

# PROTOTIPE KNOWLEDGE MANAGEMENT SYSTEM BERBASIS WIKI UNTUK PROYEK REKAYASA PERANGKAT LUNAK

Salman El Farisi dan Dana Indra Sensuse

Fakultas Ilmu Komputer, Universitas Indonesia, Kampus Baru UI Depok, 16425

Email: dana@cs.ui.ac.id

## Abstrak

Banyaknya jumlah *software house* dan pengembang perangkat lunak di Indonesia ternyata belum berbanding lurus dengan peningkatan daya saing industri TI Indonesia terhadap industri perangkat lunak negara lain. Hal ini disebabkan karena masih rendahnya inovasi produk yang dapat dihasilkan serta lambatnya proses belajar yang dilakukan oleh industri perangkat lunak di Indonesia. Untuk dapat mengatasi hal tersebut, dibutuhkan adanya proses *knowledge management* yang baik agar setiap *knowledge* dan pengalaman yang dimiliki oleh perusahaan dapat dikelola dengan baik dan dapat dimanfaatkan untuk meningkatkan inovasi produk yang akan dihasilkan. Dalam penelitian ini, penulis melakukan sebuah *knowledge audit* di sebuah unit pengembang perangkat lunak di lingkungan UI dan membuat prototipe *knowledge management system* yang berbasis wiki untuk memenuhi kebutuhan *knowledge management* di perusahaan tersebut. Dalam hal ini penulis berhasil membuat sebuah prototipe *knowledge management system* yang digunakan untuk menyimpan seluruh *knowledge* tentang penyelesaian masalah yang muncul pada sebuah produk perangkat lunak.

**Kata kunci:** *Rekayasa perangkat lunak, Knowledge Management System, KMS, Wiki*

## Abstract

A large number of software houses and software developers in Indonesia was not proportional to the increase in the competitiveness of Indonesian industry to the software industry in other countries. This is because the low innovation of products that can be produced as well as the slow process of learning is done by the software industry in Indonesia. In order to overcome this, it takes a good knowledge management process so that every knowledge and experience possessed by the company can be managed well and can be used to improve product innovation that will be generated. In this study, the authors conducted an audit knowledge in a software developer unit in UI and make knowledge management prototype wiki-based system to meet the needs of knowledge management in the company. In this case I managed to create a prototype knowledge management system that is used to store the whole knowledge about solving problems that arise in a software product.

**Keywords:** *Software engineering, Knowledge Management System, KMS, Wiki*

## 1. Pendahuluan

Memasuki era globalisasi dimana persaingan bisnis dalam skala global semakin meningkat, kebutuhan akan akses informasi dan efisiensi proses bisnis menjadi suatu hal yang sangat vital dalam sebuah organisasi. Maka tidak heran jika pada akhirnya kebutuhan akan aplikasi komputer dewasa ini semakin meningkat. Tidak hanya dalam dunia bisnis saja, melainkan juga pada sektor pemerintahan, pendidikan, dan hiburan.

Melihat kebutuhan seperti ini, maka sudah seharusnya industri perangkat lunak di Indonesia dapat bersaing untuk mengambil kesempatan yang baik ini.

Berdasarkan data yang dikeluarkan oleh *International Data Corporation (IDC)* pada tahun 2008, Indonesia memiliki 500 unit perusahaan yang bergerak di bidang perangkat lunak (*software house*) dengan total pengembang profesionalnya yang berjumlah 71.600 orang atau sekitar 0.5% dari total pengembang di Dunia [1].

Namun ternyata banyaknya perusahaan dan pengembang perangkat lunak di Indonesia masih belum mampu meningkatkan daya saing Indonesia dalam dunia industri Teknologi Informasi (TI). Hal ini dibuktikan dengan masih tertinggalnya peringkat Indonesia dalam daftar *IT competitive index* yang dikeluarkan oleh Economist Intelligence Unit pada tahun 2009.

Berdasarkan tabel *index* ini, Indonesia masih tertinggal 17 peringkat dari Malaysia yang berada di peringkat 42 mengalahkan India yang berada di peringkat 44 [2].

Salah satu hal yang dapat dilakukan oleh industri perangkat lunak di Indonesia untuk meningkatkan daya saing mereka adalah dengan cara memaksimalkan hasil pembelajaran yang mereka dapatkan selama ini melalui proses *knowledge management* (KM). Dengan diterapkannya *knowledge management* pada perusahaan perangkat lunak, maka para *knowledge worker* yang bekerja pada perusahaan tersebut akan memiliki lebih banyak waktu untuk berpikir daripada harus menyalakan waktu yang ada untuk mencari-cari informasi [3]. Salah satu alat yang dapat digunakan untuk membantu proses KM di perusahaan adalah *knowledge management system*. Sistem inilah yang akan dibuat pada penelitian kali ini dengan menggunakan unit pengembang perangkat lunak di lingkungan UI sebagai sampel dalam mencari *system requirement* yang dibutuhkan.

## 2. Knowledge

*Knowledge* merupakan kemampuan seseorang / individu dalam menghubungkan dan mengkaitkan konsep-konsep lain yang relevan dengan area tertentu [4] untuk kemudian digunakan dalam proses pengambilan keputusan [5]. *Knowledge* yang dimiliki oleh seseorang berasal dari proses pendidikan dan pengalaman yang ia dapatkan selama menggeluti suatu pekerjaan [6]. Oleh sebab itu, sudah seharusnya banyaknya proses pendidikan dan pengalaman yang dilakukan oleh seseorang akan berbanding lurus dengan *knowledge* yang mereka dapatkan.

Berdasarkan sifat alaminya, *knowledge* dapat digolongkan kedalam 3 golongan yaitu *knowledge* yang bersifat prosedural/deklaratif, *tacit/explicit*, serta *knowledge* yang bersifat umum/spesifik [4]. *Knowledge* yang bersifat prosedural merupakan *knowledge* yang dapat dijelaskan dalam rangkaian langkah-langkah atau prosedur dalam melakukan sesuatu, sedangkan

*knowledge* yang bersifat deklaratif merupakan *knowledge* yang dapat dijelaskan dengan menggunakan keterhubungan antar variabel atau faktor-faktor yang kemudian di representasikan dalam bentuk formula tertentu.

