

# VISUALISASI DAN DOKUMENTASI METODE LWBA SECARA OTOMATIS DAN KONSISTEN UNTUK *REQUIREMENT ENGINEERING*

## ABSTRAK

*Visualisasi dalam requirement engineering dibutuhkan agar penyebab permasalahan menjadi lebih terajaki (traceable). Di samping visualisasi, dibutuhkan juga deskripsi dari visualisasi. Pada fase requirement engineering, kadangkala terjadi inkonsistensi penyampaian informasi. Oleh karena itu, dibutuhkan visualisasi dan dokumentasi informasi secara otomatis dan konsisten untuk menelusuri penyebab ketidakpuasan. Untuk itu diperlukan suatu perangkat bantu agar memperoleh visualisasi dan dokumentasi secara otomatis dan konsisten. Dengan dikembangkannya perangkat bantu untuk visualisasi dan pendeskripsian informasi LWBA dengan menggunakan metode BBSDM diharapkan penyajian informasi menjadi lebih baik secara otomatis.*

*Kata Kunci: LWBA, Graf, Visualisasi, Dokumentasi, Graphviz*

Reza Chandra<sup>1)</sup>  
I Made Wiryana<sup>2)</sup>

Fakultas Ilmu Komputer  
Universitas Gunadarma

reza\_chan@staff.gunadarma.ac.id<sup>1)</sup>  
mwiryana@staff.gunadarma.ac.id<sup>2)</sup>

## PENDAHULUAN

*Software* (perangkat lunak) dikembangkan melalui siklus hidup pengembangan sistem. Beberapa metode dan siklus hidup dikenal dalam pengembangan perangkat lunak dan semuanya selalu memiliki tahap *requirement analysis*.

*Requirement Engineering* merupakan fase awal dari proses rekayasa perangkat lunak. *Requirement engineering* dilakukan untuk mengetahui masalah yang akan dipecahkan atau diberikan solusinya dalam pengembangan suatu sistem. Kegagalan suatu pengembangan sistem sering disebabkan oleh ketidaktepatan tahapan *requirement engineering*. Salah satu penyebab dari masalah yang ada adalah terdapat ketidakpuasan dalam penggunaan sistem. Oleh sebab itu, penyebab dari ketidakpuasan harus dihilangkan atau diminimalkan.

Terdapat beberapa jenis ketidakpuasan dalam rekayasa perangkat lunak, yaitu jenis ketidakpuasan yang bisa dikendalikan dan jenis ketidakpuasan yang tidak bisa dikendalikan. Ketidakpuasan yang tidak bisa dikendalikan dapat terjadi pada kondisi normal, kondisi teknik yang meliputi infrastruktur dan prosedur, dan kondisi manajemen yang meliputi kebijakan pengambilan keputusan.

Visualisasi dalam *requirement engineering* dibutuhkan agar penyebab permasalahan menjadi lebih terajaki (*traceable*) oleh pihak-pihak yang terlibat dalam pengembangan sistem. Di samping visualisasi, dibutuhkan juga deskripsi dari visualisasi. Yang menjadi permasalahan dari visualisasi dan deskripsi *requirement engineering* adalah seringkali timbul inkonsistensi dalam penyajian informasi (Chandra, 2013).

Dalam penyajian informasi secara manual, mungkin sekali terjadi inkonsistensi penyajian informasi, terutama dalam hal pendeskripsian masalah. Oleh sebab itu, dibutuhkan visualisasi dan dokumentasi dalam penyampaian informasi secara otomatis untuk menelusuri penyebab masalah yang berakibat ketidakpuasan dalam rekayasa perangkat lunak.

Untuk memperoleh visualisasi dan dokumentasi masalah secara otomatis dan

konsisten maka diperlukan alat bantu yang dapat menghasilkan visualisasi dan pendeskripsian secara otomatis. Oleh karena itu, dikembangkan suatu alat bantu yang memudahkan visualisasi dan dokumentasi untuk metode *Lightweight Why Because Analysis (LWBA)*.

*Lightweight Why Because Analysis* merupakan pengembangan dari metode *Why Because Analysis (WBA)* dan diperkenalkan sebagai perangkat bantu untuk pengembangan sistem yang berkelanjutan (*sustainable*). WBA mempertimbangkan aspek non-teknis yaitu manusia, kultur, organisasi, regulasi pada sistem nyata. WBA ini tidak terkait pada perangkat bantu khusus ataupun paradigma pemrograman apa saja, berbeda dengan UML yang terkait dengan paradigma pemrograman objek (OOP). WBA menganalisis penyebab awal, dengan cara mengetahui faktor penyebab yang diperlukan (*Necessary Caused Factor-NCF*).

