang dapat digunakan untuk query ontologi OWL. SQWRL menyediakan operasi-opereasi seperti SQL untuk memformat pengetahuan dari ontologi OWL SWRL. Setelah dilakukan penalaran sistem akan melakukan penyimpanan RDF ke dalam database Oracle.

## Rancang bangun perangkat lunak

Rancang bangun perangkat lunak merupakan tahap-tahap teknis untuk membangun perangkat lunak yang melingkupi perencanaan, analisis sistem, implementasi, serta aktivitas pengujian dan pemeliharaan perangkat lunak.

Rancang bangun perangkat lunak diperlukan untuk menentukan konsep, strategi, dan praktik yang baik diterapkan untuk menciptakan perangkat lunak yang berkualitas tinggi, sesuai anggaran biaya, mudah dalam pemeliharaannya, serta tidak membutuhkan waktu yang lama dalam pembangunannya[1]. Beberapa model rancang bangun perangkat lunak yang terkenal dan banyak dipakai antara lain model air terjun serta model iterasi.

# RINGKASAN ISI TUGAS AKHIR

Aplikasi yang akan dibangun pada tugas akhir ini adalah sebagai aplikasi yang memudahkan pengguna untuk mengetahui hubungan kekerabatan antar anggota dalam suatu silsilah keluarga.

Berikut ini adalah daftar istilah yang akan digunakan dalam perangkat lunak ini:

1. Orang
2. Jenis Kelamin
3. Nama
4. Nama Depan
5. Nama Keluarga
6. Lahir
7. Meniggal
8. Ayah
9. Ibu
10. Orang tua
11. Anak
12. Anak laki-laki
13. Anak perempuan
14. Pasangan
15. Menikah
16. Cerai
17. Kakek
18. Nenek
19. Cucu perempuan
20. Cucu laki-laki
21. Alamat
22. Kota
23. Negara
24. Umur
25. Anak angkat
26. Anak tiri
27. Adalah
28. Pria
29. Wanita

Berikut ini adalah contoh model silsilah keluarga yang akan dibangun:

Gambar 10.

Pada Gambar 10.1, Pengguna mendefinisikan Robert memiliki tipe person

Gambar 10.

Pada Gambar 10.2 menjelaskan hubungan antara anggota keluarga dimana Robert memiliki kakek bernama David. David memiliki kakek bernama William. Oleh karena itu, dapat disimpulkan bahwa Robert memiliki kakek bernama William.

Gambaran arsitektur:

Gambar 10. Gambaran Umum Arsitektur

Berikut ini penjelasan Gambar 10.3 mengenai alur kerja program sebagai berikut:

* Admin memasukkan data model ke dalam aplikasi
* Pengguna memasukkan input dan query melalui input visual yaitu dengan cara *drag and drop* diagram yang telah disediakan oleh aplikasi
* Aplikasi akan menterjemahkan visual query dan membuat rdf query
* Hasil rdf query akan diproses Jena untuk melakukan query
* Jena akan menyimpan atau mengambil data dari database
* Hasil query Jena akan dikembalikan ke aplikasi
* Aplikasi akan menampilkan hasil query tersebut ke pengguna  
  1. Input :

Pengguna memasukkan orang. Pilihan subyek dan obyek yang diberikan oleh sistem berdasarkan anggota keluarga yang telah dimasukkan pengguna. Sedangkan untuk predikat menyatakan relasi menggunakan kata-kata seperti yang telah dijelaskan pada daftar istilah yang digunakan. Pengguna memasukkan input dengan cara melakukan *drag and drop* diagram yang telah disediakan oleh sistem (Subjek, Predikat, dan Objek). Selain itu pengguna juga dapat memberikan input dari *textbox* dan *dropdownlist* yang telah disediakan.

Contoh:

Pengguna ingin memasukkan data relasi keluarga seperti berikut:

Budi mempunyai ibu bernama Ani

Ani mempunyai ayah bernama Ali

Berikut ini input relasi anggota dari pengguna:

Gambar 10.

Lalu klik tombol “Add”

Gambar 10.

