

# STUDI ANALISIS PERSYARATAN KEBUTUHAN SISTEM DALAM MENGHASILKAN PERANGKAT LUNAK YANG BERKUALITAS

SANDY KOSASI<sup>1</sup>, SUSANTI MARGARETHA KUWAY<sup>2</sup>

<sup>1,2</sup>Sekolah Tinggi Manajemen Informatika dan Komputer Pontianak  
Program Studi Sistem Informasi

Jln. Merdeka No. 372 Pontianak, Kalimantan Barat

<sup>1</sup>E-mail: [sandykosasi@yahoo.co.id](mailto:sandykosasi@yahoo.co.id) dan [sandykosasi@stmikpontianak.ac.id](mailto:sandykosasi@stmikpontianak.ac.id)

<sup>2</sup>E-mail : [Shanty\\_stmikptk@yahoo.com](mailto:Shanty_stmikptk@yahoo.com)

***Abstracts:** Requirements specification is the need for the early phases of software engineering and is very important because it affects the flexibility, reliability, accuracy and usability. No matter how good coding and design systems that do not will still be able to provide a real desire by the conditions encountered. In fact, the needs for requirements engineering and FAST approach method does not always ensure produce quality software systems so that require system migration. In this process the need for requirements analysis should also consider factors that can not be empirically expressed quantitatively. Further requires adaptability and develop alternative priority needs in terms of the requirements document. Communication skills, the use of simple terms, open to users of the system, and displays a vivid example of what is needed and avoid the pitfalls of the intelligence domain to limit the success factors in aligning with the expectations of users of the system.*

***Keywords :** requirements specification analysis, requirements engineering, FAST approach, system migration, quantitative and empirical pitfalls intelligence domain.*

## 1. PENDAHULUAN

Melakukan rekayasa perangkat lunak merupakan suatu proses yang cukup rumit dan seringkali menimbulkan masalah yang bahkan tidak pernah akan dapat mencapai suatu hasil yang maksimal. Memahami setiap persoalan sistem jelas bukan merupakan kegiatan yang mudah karena memiliki sifat yang bervariasi. Masing-masing sistem memiliki konten yang membutuhkan deskripsi layanan dan batasan tertentu dan merupakan persyaratan untuk berhasilnya sebuah sistem. Untuk memenuhi persyaratan sebuah sistem harus mendefinisikan kebutuhan dengan cara-cara yang cukup abstrak sehingga suatu solusi tidak didefinisikan sebelumnya. Persoalan yang sering terjadi dalam proses rekayasa perangkat lunak merupakan akibat dari kegagalan membuat sebuah pemisahan yang jelas antara tingkatan deskripsi dari sejumlah layanan dan batasan dari sebuah sistem. Persyaratan kebutuhan sistem merupakan atribut yang diperlukan di dalam sebuah sistem seperti persyaratan dari sisi pemakai, persyaratan sistem atau spesifikasi fungsional dan penentuan spesifikasi rancangan perangkat lunak. Kebutuhan yang jelas dan terstruktur dengan baik merupakan hal yang sangat penting sebab mereka menyediakan dasar untuk semua pengembangan pekerjaan selanjutnya.

Segera sesudah kebutuhan ditetapkan, pengembang dapat memulai pekerjaan teknis lainnya, seperti perancangan sistem, pengembang, pengujian, implementasi dan kegiatan operasional sistem dapat berjalan dengan lancar.

