DOKUMENTASI SEBAGAI BAGIAN DARI PERANGKAT LUNAK

Effan Najwaini [1], Azhari SN [2]
[1,2]Ilmu Komputer, Universitas Gadjah Mada, Yogyakarta 55281
E-mail : effan.najwaini@gmail.com, arisn@ugm.ac.id

ABSTRAK

Dokumentasi merupakan sarana peyampaian informasi tentang perangkat lunak. Suatu program komputer belum dapat dikatakan sebuah perangkat lunak tanpa adanya dokumentasi perangkat lunak tersebut. Pembuatan dokumentasi dapat membawa banyak manfaat bagi para pengembang perangkat lunak. Dokumentasi dapat mengefisienkan waktu dari perancangan, pembuatan, pengetesan dan pemanfaatan sebuah perangkat lunak. Sayangnya banyak para pengembang yang mengabaikan kualitas dari dokumentasi perangkat lunak mereka. Dokumen sering dibiarkan tanpa diperbaharui sehingga memberikan informasi yang kurang akurat. Makalah ini membahas pembuatan dokumentasi yang baik, serta penjabaran kegunaan dari dokumen tersebut.

Kata kunci : Dokumentasi, Dokumen, Perangkat Lunak

1. PENDAHULUAN

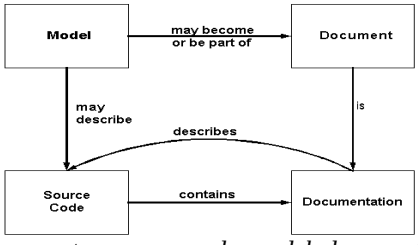
Menurut Roger S. Pressman [1], ada tiga hal yang dapat mendefinisikan suatu perangkat lunak yaitu: (1) program komputer yang bila dieksekusi akan memberikan fungsi dan kerja seperti yang diinginkan. (2) struktur data yang memungkinkan program memanipulasi informasi secara proposional, dan (3) dokumen yang menggambarkan operasi dan kegunaan program. Sehingga dapat dikatakan sebuah program komputer belum dapat disebut perangkat lunak tanpa disertai dengan dokumentasinya [2]. Hal ini menunjukkan betapa pentingnya dokumentasi pada pembuatan sebuah perangkat lunak, tetapi banyak pengembang perangkat lunak yang kurang memperhatikan masalah dokumentasi.

Dokumentasi merupakan sebuah artefak yang tujuannya untuk menyampaikan informasi tentang sistem perangkat lunak yang menyertainya [3]. Selain itu dokumentasi mempunyai fungsi sebagai berikut [4]:

1. Bertindak sebagai media komunikasi antar anggota pengembang tim,
2. Penyimpanan sistem informasi untuk digunakan oleh *maintenance engineers*,
3. Membantu manajer proyek dalam merencanakan, mengatur anggaran, dan penjadwalan dalam proses pembangunan perangkat lunak,
4. Memberi penjelasan kepada pengguna bagaimana cara menggunakan dan mengelola sistem yang dibangun.

Sebagai tempat penyimpanan informasi, dokumen semestinya harus berisi informasi yang lengkap, valid, mudah dimengerti, dan *up-to-date*. Tapi sayangnya banyak pengembang yang membiarkan dokomen yang dibuat tidak memberikan informasi yang lengkap atau informasi yang tidak diperbaharui (*out-of-date*).

Beberapa *software engineers* berpendapat bahwa *"my code is self-documenting"*. Mereka beranggapan cukup dengan *source code* sudah merupakan dokumentasinya, sehingga tidak diperlukan dokumen tambahan [5]. Hal ini mungkin dapat berlaku jika program yang dibuat untuk dirinya sendiri. Tetapi bagaimana jika program tersebut digunakan oleh orang lain atau program tersebut sebagai bagian dari sebuah sistem perangkat lunak yang dikerjakan oleh banyak orang ? *Software engineers* yang lain mungkin dapat mengerti jalannya program dengan membaca kode tersebut, tetapi tetap akan membutuhkan waktu yang lebih lama dibandingkan dengan membaca sebuah dokumen yang menjelaskan secara rinci tetang program tersebut. Scott Ambler dalam thesis Andrew Forward menjelaskan hubungan antara *source code*, model, dokumen, dan dokumentasi [3]. Ambler menjelaskan bahwa sebuah dokumentasi merupakan penjelasan dari kode yang dibuat. Sebuah model juga mungkin menjelaskan kode, dan model ini dapat menjadi dokumen atau bagian dari dokumen. Hubungan tersebut dapat dilihat pada gambar berikut.



