Requirements Engineering: Mari Pecahkan Masalah Batu !

Romi Satria Wahono
romi@romisatriawahono.net
http://romisatriawahono.net

Requirements engineering adalah fase terdepan dari proses software engineering, dimana software requirements (kebutuhan) dari user (pengguna) dan customer (pelanggan) dikumpulkan, dipahami dan ditetapkan. Para pakar software engineering (rekayasa perangkat lunak) sepakat bahwa requirements engineering adalah suatu pekerjaan yang sangat penting, terutama berdasarkan fakta bahwa kebanyakan kegagalan pengembangan software disebabkan karena adaya ketidakkonsistenan (*inconsistent*), ketidaklengkapan (*incomplete*), maupun ketidakbenaran (*incorrect*) dari requirements specification (spesifikasi kebutuhan) **[Romi-03]**. The Standish Group mencatat bahwa prosentase akumulatif kegagalan sebuah project pengembangan software sebagian besar disebabkan oleh masalah requirements dan spesifikasinya **[Standish-94]**.

Untuk merangkum masalah yang ingin dipecahkan dalam cabang ilmu requirements engineering, kebanyakan pakar mengamini ungkapan Ed Yourdon dalam *foreword* yang ditulisnya untuk buku “Managing Software Requirements - A Unified Approach” karya Dean Leffingwell **[Leffingwell-00]**. Ed Yourdon menggunakan istilah “*the rock problem*” (masalah batu) sebagai diskusi dasar masalah yang selalu muncul dalam proses pengerjaan proyek software.

Customer (pelanggan) yang datang kepada kita untuk mengerjakan sebuah proyek pengembangan software, adalah ibarat seseorang yang mengatakan kepada kita, “Tolong buatkan saya batu”. Ketika kita memberikan kepadanya sebuah batu, dia akan melihatnya sebentar dan mengatakan kepada kita, “Ya terima kasih, tapi sebenarnya yang saya inginkan adalah sebuah batu kecil berwarna biru”. Dan ketika kita bawakan untuknya batu kecil berwarna biru, dia mengatakan bahwa yang diinginkan adalah yang “bentuknya bulat”. Demikian seterusnya proses iterasi (iteration) terjadi berulangkali sampai akhirnya kita dapatkan yang sebenarnya diinginkan customer kita adalah “batu pualam kecil berwarna biru”. Meskipun mungkin sebenarnya bukan “tepat yang

diinginkan", tapi paling tidak "paling dekat" dengan yang diinginkan customer. Dan mungkin saja terjadi, customer kita mengubah pikiran tentang requirements pada saat proses interaksi dengan pengembang terjadi (dari iterasi pertama yang sekedar batu, sampai iterasi terakhir yang menghasilkan batu pualam kecil berwarna biru).

Frustasi terjadi baik di pihak pengembang maupun customer. Pengembang frustasi karena merasakan perbedaan signifikan antara requirements pertama yang dijelaskan customer dengan yang sebenarnya diinginkan customer, belum lagi waktu dan biaya besar sudah dikeluarkan pengembang dalam tiap iterasi. Di lain pihak, customer juga frustasi karena merasa sudah menjelaskan dengan baik dan pengembang tidak bisa memahami yang diinginkan, sehingga memerlukan waktu yang sangat lama.

Fenomena senada dengan masalah batu diatas sering juga disebut dengan *"Yes, But Syndrome"* (That is what I meant, but not exactly what I meant) dan *"Undiscovered Ruins Syndrome"* (Now that I see it, I have another requirement to add) **[Romi-02]**.

Dari ilustrasi diatas, muncul keinginan untuk menerapkan pendekatan engineering (engineering approach) untuk memecahkan masalah tersebut, yang akhirnya membawa arus deras kemunculan cabang ilmu requirements engineering.

Definisi

Untuk memperkuat pemahaman dari seri tulisan requirements engineering ini, alangkah baiknya apabila diskusi ini kita mulai dari definisi terminologi-terminologi yang digunakan.

Requirements
Kita mulai dari definisi requirement, menurut IEEE Standard Glossary of Software Engineering Technology (IEEE Std 610.12-1990) **[IEEE-610.12]**, 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 1 dan 2

User, Customer, Supplier
Terminologi "user" pada definisi requirements diatas bisa merupakan end-user (pengguna akhir) dari software yang akan dikembangkan atau orang-orang dibelakang layar. Termasuk didalamnya orang yang menggunakan informasi yang dihasilkan dari software, dan juga menunjuk ke customer (client) yang membayar proyek pengembangan software. Penggunaan terminologi user seperti ini (untuk end-user dan stakeholder) banyak digunakan **[Vliet-00]** dalam penjelasan masalah requirements engineering, termasuk didalamnya proses iterasi, metodologi requirements elicitation, dan sebagainya.