Pemenuhan persyaratan kebutuhan sistem seorang pengguna seringkali dipengaruhi oleh kemampuan, pemahaman dan komunikasi yang berbeda dengan cara berpikir dari personil pengembang sistem. Kondisi ini merupakan persoalan yang sering terjadi dalam pengembangan perangkat lunak selama ini, bahkan kegagalan utama dari sejumlah perangkat lunak saat ini adalah tidak selarasnya kebutuhan sistem dengan hasil nyata yang diharapkan. Kesalahpahaman mengenai persyaratan kebutuhan sistem hanya akan berujung pada usaha yang sangat sia-sia dan mengharuskan pengerjaan ulang kembali. Selain itu adanya faktor ketidakkonsistenan, ketidaklengkapan, maupun ketidakbenaran dari spesifikasi kebutuhan sistem. Melakukan identifikasi kebutuhan sistem bersifat nyata memerlukan suatu rangkaian proses kebutuhan interaktif dan iteratif yang didukung oleh praktek secara efektif, proses, mekanisme, metode, teknik dan peralatan. Sejumlah studi yang dirilis oleh The Standish Group (2001) mencatat bahwa persentase kumulatif kegagalan sebuah proses pengembangan perangkat lunak disebabkan oleh kegagalan mengantisipasi spesifikasi kebutuhan. Sementara hasil studi dari KPMG terhadap 300 perusahaan besar memperlihatkan bahwa kurang lebih 65% pengembangan perangkat lunak mengalami kegagalan sebagai akibat dari anggaran yang dikeluarkan jauh melampaui dari yang direncanakan atau ditargetkan, durasi pengerjaan proyek jauh menyimpang dari waktu yang telah ditetapkan, dan investasi teknologi informasi yang dibutuhkan tidak berhasil meningkatkan kinerja operasional perusahaan secara berarti.

## **2. TINJAUAN PUSTAKA**

Persyaratan kebutuhan sistem perangkat lunak menentukan apa yang harus dilakukan sistem dan mendefinisikan batasan-batasan operasi dan implementasinya agar dapat mengomunikasikan secara tepat semua fungsi yang diberikan (Sommerville, 2011). Melakukan analisis persyaratan adalah tugas rekayasa perangkat lunak yang menjembatani kesenjangan antara alokasi perangkat lunak tingkat sistem dan perancangan perangkat lunaknya (Pressman, 2010). Analisis kebutuhan perangkat lunak adalah proses mendapatkan informasi, model, spesifikasi sistem yang diinginkan pengguna (Simarmata, 2010). Adalah sebuah kenyataan bahwa sebagai fase awal untuk melakukan pengembangan rekayasa perangkat lunak merupakan suatu kondisi yang memiliki peran sangat penting untuk kelanjutan dari fase-fase berikutnya.

Kelalaian atau ketidakseriusan fase ini akan mengakibatkan sistem tidak dapat bekerja sebagaimana mestinya dan justru dapat menjadi jebakan untuk pengembangan perangkat lunak. Belum lagi adanya kemenduaan arti dari pengguna sistem dapat meniadakan komitmen awal dari pengembangan sistem tersebut. Perangkat lunak yang sesuai dengan kebutuhan sangat bergantung kepada keberhasilan analisis spesifikasi persyaratan kebutuhannya. Kemampuan dalam menuliskan kode perangkat lunak dan memiliki media antarmuka yang menawan bukan merupakan jaminan perangkat lunak tersebut berguna. Pernyataan ini memperlihatkan bahwa analisis persyaratan kebutuhan sistem merupakan fase yang sangat signifikan. Dalam melakukan analisis kebutuhan

memiliki berbagai faktor pembatas seperti pengguna tidak terlalu mengerti dan memahami pengetahuan mengenai perangkat komputer, teknik pemrograman dan entitas teknologi informasi. Pemerolehan kebutuhan yang jelas dan benar sesuai dengan apa yang dimaksud oleh pengguna menunjukkan langkah awal yang baik, yang akan membantu pembuatan perangkat lunak pada tahap berikutnya. Persyaratan kebutuhan sistem dapat diklasifikasikan sebagai persyaratan fungsional dan non-fungsional atau sebagai persyaratan domain yang mewakili dari sistem ini sendiri (Sommerville, 2011).

