

RENTI SUSANTI
13.3.0012

BAB 1

PENGENALAN REKAYASA KEBUTUHAN

Mengapa Perlu Rekayasa Kebutuhan

- A. Semua perangkat lunak memiliki spesifikasi
- B. Permasalahan berawal dari spesifikasi kebutuhan

Siapa yang berkepentingan terhadap Sistem

1. Pelanggan (*Customer*)
2. Pemilik Sistem (*System Owner*)
3. Pengguna (*User*)
4. Analisis kebutuhan (*Requirements Analyst*)
5. Pengembang (*Developer*)

6. Penguji (*Tester*)
7. Penulis Dokumentasi (*Documentation's Writer*)
8. Manajer Proyek (*Project Manager*)
9. Staf hukum dan non hukum
10. Staf manufaktur
11. Penjualan, pemasaran, dan bagian pendukung
12. Penyelia (*vendor*)
13. Regulator yang menetapkan batasan berupa baku muku, peraturan, panduan, atau rambu-rambu lain terkait produk, proses maupun personel

Definisi Rekayasa Kebutuhan

Sommerville(2007) mengartikan rekayasa kebutuhan (*Software Engineering*) sebagai suatu proses mewujudkan serangkaian layanan yang dibutuhkan oleh pelanggan atas suatu sistem dan batasan-batasan yang harus dipenuhi ketika diangun maupun dioperasikan.

Bray(2002) menyatakan bahwa rekayasa kebutuhan merupakan aktivitas menginvestigasi dan mendiskripsikan ranah permasalahan dan kebutuhan-kebutuhan, serta merancang dan mendokumentasikan karakteristik dari suatu sistem solusi yang nantinya diharapkan memenuhi kebutuhan-kebutuhan tersebut.

Steve Eastbook, seorang pengajar di departemen ilmu computer Universitas Toronto, mendefinisikan rekayasa perangkat lunak sebagai serangkaian aktivitas yang berkaitan dengan mengidentifikasi dan mengomunikasikan tujuan dari sistem-intensif-perangkat lunak, dan konteks dimana sistem itu digunakan.

Dari definisi di atas dapat kita simpulkan bahwa rekayasa kebutuhan meliputi aktivitas-aktivitas menyelidiki, mencari, atau mengidentifikasi spesifikasi kebutuhan sistem , serta mengomunikasikannya kepada pelanggan maupun pengembang, baik secara lisan maupun tulisan.

BAB 2

PERSPEKTIF PEMANGKU KEPENTINGAN

Klasifikasi pemangku kepentingan :

1. Pelanggan
2. Pemilik modal
3. Pemilik sistem
4. Pengguna(user)
5. Regulator, seorang atau suatu organisasi yang menetapkan aturan dan batasan
6. Penyelia
7. Pengembang
8. Analis sistem
9. Programmer

Kelompok Kebutuhan

1. Kebutuhan Bisnis
2. Kebutuhan Pengguna
3. Aturan Bisnis
4. Atribut Kualitas
5. Kebutuhan Sistem
6. Kebutuhan Fungsional
7. Antarmuka eksternal
8. Batasan

BAB 3

SKENARIO

Skenario adalah suatu cerita atau narasi yang mudah diakses untuk membuat aplikasi lebih hidup. Komponen dalam Skenario :

1. Tujuan
2. Ruang Lingkup
3. Sudut Pandang Pemangku Kepentingan
4. Visualisasi
5. Singkat , ukuran A4
6. Rekursif, dekomposisi, dan penyempurnaan

BAB 4

ELISITASI KEBUTUHAN

Elisitasi kebutuhan adalah sekumpulan aktivitas yang ditunjukkan untuk menemukan kebutuhan suatu sistem melalui komunikasi dengan pelanggan, pengguna sistem dan pihak lain yang memiliki kepentingan dalam pengembangan sistem (Sommerville and Sawyer 1997). Sejalan dengan proses rekayasa kebutuhan secara keseluruhan, elisitasi kebutuhan bertujuan untuk (Leffingwel, 2000) :

1. Mengetahui masalah apa saja yang perlu dipecahkan dan mengenali batasan-batasan sistem.
2. Mengenali siapa saja para pemangku kepentingan.
3. Mengenali tujuan dari sistem yaitu sasaran-sasaran yang harus sistem.

