

MODUL MATA KULIAH

REKAYASA PERANGKAT LUNAK 1

KP342 - 3 SKS



**FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR
JAKARTA**

TIM PENYUSUN

Noni Juliasari, M.Kom

Bima Cahya Putra, M.Kom

Basuki Hari Prasetyo, M.Kom

VERSI 1.0
N



PERTEMUAN 6

REKAYASA KEBUTUHAN

Capaian Pembelajaran	:	Mahasiswa mampu memahami kebutuhan perangkat lunak dan mampu mendokumentasikan
Sub Pokok Bahasan	:	<ul style="list-style-type: none">1.1. Pengantar rekayasa kebutuhan1.2. Definisi kebutuhan perangkat lunak1.3. Memahami jenis kebutuhan fungsional dan non fungsional1.4. Dokumentasi spesifikasi kebutuhan perangkat lunak
Daftar Pustaka	:	<ul style="list-style-type: none">1. Kung, David C., 2014. Object Oriented Software Engineering: An Agile Methodology, McGraw-Hill2. Pressman, Roger S. 2010. Software Engineering : A Practitioner's Approach, 7th, McGraw-Hill3. Schach, Stephen R. 2010. Object Oriented and Classical Software Engineering, 8th, McGraw-Hill4. Sommerville, Ian. 2010. Software Engineering, 9th, Pearson Education5. Software Engineering Body of Knowledge (SWEBOK). 20046. Romi Satria Wahono, "Menyegarkan Kembali Pemahaman tentang Requirement Engineering", http://romisatriawahono.net/2006/04/29/menyegarkan-kembali-pemahaman-tentang-requirement-engineering/7. Albert Endres, Dieter Rombach, "A Handbook of Software and Systems Engineering : Empirical

		<p>Observations, Laws and Theories”, Pearson Education Limited, England, 2003</p> <p>8. G. Kotonya and I. Sommerville, Requirements Engineering: Processes and Techniques, John Wiley & Sons, 2000.</p> <p>9. Sommerville I. & Sawyer P., 1997, "Requirements Engineering: a Good Practice. Guide", John Wiley.</p> <p>Davies A.M., 1993</p>
--	--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

1.1. Pengantar

Rekayasa Kebutuhan (Requirement Engineering) adalah bagian yang tak terpisahkan dari kegiatan rekayasa perangkat lunak. Rekayasa Kebutuhan mempunyai peran yang cukup penting, bahkan akan menentukan keberhasilan dari suatu proyek rekayasa perangkat lunak. Mengenai peran penting rekayasa kebutuhan tersebut telah banyak dikemukakan oleh para pakar.

Teori yang mendukung requirement engineering yang tercantum dalam SWEBOK edisi 2004 adalah sebagai berikut:

a. Hukum Glass (Robert Glass)

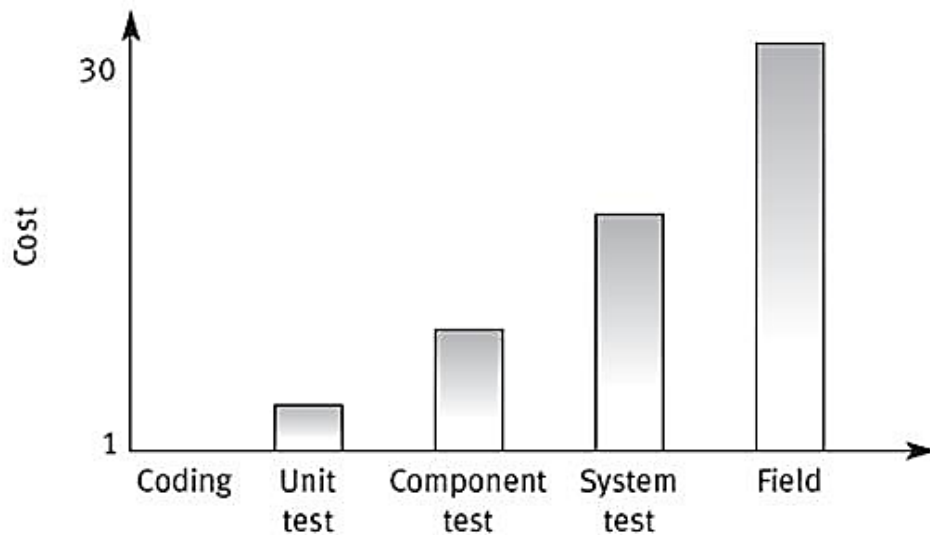
Kekurangan kebutuhan (requirement deficiencies) adalah sumber utama dari kegagalan proyek. Kekurangan kebutuhan menimbulkan masalah di banyak proyek. Kebutuhan yang ditentukan mungkin salah, atau tidak cukup perhatian yang diberikan pada definisi kebutuhan. Menetapkan tujuan dengan benar untuk setiap proyek adalah persyaratan tugas. Meskipun ada proyek dipahami dengan baik, ditentukan, dan kebutuhan stabil, lebih sering bukan ini masalahnya. Lebih khusus tidak lengkap atau kesalahan definisi kebutuhan, definisi terutama jika dilakukan oleh pihak ketiga untuk pelanggan dan pengembang.