Berbeda dengan *knowledge* yang bersifat prosedural/deklaratif yang dilihat dari cara merepresentasikan *knowledge*nya, perbedaan pada *knowledge tacit/eksplisit* lebih kepada bentuk dari *knowledge* tersebut. *Knowledge* yang bersifat *tacit* merupakan *knowledge* yang masih berbentuk asumsi, nalar, ataupun dugaan/firasat dari seseorang. Jenis *knowledge* ini masih sangat sulit untuk disebarluaskan dikarenakan bentuknya yang masih abstrak. Lain halnya dengan *knowledge* yang bersifat eksplisit, *knowledge* ini sudah direpresentasikan kedalam bentuk susunan huruf atau angka-angka sehingga lebih mudah untuk disebarluaskan ke orang lain dikarenakan bentuknya yang sudah jelas.

Terakhir adalah *knowledge* yang bersifat umum/spesifik. *Knowledge* ini dibedakan berdasarkan tingkat penguasaannya. *Knowledge* yang bersifat umum adalah *knowledge* yang umum dimiliki oleh kebanyakan orang sedangkan *knowledge* yang bersifat spesifik adalah *knowledge* yang hanya dimiliki oleh sedikit orang dikarenakan proses yang dilakukan untuk mendapatkan *knowledge* ini membutuhkan biaya yang mahal dan panjang. Selain dikategorikan berdasarkan sifat alaminya, *knowledge* juga dapat dikategorikan berdasarkan dengan tempat bersemayamnya. Perbedaan tempat bersemayam ini akan dapat mempengaruhi cara bagaimana *knowledge* ini dapat diambil atau diekstraksikan. Beberapa tempat yang menjadi tempat bersemayamnya *knowledge* antara lain adalah: Manusia, artefak, dan entitas organisasi. *Knowledge* yang bersemayam di dalam diri manusia terdapat di dalam pikiran masing-masing individu yang memiliki *knowledge* tersebut. Sedangkan *knowledge* yang bersemayam di dalam artefak merupakan *knowledge* yang telah berhasil diekstraksikan dari dalam pikiran individu dan kemudian disimpan kedalam artefak baik yang berbentuk teknologi, aktifitas, ataupun repository/tempat penyimpanan yang berbentuk kertas ataupun elektronik. Lain halnya dengan *knowledge* yang tersimpan di dalam manusia ataupun artefak, *knowledge* yang tersimpan di dalam entitas organisasi merupakan *knowledge* yang berada di dalam unit-unit organisasi, keseluruhan organisasi, ataupun keterhubungan antar satu organisasi dengan organisasi lainnya.

TABEL I  
CONTOH *KNOWLEDGE* BERDASARKAN JENISNYA.

	Umum	Spesifik Konteks	Spesifik Teknis
<b>Deklaratif</b>			
Ekplisit	Buku panduan yang berisikan informasi tentang menyusun komposisi pemain dalam pertandingan sepak bola.	Buku panduan yang menjelaskan tentang bagaimana mengatur komposisi latihan tergantung dari kondisi mental pemain yang muncul pada saat itu.	Dokumen teknis yang menjelaskan tentang faktor-faktor apa saja yang harus diperhatikan dalam melakukan pambidaian kaki pemain yang mengalami patah tulang atau terkilir.
Tacit	<i>Knowledge</i> tentang peraturan dasar dalam bermain sepak bola.	<i>Knowledge</i> tentang faktor-faktor penentu yang mempengaruhi susunan komposisi pemain dalam pertandingan sepak bola.	<i>Knowledge</i> tentang faktor-faktor apa saja yang harus diperhatikan dalam membersihkan seragam sepak bola.
<b>Prosedural</b>			
Ekplisit	Buku panduan yang berisikan informasi tentang bagaimana cara melakukan tendangan dalam permainan sepak bola.	Buku panduan yang menjelaskan tentang langkah-langkah dalam memahami pola serangan lawan.	Dokumen SOP (Standard Operational Procedure) tentang penanganan pemain yang mengalami cidera kaki pada saat pertandingan.
Tacit	<i>Knowledge</i> dasar tentang cara melakukan tendangan yang akurat.	<i>Knowledge</i> tentang langkah-langkah yang harus diambil dalam melakukan jebakan offside.	<i>Knowledge</i> tentang langkah-langkah yang harus dilakukan dalam membersihkan sepatu bola setelah melakukan permainan.

### 3. *Knowledge audit*

Proses *knowledge audit* merupakan proses yang paling penting dalam rangkaian metodologi pengembangan *knowledge management system* [7][8][9]. Dalam proses ini, seorang *knowledge auditor* berusaha untuk memahami *knowledge life cycle* pada sebuah organisasi. Dengan melakukan *knowledge audit*, seorang *knowledge auditor* akan dapat mengetahui *knowledge* apa saja yang tersimpan di dalam organisasi, dari mana asalnya, bagaimana mengaksesnya, siapa saja yang membutuhkan, serta infrastruktur apa saja yang mendukung proses *knowledge management* di organisasi tersebut.

Pada dasarnya, tidak ada aturan baku yang mengatur proses *knowledge audit*, namun secara garis besar dapat dikatakan bahwa ada 2 buah pendekatan yang dapat digunakan dalam melakukan *knowledge audit* yaitu pendekatan yang pragmatis [7] dan sistematis [8]. Pendekatan pragmatis pada *knowledge audit* merupakan pendekatan yang praktis dikarenakan hanya berfokus pada 6 area proses *knowledge management* (akuisisi, penyimpanan / pemeliharaan, eksploitasi, diseminasi, penciptan dan pengukuran performa). Sedangkan pada pendekatan sistematis merupakan sebuah pendekatan pada *knowledge audit* yang dilakukan secara terperinci dan mendetail dimulai dari mempelajari latar belakang organisasi dan orientasi dalam melakukan *knowledge management*, menganalisis kultur dan infrastruktur yang ada, sampai pada akhirnya

*knowledge management system* yang dibuat dapat diimplementasikan di dalam organisasi yang bersangkutan.

### 4. Penelitian Terkait

Isu pembuatan *knowledge management system* untuk membantu proses pembelajaran sebuah organisasi bukan sebuah hal yang baru dalam dunia TI dan manajemen. Sudah banyak penelitian yang dilakukan terkait dengan isu ini baik dari sisi manajemennya ataupun dari sisi teknologinya.