Gambar 1: Hubungan antara *source code, model, document,* dan *documentation*

 ISBN 979 - 26 - 0255 - 0

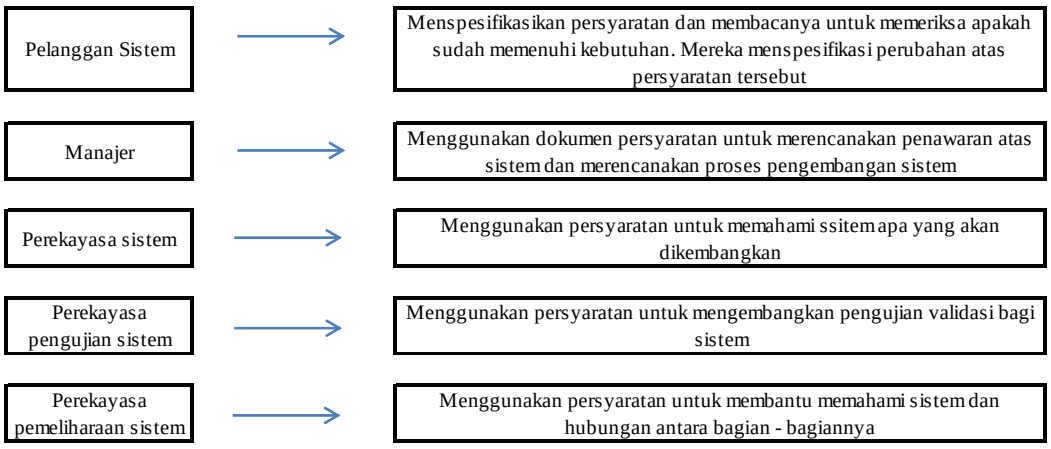
2. JENIS DOKUMENTASI DAN STRUKTUR DOKUMENTASI

Ian Sommerville mengklasifikasi dokumentasi ke dalam dua kelas, yaitu dokumentasi proses dan dokumentasi produk [4]. Dokumentasi proses merupakan dokumen yang menyimpan semua proses dari pembangunan dan pemeliharaan perangkat lunak, termasuk perencanaan, penjadwalan, lembar kerja, serta memo maupun email. Sedangkan dokumen produk yaitu dokumen yang merupakan penjelasan dari perangkat lunak yang dibangun. Dokumentasi pengguna dan dokumentasi sistem termasuk dalam dokumen produk. Dokumentasi pengguna yaitu dokumen yang menjelaskan tentang bagaimana penggunaan dari produk perangkat lunak tersebut, sedangkan dokumen sistem yaitu semua dokumen yang menjelaskan tentang sistem yang dibagun, mulai dari spesifikasi kebutuhan sampai dengan pengujian perangkat lunak.

Pada sumber lain ada yang mengklasifikasikan dokumentasi ke dalam empat bagian yaitu dokumen kebutuhan, arsitektur dan desain, dokumen teknis, dokumen end user, dan dokumen pemasaran [6]. Dokumen kebutuhan merupakan dokumen yang menjelaskan tentang atribut, kemampuan, karakterisitik, atau kualitas dari suatu sistem yang merupakan dasar dari pembuatan suatu perangkat lunak. Dokumen arsitektur dan disain yaitu dokumen yang menjelaskan tentang arsitektur sistem dan prinsip – prinsip konstruksi yang akan digunakan dalam desain komponen perangkat lunak. Dokumen teknis merupakan dokumentasi dari kode, algoritma dan interface. Dokumen end user merupakan dokumen manual tentang bagaimana perangkat lunak tersebut digunakan. Dokumen pemasaran berisi bagaimana cara pemasaran dari produk dan analisis permintaan pasar.

2.1 Dokumen Persyaratan Perangkat Lunak

Dokumen persyaratan perangkat lunak (SRS/*Software Requirements Specification*) merupakan persyaratan resmi mengenai apa yang dituntut dari pengembang sistem [7]. Dokumen berisi persyaratan user untuk sistem dan spesifikasi secara rinci dari persyaratan sistem. Berikut ilustrasi contoh dokumen persyaratan perangkat lunak dan bagaimana pemanfaatannya [7].



