

# Membuat Software Requirement

**Indra Kharisma Raharjana**

*indra.kharisma@gmail.com*

*http://rekayasaperangkatlunak.com*

## ***Lisensi Dokumen:***

*Copyright © 2003-2007 IlmuKomputer.Com*

*Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.*

Dalam IEEE Standard Glossary of Software Engineering Technology (IEEE Std 610.12-1990) [IEEE] requirement dapat diartikan sebagai berikut.

1. Suatu kondisi atau kemampuan yang diperlukan oleh user untuk memecahkan masalah atau mencapai tujuan
2. Suatu kondisi atau kemampuan yang harus dipenuhi atau dimiliki oleh sistem atau komponen sistem untuk memenuhi kontrak, standard, spesifikasi atau dokumen formal lain
3. Gambaran yang terdokumentasi dari kondisi atau kemampuan yang disebut pada kondisi 1 dan 2 diatas

Menurut sommerville [SOMM] *requirement* adalah spesifikasi dari apa yang harus diimplementasikan, deskripsi bagaimana sistem harusnya berkerja atau bagian-bagian yang ada didalam sistem, bisa juga dijadikan batasan dalam proses pengembangan sistem.

Ada beberapa macam *requirement* menurut sommerville [SOMM] yaitu:

a. *User Requirement* (kebutuhan pengguna)

Pernyataan tentang layanan yang disediakan sistem dan tentang batasan-batasan perasionalnya. Pernyataan ini dapat dilengkapi dengan gambar/diagram yang dapat dimengerti dengan mudah.

b. *System requirement* (kebutuhan system)

Sekumpulan layanan/kemampuan sistem dan batasan-batasannya yang ditulis secara detil. System requirement document sering disebut functional

specification (spesifikasi fungsional), harus menjelaskan dengan tepat dan detail. Ini bisa berlaku sebagai kontrak antara klien dan pembangun.

c. *Software design specification* (spesifikasi rancangan perangkat lunak)

Gambaran abstrak dari rancangan software yang menjadi dasar bagi perancangan dan implementasi yang lebih detail.

Disitus Wikipedia [WIKI] menjelaskan bahwa *requirement* khususnya dalam *engineering* mempunyai arti *a singular documented need of what a particular product or service should be or do*. Istilah tersebut sering digunakan dalam bidang *system engineering* dan *software engineering*. Fase dalam pengembangan *requirement* dapat dipecah-pecah menjadi : *requirements elicitation* (mengumpulkan kebutuhan *stakeholders*), *analysis* (memeriksa konsistensi dan keterlengkapan), *specification* (mendokumentasikan *requirements*) and *verification* (memastikan bahwa *requirements* yang telah dispesifikasikan benar)

Secara umum, *requirement* dibagi menjadi 2 yaitu

- *Functional requirement* : menjelaskan tentang fungsional dari sistem
- *Non-Functional requirement* : yang berperan untuk member batasan pada solusi atau biasa disebut *quality requirement*.

*Requirement* adalah pernyataan yang mengidentifikasi kebutuhan yang penting dalam sistem dan didalamnya mencakup aspek kebenaran, Realistis, Dibutuhkan, tidak ambigu, dan terukur. Langkah yang paling penting dalam proses *requirement* adalah komunikasi yang akurat antara *user* yang memerlukan sistem dengan pembuat sistem.

Banyak orang sering salah dalam mendefinisikan *requirement* untuk sistem yang mereka bangun, kerana mereka kurang mendapat pelatihan dan pengalaman dalam membuat *requirement*. Bahkan di bangku kuliah yang mengajarkan mata kuliah *system engineering* hanya mengajarkan pengenalan untuk materi menulis *requirement*.

Aspek yang penting dari sistem engineering adalah merubah kebutuhan user menjadi jelas, ringkas, dan dapat diverifikasi.