Beberapa penelitian yang terkait dengan pembuatan *knowledge management system* berbasis wiki antara lain adalah:

1. RISE (*Reuse In Software Engineering*) Project. Proyek ini merupakan sebuah proyek pembuatan aplikasi *knowledge management system* yang berfokus pada isu pemanfaatan kembali setiap *knowledge* yang ada di dalam sebuah organisasi. Hasil akhir dari penelitian ini adalah terbentuknya sebuah sistem yang bernama RIKI (*Reuse-oriented WIKI*) yang merupakan sebuah *knowledge management system* berbasis wiki untuk keperluan pemanfaatan kembali *knowledge* yang berkaitan dengan rekayasa perangkat lunak [10].
2. Riset tentang pengembangan fitur semantik pada aplikasi wiki untuk menunjang proses *knowledge management* secara kolaboratif. Hasil akhir dari penelitian ini adalah sebuah konsep wiki yang memiliki fitur semantik

untuk memudahkan pengguna dalam melakukan pencarian *knowledge* yang mereka butuhkan. Selain untuk memudahkan pencarian, tujuan utama dari pemanfaatan teknologi semantik pada sistem wiki adalah agar setiap *knowledge* yang tersimpan di dalam sistem wiki tidak hanya dapat dipahami oleh manusia, melainkan juga dapat dipahami oleh mesin [11].

3. Implementasi *knowledge management system* berbasis wiki pada kelompok riset kecil. Penelitian ini berfokus pada implementasi teknologi wiki untuk menunjang proses akuisisi, penyimpanan, penyebaran, serta penggunaan *knowledge* yang dapat digunakan untuk keperluan riset. Dalam penelitian ini, peneliti memanfaatkan 2 buah jenis sistem wiki (tidlywiki dan media wiki) sebagai sebuah *tools* untuk menunjang proses *management* pada kelompok riset [12].

## 5. Kerangka Berpikir

Untuk dapat membuat sebuah *knowledge management system* berbasis wiki dibutuhkan sebuah *system requirement* yang jelas agar nantinya setiap fitur yang diimplementasikan kedalam sistem merupakan fitur yang benar-benar dibutuhkan dalam melakukan proses *knowledge management*. *System requirement* tersebut akan sangat erat kaitannya dengan kebutuhan *knowledge* yang ada di dalam organisasi. Dengan alasan itulah sebelum dapat melakukan proses pembuatan sistem, dibutuhkan adanya proses *knowledge audit* terlebih dahulu untuk dapat mengetahui medan *knowledge management* yang ada di dalam organisasi. Dalam hal ini, organisasi yang dijadikan sampel untuk mendapatkan *system requirement* yang dibutuhkan adalah unit pengembang perangkat lunak di lingkungan UI.

Dengan dilakukannya proses *knowledge audit*, maka penulis dapat mengetahui sejauh mana proses *knowledge management* yang sudah diterapkan beserta infrastruktur pendukungnya. Tidak hanya itu saja, tujuan dari dilakukannya proses *knowledge audit* adalah untuk memahami *knowledge* apa saja yang ada di unit tersebut, sumber utama *knowledge* yang biasa diakses oleh staf mereka, tempat penyimpanan *knowledge*,

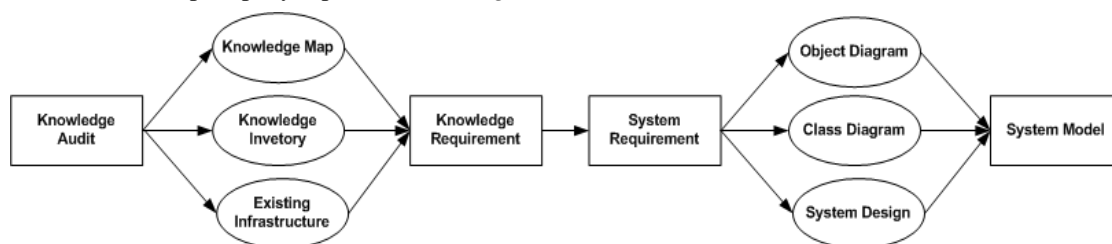
serta cara pendistribusian dan tranfer *knowledge* dari satu tempat ke tempat yang lain. Seluruh informasi tersebut dibutuhkan oleh penulis untuk dapat memahami kebutuhan *knowledge* pada unit tersebut.

Hasil akhir yang diperoleh dari proses *knowledge audit* ini nantinya adalah terbentuknya peta *knowledge* (*knowledge map*), daftar inventaris *knowledge* (*knowledge inventory*), serta daftar infrastruktur *knowledge management*. Seluruh hasil *knowledge audit* tersebut nantinya akan digunakan untuk menganalisis kebutuhan *knowledge* dan dari hasil analisis kebutuhan *knowledge* inilah penulis akan membuat *system requirement* yang nantinya akan diimplementasikan pada *system model* KMS berbasis wiki yang akan dibuat.

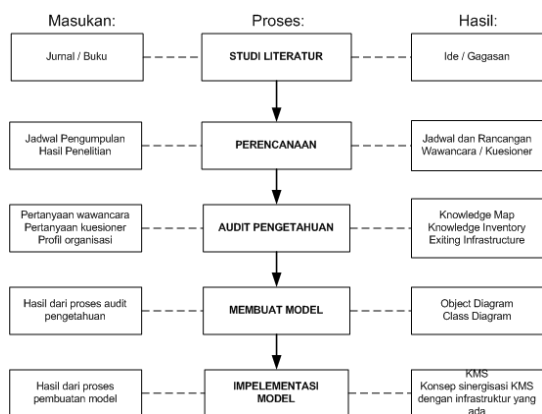
## 6. Tahapan Penelitian

Untuk dapat menghasilkan sebuah penelitian yang baik, maka sudah seharusnya kerangka berpikir yang digunakan harus didukung dengan tahapan penelitian yang sesuai. Tujuannya adalah agar tahapan penelitian yang dilakukan sesuai dengan kerangka berpikir yang digunakan. Dalam penelitian ini, penulis menggunakan 5 buah tahapan penelitian yang diantaranya adalah: Studi literatur, perencanaan, *knowledge audit*, membuat model, serta implementasi model.

Pada tahap studi literatur, penulis mencari referensi terkait dengan penelitian yang akan dilakukan. Tujuan dari tahapan ini adalah untuk mencari teori serta ide/gagasan yang dapat digunakan oleh penulis dalam melakukan penelitian. Pada tahap perencanaan penulis membuat timeline kerja untuk memastikan bahwa penelitian yang dilakukan tidak melewati jadwal yang seharusnya. Sedangkan tahap *knowledge audit* merupakan tahap dimana penulis akan melakukan wawancara dan *interview* untuk menganalisis kebutuhan *knowledge*. Terakhir, penulis akan membuat model dan mengimplementasikan model tersebut kedalam KMS yang akan dibuat. Keseluruhan tahapan penelitian tersebut akan berakhir pada terciptanya sebuah prototipe *knowledge management system* berbasis wiki yang nantinya akan digunakan untuk keperluan rekayasa perangkat lunak.