LWBA disebut "*lightweight*" (ringan) karena analisis ini tidak mendetail dan tidak "formal" seperti WBA. LWBA adalah analisis "semi-formal" yang menyelidiki kendala-kendala tanpa cara yang menghakimi. LWBA juga digunakan untuk memahami kebutuhan dari suatu metode pengembangan yang baru dengan bertumpu pada keberlanjutan (*sustainability*). (Zave, 1997)

Ide utama dari analisis LWBA adalah mengenali faktor kausal yang dapat diganti untuk membuat sebuah sistem menjadi lebih baik. Analisis pada LWBA juga mencakup pada aspek nonteknis, misalkan sumber daya manusia, regulasi dan organisasi. Analisis LWBA mempunyai beberapa karakteristik, antara lain merupakan analisis yang bersifat semi-formal. Berbeda dengan analisis WBA yang bersifat formal.

## METODE PENELITIAN

Metode penelitian yang digunakan adalah studi pustaka dan pengembangan solusi. Studi pustaka yang dilakukan adalah dengan cara penelusuran jurnal, artikel dan tutorial yang terkait dengan *requirement engineering, web parsing, Lightweight Why Because Analysis*

(LWBA), Bandung Bondowoso System Development Method (BBSDM) serta perangkat bantu yang digunakan dalam penelitian ini.

## HASIL DAN PEMBAHASAN

### Perancangan Sistem

Penggunaan model formal, yang merupakan cara abstrak untuk menentukan sistem komputer, merupakan realitas industri. Penggunaan notasi abstrak sebelum memulai implementasi sangat membantu untuk pemahaman masalah yang lebih baik.

Sebagai permulaan dalam pengembangan perangkat lunak, dibutuhkan suatu *requirement* yang menghasilkan dokumen berkualitas tinggi sebagai masukan dari rekonstruksi model. Namun demikian, jika *requirement* telah didapat masih merupakan tugas yang sulit untuk membangun model dan implementasi yang mencerminkan masalah. Transisi dari persyaratan untuk analisis atau model desain adalah proses manual, dan karena itu rawan kesalahan. (Cabral and Sampaio, 2008)

Untuk memudahkan penelusuran penyebab permasalahan dalam fase *requirement engineering*, maka dibuatlah suatu visualisasi grafis. Namun kebanyakan visualisasi di buat dengan cara yang manual. Hal ini tentu bisa menimbulkan inkonsistensi dalam perancangan sistem seperti timbulnya kerangkapan analisis sistem. Berkaitan dengan hal tersebut maka dikembangkanlah suatu perangkat bantu untuk mencegah terjadinya inkonsistensi dan informasi menjadi lebih mudah ditelusuri.

### Konteks

Penggunaan perangkat bantu ini akan digunakan dalam konteks :

- 1 Membantu proses pengembangan sistem.
- 1 Melakukan validasi atau kecocokan antara pengembang dan klien.
- 1 Membantu proses kontrak.

## Konstraint

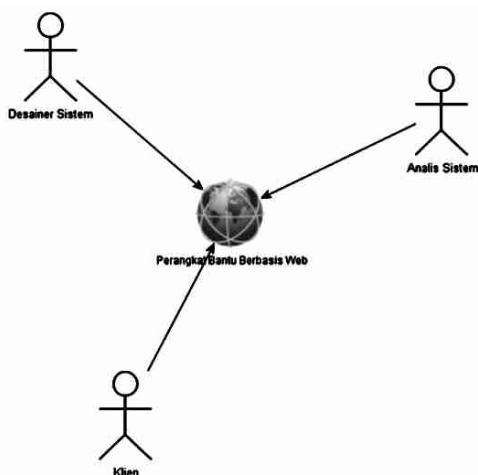
Perangkat bantu ini dikembangkan dengan pertimbangan konstraint :

- 1 Pengembang dan kustomer terkendala masalah geografis.
- 1 Waktu yang terbatas dalam pengembangan sistem.
- 1 Harus mudah digunakan.