Lalu klik tombol “Add”

* 1. Proses :

Admin memasukkan data model ke dalam aplikasi. Pengguna memasukkan input dan query melalui input visual yaitu dengan cara *drag and drop* diagram yang telah disediakan oleh aplikasi. Aplikasi akan menterjemahkan visual query dan membuat rdf query. Hasil rdf query akan diproses Jena untuk melakukan query. Jena akan menyimpan atau mengambil data dari database. Hasil query Jena akan dikembalikan ke aplikasi. Aplikasi akan menampilkan hasil query tersebut ke pengguna

* 1. Output :

Terdapat graph yang menampilkan keseluruhan silsilah keluarga yang terjadi. Jika pengguna ingin melakukan query, pengguna melakukan hal yang sama seperti saat input relasi keluarga hanya saja pada bagian obyek pengguna memilih tanda “?”. Berikut ini contoh query dari pengguna

Gambar 10. Query Diagram User

Sistem akan menampilkan hasil query dari pengguna berupa daftar nama anggota keluarga hasil dari pencarian.

# METODOLOGI

## Penyusunan proposal tugas akhir

Pada tahap ini penulis menyusun proposal tugas akhir sebagai langkah awal dalam pengerjaan tugas akhir. Pada proposal ini penulis menggagas penulisan tugas akhir untuk merancang bangun aplikasi untuk mengetahui silsilah keluarga menggunakan ontologi*.*

## Studi literatur

Pada tahap ini dilakukan pengumpulan informasi yang diperlukan untuk perancangan sistem. Informasi tersebut dapat diperoleh dari literatur, paper maupun buku bahasa pemrograman.

## Analisis dan desain perangkat lunak

Analisis kebutuhan dan perancangan sistem dilakukan untuk merumuskan solusi yang tepat dalam pembuatan aplikasi serta kemungkinan yang dapat dilakukan untuk mengimplementasikan rancangan tersebut. Fase desain meliputi arsitektur perangkat lunak yang digunakan, desain class-class yang terlibat dalam aplikasi, desain basis data, dan lain-lain.

## Implementasi perangkat lunak

Pada tahap ini, desain perangkat lunak yang telah dibuat pada tahap sebelumnya, diimplementasikan ke dalam bentuk kode program. Pembangunan aplikasi dilakukan dengan menggunakan bahasa pemrograman yang digunakan adalah Java. Pada fase ini, penulis menggunakan tools *Protege* sebagai alat bantu untuk proses penalaran.

## Pengujian dan evaluasi

Pada tahap ini dilakukan uji coba terhadap perangkat lunak yang dibuat. Tujuan uji coba perangkat lunak adalah untuk menemukan kesalahan-kesalahan (*bug*) sedini mungkin sehingga dapat diperbaiki sesegera mungkin [3]. Data silsilah dan query pertanyaan yang dimasukkan oleh pengguna berupa kalimat dengan aturan-aturan seperti yang telah dijelaskan pada bagian ringkasan tugas akhir.

## Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

1. Pendahuluan
   1. Latar Belakang
   2. Rumusan Masalah
   3. Batasan Tugas Akhir
   4. Tujuan
   5. Metodologi
   6. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Uji coba dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

# JADWAL KEGIATAN

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tahapan | 2012 | | | | | | | | | | | | | 2013 | | | |
| Oktober | | | | Nopember | | | | Desember | | | | | Januari | | | |
| Penyusunan Proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Studi Literatur |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Perancangan sistem |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Uji coba dan evaluasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Penyusunan buku |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# DAFTAR PUSTAKA

1. **Sommerville, Ian.** *Perangkat lunak Engineering.* Boston : Addison-Wesley, 2007. 9780321313799.
2. **Horrocks, Ian.** [Dikutip : 30 September 2012] <http://www.kde.cs.uni-kassel.de/conf/iccs05/horrocks_iccs05.pdf>
3. **Godbole, Nina.** *Perangkat lunak Quality Assurance: Principles and Practice.* Oxford : Alpha Science, 2007. 1-84265-176-5.
4. **Schlagwein, Daniel.,Schoder, Detlef., Fischbach Kai**. 2011. Social Information Sistems: Review, Framework, and Research Agenda

**[5] Apache Jena.** Jena documentation. <http://jena.apache.org/documentation/>. Diakses pada 24 Oktober 2012