Gambar 1. Kerangka berpikir yang digunakan



Gambar 2. Tahapan penelitian

### 7. Hasil Knowledge Audit

Proses *knowledge audit* dilakukan dengan 2 cara yaitu dengan cara kuesioner dan wawancara. Pada penelitian ini penulis melakukan wawancara dengan GM Operation di unit pengembangan perangkat lunak. Tujuan dari dilakukannya wawancara ini adalah untuk mengetahui sejauh mana pihak manajemen memiliki kesadaran untuk menerapkan *knowledge management* di lingkungannya serta strategi apa saja yang sudah dijalankan untuk mendukung proses *knowledge management* tersebut. Sedangkan untuk pengisian kuesioner, penulis menjadikan seluruh staf unit pengembang perangkat lunak untuk menjadi responden dari kuesioner yang diberikan. Tujuan dari pengisian kuesioner ini adalah untuk mengetahui *knowledge* apa saja yang paling sering dibutuhkan oleh staf unit tersebut serta dari mana sumber *knowledge* tersebut berasal. Selain itu, kuesioner ini juga ditujukan untuk mengetahui sejauh mana kultur berbagi *knowledge* dapat dibangun di lingkungan unit pengembang perangkat lunak tersebut.

Hasil kuesioner yang diperoleh dari 14 responden yang menjadi staf di unit pengembang perangkat lunak dapat dilihat pada tabel II.

Berdasarkan hasil kuesioner yang dilakukan, dapat diketahui bahwa *knowledge* yang banyak digunakan adalah *knowledge* tentang *programming*. Hal ini terjadi dikarenakan mayoritas pekerjaan yang dilakukan oleh staf mereka adalah pekerjaan yang berhubungan dengan *programming*. Jika melihat dari tingkat komitmen dan intensitas interaksi sosial yang terjadi diantara staf, maka dapat disimpulkan bahwa kultur berbagi *knowledge* sangat mungkin untuk dibangun di dalam lingkungan kerja unit pengembang perangkat lunak tersebut. Kesimpulan ini diambil karena faktor komitmen organisasi dan intensitas interaksi sosial

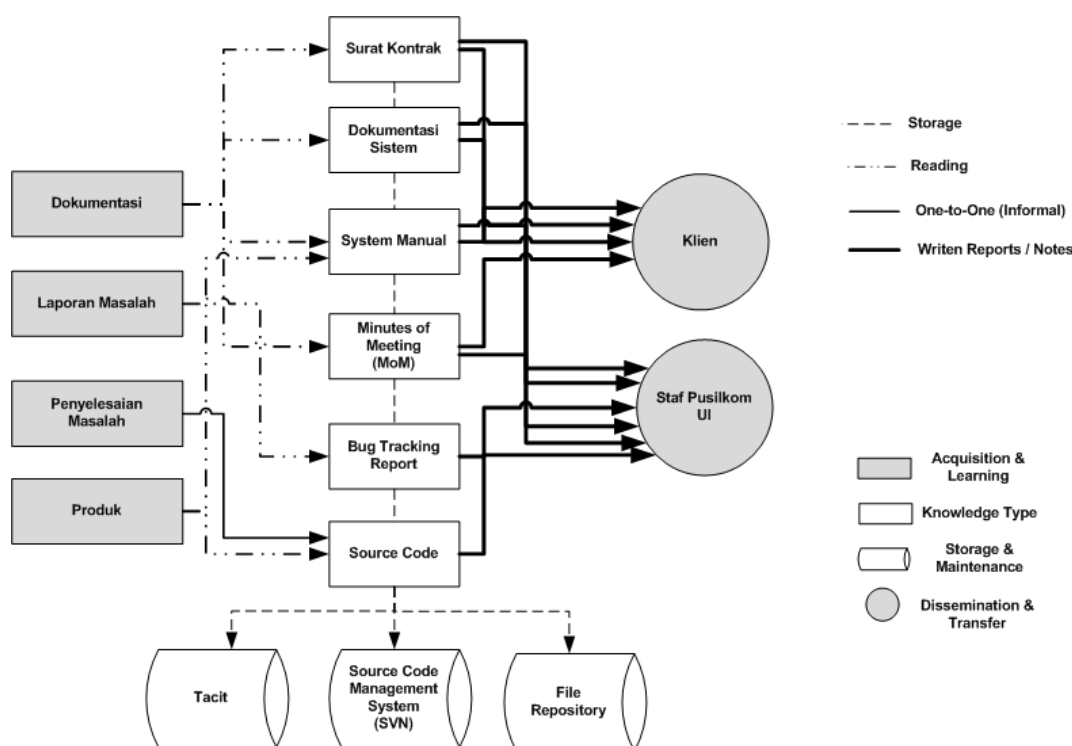
merupakan faktor yang dibutuhkan dalam membangun kultur berbagi *knowledge* di dalam sebuah organisasi [13].

TABEL II  
HASIL PENGISIAN KUESIONER

Pekerjaan yang sering digeluti	
1. Programmer	63%
2. Lainnya	37%
Pengetahuan yang paling sering digunakan	
1. Programming	67%
2. Database	13%
3. Lainnya	20%
Sumber Mencari Pengetahuan	
1. Web	32%
2. Bertanya	25%
3. Buku	15%
4. Dokumen Proyek	13%
5. Diskusi	10%
6. Lainnya	5%
Perilaku dalam berbagi pengetahuan	
1. Biasa berbagi	78%
2. Tidak biasa berbagi	22%
Fasilitas yang paling sering digunakan dalam berbagi <i>knowledge</i>	
1. Diskusi informal	72%
2. Blog	28%
Tingkat komitmen terhadap organisasi	
1. Tinggi	79%
2. Sedang	7%
3. Rendah	14%
Intensitas interaksi sosial dan tingkat kepercayaan	
1. Tinggi	79%
2. Sedang	7%
3. Rendah	14%

Untuk mendapatkan gambaran yang lebih komprehensif terkait dengan medan yang akan dihadapi dalam melakukan *knowledge management*, penulis membuat daftar inventaris *knowledge* (Tabel II) beserta peta *knowledge* (Gambar 3.). Tujuan dari dibentuknya kedua hal tersebut adalah untuk melihat keterhubungan antara *knowledge*, infrastruktur, entitas organisasi, serta proses yang berada di unit kerja tersebut.