Teori: Menentukan kebutuhan yang tepat merupakan masalah berat. Alasan utama untuk ini adalah kebutuhan yang berbeda dari berbagai kelompok pengguna, konflik kepentingan antara orang atau kelompok yang terlibat, dan kesulitan dari konflik antara prioritas kebutuhan. Definisi kebutuhan adalah proses belajar dan negosiasi. Kedua para pengembang dan pengguna belajar sambil menerapkan atau menggunakan sistem. Pengetahuan dari setiap orang yang terlibat sangat terbatas. Orang tidak tahu semuanya dan banyak lupa. Berbagi pengetahuan tidak terjadi dengan sendirinya. Masalah-masalah yang melekat dan tidak akan hilang sebagai kelangsungan teknologi.

b. Hukum Boehm pertama

Kesalahan yang paling sering selama menentukan kebutuhan (requirements) adalah kegiatan desain yang lebih mahal. Studi ini berkaitan dengan analisis

kesalahan yang dibuat oleh pengembang. Ketika menganalisis kesalahan, pertanyaan pertama adalah: "Di mana dalam proses pembangunan kesalahan ini telah dibuat?" Ini mengarah ke salah satu pekerjaan dari kesalahan ke setiap tahapan atau kegiatan di Lifecycle. Hukum ini menggabungkan dua observasi yang erat kaitannya.



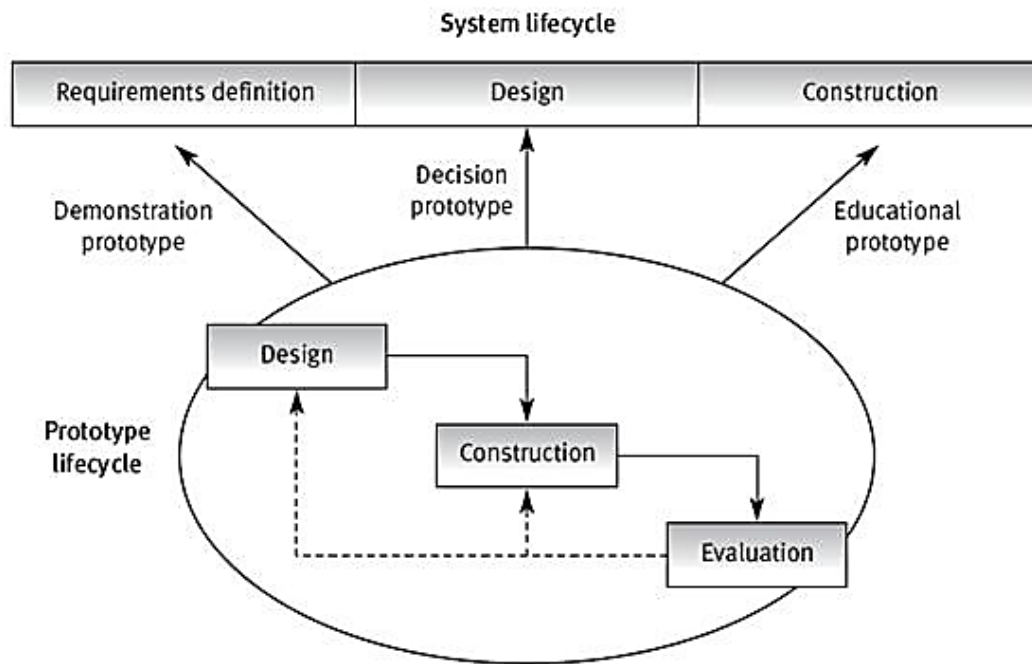
Gambar 5.1 Cost of problems per phase

Teori: Manusia biasanya mempunyai masalah jika banyak situasi perlu pemikiran pada saat yang bersamaan. Kita cenderung untuk berpikir baris utamanya saja, dan melupakan kasus khusus. Bahkan jika pikiran manusia mendukung pemrosesan paralel, ini tidak berarti bahwa perbedaan berbagai unit investigasi di berbagai penjuru. Kami memiliki arti yang tidak melekat atau mekanisme untuk mencari domain secara mendalam (kecuali dapat diwakili secara visual). Kesalahan dari kelalaian lebih sering daripada kesalah-pahaman.