Gambar 2: Ilustrasi pemanfaatan dokumen persyaratan perangkat lunak

Heninger dalam buku Ian Sommerville [7] mengusulkan bahwa ada enam persyaratan yang harus dipenuhi oleh dokumen persyaratan perangkat lunak yaitu:

- Menspesifikasi perilaku sistem eksternal.
- Menspesifikasi batasan – batasan implementasi.
- Mudah diubah
- Berfungsi sebagai alat bantu referensi bagi pemelihara sistem.

Lembaga IEEE telah membuat standar untuk dokumen persyaratan perangkat lunak (IEEE/ANSI 830-1993). Berikut outline yang disarankan oleh IEEE untuk dokumen persyaratan perangkat lunak:

1. Pendahuluan
- 1.1 Tujuan dokumen persyaratan
- 1.2 Cakupan produk
- 1.3 Definisi, akronim, dan singkatan
- 1.4 Referensi
- 1.5 Tinjauan bagian dokumen berikutnya

2. Deskripsi umum
- 2.1 Perspektif produk
- 2.2 Fungsi produk
- 2.3 Karakteristik user
- 2.4 Batasan-batasan umum
- 2.5 Asumsi dan ketergantungan

3. Persyaratan khusus

4. Lampiran

5. Indeks

Gambar 3: Outline dokumen p**e**rsyaratan perangkat lunak [7]

Persyaratan khusus mencakup persyaratan fungsional, non-fungsional dan interface yang merupakan bagian penting dari dokumen persyaratan perangkat lunak. Standar dari IEEE memberikan saran apa saja yang perlu ditulis di dokumen persyaratan perangkat lunak, tetapi pemanfaatannya tergantung dari kebutuhan pengembang dan pengguna perangkat lunak tersebut.

2.2 Dokumentasi Desain

Dokumentasi desain berisi penjelasan rinci tentang inti teknis dari rekayasa perangkat lunak yang meliputi struktur data, arsitektur program, interface dan detail prosedural [1]. Gambar 3 menunjukan contoh outline dari dokumen desain yang diambil dari buku Pressman [1]. Berikut penjelasan perbagian dari Pressman mengenai outline tersebut:

- Bagian I berisi ruang lingkup dari kerja desain.
- Bagian II berisi desain data, struktur file eksternal dan referensi silang yang menghubungkan objek data dengan file tertentu.
- Bagian III berisi desain arsitektur.
- Bagian IV dan V, pada bagian ini berkembang pada saat desain interface dan procedural dimulai.
- Bagian VI berisi referensi silang yang bertujuan utnuk menetapkan bahwa semua persyaratan dipenuhi oleh desain perangkat lunak dan menunjukkan modul mana yang krites terhadap implementasi persyaratan spesifik.
- Bagian VII berisi tahap pertama dari pembuatan dokumentasi pengujian.
- Bagian VIII dan IX berisi data tambahan meliputi deskripsi algoritma, prosedur alternative, data dalam bentuk tabel, kutipan dari dokumen lain, dan informasi relevan lainnya.

 ISBN 979 - 26 - 0255 - 0

I. Ruang lingkup
   A. Sasaran sistem
   B. Persyaratan utama perangkat lunak
   C. Batasan-batasan dan pembatasan desain
II. Desain Data
   A. Objek data dan struktur data resultan
   B. Struktur file dan database
      1. struktur file eksternal
         a. struktur logis
         b. deskripsi record logis
         c. metode akses
      2. data global
      3. file dan referensi lintas data
III. Desain Arsitektural
   A. Kajian data dan aliran kontrol
   B. Struktur program yang diperoleh
IV. Desain Interface
   A. Spesifikasi interface manusia-mesin
   B. Aturan desain interface manusia-mesin
   C. Desain interface eksternal
      1. interface untuk data eksternal
      2. interface untuk sistem atau peralatan eksternal
V. Desain Prosedural
   untuk masing-masing modul
   A. Naratif pemrosesan
   B. Deskripsi interface
   C. Deskripsi bahasa (atau lainnya) desain
   D. Modul-modul yang digunakan
   E. Struktur data internal
   F. Kererangan/larangan/pembatasan
VI. Persyaratan Lintas-Referensi
VII. Ketentuan Pengujian
   1. Panduan pengujian
   2. Strategi integrasi
   3. Pertimbangan khusus
VIII. Catatan Khusus
IX. Lampiran

Gambar 4: Outline dokumen desain [1]

2.3 Dokumentasi Pengujian

Pengujian perangkat lunak merupakan sederetan langkah yang digunakan untuk melakukan pengujian atau pengecekan terhadap unit program ataupun sistem lengkap dari perangkat lunak untuk menjamin bahwa persyaratan sistem telah dipenuhi. Pengujian memastikan bahwa program tersebut telah berfungsi sebagaimana mestinya. Rencana, hasil serta prosedur pengujian harus didokumentasikan dalam suatu dokumen pengujian. Gambar 5 menunjukkan outline dari dokumen pengujian.