Pada tabel inventaris *knowledge*, dapat terlihat daftar aset *knowledge* yang ada di unit kerja tersebut beserta format dan tempat penyimpanan *knowledge* tersebut. Dalam hal ini, penulis memberikan peringkat pada setiap aset *knowledge* berdasarkan pada intensitas penggunaannya. Pada tabel inventaris *knowledge* yang dibuat, penulis memberikan peringkat 1 pada aset *knowledge* yang berbentuk dokumentasi pengembangan sistem. Hal ini dilakukan karena dokumentasi pengembangan sistem merupakan *knowledge* yang sering diakses oleh staf pengembang dikarenakan unit ini tidak memiliki *template* baku dalam membuat dokumentasi sistem sehingga setiap staf harus melihat kembali dokumentasi yang lalu sebagai referensi.



Gambar 3. Peta knowledge

TABEL III  
DAFTAR INVENTARIS KNOWLEDGE

No.	Aset Pengetahuan	Format	Pemilik	Lokasi	Jenis	Peringkat
1.	Source code program	Source code	Unit pengembang	Source code repository (SVN)	Know-how	3
2.	Minute of Meeting (MoM)	Dokumen	Unit pengembang	Server Dokumen	Know-what, know-when, know who	5
3.	Dokumentasi pengembangan sistem	Dokumen	Unit pengembang	Server Dokumen	Know-what	1
4.	Dokumen instalasi sistem.	Dokumen	Unit pengembang	Server Dokumen	Know-how	4
5.	Bug tracking report	Aplikasi	Unit pengembang	Aplikasi (JIRA)	Know-when, know-where, know why	2

Berdasarkan daftar inventaris *knowledge* yang telah dibuat, berikutnya penulis membuat peta *knowledge* untuk melihat keterhubungan setiap entitas organisasi dengan *knowledge* yang ada. Berdasarkan gambar peta *knowledge* tersebut nantinya akan terlihat bagaimana sebuah *knowledge* berpindah dari satu tempat ke tempat lain dan infrastruktur apa saja yang membantu proses perpindahan tersebut. Gambar peta *knowledge* dapat dilihat pada Gambar 3 di atas.

Pada gambar peta *knowledge* (Gambar 3) terlihat bahwa mayoritas *knowledge* yang dimiliki oleh unit ini berbentuk tertulis (*written report*).

Oleh karenanya mayoritas *knowledge* tersebut diakses dengan cara dibaca. Satu hal yang berbeda pada gambar peta *knowledge* tersebut adalah pada jenis *knowledge* ‘penyelesaian masalah’ dimana cara pengaksesan *knowledge* tersebut tidak dengan cara dibaca melainkan dengan cara pertemuan informal (*one-*

to-one conversation). Hal ini menunjukkan bahwa *knowledge* tentang penyelesaian masalah belum terdokumentasikan dengan baik sehingga masih bersifat *tacit knowledge*. Untuk dapat mengakses *knowledge* tersebut harus dilakukan dengan cara interaksi langsung bersama pemilik *knowledgenya*.

Selain itu, dari gambar peta *knowledge* tersebut dapat terlihat juga bahwa setiap tempat penyimpanan *knowledge* (*knowledge repository*) belum saling terintegrasi. Hal ini menyebabkan sulitnya mencari *knowledge* yang saling berkaitan namun tersimpan di dalam tempat yang berbeda. Jika setiap tempat penyimpanan *knowledge* yang ada dapat diintegrasikan, maka proses pencarian *knowledge* akan menjadi lebih mudah.

Selain melakukan pembuatan peta *knowledge*, untuk mengetahui keterhubungan antara setiap *knowledge*, proses, entitas, serta tempat penyimpanan yang ada, untuk dapat memahami medan *knowledge management* di unit pengembang ini, penulis juga harus bisa memahami alur *knowledge* yang ada di unit tersebut. Mulai dari proses ekstraksi pada sumbernya, hingga sampai pada proses penyimpanan dan pemeliharaan *knowledge* tersebut. Untuk dapat memahami alur perpindahan *knowledge*, maka penulis membuat sebuah diagram alir *knowledge* yang digunakan untuk melihat bagaimana *knowledge* mengalir dari satu tempat ke tempat yang lain. Gambar diagram alir *knowledge* dapat dilihat pada Gambar 4.

Pada gambar diagram alir di atas terlihat bahwa setiap individu dapat menciptakan sebuah *knowledge* baru dari *knowledge* yang ia dapatkan sebelumnya. Sebagai contoh adalah, seorang *programmer* dapat menciptakan sebuah *knowledge* baru yang berupa kode program berdasarkan *knowledge* tentang desain sistem

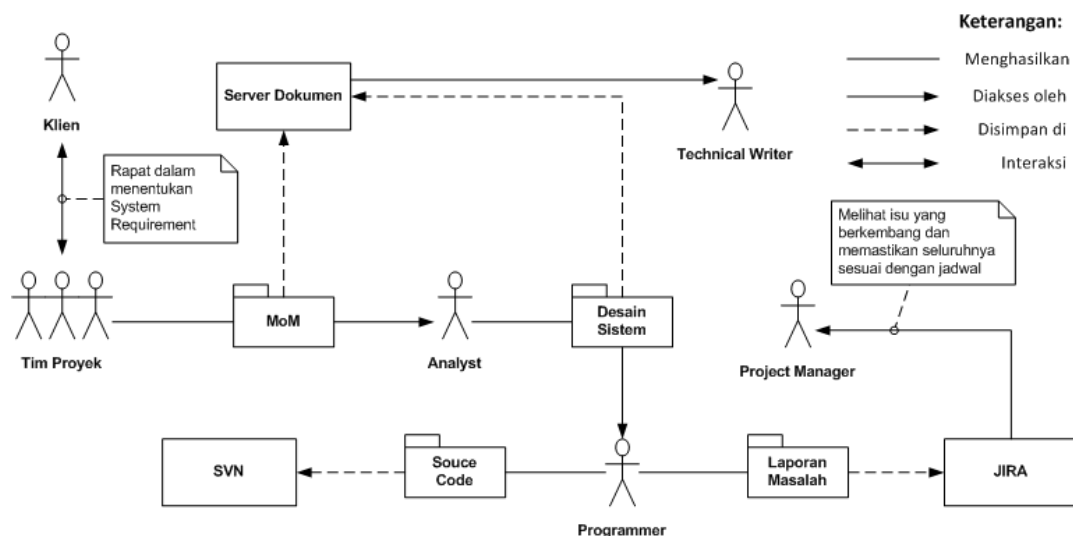
yang dibuat oleh seorang *analyst*. Setiap *knowledge* yang mengalir dari satu tempat ke tempat lain pada akhirnya akan bermuara pada sebuah tempat penyimpanan (*repository*) seperti misalnya *server* dokumen ataupun *source code management system*. Tempat penyimpanan inilah yang nantinya digunakan sebagai tempat penyimpanan dan pemeliharaan *knowledge*. Berawal dari tempat penyimpanan tersebut nantinya *knowledge* akan disebarluaskan dan dipergunakan kembali (*reuse*).