Tahap elisitasi termasuk tahap yang sulit dalam spesifikasi perangkat lunak. Secara umum kesulitan ini disebabkan tiga masalah, yakni : masalah cakupan, masalah pemahaman, dan masalah perubahan (Nuiseibeh and Eastbrook, 2000). Ketiga masalah tersebut muncul karena (Sommerville, 2007) :

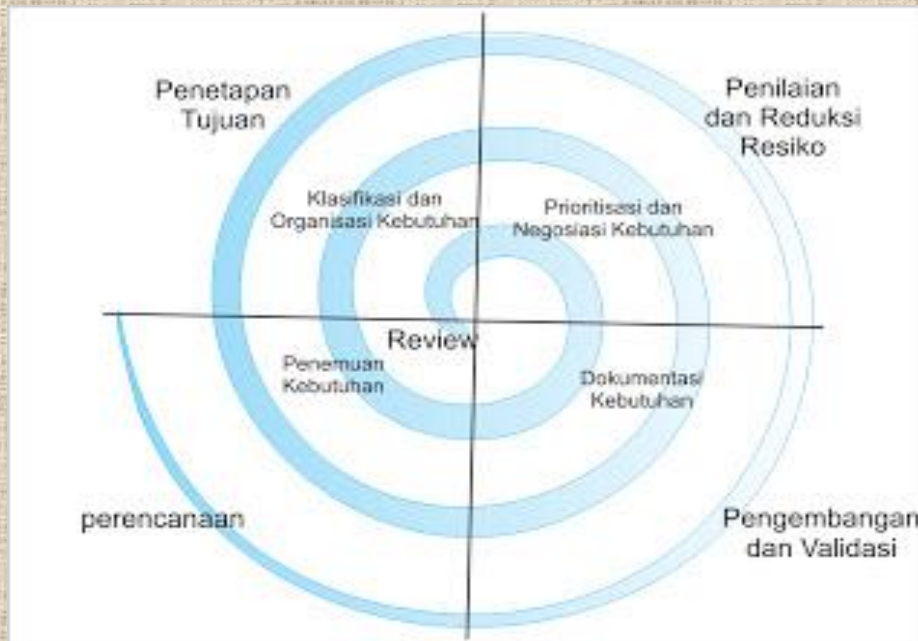
1. Pemangku kepentingan sering tidak mengetahui apa yang diinginkan dan mengungkapkan keinginannya dalam kalimat yang umum.
2. Pemangku kepentingan mengungkapkan permintaan dalam istilah bidang pekerjaannya, sehingga *perekayasa kebutuhan* yang tidak memiliki pengalaman di bidang kerja pemesan harus memahami permintaan tersebut.
3. Beberapa pemangku kepentingan memiliki permintaan yang berbeda-beda yang dinyatakan dalam cara yang berbeda pula.
4. Faktor politik dapat mempengaruhi kebutuhan sistem.
5. Lingkungan bisnis dan ekonomi yang bersifat dinamis.

Seorang analisi kebutuhan harus dibekali landasan teori ilmu social dan teknik praktik elisitasi kebutuhan yang baik. Ilmu social tersebut antara lain (Nuseibeh and Eastbrook, 2000):

1. *Cognitive Psychology*, yang menekankan tentang kesulitan seseorang dalam mendeskripsikan kebutuhannya.
2. *Antropologi* memberikan pendekatan metodologis untuk mengamati kegiatan manusia yang membantu pemahaman lebih mendalam tentang bagaimana sistem komputer membantu atau mengganggu kegiatan.
3. *Sosiologi* memberikan pemahaman tentang perubahan politik dan budaya disebabkan oleh kompetetisasi.
4. Ilmu bahasa sangat penting karena elisitasi kebutuhan berfokus pada komunikasi.