Persyaratan fungsional merupakan pernyataan layanan yang harus diberikan kepada sistem agar dapat melakukan keperilakuannya dalam bereaksi terhadap masukan tertentu dan pada situasi tertentu. Persyaratan ini harus dapat mengilustrasikan secara terperinci pada masing-masing tingkatan sistem. Kemenduaan arti terjadi karena ketidaktepatan memformulasikan dan menerjemahkan spesifikasi persyaratan kebutuhan sistem dan adanya indikasi untuk memudahkan dalam penerapannya. Dalam beberapa kasus menghasilkan perangkat lunak juga memerlukan suatu pembatasan yang jelas dan terstruktur mengenai perilaku sebuah sistem. Spesifikasi persyaratan fungsional harus lengkap dan konsisten. Namun ada kalanya untuk sistem yang skalanya besar, kedua aspek ini sulit untuk dilaksanakan karena kompleksitas sistem dan ketidaksesuaian dari sudut pandang masing-masing. Untuk persyaratan non-fungsional lebih mengarah kepada batasan layanan atau fungsi yang diberikan sistem. Dokumen pernyataan non-fungsional ini mencakup batasan waktu, proses pengembangan dan standarisasi keluaran sebuah sistem. Umumnya persyaratan ini berasal dari kebutuhan pengguna sistem seperti persyaratan produk, organisasi, dan eksternal. Kenyataan yang sering dihadapi adalah persyaratan kebutuhan sistem sulit untuk diverifikasi sehingga membutuhkan pernyataan yang bersifat kuantitatif dan dapat dilakukan pengujian secara objektif. Sementara persyaratan domain berhubungan langsung dengan proses komputasinya dan jika tidak terpenuhi dapat menyebabkan sistem bekerja dengan tidak memuaskan. Namun kenyataannya semua klasifikasi ini adalah saling menyatu satu dengan lainnya dan pembedaannya justru lebih kearah artifisial saja.

Memahami persyaratan kebutuhan sistem juga tidak akan menghasilkan sebuah perangkat lunak yang ideal seandainya tidak diikuti dengan pernyataan dari persyaratan kebutuhan pengguna. Dalam menetapkan persyaratan pengguna, pengembang sistem harus memiliki indikator pengukuran yang jelas untuk menghindari cakupan informasi yang terlalu banyak namun tidak signifikan dan bahkan bisa menimbulkan kemenduaan arti dan kesalahan dalam pemahaman persepsi terkait dengan berbagai istilah teknis bidang komputer. Untuk meminimalkan kesalahpahaman dalam membuat pernyataan persyaratan, pengguna dapat menggunakan sebuah form standar dan bahasa yang konsisten terutama antara persyaratan yang diperintahkan dan yang diinginkan. Standarisasi format berguna untuk meminimalisasi kemungkinan terjadinya kelalaian dan juga untuk membuat persyaratan lebih mudah diperiksa serta kelancaran dalam proses verifikasinya (Sommerville, 2011).

Persyaratan sistem harus menyatakan apa yang harus dilakukan sistem dan bukan bagaimana sistem tersebut harus diterapkan. Caranya mulai menggunakan bahasa natural terstruktur, bahasa deskripsi desain, notasi grafis, dan spesifikasi matematis.

Namun pengaruh dari aspek variabilitas cenderung masih menimbulkan kemenduaan arti dalam sebuah persyaratan kebutuhan sistem. Adapun upayanya adalah menggunakan bahasa deskripsi program. Namun cara ini terlalu teknis bagi pengguna sistem dan cenderung lebih mengarah kepada perancangan spesifikasi sistem daripada membantu pengguna memahami fungsionalitas sebuah sistem. Persyaratan kebutuhan sistem harus memiliki perangkat dokumentasi yang lengkap. Dalam dokumen tersebut harus berisi entitas pengguna sistem dengan tujuan menspesifikasi persyaratan dan untuk memeriksa apakah sudah memenuhi kebutuhan, entitas manajer dengan tujuan untuk merencanakan penawaran atas sistem dan merencanakan proses pengembangan sistem, entitas perekayasa sistem dengan tujuan untuk memahami sistem apa yang akan dikembangkan, entitas perekayasa pengujian sistem dengan tujuan mengembangkan pengujian validasi bagi sistem, dan entitas perekayasa pemeliharaan sistem dengan tujuan untuk membantu memahami sistem dan hubungan antara bagian-bagiannya (Sommerville, 2011).

Analisis persyaratan perangkat lunak dapat dibagi menjadi lima area kerja, yaitu pengenalan masalah, evaluasi dan sintesis, pemodelan, spesifikasi dan kajian spesifikasi (Pressman, 2010). Analisis persyaratan perangkat lunak selalu dimulai dengan komunikasi antara dua bagian atau lebih melalui media wawancara. Gause dan Weinberg (1989), menyarankan analisis memulainya dengan mengajukan pertanyaan-pertanyaan bebas konteks agar lebih mudah mengarahkan kepada pemahaman yang mendasar (Pressman, 2010). Namun kegiatan ini tidak selamanya berjalan dengan lancar, karena pengguna dan pengembang sistem perangkat lunak tanpa mereka sadari, sering memiliki konteks pemikiran antara “kita dan mereka”. Lebih lanjut Zahniser (1990) menyarankan penggunaan pendekatan FAST (*Facilitated Application Specification Techniques*). Pendekatan ini mendorong munculnya tim gabungan antara pengguna dan pengembang sistem yang bekerja bersama untuk mengidentifikasi masalah, mengusulkan elemen pemecahan, menegosiasikan pendekatan yang berbeda dan menghususkan rangkaian persyaratan pemecahan awal (Pressman, 2010).