## 8. System Requirement

Berdasarkan hasil *knowledge audit* yang telah didapatkan, maka dapat disimpulkan bahwa beberapa *system requirement* dari KMS berbasis wiki yang akan dibuat antara lain adalah:

1. KMS harus dapat menyimpan informasi tentang permasalahan yang terjadi pada sebuah produk perangkat lunak.
2. KMS harus dapat menyimpan seluruh *knowledge* tentang penyelesaian masalah yang terjadi.
3. KMS harus dapat mengintegrasikan seluruh *explicit knowledge* yang berkaitan dengan kegiatan pembuatan produk perangkat lunak (dokumentasi, manual, dan *source code*).
4. KMS harus dapat merelasikan setiap *knowledge* yang tersimpan di dalamnya.
5. KMS harus memiliki fungsi pencarian yang dapat memudahkan pengguna dalam mendapatkan *knowledge* yang dibutuhkan.

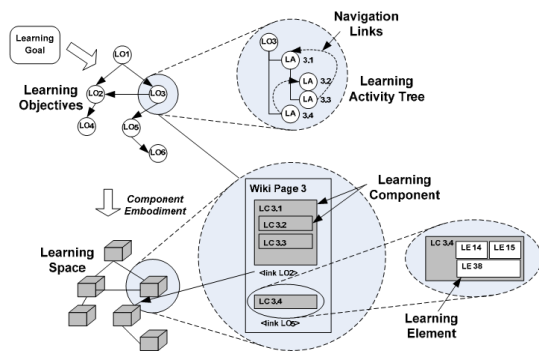
Seluruh *system requirement* yang didapatkan dari hasil *knowledge audit* ini nantinya akan menjadi landasan dalam membuat rancangan sistem KMS berbasis wiki yang akan dibuat.



Gambar 4. Diagram alir *knowledge*

### 9. Konsep KMS Berbasis Wiki

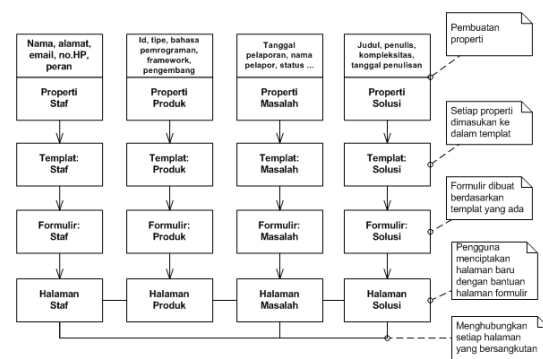
Wiki Web atau yang biasa disebut dengan sebutan wiki merupakan sebuah sistem berbasis web yang dibangun oleh Ward Cunningham pada tahun 1990. Sistem ini dibuat dengan tujuan menciptakan sebuah sistem yang dapat memfasilitasi proses penulisan konten pada halaman web secara kolaboratif [14][15]. Seperti aplikasi *content management system* (CMS) lainnya, pada dasarnya wiki merupakan sebuah sistem yang bertugas untuk mengatur setiap halaman artikel yang dibuat oleh sistem tersebut. Apabila sistem ini akan digunakan dalam pembuatan KMS, maka representasi dari setiap *knowledge* yang tersimpan di dalam KMS berbasis wiki ini pastinya dalam bentuk halaman-artikel.



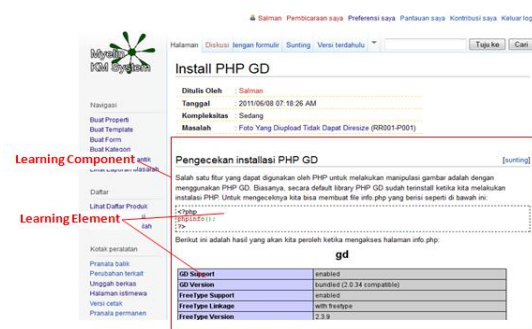
Gambar 5. Representasi *knowledge* pada wiki [10]

gambar, dan potongan *source code*). Sedangkan *learning component* adalah kumpulan *learning element* yang berada di dalam satu buah section tertentu. Setiap artikel yang tersimpan di dalam sistem wiki ini nantinya akan saling berkaitan satu sama lainnya hingga membentuk sebuah learning space.

Dikarenakan setiap *knowledge* yang tersimpan di dalam wiki nantinya akan direpresentasikan kedalam bentuk halaman-halaman wiki, maka desain sistem yang dibuat haruslah menggambarkan struktur-struktur halaman wiki yang nantinya akan merepresentasikan *knowledge* yang disimpan. Berikut ini adalah konsep pembagian halaman-halaman pada KMS berbasis wiki untuk merepresentasikan setiap *knowledge* yang tersimpan di dalamnya.



Gambar 7. Konsep pembagian halaman pada wiki



Gambar 6. Contoh representasi *knowledge* pada halaman wiki

Pada gambar representasi *knowledge* di atas, dapat terlihat bahwa setiap halaman wiki yang akan dibuat nantinya akan mengandung *learning component* dan *learning element* yang merupakan bagian terpenting dalam merepresentasikan *knowledge* pada KMS berbasis wiki. *Learning element* yang dimaksudkan dalam representasi *knowledge* ini adalah setiap entitas artikel yang merepresentasikan setiap *knowledge* yang tersimpan di dalam artikel tersebut (misal: video,

Pada gambar konsep pembagian halaman di atas terlihat bahwa terdapat 4 buah jenis halaman yang akan dibuat untuk merepresentasikan *knowledge* yang tersimpan di dalam wiki yaitu: Halaman staf, produk, masalah, dan halaman solusi. Setiap halaman-halaman tersebut merepresentasikan *knowledge* yang berbeda-beda namun saling berkaitan. Halaman produk digunakan untuk merepresentasikan *knowledge* tentang produk yang dibuat oleh unit pengembangan perangkat lunak, halaman masalah digunakan untuk merepresentasikan *knowledge* tentang setiap masalah yang muncul pada sebuah produk, halaman solusi digunakan untuk merepresentasikan setiap *knowledge* tentang bagaimana menyelesaikan sebuah masalah yang muncul pada sebuah produk, sedangkan halaman staf merupakan halaman yang merepresentasikan *knowledge* tentang orang-orang yang terlibat dalam pembuatan produk perangkat lunak.