Model Elisitasi Kebutuhan

Gambar dibawah ini adalah ilustrasi dari model proses elisitasi dan analis secara umum. Aktivitas-aktivitas yang digambarkan dalam model tersebut diilustrasikan sebagai suatu spiral dimana proses berjalan dari cincin terdalam menuju cincin terluar spiral.



Gambar. Elisitasi kebutuhan dan proses analisis

1. Penemuan Kebutuhan

Ini adalah proses interaksi dengan para pemangku kepentingan sistem untuk mengumpulkan kebutuhan mereka. Ranah kebutuhan dari para pemangku kepentingan dan dokumentasi juga didapatkan selama aktivitas ini.

2. Pengelompokan dan pengorganisasian kebutuhan

Aktivitas ini mengoleksi kebutuhan yang belum terstrukturkan, mengelompokkan kebutuhan yang saling terkait, dan kemudian mengorganisasikannya ke dalam kelompok yang koheren .

3. Prioritas dan negosiasi kebutuhan

Dalam tahapan ini, aktivitas manajemen yang dilakukan adalah analisis risiko dari masing-masing kebutuhan, yang meliputi penilaian risiko serta identifikasi control yang dapat diterapkan untuk mereduksi risiko dari setiap kebutuhan.

4. Dokumentasi kebutuhan

Dalam tahapan ini, aktivitas manajemen yang dilakukan adalah validasi dan pengembangan sistem.

Win-Win Spiral Model

Model ini diinisialisasi dengan mengidentifikasi pemangku kepentingan serta menetapkan kondisi-kondisi menang dari masing-masing pemangku kepentingan tersebut. Kemudian perekayasa kebutuhan mendeteksi setiap kemungkinan munculnya konflik dari kondisi-kondisi tersebut dan mengarahkan para pemangku kepentingan yang terlibat dalam konflik tersebut untuk menemukan resolusinya, baik melalui negosiasi dan kompromi. Dari hasil proses tersebut muncullah spesifikasi sistem, yang merupakan kondisi-kondisi menang yang tidak konflik serta hasil kompromi tadi. Pada akhirnya, setelah para pemangku kepentingan menyetujui atau menyepakati spesifikasi yang telah dibangun bersama, maka proses ini diakhiri. Dari situlah kemudian iterasi baru dimulai kembali.

I*Frame

Kerangka ini memiliki dua komponen utama, yaitu model kebergantungan strategis dan model rasional strategis. Model kebergantungan strategis merepresentasikan sejumlah kebergantungan antar actor-aktor di dalam suatu konteks organisasi. Sedangkan model rasional strategis merepresentasikan kebutuhan-kebutuhan serta perhatian dari para pemangku kepentingan.

Metode	Pro	Kontra
Win-Win Spiral	Resolusi konflik secara aktif Sangat person oriented Tidak ada pembatasan pada metode yang digunakan	Perkakas hanya sebagai basis pengetahuan Hanya untuk metode yang ringan
I*Frame	Penekanan pada integrasi organisasional Hanya dapat diterapkan pada awal sekali	Pemangku kepentingan sebagai actor yang direncanakan dalam proses rekayasa kebutuhan

Langkah-Langkah Elisitasi

Berikut ini merupakan langkah-langkah untuk elisitasi kebutuhan (Sommerville and Sawyer, 1997) :

1. Identifikasi orang-orang yang akan membantu menentukan kebutuhan dan memahami organisasi mereka.

2. Menentukan lingkungan teknis kemana sistem atau produk akan ditempatkan.
3. Identifikasi ranah permasalahan
4. Menentukan satu atau lebih metode elisitasi kebutuhan
5. Meminta partisipasi dari banyak orang sehingga mereduksi dampak dari kebutuhan yang bias yang teridentifikasi dari sudut pandang yang berbeda.
6. Mengidentifikasi kebutuhan yang ambigu dan menyelesaikannya
7. Membuat scenario penggunaan