Romi Satrio Wahono dalam sebuah artikel di [ilmukomputer.com](http://ilmukomputer.com) memaparkan bahwa sebuah definisi yang cukup jelas dan diterima secara umum adalah yang diuraikan oleh Pamela Zave (1997). *Requirements engineering* adalah cabang dari *software engineering* yang mengurus 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. Hasil dari fase *requirements engineering* terdokumentasi dalam bentuk *requirements specification*. Kegiatan ini berisi kesepakatan bersama tentang permasalahan yang ingin dipecahkan antara pengembang dan pengguna, dan merupakan titik mulai menuju proses berikutnya yaitu *software design*. Sistemisasi proses negosiasi pengembang dan pengguna dalam *requirements engineering* dibagi dalam 3 proses besar yaitu: *elicitation, specification, validation and verification*.

### 3. METODOLOGI PENELITIAN

Penelitian ini menggunakan bentuk survei dengan metode deskriptif. Metode pengumpulan datanya terdiri dari data primer dan sekunder. Data primer berasal dari

hasil wawancara dan melakukan pengamatan secara langsung. Tipe wawancara yang digunakan adalah wawancara secara langsung dengan sejumlah responden, dalam hal ini respondennya adalah mereka yang bekerja sebagai programmer lepas dan beberapa karyawan yang bekerja di bagian teknologi informasi untuk beberapa perusahaan swasta/BUMN yang ada di kota Pontianak. Kriteria penarikan sampel adalah mereka yang sudah bekerja dibidangnya lebih dari 5 tahun. Sementara untuk penarikan sampelnya menggunakan teknik *purposive sampling*. Untuk kebutuhan data sekunder diperoleh dari sejumlah dokumen dan laporan mengenai aktivitas melakukan proses rekayasa perangkat lunak. Penelitian ini menggunakan variabel tunggal yaitu studi analisis persyaratan kebutuhan sistem dalam menghasilkan perangkat lunak yang berkualitas. Metode analisisnya menggunakan kombinasi pendekatan rekayasa kebutuhan (*requirement engineering approach*) dan pendekatan FAST (*Facilitated Application Specification Techniques*).

#### **4. KONTRIBUSI PENELITIAN**

Menentukan pendekatan rekayasa persyaratan sangat mempengaruhi keandalan, reliabilitas dan usabilitas dari sebuah perangkat lunak. Namun demikian juga harus mempertimbangkan kondisi dan faktor-faktor lain yang cenderung sering tidak dapat diungkapkan secara empiris saat komunikasi berlangsung. Memberikan wacana dan pemikiran baru bahwa menggunakan setiap pendekatan yang sudah ada belum tentu dapat menjamin hasil analisis sesuai kebutuhan pengguna sistem. Kondisi ini membutuhkan dimensi pemikiran yang cerdas dalam membentuk kembali pola-pola saat melakukan analisis persyaratan kebutuhan sistem. Analisis ini penting untuk menghilangkan kesalahpahaman atau ketidakkonsistenan dalam perolehan data dan informasi yang dibutuhkan agar dalam proses pengembangannya dapat sesuai dengan rencana kerja yang telah ditetapkan. Penelitian ini mencoba membangun sebuah pemikiran yang lebih inovatif untuk mencapai keselarasan antara persepsi kebutuhan menurut pengembang dan pengguna sistem.