c. Hukum Boehm kedua

Prototyping (secara signifikan) mengurangi kebutuhan dan kesalahan desain, terutama untuk user interface. Hukum ini menempatkan penekanan pada pengurangan kesalahan. Pengurangan kesalahan membawa penurunan biaya juga. Jumlah pengurangan tidak terukur, namun, menjadi signifikan, setidaknya 20-30 persen harus terjadi. Hal ini berlaku untuk semua hukum, meskipun kata 'signifikan' akan diabaikan. Perubahan dalam kisaran 5-20 persen karena

perbedaan pengukuran atau setup, atau dapat disebabkan oleh gangguan tak terkendalikan.



Gambar 5.2 Prototypes in the system lifecycle

Teori: Prototip memberikan pandangan dari sistem yang tampak nyata bagi pengguna. Berbeda dengan representasi desain lainnya, prototip tidak bergantung pada kekuatan imajinasi orang untuk memvisualisasikannya. Ini adalah perwujudan sebagian sistem yang sesungguhnya, bukan yang abstrak. Mungkin lebih menekankan detail dan dengan demikian tidak menyembunyikan atau merusak penampakan total dari sistem. Prototip perlu dibuat untuk sistem di bawah pengembangan saja, bukan untuk sistem yang ada.

d. Hukum Davis

Nilai dari sebuah model tergantung pada pandangan diambil, tetapi tidak ada yang terbaik untuk semua tujuan. Model adalah bentuk yang sangat berguna untuk menjelaskan sistem. Hal ini berlaku sebelum, selama, dan sesudah pengembangan sistem. Contoh model yang digunakan dalam ilmu alam adalah model yang menggambarkan evolusi bintang, model atom atau pengoperasian sel. Model tersebut konsep intelektual yang pertama, namun dapat diwujudkan atau dinyatakan dalam sebuah representasi yang terlihat. Dalam ilmu komputer,

kita dapat menggunakan model untuk mempelajari struktur statis objek sistem atau komponen, struktur logika data yang digunakan, atau struktur dinamis interaksi tugas dan proses.

Teori: Sebuah model dari realitas membantu untuk menjelaskan pemahaman. Model merupakan penjelasan dari sistem. Model taklangsung terlihat atau abstrak, berangkat dari hal-hal yang tidak dianggap penting untuk sementara waktu. Abstraksi berguna untuk beberapa jenis pemahaman manusia saja. Abstraksi merupakan pengetahuan konseptual yang ditingkatkan. Tidak semua pengguna perlu, ingin atau bahkan akan mentolerir abstraksi ini. Dari sudut pandang sistem yang akan dibangun, abstraksi berangkat dari kenyataan yang tergantung pada notasi yang digunakan, yang sering menipu pengamat. Gerakan bintang dalam gugus bintang, atau orbit elektron dalam model atom, hanya ada satu persamaan kusam pada kenyataan. Namun demikian, model-model seperti itu sering digunakan untuk tujuan yang bermanfaat.