Viewpoints

Viewpoints atau sudut pandang adalah pihak-pihak yang meminta atau menggunakan layanan yang diberikan atau disediakan oleh sistem. Ada 3 viewpoints yang umum :

1. Interactor viewpoints yaitu orang atau sistem yang lain yang berinteraksi secara langsung dengan sistem.
2. Indirect viewpoints yaitu pemangku kepentingan yang tidak menggunakan sistem tetapi mempengaruhi jalannya sistem.
3. Domain viewpoints yaitu karakteristik ranah dan batasan yang memengaruhi kebutuhan sistem.

TEKNIK ELISITASI

Ada beberapa jenis teknik elisitasi yang dapat di kombinasikan dalam proses penspesifikasian kebutuhan perangkat lunak :

1. Teknik – Teknik Tradisional, yaitu teknik pengumpulan kebutuhan yang meliputi kuesioner dan survey, wawancara, observasi, sampling, dan analisis dokumen yang ada.
2. Teknik – Teknik Elisitasi Berkelompok, yaitu Teknik yang bertujuan untuk mendorong kesepakatan pemangku kepentingan dan memanfaatkan dinamika tim elisitasi untuk menggali pemahaman kebutuhan yang lebih mendalam.
 - a. Brainstorming
 - b. Join Application Design (JAD)
 - c. Prototyping
3. Teknik – Teknik Model-driven, yaitu teknik yang memberikan model tertentu dari jenis informasi yang dikumpulkan dan menggunakan model ini untuk proses elisitasi.
 - a. Goal-based methods
 - b. Scenario-Based Methods

4. Teknik – Teknik kognitif, yaitu, teknik – teknik yang terdiri dari sekumpulan teknik yang awalnya dikembangkan untuk akuisisi pengetahuan untuk Knowledge Based System.
5. Teknik – teknik Kontekstual, adalah teknik – teknik yang muncul pada tahun 1990-an sebagai alternatif teknik dari teknik – teknik tradisional dan kognitif

Jenis kebutuhan dan Para Pembacanya

Dari aktivitas elisitasi kebutuhan dapat dikelompokkan kebutuhan ke dalam beberapa level kebutuhan yang didasarkan kepada siapa pembacanya, yaitu :

1. Kebutuhan pengguna . pernyataan tentang layanan yang disediakan sistem dan tentang batasan-batasan operasionalnya.
2. Kebutuhan sistem. Sekumpulan layanan, kemampuan, dan batasan-batasan sistem yang ditulis secara rinci.
3. Spesifikasi Rancangan Perangkat Lunak. Gambaran abstrak dari rancangan perangkat lunak yang menjadi dasar bagi perancangan dan implementasi yang lebih rinci.

Perkakas Bantu untuk Elisitasi

Athena Tool

Athena tool termasuk perkakas bantu berbasis Web yang dikembangkan dengan bahasa Java menggunakan Vraptr Framework. Dalam perkakas bantu ini, pengguna dibedakan menjadi lima yaitu moderator, editor, komentator, converter, dan administrator. Athena tool dapat digunakan untuk mengelola beberapa proyek pada saat yang bersamaan dan masing-masing proyek terdiri dari sekelompok orang yang memiliki peran berbeda.

FGD-Relicit

Focus Group Discussion technique in Requirements Elicitation (FGD-Relicit) merupakan suatu perkakas bantu yang mendukung elisitasi kebutuhan menggunakan teknik Focus Group Discussion. Fitur yang dimiliki perkakas. Fitur yang dimiliki perkakas bantu ini ada 12 buah yaitu :

- ◆ Autorisasi peserta yang mengikuti FGD
- ◆ Permulaan Sidang
- ◆ Focus Group Discussion
- ◆ Concern-Viewpoint Polling
- ◆ Concern-Viewpoint Integration
- ◆ Pencarian Concern-viewpoint.