#### **5. HASIL PENELITIAN DAN PEMBAHASAN**

Berdasarkan hasil survei melalui wawancara dan pengamatan secara langsung terhadap sejumlah responden yang dipilih untuk maksud dan tujuan tertentu dan hasil kajian dari beberapa teori atau pendekatan mengenai konteks analisis persyaratan kebutuhan sistem, memperlihatkan bahwa persyaratan sebuah sistem dapat diidentifikasi dan ditetapkan. Beberapa pendekatan memiliki tujuan yang sama, tetapi langkah pengerjaan analisis persyaratan agak sedikit berbeda. Jika sebelumnya telah dilakukan pemahaman terhadap sistem tersebut secara spesifik dan memiliki fokus kebutuhan yang jelas sehingga mudah melakukan penelusuran permasalahan sistem, keinginan dan harapan pengguna sistem sekarang maupun di waktu yang akan datang. Pemahaman sistem sekarang penting untuk mempelajari dan memahami tempat masalah, kesempatan dan peluang untuk peningkatan sistem, lingkup dan batasan yang ada sistem sekarang, proses-proses yang berjalan di sistem, data yang mengalir dan disimpan, serta apa dan siapa yang berhubungan langsung dengan sistem.

Kecenderungan dari sejumlah pendekatan dalam melakukan proses rekayasa kebutuhan sistem lebih mengarah pada upaya untuk mencari solusi dari permasalahan

yang dihadapi. Kegiatan analisis persyaratan kebutuhan fokus kepada kebutuhan perangkat lunak yang akan dihasilkan tanpa mencoba untuk menelusuri apa sesungguhnya penyebab dan akibat dari akar permasalahan tersebut yang mempengaruhi jalannya sistem. Melalui analisis sebab dan akibat harus menjadi pedoman awal saat mencoba untuk memahami perilaku sebuah sistem. Kenyataannya hal ini memang tidak mudah karena perilaku sistem mencakup banyak entitas, mulai dari para stakeholder dengan ilustrasi kebutuhan sistem yang bervariasi baik internal maupun eksternal, pihak manajemen dari setiap divisi atau unit fungsi bisnis yang saling berinteraksi dengan sasaran yang berbeda-beda, kesiapan sumberdaya manusianya dari sisi pengetahuan dan pendidikan serta pengalaman, kesadaran dan kesiapan untuk memulai dan mau menerima sesuatu yang baru dengan resiko penerapannya. Sisi pengembang juga harus mempunyai pemahaman yang sama sehingga solusi yang mereka gambarkan sesuai dengan harapan semua orang. Kesalahpahaman mengenai kebutuhan hanya akan berujung pada usaha yang sangat sia-sia dan mengharuskan pengerjaan ulang tanpa memiliki nilai-nilai positif yang signifikan. Identifikasi kebutuhan nyata memerlukan suatu proses kebutuhan interaktif dan iteratif agar proses menjadi lebih efektif dan memiliki standarisasi yang jelas dan memiliki sistematika yang jelas.

Selama ini memang kita sadari bahwa banyak dijumpai analisis dan perancang sistem gagal dalam menghasilkan perangkat lunak yang benar-benar dapat menjawab permasalahan sistem pada perusahaan, walaupun sudah menggunakan pendekatan rekayasa kebutuhan dan pendekatan FAST. Memang kedua pendekatan ini sering digunakan untuk menggali sebanyak dan semendetil mungkin persyaratan kebutuhan sebuah sistem agar tidak lagi menjadi dilema yang cenderung membingungkan pengguna dan pengembang sistem. Namun untuk menspesifikasikan sebuah persyaratan kebutuhan sering terjadi konflik bahkan konflik kepentingan, karena masing-masing terperangkap kepada kecerdasan yang membawa pada asumsi ketidaksesuaian dari pengetahuan dan pemahaman sosioteknis dan dimensi bisnis perusahaan. Kondisi ini menimbulkan ketidakpercayaan dan kebingungan yang pada akhirnya ada keengganan untuk menggunakan perangkat lunak tersebut. Kenyataan ini menjadi semakin sulit untuk dihindari karena adanya ketidakkonsistenan dalam merepresentasikan persyaratan fungsional dan non-fungsional. Hal ini menyebabkan terjadinya kemenduaan arti yang pada penerapannya kurang mendapatkan perhatian awal. Jadi menggunakan pendekatan rekayasa kebutuhan dan pendekatan FAST tidak senantiasa dapat menjamin sebuah perangkat lunak memiliki validitas, reliabilitas, dan usability yang layak dan selaras dengan keinginan dan harapan pengguna dan pengembang sistem.