I. Lingkup Pengujian
II. Rencanan Pengujian
   A. Phase dan build pengujian
   B. Jadwal
   C. Perangkat lunak overhead
   D. Lingkungan dan sumber daya
III. *n* Prosedur Pengujian (deskripsi pengujian untuk *n* build)
   A. Urutan integrasi
      1. tujuan
      2. modul untuk diuji
   B. Pengujian unit untuk modul-modul dalam build
      1. deskripsi pengujian untuk *n* modul
      2. deskripsi perangkat lunak overhead
      3. hasil yang diharapkan
   C. Lingkungan pengujian
      1. peranti atau teknik khusus
      2. deskripsi perangkat lunak oeverhead
   D. Data test case
   E. Hasil yang diharapkan untuk *n* build
IV. Hasil Pengujian Sesungguhnya
V. Referensi
VII. Lampiran

Gambar 5: Outline dokumen desain [1]

2.4 Dokumentasi Pengguna

Dokumentasi pengguna merupakan dokumen yang menyertai sebuah perangkat lunak yang berisi penjelasan secara detail tentang perangkat lunak tersebut. Dokumen pengguna menjelaskan setiap *feature* dari perangkat lunak dan menjelaskan bagaimana cara menggunakan setiap *feature* tersebut. Selain itu dokumen pengguna juga dapat memberikan penjelasan terhadap setiap masalah atau error yang terjadi dan bagaimana cara menanganinya. Dokumen pengguna dapat berupa dokumen cetak, elektronik, dokumen online yang mudah diakses ataupun gabungan dari semuanya. Dengan adanya dokumen pengguna ini, pengguna dapat dimudahkan dalam menggunakan perangkat lunak tersebut.

IEEE telah mendefinisikan standar untuk dokumentasi pengguna. Pada standar tersebut, IEEE mendefinisikan komponen-komponen yang semestinya ada pada dokumentasi pengguna. Komponen yang disarankan oleh IEEE dapat dijadikan panduan untuk membuat dokumentasi pengguna. Komponen tersebut dapat dilihat di tabel berikut:

Tabel 1:Komponen pada dokumen pengguna perangkat lunak [4]; [8].

Component	Description	Required?
Identification data (package label/title page)	Data such as a title and identifier that uniquely identifies the document.	Yes
Table of contents	Chapter/section names and page numbers.	Yes, in documents of more than eight pages after the identification data
List of illustrations	Figure numbers and titles	Optional
Introduction	Defines the purpose of the document and a brief summary of the contents	Yes
Information for use of the documentation	Suggestions for different readers on how to use the documentation effectively.	Yes
Concept of operations	An explanation of the conceptual background to the use of the software.	Yes
Procedures	Directions on how to use the software to complete the tasks that it is designed to support.	Yes (instructional mode)
Information on software commands	A description of each of the commands supported by the software.	Yes (reference mode)
Error messages and problem resolution	A description of the errors that can be reported and how to recover from these errors.	Yes
Glossary	Definitions of specialized terms used.	Yes, if documentation contains unfamiliar
Related information sources	References or links to other documents that provide additional information	Optional
Navigational features	Features that allow readers to find their current location and move around the document.	Yes
Index	A list of key terms and the pages where these terms are referenced.	Yes, in documents of more than 40 pages
Search capability	In electronic documentation, a way of finding specific terms in the document.	Yes, in electronic documents

3. KUALITAS DOKUMENTASI

Berdasarkan hasil survei yang dilakukan oleh Andrew Forward, *software engineers* mengungkapkan dokumen seperti apa yang dianggap berkualitas bagus, jelek dan sangat buruk [9].