Kumpulan halaman-halaman inilah yang nantinya saling melengkapi satu sama lainnya dalam merepresentasikan *knowledge* yang berkaitan tentang masalah dan penyelesaian permasalahan pada produk perangkat lunak yang



dibuat. Harapannya dengan adanya KMS berbasis wiki ini, setiap staf dapat mengekspresikan setiap *knowledge* yang dimilikinya kedalam halaman-halaman wiki sehingga *knowledge* yang dimiliki dapat ditransfer ke orang lain untuk kemudian digunakan kembali dalam melakukan penyelesaian masalah.

Dalam mengimplementasikan konsep KMS berbasis wiki ini, penulis menggunakan wiki engine yang bernama mediawiki dengan tambahan (*extension*) semantic mediawiki untuk mengimplementasikan fitur *web* semantik pada KMS yang dibuat.

### 10. Pencarian Semantik Pada KMS

Untuk dapat meningkatkan relevansi pencarian dan memudahkan pengguna dalam mendapatkan *knowledge* yang dibutuhkan, maka harus ada metode pencarian khusus yang dapat memfasilitasi hal tersebut. Dalam penelitian ini, penulis menerapkan pencarian semantik pada KMS yang dibuat. Konsep dasar dari pencarian semantik yang diimplementasikan pada KMS ini adalah menggunakan atribut-atribut yang tersimpan di dalam setiap halaman untuk mendefinisikan setiap *knowledge* yang tersimpan di dalam sistem. Dengan menggunakan cara ini maka tidak hanya manusia saja yang mengerti isi halaman yang dibuat, melainkan juga dapat dimengerti oleh KMS itu sendiri.

Dalam mengimplementasikan fitur pencarian semantik pada KMS berbasis wiki ini, penulis menggunakan fitur pencarian semantik bawaan yang sudah ada di dalam *extension* semantic mediawiki yang di install oleh penulis. Hal yang perlu dipersiapkan selanjutnya adalah mengatur properti yang akan mendefinisikan setiap halaman wiki agar setiap halaman yang dibuat dapat dimengerti oleh mesin.

Dalam melakukan pencarian semantik, nantinya pengguna yang ingin mencari *knowledge* akan membuat sebuah *query* yang isinya memanggil properti-properti tertentu sesuai dengan kriteria *knowledge* yang diinginkan olehnya. Sebagai contoh, apabila pengguna ingin mendapatkan *knowledge* tentang masalah apa saja yang muncul dari produk-produk yang menggunakan framework codeigniter, maka pengguna harus membuat *query* sebagai berikut:

```
[[Kategori:Masalah]][[Relasi: <q>
[[Kategori:Produk]] [[Framework::Codeigniter]]
</q>]]
```

Catatan: “<q></q>” adalah *subquery*

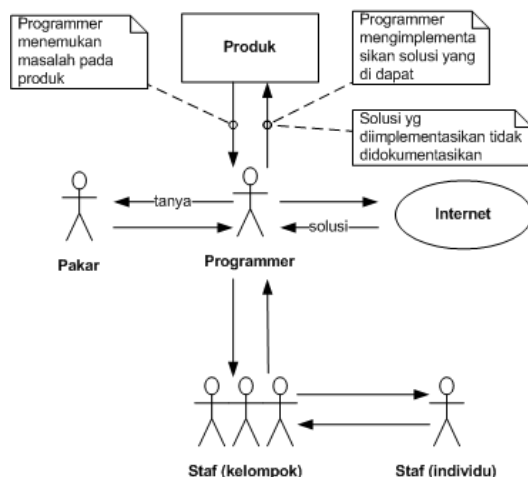
Dengan menggunakan *query* tersebut nantinya sistem akan mencari setiap halaman wiki yang menyimpan properti seperti yang didefinisikan oleh pengguna. Hasil akhir dari *query* tersebut nantinya adalah daftar halaman masalah dari setiap produk yang menggunakan framework Codeigniter. Dengan cara ini, pengguna dapat melakukan pencarian yang lebih spesifik untuk mendapatkan *knowledge* yang dibutuhkan.

### 11. Mensinergikan KMS

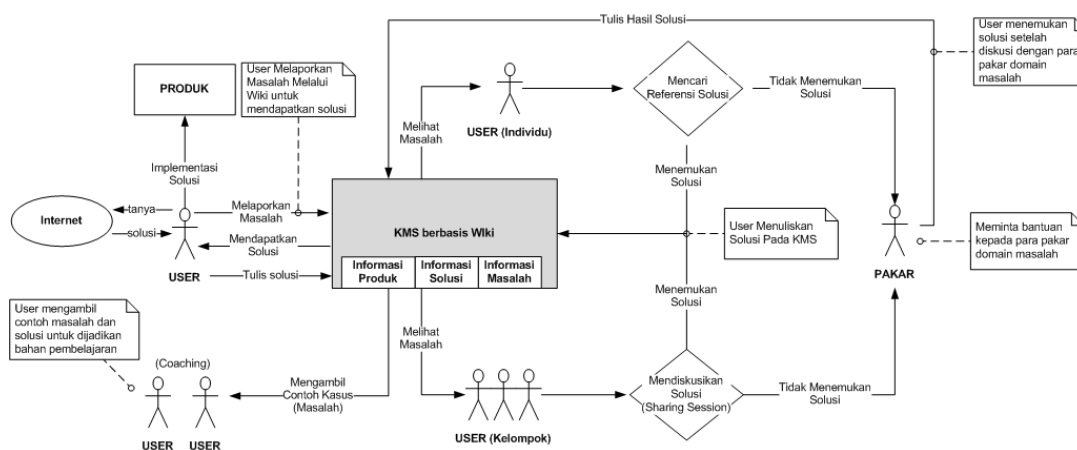
Agar KMS yang dibuat dapat dimanfaatkan dengan baik, maka perlu adanya sinergisasi antara KMS yang dibuat dengan infrastruktur lain baik berupa infrastruktur sosial maupun infrastruktur teknis. Pada dasarnya, unit pengembang perangkat lunak ini sudah memiliki beberapa infrastruktur yang dapat mendukung proses berjalannya KM di organisasi tersebut baik berupa infrastruktur sosial seperti *knowledge sharing session* yang diadakan setiap minggunya,