1.2. Definisi Rekayasa Kebutuhan PL

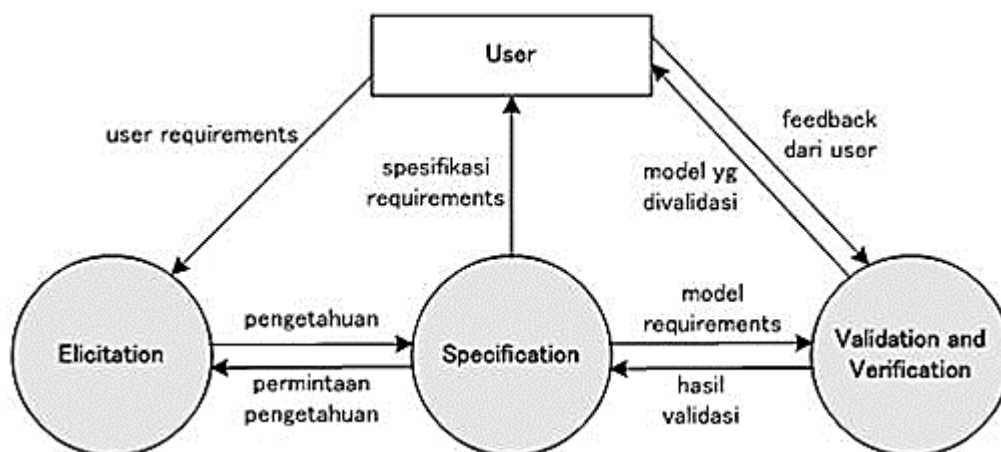
Requirements engineering merupakan fase terdepan dari proses rekayasa perangkat lunak (software engineering), dimana software requirements (kebutuhan) dari user (pengguna) dan customer (pelanggan) dikumpulkan, dipahami dan ditetapkan. Para pakar software engineering sepakat bahwa requirements engineering adalah suatu pekerjaan yang sangat penting. Fakta membuktikan bahwa kebanyakan kegagalan pengembangan software disebabkan karena adanya ketidakkonsistenan (inconsistent), ketidaklengkapan (incomplete), maupun ketidakbenaran (incorrect) dari requirements specification (spesifikasi kebutuhan). Ada sejumlah definisi yang telah diajukan oleh beberapa penulis sebelumnya tentang rekayasa kebutuhan. Terminologi rekayasa kebutuhan sendiri terdiri dari dua kata, yaitu kata "rekayasa" atau engineering, dan kata "kebutuhan" atau requirements. Berikut ini adalah beberapa definisi dari rekayasa kebutuhan :

- Requirement Engineering adalah proses menentukan properti tertentu dari sistem yang harus ada, dengan kata lain, menentukan komponen-komponen sistem. Kebutuhan proses menghasilkan informasi tentang desain yang akan menjadi dasar. Untuk ini, harus mengetahui dimana sebuah sistem akan

digunakan, oleh siapa, dan layanan apa yang harus disediakan. Juga penting untuk menentukan kompromi apa yang dapat dilakukan jika terjadi konflik kebutuhan. Kita berasumsi bahwa setiap sistem memiliki kumpulan fungsi yang berguna, yang penting untuk keberhasilan [Alb-2003].

- Sommerville mengartikan rekayasa kebutuhan (requirement engineering) sebagai suatu proses mewujudkan serangkaian layanan yang dibutuhkan oleh pelanggan atas suatu sistem dan batasan-batasan yang harus dipenuhi ketika dibangun maupun dioperasikan
- Menurut Roger S. Pressman rekayasa Kebutuhan meliputi kumpulan dari tugas-tugas yang mengarah ke pemahaman tentang apa yang akan menjadi dampak dari bisnis perangkat lunak, apa yang diinginkan oleh pelanggan, dan bagaimana pemakai akan berinteraksi dengan perangkat lunak. Rekayasa kebutuhan membantu para software engineer untuk lebih memahami masalah yang mereka coba selesaikan.

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. Proses requirements engineering ini dilakukan secara iterasi dengan mengakomodasi adanya feedback dari customer (user).



Gambar 5.3 The three dimensions of requirement engineering

1) 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.

2) 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.

3) 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.

1.3. Kebutuhan Perangkat Lunak

1.3.1. Definisi Kebutuhan

Dalam proses produksi perangkat lunak, hal utama yang menjadi bahan baku dalam proses tersebut adalah adanya kebutuhan (requirement) dari perangkat lunak yang dibutuhkan tersebut dari pengguna. Adapun definisi dari kebutuhan (requirement) antara lain :

- Sommerville dan Sawyer (1997) mengartikan kebutuhan sebagai berikut : sekumpulan kebutuhan yang merupakan suatu spesifikasi dari apa yang hendak diimplementasikan.
- Lawrence (1996) mendefinisikan kebutuhan sebagai segala sesuatu yang mengemudikan segala rancangan.
- Sementara IEEE Standard Glossary of Software Engineering Technology (1977) mendefinisikan kebutuhan sebagai 3 definisi berikut:
 - 1) Suatu kondisi atau kemampuan yang diperlukan pengguna untuk menyelesaikan masalah atau mencapai sesuatu.
 - 2) Suatu kondisi atau kemampuan yang harus dipenuhi atau dimiliki oleh suatu sistem atau komponen sistem untuk memenuhi suatu kontrak, standard, spesifikasi, atau dokumen formal lain yang diberlakukan.
 - 3) Suatu representasi terdokumentasi dari suatu kondisi atau kemampuan seperti dalam 1) dan 2).