Untuk mencapai keselarasan antara pengguna dan pengembang sistem perlu melakukan kegiatan dekomposisi fase-fase dari kedua pendekatan tersebut dan melakukan suatu proses migrasi pendekatan sebagai upaya untuk mengurangi ketidaksesuaian pemahaman mengenai masing-masing perilaku sistem. Baik pendekatan rekayasa kebutuhan dan FAST pertama kali harus menerjemahkan sasaran dan tujuan sistem ke dalam kebutuhan dan persyaratan yang diperlukan. Dalam hal ini batasan sistem dari sebuah perangkat lunak harus mendapat perhatian utama. Hal-hal yang membatasi fleksibilitas kita dalam mendefinisikan sebuah solusi. Pada dasarnya batasan tidak dapat diubah, seperti tanggal, waktu, anggaran dan kebijakan sistem sehingga

membutuhkan perlakuan khusus dalam mengontrol dari setiap batasan yang ada seperti biaya, perencanaan, teknologi, dan kebijakan.

Faktor-faktor tersebut harus menjadi perhatian yang serius sebagai bentuk mengantisipasi kesenjangan dalam menemukan dan membangun sebuah kesepakatan dari fitur masing-masing pendekatan untuk menghasilkan sebuah perangkat lunak yang berkualitas. Melakukan proses diversifikasi sesuai dengan kebutuhan memang merupakan suatu kondisi yang tidak mudah karena banyak bermunculan berbagai anggapan, asumsi dan hipotesa yang keliru. Kenyataan proses migrasi ini akan dapat menghindari pembentukan pemikiran yang hanya menguntungkan beberapa pihak tanpa suatu keselarasan akan pemahaman dari setiap proses bisnis. Dekomposisi setiap fase dengan mengintegrasikan kedua pendekatan tersebut dipandang sebagai suatu upaya untuk mempermudah dan meningkatkan tingkat keluwesan dari setiap kepentingan antara stakeholder dan pengembang sistem. Hakekatnya sebuah sistem yang sudah melakukan proses asimilasi dari sejumlah kepentingan dapat memperkaya khasanah dari sebuah kenyataan saat melakukan penerapan sebuah perangkat lunak. Kualitas sebuah perangkat lunak tidak bisa mengabaikan dari sebuah struktur dan mekanisme sistem yang sudah ada sebelum melakukan pengembangan dan rekayasa perangkat lunak. Sejumlah persyaratan kebutuhan hanya bisa dan dapat diselesaikan jika masing-masing pihak yang terlibat mau melakukan knowledge sharing dengan tanpa batasan yang mau menang sendiri. Senantiasa melakukan proses brainstorming untuk menemukan berbagai celah yang di waktu yang akan datang dapat menjadi hambatan saat menerapkan perangkat lunak tersebut.

Indikator berikutnya adalah harapan pengguna sistem. Harapan dan keinginan pengguna diperoleh dari hasil wawancara. Hal yang harus diperhatikan adalah apa keinginan pengguna tersebut merupakan kebutuhan atau hanya sebagai keinginan yang tidak ada kaitannya dengan kebutuhan pengguna dari sistem. Selanjutnya proses migrasi mencakup fase-fase berikut ini, yaitu mengidentifikasi stakeholder, mendapatkan pemahaman kebutuhan dan harapan dari pelanggan dan pengguna atas suatu sistem terencana atau mengungkapkan kebutuhan, dan mengidentifikasi kebutuhan tersebut. Kemudian memetakan tingkat dukungan kebutuhan bisnis dan fasilitas suatu organisasi untuk mencapainya. Memperjelas dan menyatakan kembali kebutuhan, menganalisis kebutuhan, mendefinisikan kebutuhan yang mempunyai arti yang sama bagi semua stakeholder, menetapkan dan memprioritaskan kebutuhan, dan memperoleh kebutuhan, melacak kebutuhan melalui pengelolaan kebutuhan, menguji dan memverifikasi kebutuhan, dan memvalidasi kebutuhan. Kriteria kebutuhan harus dapat dilaksanakan, benar, singkat, jelas dan tidak bersifat ambiguitas, lengkap, konsisten, dapat dibuktikan, dapat ditelusuri, mudah dialokasikan, tidak ada kerangkapan, menggunakan standarisasi tertentu. Rekayasa kebutuhan harus menjadi penengah antara domain pengguna sistem dan teknis rekayasa perangkat lunak harus menguasai keahlian teknis, kemampuan untuk mendapatkan pemahaman domain aplikasi, dan keahlian interpersonal untuk membangun konsensus antara kelompok stakeholder yang bersifat heterogen.