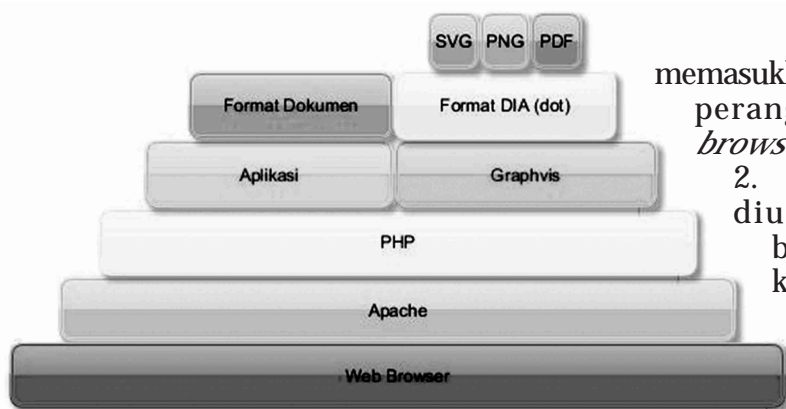
Dengan pertimbangan konstraint ini maka perangkat bantu ini dikembangkan berbasis web karena dapat berjalan di platform yang berbeda seperti Unix, Linux, Machintosh atau Windows dan pemasukkan data dibuat dalam bentuk *spreadsheet*. Perangkat bantu ini akan menampilkan notasi-notasi yang sederhana yang dapat disesuaikan dengan kebutuhan pengembangan sistem.

## Environment

Sistem ini dapat diterapkan untuk lingkungan yang memiliki keterbatasan waktu pengembangan sistem dengan jumlah SDM yang terbatas untuk memecahkan suatu masalah. Adapun penggunaannya adalah orang-orang yang bekerja di lingkungan pengembangan sistem.



Gambar 1. Lingkungan Pemakai Perangkat Bantu Sistem Arsitektur



Gambar 2. Arsitektur Perangkat Bantu

Arsitektur yang diterapkan untuk perancangan sistem ini adalah arsitektur pada *client*. Pengguna dapat langsung men-*generate* file *spreadsheet* dengan format atribut yang telah disediakan melalui web browser. Apabila pengguna sudah benar menganalisis permasalahan dan menghubungkan relasi-relasinya, maka tampilan dalam format dokumen akan muncul dan pengguna dapat men-*generate* dokumen tersebut ke dalam bentuk graf dengan format *svg*, *png* ataupun *pdf*. Tampilan dari arsitektur terlihat seperti Gambar 2.

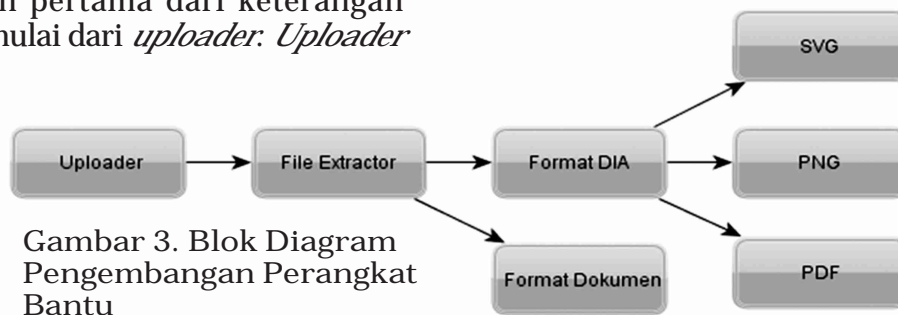
Setelah dilakukan analisis, peran-

cangan, dan arsitektur pada sistem, tahap selanjutnya adalah implementasi ke dalam pengembangan perangkat lunak. Implementasi dari perangkat bantu yang dibuat ini diterapkan untuk analisis menggunakan LWBA dengan menggunakan metode BBSDM.

Untuk mengurangi atau menghilangkan permasalahan tersebut, maka dibuatlah suatu *engine* yang dimaksudkan untuk mengurangi atau menghilangkan inkonsistensi penyajian informasi agar informasi terlihat lebih jelas. Adapun alur dari blok diagramnya adalah seperti Gambar 3.