Berdasarkan definisi tersebut, dapat dikatakan bahwa spesifikasi kebutuhan merupakan deskripsi bagaimana sistem harus berlaku, properti atau atribut sistem, dan karakteristik dan kualitas sistem sehingga dapat memberi nilai atau berguna bagi penggunaannya. Spesifikasi kebutuhan juga dapat dilihat sebagai suatu kesatuan kebutuhan (need) terdokumentasi dari suatu produk atau layanan tertentu yang harus ada atau dilakukan atau dibuat.

1.3.2. Jenis Kebutuhan Perangkat Lunak

Kebutuhan Perangkat Lunak adalah kondisi, kriteria, batasan-batasan, syarat, atau kemampuan yang harus dimiliki oleh produk perangkat lunak yang akan dibangun

untuk memenuhi apa yang diinginkan atau disyaratkan pemakai. Kebutuhan perangkat lunak dapat dibedakan menjadi :

1) Kebutuhan Fungsional

Kebutuhan Fungsional adalah kebutuhan yang berisi proses-proses apa saja / layanan apa saja yang nantinya harus disediakan oleh sistem, mencakup bagaimana sistem harus bereaksi pada input tertentu dan bagaimana perilaku sistem pada situasi tertentu.

Kebutuhan Fungsional Sangat bergantung dari jenis perangkat lunak, pengguna sistem, dan jenis sistem dimana perangkat lunak tersebut digunakan. Kebutuhan Fungsional dapat dicari dari pertanyaan: Apa yang harus sistem lakukan?. karena kebutuhan fungsional harus dapat menggambarkan layanan-layanan yang bisa diberikan sistem kepada pengguna secara mendetail.

2) Kebutuhan Non Fungsional

Kebutuhan Non Fungsional adalah kebutuhan yang menitikberatkan pada properti perilaku yang dimiliki oleh sistem. Kebutuhan fungsional juga sering disebut sebagai batasan layanan atau fungsi yang ditawarkan sistem seperti batasan waktu, batasan pengembangan proses, standarisasi dan lain lain. Secara umum kebutuhan non fungsional suatu perangkat lunak terdiri dari empat macam, yaitu:

a. Usability

Usability adalah kebutuhan non fungsional terkait dengan kemudahan penggunaan sistem atau perangkat lunak oleh user.

b. Portability

Portability ialah kemudahan dalam pengaksesan sistem khususnya terkait dengan faktor waktu dan lokasi pengaksesan, serta perangkat atau teknologi yang digunakan untuk mengakses. Perangkat atau teknologi tersebut meliputi perangkat lunak, perangkat keras, dan perangkat jaringan.

c. Reliability

Reliability merupakan kebutuhan terkait kehandalan sistem atau perangkat

lunak termasuk juga faktor keamanan (security) sistem.

d. Supportability

Supportability ialah kebutuhan terkait dengan dukungan dalam penggunaan sistem atau perangkat lunak.

1.4. Dokumentasi Software Requirement Specification

Hasil dari fase requirements engineering terdokumentasi dalam software requirements specification (SRS). Requirements specification berisi kesepakatan bersama tentang permasalahan yang ingin dipecahkan antara pengembang dan pelanggan, dan merupakan titik start menuju proses berikutnya yaitu software design. Dokumen kebutuhan merupakan pernyataan resmi dari apa yang dibutuhkan dari pembangun sistem, berisi definisi dan spesifikasi requirement dan bukan dokumen desain. Sebisa mungkin berupa kumpulan dari APA yang harus dikerjakan sistem, BUKAN BAGAIMANA sistem mengerjakannya. Dokumen kebutuhan sebaiknya memenuhi 6 hal berikut :

- 1) Menjelaskan perilaku eksternal system
- 2) menjelaskan batasan pada implementasi
- 3) mudah diubah
- 4) sebagai alat referensi untuk pemelihara system
- 5) mencatat peringatan awal tentang siklus dari system
- 6) menjelaskan bagaimana sistem merespon hal-hal yang tidak biasa/normal.