Setelah itu, melakukan eksplorasi kebutuhan dari berbagai pandangan masing-masing stakeholder. Kebutuhan yang sama dan berbeda dari masing-masing stakeholder

akan dicatat dan perbedaan persepsi diluruskan. Mengungkapkan kebutuhan pengguna merupakan proses yang tidak mudah dilakukan karena batasan sistem sering tidak jelas. Pengguna tidak cukup paham mengenai apa yang dibutuhkan dan kebutuhan sering berubah. Aktivitas dilakukan dalam bentuk pertemuan dan setiap stakeholder yang diundang diminta untuk membuat daftar kebutuhan perangkat lunak yang disertai penjelasan mengenai karakteristik atau detail dari kemampuan tersebut. Validasi akan menguji kualitas dari spesifikasi untuk memastikan bahwa kebutuhan yang dinyatakan dapat diterima/sepaham, konsisten, lengkap dan bebas dari kesalahan. Tiga faktor yang harus dipenuhi ketika melakukan analisis kebutuhan, yaitu bersifat menyeluruh, terperinci dan memiliki tingkat keakuratan yang tinggi. Dalam melakukan analisis kebutuhan yang sesungguhnya di lapangan, akan sering menemukan hal-hal yang kurang diharapkan. Misalnya, pengguna sistem yang membutuhkan perangkat lunak namun tidak memiliki pengetahuan yang cukup tentang segala hal yang berkaitan dengan rekayasa perangkat lunak, pemrograman dan teknologi informasi. Keadaan ini tentu cukup merepotkan karena akan memerlukan banyak usaha untuk menggali keinginan mereka yang sesungguhnya dan menghindari kondisi terperangkap kedalam domain kecerdasan masing-masing. Masalah ini paling sering dihadapi dan merupakan aspek validitas yang sangat penting untuk mengungkapkan kebutuhan pengguna sistem.

Salah satu hal yang perlu diperhatikan adalah membangun hubungan yang baik dengan pengguna sistem. Kemampuan seorang analis dalam menciptakan hubungan sosial dengan pihak lain, yang dalam situasi ini adalah dengan pengguna, menjadi bantuan yang signifikan. Hubungan sosial yang baik dengan pengguna akan menjadikan komunikasi yang terbuka dan lancar. Mengenal lebih dekat dengan pengguna sistem berarti menciptakan komunikasi lebih lancar dan tidak terlalu formal sehingga kemungkinan kesalahan dalam berkomunikasi dapat dikurangi. Yang perlu diketahui adalah tentang apa yang dikerjakannya, data apa yang menjadi masukan, dan yang dihasilkan. Jangan terburu-buru menanyakan bagaimana cara mengerjakannya. Bisa mengetahui apa yang sebenarnya dihasilkan dan apa yang menjadi masukan merupakan suatu langkah yang baik untuk memperoleh gambaran yang lebih spesifik tentang sebuah perangkat lunak.

Dalam berkomunikasi dengan pengguna sistem, terutama ketika dalam proses analisis kebutuhan harus menggunakan bahasa dan istilah yang mudah untuk dipahami. Menggunakan istilah dalam bidang teknologi informasi dan komputer dapat membuat orang kagum dan salut dengan anda, namun hal ini dapat menimbulkan masalah serius. Komunikasi bisa menjadi tidak lancar dan berpeluang menimbulkan kesalahpahaman dari persepsi masing-masing. Senantiasa harus memiliki sikap yang terbuka dan dapat menerima pendapat dan masukan pengguna sistem. Harus menggunakan contoh nyata yang bisa dibuktikan dan mudah untuk dipahami sekalipun orang awam yang tidak mengerti dalam bidang teknologi informasi dan komputer. Bisa jadi, contoh nyata ini dibuat oleh analis atau dari apa yang sudah dimiliki oleh pengguna, misalnya untuk proses memasukkan data dari sebuah transaksi perusahaan. Pengguna sistem dan analis sama-sama menyaksikan proses memasukkan data tersebut. Jika hanya disampaikan melalui deskripsi saja, kemungkinan interpretasi yang salah cukup terbuka lebar dan dapat mengaburkan persyaratan kebutuhan dari yang seharusnya. Sementara melalui



aktivitas nyata kinerja dari sebuah sistem dapat memperkuat dan memberi keyakinan positif perangkat lunak yang berkualitas.