BAB 5

ANALISIS KEBUTUHAN

Tujuan

1. Mengolah hasil elastisitasi kebutuhan
2. Mengembangkan persyaratan kualitas yang memadai dan rinci
3. Membangun pemahaman tentang karakteristik ranah permasalahan dan sekumpulan kebutuhan

Tahapan Analisis Kebutuhan

1. Domain Understanding
2. Requirements Collection
3. Classification
4. Conflict Resolution
5. Prioritisation
6. Requirements Checking

Prinsip-Prinsip Analisis

Menurut Pressman (2008) :

1. Ranah informasi dari suatu masalah harus direpresentasikan dan dipahami
2. Fungsi-fungsi yang akan dilakukan oleh perangkat lunak harus didefinisikan
3. Tingkah laku perangkat lunak harus terwakilkan
4. Model-model yang merepresentasikan informasi, fungsi, dan tingkah laku sistem harus dipecah-pecah ke dalam tingkat yang lebih rinci
5. Dimulai dari informasi dasar menuju implementasi rinci

Menurut Davis (1993) :

1. Memahami masalah sebelum anda mulai menciptakan model analisis
2. Mengembangkan prototype yang memungkinkan seorang pemakai memahami bagaimana interaksi manusia dengan mesin terjadi
3. Mencatat asal dan tujuan untuk setiap kebutuhan
4. Menggunakan pandangan kebutuhan berjenjang
5. Memprioritaskan kebutuhan
6. Berusaha mengurangi kerancuan

Sepuluh Perangkap Dalam Rekayasa Kebutuhan

1. Bingung tentang yang dimaksud “kebutuhan”
2. Kurangnya keterlibatan pelanggan
3. Kebutuhan yang kabur atau ambigu
4. Tidak ada prioritas
5. Membangun fungsionalitas yang tidak digunakan siapapun
6. Paralis analisis
7. Scope Creep
8. Proses kebutuhan yang tidak memadai
9. Analisis dampak perubahan yang tidak memadai
10. Kontrol versi yang tidak memadai

Praktek yang Baik dalam Analisis Kebutuhan

1. Menggambar Diagram Konteks
2. Membangun prototype
3. Menganalisis Kelayakan
4. Memberikan Prioritas
5. Memodelkan kebutuhan
6. Membuat kamus data
7. Mengalokasi kebutuhan-kebutuhan ke dalam sub-sub sistem
8. Mengaplikasikan Quality Function Deployment (QFD), QFD adalah teknik yang menghubungkan fitur produk dan atribut dengan suatu metric penilaian tertentu

BAB 6

SPESIFIKASI KEBUTUHAN

Gambaran Luas Tentang Spesifikasi

Dokumen Spesifikasi yang baik harus memiliki empat tujuan utama, yaitu :

1. Menyediakan umpan balik kepada konsumen
2. Memecah permasalahan dalam komponen yang lebih mudah dipecahkan
3. (Merupakan) masukan untuk tahap spesifikasi rancangan.
4. (bisa untuk) melakukan pengecekan validasi produk

Dalam membuat spesifikasi perangkat lunak terdapat empat kelompok metode pendekatan yang umum, yaitu : Bahasa alamiah (Natural Language), bahasa alamiah terstruktur (structured natural language), Bahasa semi – formal (semi-formal language), dan bahasa formal (Formal Language).

Apakah Spesifikasi kebutuhan?

Spesifikasi kebutuhan merupakan salah satu aktivitas yang dilakukan ketika merekayasa kebutuhan. Ada sejumlah standard yang dapat digunakan ketika mengembangkan dokumen spesifikasi kebutuhan, misal IEEE Standard 830 (IEEE, 1998), ISO 9116 (ISO, 2008), dan Software Standards PSS-05-0 (ESA, 1991).