Adapun langkah-langkah dari Gambar 3 adalah sebagai berikut :

1. Langkah pertama dari keterangan Gambar dimulai dari *uploader*. *Uploader*



Gambar 3. Blok Diagram Pengembangan Perangkat Bantu

no	node	type	label	description	relation
1	1.3.1	1	tidak bisa berkomunikasi dengan keluarga	Karena berada jauh dengan keluarga, dan tidak dapat berkomunikasi	1.3
2	1.3	1	terdapat pelaksana haji dan umroh yang rindu dengan keluarganya di tanah air	Rasa rindu tersebut dapat disebabkan oleh faktor kausal	1
3	1.2.2	3	kurangnya pengertian kepada pihak penyelenggara akan sarana yang dibutuhkan	Seharusnya para jemaah mengerti akan kerja keras yang telah dilakukan	1.2
4	1.2.1	1	pelayanan yang didapat tidak sesuai dengan yang tertera di brosur	Jemaah merasa kecewa karena pelayanan yang mereka dapatkan tidak sesuai	1.2
5	1.2	1	masih terdapat jemaah yang komplain	Terdapatnya jemaah haji yang komplain tentang sistem yang ada pada sistem	1
6	1.1.2	1	jemaah tidak mengetahui sedang berada dimana	Faktor kausal yang menyebabkan jemaah tidak mengetahui sedang berada di mana	1.1
7	1.1	1	terdapat jemaah yang tersesat	Bepergian atau melakukan kegiatan lainnya di negeri orang, dapat menimbulkan	1.1
8	1	1	sistem tidak menjamin kenyamanan para pelaksana haji dan umroh	Setiap tahunnya Indonesia mengirimkan para calon jemaah haji dan umroh ke Arab Saudi. Namun	

Gambar 4. Format File Spreadsheet

memasukkan file *spreadsheet* ke dalam perangkat bantu melalui *web browser*.

2. File *spreadsheet* yang telah diunggah akan di *extract* berdasarkan kolom-kolomnya.
3. Hasil *extract* tersebut menghasilkan dua jenis file, yaitu file berformat *diagraph* dan file berformat dokumen.
4. File berformat *diagraph* membentuk suatu relasi-relasi untuk digambarkan ke bentuk *SVG*, *PNG* dan *PDF*.
5. File berformat dokumen akan membentuk suatu penjelasan dari relasi-relasi yang terjadi.

## File Extractor

Setelah permasalahan dijabarkan dalam bentuk *spreadsheet*, file *spreadsheet* diunggah melalui *web browser* untuk mendapatkan dokumen deskripsi permasalahan dan visualisasi dalam bentuk grafis. Bentuk potongan programnya adalah sebagai berikut:

```
$tmp = $_FILES['file']['tmp_name'];
require_once 'excel_reader2.php';
$xmls = new
Spreadsheet_Excel_Reader($tmp);
$data = array();
$stabel=array();
$slabel=array();
$relation=array();
$jumlah=$xmls->rowCount();
for ($i = 2; $i <= $xmls->rowCount(); $i++)
{
$stabel[$i][2]= $xmls->val($i, 2);
$stabel[$i][3]= $xmls->val($i, 3);
$slabel[$i]="".$xmls->val($i, 2)."";
$stabel[$i][4] = $xmls->val($i, 4);
$stabel[$i][5] = $xmls->val($i, 5);
$sv = $xmls->val($i, 6);
if (empty($sv)) continue;
$relation[$i] = $xmls->val($i, 6); }
```

Untuk mengeskrak file *spreadsheet* dibutuhkan pustaka *php* yang bernama *excel\_reader2.php*. File *spreadsheet* yang diunggah masuk ke direktori *tmp* untuk pemrosesan. Variabel data, label, relation dideklarasikan sebagai array. Variabel jumlah sama dengan variabel *xls* yang berfungsi untuk menghitung jumlah baris untuk pengulangan. Pengulangan dimulai dari baris kedua pada file *spreadsheet*. Setelah melakukan perulangan, selanjutnya program membaca *array* per kolom.

## Format DIA

File yang sudah di ekstrak dari *spreadsheet* akan dicetak juga dalam bentuk notasi pada *digraph* untuk seterusnya di konversi ke dalam bentuk *SVG*, *PNG* ataupun *PDF* menggunakan *Graphviz*. Potongan program untuk men-*generate* ke dalam bentuk dot dalam *digraph* adalah sebagai berikut :

```
$fp = fopen('data.txt', 'w');
fwrite($fp, "digraph G \n");
fwrite($fp, "{\n");
for ($i = 2; $i <= $xmls->rowCount(); $i++)
```

```

{
fwrite($fp, $stabel[$i][2]); //node
fwrite($fp, ' [label="');
fwrite($fp, $stabel[$i][4]); //label
fwrite($fp, '" , ');
switch ($stabel[$i][3]) {

case 1: $shape="box";
break;
case 2: $shape="diamond";
break;
case 3: $shape="circle";
break; }
fwrite($fp, 'shape=');
fwrite($fp, $shape);
fwrite($fp, ');');
fwrite($fp, "\n");
}