Kenyataan memperlihatkan bahwa faktor kritis sukses untuk berhasilnya analisis persyaratan kebutuhan antara lain persyaratan tidak boleh ambigu, harus dapat mendeskripsikan semua input dan merespon sistem yang mungkin terjadi, dapat dipenuhi berdasarkan sumberdaya dan batasan yang ada, memenuhi kebutuhan sistem, akurat, memiliki fungsi dan fitur-fitur sistem, dan kesesuaian dalam pengujian. Memiliki performa sistem dari perangkat lunak yang dibutuhkan, kesesuaian dan kehandalan distribusi informasi, dan analisis kebutuhan ini menjadikan sistem menjadi lebih fleksibel, reliabel dan dapat diperluas. Supaya proses rekayasa kebutuhan ini menjadi lebih adaptif dengan perubahan sistem memerlukan penetapan prioritas persyaratan, baik yang bersifat tetap atau harus dipenuhi oleh sistem atau hanya berupa persyaratan yang menjadi keinginan saja. Ketika salah satu faktor tersebut tidak dapat dipenuhi oleh sebuah persyaratan yang diusulkan, maka kita harus melakukan kesepakatan dengan pengguna sistem dengan mengusulkan kembali beberapa alternatif persyaratan lainnya. Semuanya harus memiliki dokumen persyaratan yang lengkap agar memudahkan dalam pengembangan perangkat lunak selanjutnya. Dimana dalam dokumen tersebut memiliki kejelasan persyaratan fungsional dan non-fungsional sehingga dapat menghilangkan sikap keragu-raguan, ketidakpastian, kekurangyakinan, dan kegagalan penerapan perangkat lunak.

## **6. KESIMPULAN**

Hasil penelitian ini memberikan kontribusi nyata, dimana analisis persyaratan kebutuhan sistem merupakan fase awal proses rekayasa perangkat lunak yang perlu mendapatkan perhatian serius, karena sebaik apapun pengkodean dan perancangan sistem tetap tidak akan dapat memberikan keinginan yang sesungguhnya dengan kondisi yang dihadapi. Hal ini semakin penting karena banyak sekali ditemukan kegagalan demi kegagalan saat menggunakan perangkat lunak tersebut. Pemahaman yang konsisten dapat mencerminkan fleksibilitas, reliabilitas, keakuratan dan usability yang handal. Penelitian ini memberikan sejumlah kaidah dan pertimbangan mengenai persyaratan kebutuhan melalui proses migrasi sistem dari kedua pendekatan tersebut agar analisis persyaratan kebutuhan dapat sesuai dengan kondisi nyata. Memiliki kemampuan komunikasi dan interaksi yang baik dan lancar, istilah yang mudah dipahami, bersifat terbuka dan fleksibel menerima pendapat dan masukan pengguna sistem, menggunakan contoh nyata mendeskripsikan apa yang dibutuhkan dan menghindari perangkat domain kecerdasan dari para pelaku sistem merupakan faktor yang sangat penting dalam menyelaraskan batasan kebutuhan perangkat lunak dengan pengguna sistem.

## DAFTAR PUSTAKA

- Aybuke Aurum and Claes Wohlin., 2005, Engineering and Managing Software Requirements, *Springer-Verlag* Boili Heidelberg.
- Pressman, Roger S., 2010, *Software Engineering: A Practitioner's Approach*, Seventh Edition, McGraw-Hill, Series in Computer Science.
- Sommerville, Ian., 2011, *Software Engineering*, Ninth Edition, Addison-Wesley.
- Simarmata, Janner., 2010, *Rekayasa Perangkat Lunak*, Penerbit Andi Offset Yogyakarta.
- Wahono, Romi Satria., 2003, Analyzing Requirements Engineering Problems, *IECI Japan Workshop 2003 (IJW-2003)*, Japan.
- Zave, Pamela., 1997, Classification of Research Efforts in Requirements Engineering, *ACM Computing Surveys*, 29(4), pp. 315-321.