Secara umum, Kategori - kategori yang biasanya diajukan antara lain :

- ◆ Kemudahan Pemeliharaan (Maintainability)
- ◆ Kemudahan Verifikasi (Verifiability),
- ◆ Kelengkapan (completeness),
- ◆ Kebenaran (correctness),
- ◆ Konsistensi (consistency),
- ◆ Kejelasan (clarity),
- ◆ Kemudahan Pelacakan (traceability),
- ◆ Kemudahan Perubahan (modifiability),
- ◆ Kemudahan Membaca (Readability), dan
- ◆ Kemudahan Penggunaan (Ease of use).

Metode Pembuatan Spesifikasi Perangkat Lunak

- ◆ Bahasa Alamiah, (Natural Language) adalah bahasa yang sehari – hari digunakan oleh manusia.
- ◆ Structured Natural Language, Adalah bahasa ilmiah dengan struktur dan batasan tertentu.
- ◆ Semi – Formal Language, adalah pendekatan yang menggunakan diagram untuk mendeskripsikan spesifikasi kebutuhan perangkat lunak (sommerville, 2007).
- ◆ Metode Formal

Spesifikasi Formal dalam Proses Perangkat Lunak

Ada dua pendekatan Mendasar untuk spesifikasi formal yang digunakan untuk menulis spesifikasi rinci untuk industry system perangkat lunak, yaitu :

1. Pendekatan Aljabar
Sistem di spesifikasikan dalam bentuk operasi – operasi dan hubungan – hubungannya
2. Spesifikasi berbasis Model
sistem dispesifikasikan dalam bentuk *state model* yang dibangun dengan menggunakan konstruksi matematika seperti himpunan dan urutan – urutan. Operasi didefinisikan dengan modifikasi terhadap *state* dari system.

Kerancuan dalam Spesifikasi Kebutuhan

Menurut Husain et al (2007), kerancuan dalam spesifikasi kebutuhan dapat terjadi dalam dua bentuk, yaitu :

1. Pengertian Permukaan (*Surface Understanding*)
2. Pengertian Konseptual (*conceptual Understanding*)

Templat SKPL

Ada beberapa baku dokumen spesifikasi kebutuhan perangkat Lunak (SKPL) yang sering digunakan untuk pembuatan dokumen Spesifikasi kebutuhan perangkat lunak, berikut diantaranya :

IEEE Std 830-1998

Hal – hal mendasar yang harus dimuat dalam SKPL adalah :

1. Fungsionalitas
2. Antarmuka Eksternal
3. Unjuk Kerja
4. Atribut
5. Batasan Rancangan pada implementasi

Karena SKPL memiliki peran spesifik dalam proses pembangunan perangkat lunak, maka SKPL harus dibuat berdasarkan batasan tertentu. Hal ini berarti sebuah SKPL :

1. Harus mendefinisikan semua kebutuhan perangkat lunak dengan benar
2. Tidak perlu mendeskripsikan rincian rancangan atau implementasi karena hal ini akan dideskripsikan pada tahapan rancangan
3. Tidak perlu menuliskan batasan tambahan perangkat lunak karena akan dispesifikasikan pada dokumen lain, misalnya rencana jaminan kualitas perangkat lunak.

KARAKTERISTIK SKPL YANG BAIK

1. Correct
2. Unambiguous
3. Complete
4. Consistent
5. Ranked for importance and/ or stability
6. Verifiable
7. Modifiable
8. Traceable

BAB 7

KEBUTUHAN YANG SMART

MASALAH DALAM REPRESENTASI KEBUTUHAN

Atribut kualitas dari sebuah spesifikasi kebutuhan tersebut adalah :

1. Mudah dikelola
2. Dapat diverifikasi
3. Lengkap
4. Benar/tepat
5. Konsisten
6. Jelas
7. Dapat dilacak
8. Dapat di Modifikasi
9. Mudah dibaca
10. Mudah digunakan

SPECIFIK, MEASURABLE, ATTAINABLE, REALIZABLE, and TIME-BOUNDED/TRACABLE (SMART)