```

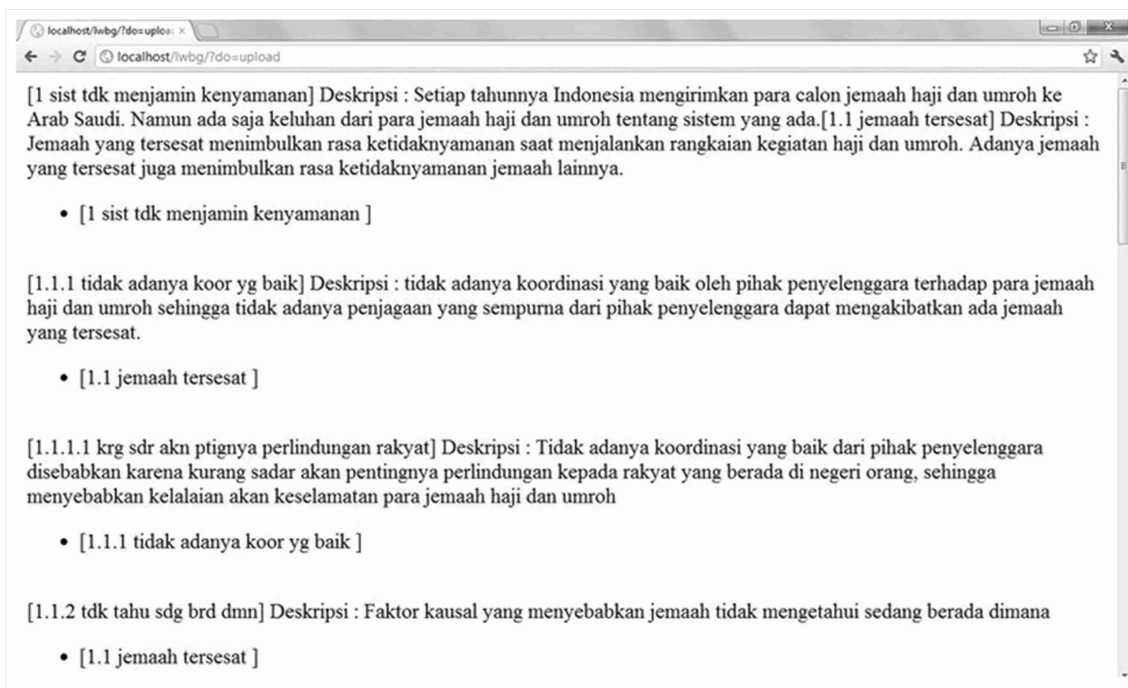
Hasil eksekusi program akan disimpan ke dalam .txt, lalu program akan mencetak kolom dari file *spreadsheet*. Yang dibutuhkan untuk format dot adalah kolom 2 (node), kolom 4 (label) dan kolom 3 (type) Data didapat dari file diparsing kedalam bentuk. Potongan format dot pada digraph adalah sebagai berikut :

```

digraph G {
1.2 -> 1 [];
1.2.1 -> 1.2 [];
}