1. Dokumen berkualitas bagus
    - Arsitektur dan informasi dokumentasi lainnya selalu valid atau setidaknya menyediakan panduan sejarah yang dapat berguna untuk pemeliharaan perangkat lunak.
    - Inline comments pada kode program cukup baik dalam memberikan informasi yang berguna untuk pemeliharaan perangkat lunak.
2. Dokumen berkualitas jelek
    - Dokumentasi untuk semua jenis sering sekali tidak diperbaharui (*out of date)*
    - Sistem mempunyai terlalu banyak dokumentasi
    - Penulisan dokumentasi yang buruk
    - Pengguna kesulitan menemukan isi yang berguna dalam dokumentasi
    - Pembuatan dokumentasi yang memakan waktu yang tidak sebanding dengan keuntungan dari dokumentasi tersebut
3. Dokumen berkualitas sangat buruk
    - Sebuah dokumentasi yang informasinya tidak dapat dipercaya

Secara umum dokumentasi yang bagus yaitu dokumen yang ditulis dengan baik, mudah dibaca dan dimengerti serta memberikan informasi yang lengkap dan akurat. Walaupun pembuatan dokumen yang seperti ini mungkin akan menyita waktu yang lebih banyak, tetapi dengan dokumen yang baik akan sangat membantu baik itu pengembang maupun pengguna program tersebut.

Berdasarkan survei Andrew Forward [10] menunjukkan bahwa isi dokumen merupakan atribut dokumen yang paling penting dari sebuah dokumentasi perangkat lunak. Tiga atribut lainnya yang dianggap penting yaitu *up-to-date, availability, use of examples*. Atribut-atribut tersebut yang sangat menentukan kualitas suatu dokumen, walaupun atribut lainnya juga tidak kalah pentingnya.

4. ALAT BANTU

Ada banyak *software tool* yang dapat digunakan untuk membantu membuat dokumentasi perangkat lunak. Penggunaan tool dapat mempercepat dan mempermudah dalam pembuatan dokumentasi. Berdasarkan survei yang dilakukan oleh Andrew Forward, *software tools* yang sering digunakan oleh para pengembang terlihat dari tabel berikut [11]:

Tabel 2: Useful Documentation Technologies [11]

Documentation Technology	Frequency	Percentage of Participants
MS Word (and other word processors)	22	54
Javadoc and similar tools (Doxygen, Doc++)	21	51
Text Editors	9	22
Rational Rose	5	12
Together (Control Centre, IDE)	3	7

Teknologi lainnya yang digunakan pengembang berdasarkan survei tersebut yaitu Ar goUML, Visio, FrameMaker, Author-IT, whiteboards dan digital cameras, JUnit dan XML editors. Word processors paling banyak digunakan karena merupakan *tool* yang gampang digunakan dan lebih fleksibel. Tool yang berguna lainnya yaitu software sejenis mindmap (freemind). Software tersebut dapat membantu untuk pengembang dalam menuliskan dokumentasi terkait dengan pembuatan perangkat lunak.

5. HASIL DAN PEMBAHASAN

Beberapa lembaga maupun peneliti telah memberikan outline yang dapat digunakan sebagai dasar pembuatan dokumen perangkat lunak. Outline tersebut sangat membatu untuk menentukan hal apa yang seharusnya ditulis dalam dokumen tersebut. Standar dalam pembuatan dokumen bukan hal yang mutlak harus diikuti, tetapi alangkah lebih baik jika mengikuti standar yang telah dibuat oleh lembaga tertentu, misalnya standar IEEE. Jika tidak mengikut standar tersebut, tujuan utama dokumentasi yaitu memberikan informasi yang lengkap dan akurat harus tetap dipenuhi.

Dokumentasi yang baik akan membawa manfaat yang cukup besar baik itu bagi pengembang maupun bagi pengguna perangkat lunak. Dokumentasi yang baik yaitu dokumen yang memberikan informasi yang lengkap dan akurat, mudah dibaca dan dimengerti, serta ditulis dengan baik. Berikut beberapa manfaat dari pembuata dokumen perangkat lunak yang baik:

- Seorang *software engineers* untuk memahami cara kerja suatu program atau perangkat lunak mungkin cukup dengan membaca kode yang dibuat. Tetapi hal itu akan memakan waktu yang lebih lama dibandingkan dengan membaca sebuah dokumen yang berisi data lengkap tentang alur program tersebut. Begitu juga ketika melakukan pengujian perangkat lunak. Ketika ditemukan adanya kesalahan atau bug dalam program tersebut, maka diperlukan perbaikan kode program. Dengan adanya dokumentasi yang baik mungkin akan mempersingkat waktu perbaikan dari kode program tersebut [12].

- Dengan adanya dokomentasi perencanaan, persyaratan, desain yang baik akan lebih mempercepat pembuatan sebuah perangkat lunak. Pembuatan perangkat lunak juga lebih terstruktur sehingga dapat membuat perangkat lunak yang berkualitas baik.

- Dari sisi pengguna, pengguna dapat dengan cepat mengerti cara kerja dari perangkat lunak tersebut dan dapat memanfaatkan semua *feature*-nya dengan maksimal. Pengguna juga dapat dengan cepat menangani berbagai masalah (*error*) dari perangkat lunak tersebut.