SPESIFIK (SPECIFIK)

Karakteristik Spesifik ini meliputi sejumlah Area sebagai berikut :

1. Jelas
2. Konsisten
3. Sederhana
4. Sampai pada tingkat rinci yang diperlukan
5. Jika kebutuhan dinyatakan dengan sebuah prototype program, pastikan bahwa setiap prototype haruslah disertai dengan dokumennya.
6. Menggunakan glossary apabila istilah – istilahnya tidak umum
7. Hindari penggunaan “Akan didefinisikan kemudian”
8. Hendaklah menggunakan kata “rinci”, “informasi”, dan “data” dalam suatu kebutuhan.
9. Hendaklah menempatkan kebutuhan individual dalam suatu paragraph terpisah dan tandailah secara individual pula.

TERUKUR (MEASURABLE)

Kebutuhan – kebutuhan yang seperti itu masuk dalam dua kelompok besar berikut :

1. Kebutuhan – kebutuhan yang tidak dapat di instrumentasikan (*Instrumentation interferences*)
2. Kebutuhan – kebutuhan yang spesifik tetapi tidak tersedia alat pengukurnya.

DAPAT DICAPAI (ATTAINABLE)

Suatu kebutuhan dapat dicapai apabila secara fisik system tersebut mungkin mencapai kebutuhan – kebutuhan bersangkutan dalam kondisi – kondisi yang telah dicapai. Berikut adalah panduan yang dapat di ikuti :

- ◆ Apakah solusi teoritis tersebut di implementasikan? Jika belum pernah mengapa belum pernah?
- ◆ Apakah solusi teoritis telah ada untuk permasalahan tersebut?
- ◆ Apakah ada batasan atau halangan utama yang menyebabkan kebutuhan ini tidak dapat dicapai
- ◆ Apakah ada batasan fisik pada ukuran memory, processor, atau perangkat lainnya?
- ◆ Apakah ada batasan Lingkungan, seperti suhu, tekanan udara dan lain – lain?

DAPAT DIREALISASIKAN (REALIZABLE)

Dalam konteks rekayasa kebutuhan, suatu kebutuhan dianggap dapat direalisasikan apabila kebutuhan tersebut dapat dicapai atau dibuat berdasarkan pengetahuan kita tentang batasan yang diberlakukan dalam pengembangan system maupun proyeknya.

MEMILIKI DIMENSI WAKTU/DAPAT DILACAK (TIME BOUNDED/TRACEABLE)

Untuk membantu proses pelacakan yang baik, suatu dokumen spesifikasi kebutuhan hendaknya menyediakan informasi – informasi berikut :

1. Originator dari setiap kebutuhan (Institusi atau Pribadinya)
2. Asumsi yang mungkin digunakan sebagai dasar
3. Justifikasi kebutuhan
4. Hubungan antar kebutuhan, seperti subsumption, kebergantungan dan implikasi
5. Tingkat Kekritisannya

PRIORITAS

1. **Essensial**, Merupakan kebutuhan yang sifatnya harus direalisasikan oleh pengembang
2. **Desirable**, Merupakan kebutuhan yang sifatnya boleh direalisasikan oleh pengembang. Jika kebutuhan ini tidak direalisasikan, maka system tetap dapat menjalankan fungsionalitasnya secara utuh.

PIHAK YANG BERKEPENTINGAN

Dari Pihak pelanggan kita melihat bahwa baik pemilik system maupun pengguna memiliki kepentingan yang berbeda. Bagi seorang pengguna yang penting adalah apa yang ia butuhkan sudah disebutkan dengan tepat dan lengkap dalam dokumen spesifikasi kebutuhan. Dari pihak Pengembang, kita juga melihat bahwa tiap – tiap pemangku kepentingan memiliki kepentingan yang berbeda – beda pula. Manajer proyek memiliki kepentingan atas setiap aspek dari kebutuhan yang SMART.