TABEL IV  
DAFTAR PROPERTI UNTUK SETIAP HALAMAN

Halaman Produk	Halaman Masalah	Halaman Solusi	Halaman Staf
Id produk:	Id masalah:	Ditulis oleh:	Telepon:
Versi:	Tanggal pelaporan:	Tanggal penulisan:	Email:
Pengembang:	Tenggat waktu:	Kompleksitas:	Peran:
Dokumentasi:	Dilaporkan oleh:	Id masalah relasi:	Keahlian:
Manual:	Ditujukan ke:	Relasi:	
Logo:	Id produk relasi:		
Deskripsi singkat:	Komponen:		
Platform:	Url masalah:		
Tipe produk:	Prioritas:		
Tanggal rilis:	Status:		
Pengembang:	Pemicu:		
Website:	Kata kunci:		
	Lampiran:		
	Relasi:		



Gambar 8. Knowledge flow sebelum ada KMS



Gambar 9. Knowledge flow sesudah ada KMS

ataupun yang berupa infrastruktur teknis seperti server dokumen dan *source code management system* (SVN).

Sebelum adanya KMS berbasis wiki, setiap *knowledge* tentang penyelesaian masalah pada sebuah produk perangkat lunak hanya tersimpan di dalam pikiran masing-masing individu yang terlibat di dalam proses penyelesaian masalah. Hal ini menyebabkan sulitnya untuk mengakses *knowledge* tersebut dikarenakan harus ada interaksi secara langsung antara orang yang membutuhkan *knowledge* dengan pemilik *knowledge*nya. Dengan adanya KMS berbasis wiki, maka setiap *knowledge* tersebut akan disimpan kedalam wiki baik itu *knowledge* yang diperoleh dari hasil diskusi, studi literatur, ataupun pengalaman-pengalaman yang didapatkan oleh seseorang. Dengan cara ini, maka *knowledge* yang tadinya bersifat *tacit* dapat diubah menjadi

eksplisit sehingga lebih mudah diakses dan dipergunakan kembali.

## 12. Kesimpulan

Kesimpulan yang didapat dari hasil penelitian yang dilakukan antara lain adalah:

1. Penulis berhasil membangun sebuah KMS berbasis wiki yang dapat digunakan untuk mendokumentasikan setiap *knowledge* tentang masalah dan penyelesaian masalah pada setiap produk perangkat lunak yang dibuat.
2. Untuk dapat memastikan bahwa KMS yang dibuat dapat dimanfaatkan dengan baik, maka penulis telah menyusun konsep sinergisasi antara KMS yang dibuat dengan infrastruktur yang ada.
3. Untuk dapat meningkatkan relevansi pencarian dan memudahkan pengguna dalam mengakses *knowledge* yang dibutuhkan,

4. penulis mengimplementasikan pencarian semantik pada KMS yang dibuat.

#### Referensi

- [1] B. S. A. Mohammad. (2010, Februari 28). "Gelombang Baru Digitalpreneur Indonesia" [Online]. Tersedia di: <http://swa.co.id/2010/02/gelombang-baru-digitalpreneur-indonesia>. (Diakses pada tanggal 24 Februari 2011)
- [2] Anonymous, "Resilience amid turmoil: Benchmarking IT industry competitiveness 2009". The Economist Intelligence Unit. London, UK. Agustus 2009.
- [3] Dalkir, Kimiz. *Knowledge Management Theory and Practice*. USA: Elsevier, 2005.
- [4] Bacerra-Fernandez, Irma, Avelino Gonzalez, and Rajiv Shaberwal. *Knowledge Management: Challenges, Solution and Technologies*. New Jersey: Pearson Education, 2003.
- [5] Tiwana, Amrit. *The Knowledge Management Toolkit*. USA: Prentice Hall, 2000
- [6] Awad, Elias M., Ghaziri, Hasan M. *Knowledge Management*. New Jersey: Pearson Education, 2004.
- [7] Burnett, Simon, L. Illingworth dan Linda Webster. "Knowledge auditing and Mapping: A Pragmatic Approach" *Knowledge and Process Management* Vol. 11 (2004): 25-37. doi:10.1002/kpm.194.
- [8] F. Cheung et al. "A Systemic Approach for Knowledge auditing: A Case Study In Transportation Sector" *Journal of Knowledge Management* Vol.11 No.4 (2007): 140-58. doi:10.1108/13673270710762774.
- [9] J. Liebowitz et al. "The Knowledge audit" *Knowledge and Process Management* Vol.7 No.1 (2000): 3-10.
- [10] Lytras, Miltiadis, Naeve, Ambjorn. "Riki: A System for Knowledge Transfer and Reuse in Software Engineering Projects" dalam buku *Open Source for Knowledge and Learning Management*. (US: Idea Group Publishing, 2007), 52. *Knowledge Management* Vol.11 No.4 (2007): 140-58. doi:10.1108/13673270710762774.
- [11] Junjie, Wang et al. "Research for Collaborative Knowledge Management Based On Semantic Wiki Technology" Second International Workshop on Education Technology and Computer Science, IEEE Computer Society (2010). doi: 10.1109/ETCS.2010.82
- [12] Yang, Chia-Han et al. "Implementation of Wiki-based Knowledge Management System for Small Research Group" Eight International Conference on Intelligent System Design and Application, IEEE Computer Society (2008). doi: 10.1109/ISDA.2008.47
- [13] Juan-Ru, Wang, dan Yang Jin. "Study On Knowledge Sharing Behaviour In Software Development Team" The 4th International Conference on Wireless Communications, Networking and Mobile Computing (2008). doi: 10.1109/WiCom.2008.2
- [14] Sceffert et al., "Semantic Wikis." *IEEE Software*, Juli/Agustus, 2008, 8.
- [15] Lauridas, Panagiotis, "Using Wikis in Software Development" *IEEE Software*, Maret/April, 2006, 88.