- Dokumentasi persyaratan perangkat lunak juga sebagai sarana komunikasi dengan pelanggan. Pembuatan dokumen persyaratan yang baik mampu memberikan gambaran secara detail mengenai kebutuhan dari pelanggan tersebut. Dari dokumen ini kemudian para pengembang dapat membuat perangkat lunak yang benar-benar sesuai dengan kebutuhan pelanggan tersebut

- Dokumentasi yang baik, nantinya dapat menjadi referensi dalam pembuatan program atau perangkat lunak berikutnya. Pengembang dapat mempelajari kekurangan-kekurangan dalam pembuatan program sebelumnya sehingga pada proyek pembuatan program berikutnya dapat berjalan lebih baik, lebih cepat serta efektif dan efisien dari segi biaya, waktu dan tenaga kerja.

Dalam pembuatan dokumen perangkat lunak dapat dibantu dengan *tools* (alat atau perangkat lunak) sehingga pembuatan dokumen dapat lebih cepat. Pengembang juga dapat menggunakan layanan kolaboratif dokumen seperti google docs, etherpad, zoho, serta layanan online lainnya. Aplikasi tersebut dapat membatu untuk membuat suatu dokumen secara bersama-sama real-time. Jika tidak ingin membuat dokumen kolaboratif secara online lewat internet dapat membuat *software* kolaboratif *client-server* yang dibuat di jaringan pribadi. Atau solusi sederhana dapat memanfaatkan ftp server atau samba server pada jaringan lokal untuk berbagi dokumen yang nantinya dapat digunakan untuk pembuatan dokumen secara bersama-sama.

6. PENUTUP

Dokumentasi merupakan artefak yang berisi informasi dari sebuah perangkat lunak yang menyertainya. Sebuah dokumen yang baik yaitu dokomen yang dapat memberikan informasi secara lengkap dan akurat, ditulis dengan baik, mudah dibaca dan gampang dimengerti. Sebuah dokumen yang baik dapat menunjukkan kualitas dari perangkat lunak tersebut. Perangkat lunak belum dapat dikatakan berkualitas tinggi jika disertai dengan dokumentasi yang tidak lengkap, tidak update atau memberikan informasi yang tidak benar. Pembuatan dokumentasi yang baik dapat memberikan banyak manfaat bagi pembuatan perangkat lunak. Pembuatan dokumen yang buruk pada awal perancangan perangkat lunak, dapat mempengaruhi kualitas dari perangkat lunak yang dibuat. Oleh sebab itu, para pengembang tidak boleh mengabaikan pembuatan dokumentasi yang baik atau memenuhi standar.

DAFTAR PUSTAKA

[1] R. S. Pressman, *Software Engineering: A Practitioner's Approach*. New York: Mc Graw-Hill Companies, Inc, 1997.
[2] R. A.S and M. Shalahuddin, *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung: Modula, 2011.
[3] A. Forward, "Software Documentation – Building and Maintaining Artefacts of Communication," University of Ottawa, 2002.
[4] I. Sommerville, "Software Documentation," 2011.
[5] D. L. Parnas, "Precise Documentation : The Key To Better Software," *Joburg Centre for Software Engineering*.
[6] R. Shujaat, "Types of Software Documentation," 2009. [Online]. Available: http://rafia-shujaat.suite101.com/types-of-software-documentation-a107716. [Accessed: 23-Apr-2012].
[7] I. Sommerville, *Software Engineering*, 06 ed. London: Pearson Education, 2000.
[8] IEEE, *IEEE Standard for Software User Documentation, IEEE-Std1063-2001*. New York: Institute of Electrical and Electronics Engineers, 2001.
[9] T. C. Lethbridge, J. Singer, A. Forward, and D. Consulting, "How Software Engineers Use Documentation : The State of the Practice Documentation :," *IEEE Computer Society*, pp. 35-39, 2003.
[10] A. Forward, K. Edward, and T. C. Lethbridge, "Software Engineering Documentation Priorities : An Industrial Study," *University of Ottawa*, pp. 1-13, 2002.
[11] A. Forward, K. Edward, and T. C. Lethbridge, "The Relevance of Software Documentation , Tools and Technologies : A Survey," *University of Ottawa*, 2002.
[12] M. D. Ernst, "Automated documentation inference to explain failed tests," *2011 26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011)*, pp. 63-72, Nov. 2011.