*Requirement* yang baik menyatakan sesuatu yang **dibutuhkan, dapat diverifikasi, memungkinkan, dan Jelas.**

Terdapat beberapa masalah yang sering ditemui dalam membuat *requirement*, diantaranya adalah : membuat asumsi yang buruk, menulis implementasi (HOW) daripada *requirement* (WHAT), menjelaskan operasional daripada kebutuhan, menggunakan istilah yang salah, menggunakan bahasa yang kurang tepat, *requirement* tidak lengkap, dan menspesifikasikan *requirement* secara berlebihan.

Asumsi yang buruk terjadi karena pembuat requirement tidak mempunyai akses yang cukup terhadap informasi yang dibutuhkan. Hal ini bisa dikurangi dengan cara mendokumentasikan informasi yang kritis seperti kebutuhan, tujuan, batasan, misi dan lain-lain. Selain itu semua asumsi juga harus didokumentasikan sehingga ketika direview asumsi bisa diperbaiki jika tidak sesuai.

Spesifikasi requirement harus menyatakan Apa yang dibutuhkan (WHAT), bukan bagaimana hal tersebut di sediakan (WHY). Caranya dengan menanyakan mengapa (WHY) anda membutuhkan requirement tersebut.

Requirement biasanya ditulis dalam format (contoh dalam bahasa inggris)

- The System shall provide .....
- The System shall be capable of .....
- The System shall weigh .....
- The Subsystem #1 shall provide .....

Semua requirement harus bisa diverifikasi, kata-kata yang ambigu seperti dibawah ini (dalam bahasa inggris) harus dihindari karena tidak bisa diuji kecocokan sistem dan requirement nantinya.

- Minimize
- Maximize
- Rapid
- User-friendly
- Easy
- Sufficient
- Adequate
- Quick

Requirement yang tidak lengkap sangat mungkin terjadi karena team yang menulis requirement terfokus pada bagian tertentu pada sistem, untuk menghindari ketidaklengkapan requirement perlu diperhatikan detail-detail berikut ini

- Functional · Reliability
- Performance · Maintainability
- Interface · Operability
- Environment · Safety
- Facility · Regulatory
- Transportation · Security
- Deployment · Privacy
- Training · Design constraints
- Personnel

menspesifikasikan requirement secara berlebihan bisa menyebabkan biaya yang berlebihan dalam pengembangan sistem, biasanya berawal dari sesuatu yang tidak dibutuhkan atau requirement yang terlalu ketat.

### Kesimpulan

Requirement yang baik menyatakan sesuatu yang dibutuhkan, dapat diverifikasi, memungkinkan, dan Jelas. Namun dengan definisi tersebut masih banyak orang yang salah dalam menulis requirement, untuk mengurangi kesalahan orang dalam menulis requirement dibutuhkan arahan dan contoh yang benar dalam menulis requirement. Namun sayangnya arahan yang diberikan diatas terdapat point-point yang spesifik ke bahasa inggris, maka perlu juga di buat arahan yang spesifik ke bahasa Indonesia.

### Referensi

[IEEE] IEEE Standard Glossary of Software Engineering Technology, IEEE Std 610.12-1990, Institute of Electrical and Electronics Engineers, New York, 1990

[WEBS] Merriam-Webster Online, <http://www.merriam-webster.com/>

[SOMM] Sommerville, Ian. (2001), "Software Engineering" 6<sup>th</sup>. Addison Wesley.

[WIKI] "requirements", <http://en.wikipedia.org/wiki/Requirement>

[HOOKS] Ivy Hooks , "Writing Good Requirements", *Proceedings of the Third International Symposium of the INCOSE - Volume 2*, 1993.

### Biografi Penulis



**Indra Kharisma Raharjana.** Menyelesaikan program sarjana di [Institut Teknologi Sepuluh Nopember](#) jurusan teknik Informatika pada tahun 2005 dan saat ini sedang menyelesaikan program master Informatika konsentrasi bidang rekayasa perangkat lunak dan data di [Institut Teknologi Bandung](#). Pengajar di program studi sistem informasi [Universitas Airlangga](#), saat ini sedang merintis situs e-learning <http://rekayasaperangkatlunak.com> yang fokus pada pengembangan bidang ilmu rekayasa perangkat lunak di Indonesia.