```

Adapun tampilan output dari ekstrasi file spreadsheet adalah seperti Gambar 5



Gambar 5. Output Dokumen Secara Otomatis

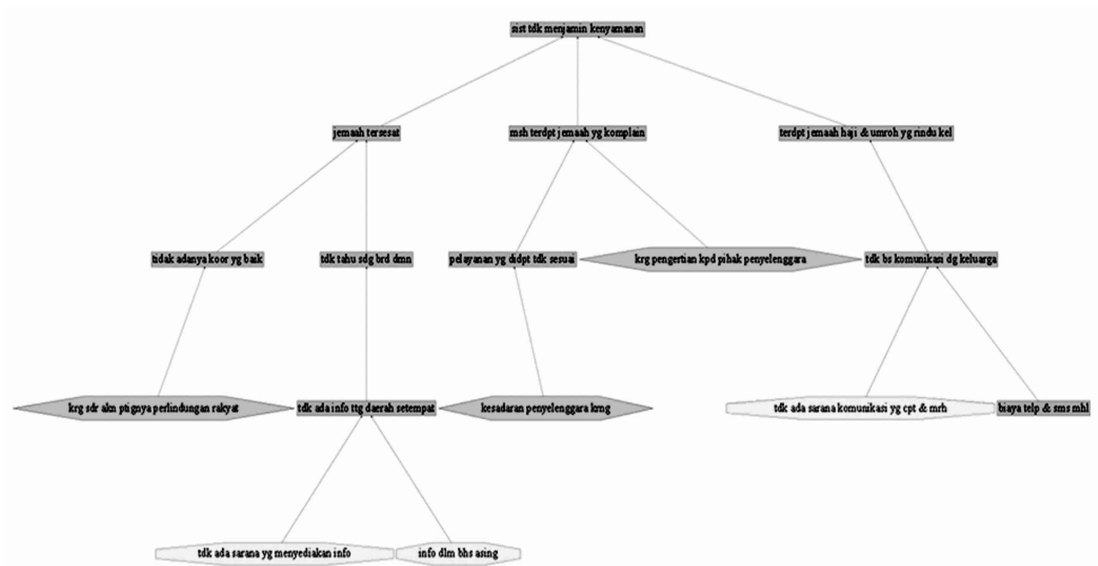
## KESIMPULAN DAN SARAN

### Kesimpulan

Perangkat bantu untuk visualisasi dan pendeskripsian informasi LWBA dengan menggunakan metode BBSDM telah dikembangkan dan diharapkan penyajian informasi menjadi lebih baik secara otomatis.

### Saran

Perangkat bantu ini masih memiliki keterbatasan karena tampilan graf yang dihasilkan pada setiap node masih besar. Diharapkan terdapat pengembangan selanjutnya pada perangkat bantu ini. Misalnya dalam satu dokumen juga terdapat grafik analisis, karena perangkat bantu ini masih memisahkan antara dokumen dan visualisasi grafik.



Gambar 6. Tampilan Visualisasi Otomatis

## DAFTAR PUSTAKA

Cabral, G. & Sampaio, A. 2008. *Automated Formal Specification Generation and Refinement from Requirement Documents*, s.l.: s.n.

Chandra, R. 2013. *Pengembangan Tools pada Fase Requirement Engineering dengan Metode LWBA*

Chris, E. R. & Levine, J. 1995. *Automatic Generation of Technical Documentation, Applied Artificial Intelligence*, Volume 9, pp. 259-287.

*review, Information and Software Technology*, Volume 51, pp. 1291-1307.

Pratiwi, N. L. 2010. *Behavior Engineering*.

Rasmussen, J. 1986. *Information Processing And Human-Machine Interaction. An Approach To Cognitive Engineering*. s.l.:Elsevier Science Publishing Co., Inc..

Stephen, E. K. & Woodhull, G. 2004. *Graphviz and Dynagraph – Static and Dynamic Graph Drawing Tools*.

Thimbleby, H. & Ladkin, P. 1996. From logic to manuals. *Software Engineering Journal*, Volume 11, pp. 347-354.

Viégas, F. B. & Wattenberg, M. 2012. *Artistic Data Visualization: Beyond Visual Analytics*.

Wahono, R. S. 2006. *Menyegarkan Kembali Pemahaman tentang Requirement Engineering*.

Wasson, C. S. 2006, *System analysis, design, and development. Concepts, principles, and practices*. John Willey & Sons, Inc.

Wirjana, I. M. 2009. *A Sustainable System Development Method with Applications*.

Wirjana, I. M. & Hasibuan, E. 2007. *Pengelolaan Pustaka Menggunakan BibTex*, Graha Ilmu: Jakarta.

Wybrow, M. 2008. *Using semi-automatic layout to improve the usability of diagramming software*.

Zahara, R. 2011. *Pengembangan Layanan Location Base Service Dan Chat Pada Aplikasi Panduan Haji Dengan Pendekatan Lightweight Why Because Analysis*.

Zave, P. 1997. *Classification of Research Efforts in Requirements Engineering*, *ACM Computing Surveys*, Volume 29, pp. 315-321.

Darmayantie, A. 2012. *Repository Model and Specification Matching Strategy for Requirement Engineering in Mobile Manufacturing*.

Few, S., 2006, *Information Dashboard Design*, s.l.:O'Reilly.

Few, S., 2007, *Perceptual Edge*, s.l.: Cognos.

Fry, B., 2004, *Computational Information Design*.

Hendrawan, W. 2009 *Software System Requirement Management Planning*.

Nicolás, J. & Toval, A. 2009, *On the generation of requirements specifications from software engineering models: A systematic literature*