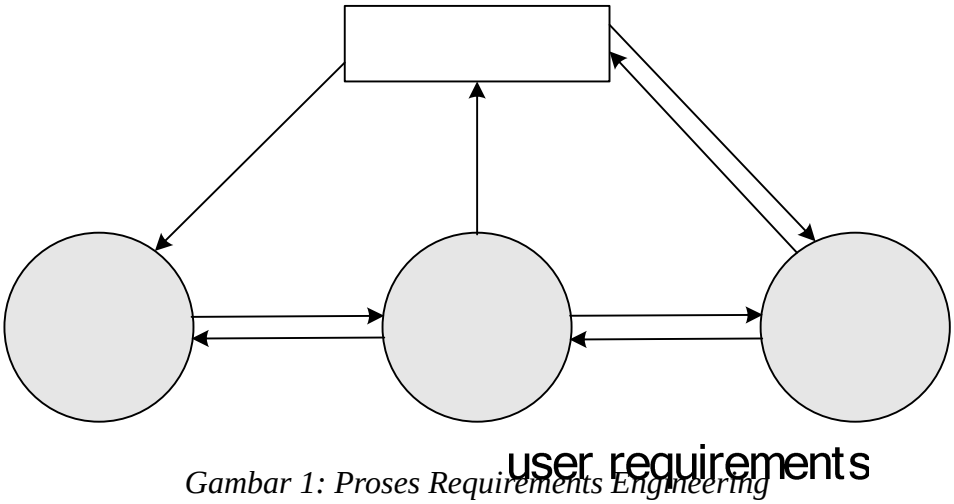
Meskipun kalau kita mau mempelajari kembali, sebenarnya ada perbedaan mendasar definisi formal dari terminologi berhubungan dengan orang-orang dalam requirements engineering **[IEEE-610.12]**:

- *Customer*: Orang atau orang-orang yang membayar produk dan biasanya (tidak selalu yang memutuskan requirements. Dalam konteks ini direkomendasikan bahwa customer dan supplier adalah anggota dari organisasi yang sama.
- *Supplier*: Orang atau orang-orang yang memproduksi produk untuk customer. Dalam konteks ini direkomendasikan bahwa customer dan supplier adalah anggota dari organisasi yang sama.
- *User*: Orang atau orang-orang yang mengoperasikan atau berinteraksi secara langsung dengan produk. User dan customer kadang bukanlah orang yang sama.

Requirements Engineering
Banyak definisi yang diungkapkan oleh para peneliti tentang requirements engineering. Pada kesempatan kali ini penulis sajikan satu definisi yang cukup jelas dan banyak digunakan, yaitu yang diuraikan oleh Pamela Zave **[Zave-97]**:

> Requirements engineering adalah cabang dari software engineering yang mengurusi masalah yang berhubungan dengan: tujuan (dunia nyata), fungsi, dan batasan-batasan pada sistem software. Termasuk hubungan faktor-faktor tersebut dalam menetapkan spesifikasi yang tepat dari suatu software, proses evolusinya baik berhubungan dengan masalah waktu maupun dengan software lain (dalam satu famili).

Proses Requirements Engineering
Hasil dari fase requirements engineering terdokumentasi dalam requirements specification. Requirements specification berisi kesepakatan bersama tentang permasalahan yang ingin dipecahkan antara pengembang dan customer, dan merupakan titik start menuju proses berikutnya yaitu software design. Sistemisasi proses negosiasi pengembang dan customer dalam requirements engineering dibagi dalam 3 proses besar yaitu: elicitation, specification, validation and verification. Formula ini kemudian juga dikenal dengan nama *The Three Dimensions of Requirements Engineering* (Gambar 1). Proses requirements engineering ini dilakukan secara iterasi dengan mengakomodasi adanya feedback dari customer (user). Perlu dicatat bahwa terminologi user pada Gambar 1 adalah melingkupi juga client/customer.



Gambar 1: Proses Requirements Engineering

Requirements Elicitation
Adalah proses mengumpulkan dan memahami requirements dari user. Kadang masalah yang muncul berakar dari gap masalah knowledge domain (perbedaan disiplin ilmu yang dimiliki). Customer adalah expert pada domain yang softwarenya ingin dikembangkan (domain specialist), dilain pihak sang pengembang (requirements analyst) adakalanya sama sekali buta terhadap knowledge domain tersebut, meskipun tentu memahami dengan benar bagaimana sebuah software harus dikembangkan. Gap knowledge domain tersebut yang diharapkan bisa diatasi dengan adanya interaksi terus menerus dan berulang (iterasi) antara pengembang dan customer. Proses interaksi tersebut kemudian dimodelkan menjadi beberapa teknik dan metodologi diantaranya adalah interviewing, brainstorming, prototyping, use case, dsb.

Requirements Specification
Setelah masalah berhasil dipahami, pengembang mendeskripsikannya dalam bentuk dokumen spesifikasi dokumen. Spesifikasi ini berisi tentang fitur dan fungsi yang diinginkan oleh customer, dan sama sekali tidak membahas bagaimana metode pengembangannya. IEEE mengeluarkan standard untuk dokumen spesifikasi requirements yang terkenal dengan nama IEEE Recommended Practice for Software Requirements Specifications **[IEEE-830]**. Dokumen spesifikasi requirements bisa berisi functional requirements, performance requirements, external interface requirements, design constraints, maupun quality requirements.

Requirements Validation and Verification
Setelah spesifikasi requirements berhasil dibuat, perlu dilakukan dua usaha:

- Validation (validasi), yaitu proses untuk memastikan bahwa requirements yang benar sudah ditulis
- Verification (verifikasi), yaitu proses untuk memastikan bahwa requirements sudah ditulis dengan benar

Proses validasi dan verifikasi ini melibatkan customer (user) sebagai pihak yang menilai dan memberi feedback berhubungan dengan requirements.

Penutup dan Diskusi

Hakekat dari cabang ilmu requirements engineering adalah bagaimana kita bisa memecahkan masalah batu (the rock problem) seperti yang telah dijelaskan diawal artikel. Pada artikel ini diuraikan tentang hakekat munculnya cabang ilmu requirements engineering, definisi dan pengertian, serta proses-prosesnya.

Referensi:

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

[IEEE-830] IEEE Recommended Practice for Software Requirements Specifications, IEEE Std 830-1998, Institute of Electrical and Electronics Engineers, New York, 1998.

[Leffingwell-00] Dean Leffingwell and Don Widrig, *Managing Software Requirements – A Unified Approach*, Addison Wesley, 2000.

[Loucopoulos-95] P. Loucopoulos and V. Karakostas, *Software Requirements Engineering*, McGraw-Hill, 1995.

[Vliet-00] Hans Van Vliet, *Software Engineering - Principles and Practice*, John Wiley & Sons, 2000.

[Romi-02] Romi Satria Wahono and Jingde Cheng, *Extensible Requirements Patterns of Web Application*, IEEE International Symposium on Cyber Worlds (CW 2002), Japan, 2002.

[Romi-03] Romi Satria Wahono, *Analyzing Requirements Engineering Problems*, IECI Japan Workshop 2003 (IJW-2003), Japan, 2003.

[Standish-94] The Standish Group, *Charting the Seas of Information Technology – Chaos*, The Standish Group International, 1994.

[Zave-97] Pamela Zave, *Classification of Research Efforts in Requirements Engineering*, ACM Computing Surveys, 29(4), pp. 315-321, 1997.

Biografi Penulis

Romi Satria Wahono. Menamatkan SMU di SMU Taruna Nusantara, Magelang pada tahun 1993. Menyelesaikan program S1 dan S2 di *Department of Computer Sciences*, *Saitama University*, Jepang tahun 1999 dan 2001. Saat ini program program S3 pada jurusan yang sama. Peneliti di Lembaga Ilmu Pengetahuan Indonesia (LIPI). Kompetensi inti pada bidang *Software Engineering*, *eLearning System*, dan *Knowledge Management*. Aktif sebagai penulis, dimana ratusan tulisan berupa scientific paper, artikel, dan tutorial telah diterbitkan dalam berbagai proceedings conference, jurnal ilmiah, majalah, koran dan portal, bertaraf nasional maupun internasional. Mendapatkan penghargaan dari PBB pada pertemuan puncak WSIS (*World Summit on Information Society*) tahun 2003 di jenewa, sebagai pendiri dari IlmuKomputer.Com. CEO dari PT Brainmatics Cipta Informatika, perusahaan IT yang bergerak di bidang training dan consulting.

Informasi lebih lanjut tentang penulis bisa didapat melalui:
Email: *romi@romisatriawahono.net*
URL: *http://romisatriawahono.net*
YM: *romi_sw*