IEEE menyarankan standar struktur dari dokumen kebutuhan sebagai berikut :

- 1) introduction
 - purpose of the requirement document
 - scope of the product
 - definitions, acronyms and abbreviations
 - references
 - overview of the remainder of the document
- 2) General description
 - product perspective

- product functions
 - user characteristics
 - general constrains
 - assumptions and dependencies
- 3) appendices
 - 4) index

Sekalipun standar IEEE belumlah ideal tetapi telah memberikan masukan format dokumen yang cukup lengkap. Informasi yang dimasukkan ke dalam dokumen tergantung pada tipe software yang dibangun dan pendekatan yang digunakan untuk membangun software tersebut.

Rangkuman

Definisi Kebutuhan (Requirement Definitions) adalah pernyataan yang mengidentifikasi kebutuhan yang penting dalam sistem dan di dalamnya 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 pengembang (developer). RE yang baik adalah penting karena dampaknya mampu mengurangi biaya proyek, dan diterimanya sistem oleh stakeholder sehingga bisa mengarah kepada keuntungan yang tinggi. Namun juga harus diakui dibutuhkan tenaga dan waktu yang tidak sedikit untuk berinvestasi dalam pembuatan requirement yang benar-benar baik. Untuk mendapatkan requirement yang baik, ada banyak pekerjaan/tasks harus dilakukan, untuk itu tim Requirements Engineering tidak hanya bekerja pada awal dari proyek namun bekerja melalui tahap pengembangan sampai tahap delivery untuk memastikan requirement benar-benar sesuai. Karena kompleksitas, ragam pengetahuan dan keahlian khusus serta bidang kerja yang banyak, maka Requirement Engineering telah menjadi cabang ilmu baru pada tahun 1990an.

Latihan

Berdasarkan studi kasus yang diberikan, lakukan analisis kebutuhannya dan tuliskan daftar kebutuhan fungsional dan non fungsionalnya. Anda boleh menambahkan asumsi yang relevan dari kasus yang diberikan!.

"CIPAGANTI" adalah perusahaan penyewaan mobil.

Berikut adalah proses bisnis yang terjadi pada "CIPAGANTI":

- Pelanggan yang ingin menyewa mobil harus mengisi Form Sewa Kendaraan (FSK). Setiap penyewa harus menggunakan supir. Setiap jenis kendaraan memiliki harga sewa yang berbeda-beda. Harga sewa jasa sopir untuk daerah Jabodetabek dan diluar jabodetabek berbeda (lihat tabel harga sewa). Setelah mengisi FSK, pelanggan menyerahkan FSK kepada Staf Administrasi untuk dihitung jumlah uang sewa yang harus dibayarkan. Setelah Staf Administrasi mengkonfirmasi jumlah uang sewa kemudian pelanggan melunasi uang sewa dimuka dan sebagai bukti pembayarannya pelanggan mendapatkan Kwitansi dari Staf Administrasi.
- Setiap bulan Staf Administasi membuat laporan penyewaan mobil dan laporan pendapatan yang diberikan kepada Pimpinan.
- Berikut adalah dokumen yang dapat membantu untuk analisis kebutuhan dari proses bisnis yang ada :

Tabel Harga Sewa Kendaraan

Jenis Kendaraan	Merk	Kapasitas Max Penumpang	Sewa/hari
- Minibus			
	Toyota Innova	8 orang	Rp . 600.000
	Isuzu Panther	8 orang	Rp. 500.000
	Mitsubishi Kuda	8 orang	Rp. 500.000
	Suzuki APV	10 orang	Rp. 400.000
- Sedan			
	Toyota Camry	5 orang	Rp. 800.000
	Suzuki Esteem	5 orang	
- Mini Van			
	Mitsubishi Elf	15 orang	Rp. 1.000.000
	Nissan Serena	12 orang	Rp. 750.000

FORM SEWA KENDARAAN (FSK)

No :

PELANGGAN

No.KTP :
Nama :
Alamat :
Telp :

KENDARAAN

No. Polisi :
Jenis Kendaraan :
Merk :
No. Mesin :

TUJUAN :

Jumlah Penumpang :
Alamat :
Kota : ☐ Jabodetabek ☐ Luar Jabodetabek
Telpon/Hp :

WAKTU & HARGA SEWA

Tanggal Mulai :
Tanggal Selesai :

SUPIR

Nama :
Telp :

Sewa/Hari : Rp
Jumlah Hari Sewa :

Sewa/hari : Rp.
Jumlah Hari :

Total : Rp.

Total : Rp.

TOTAL SELURUH (SEWA MOBIL+SUPIR) : Rp.

Jakarta, - - 2019
Penyewa

Ketentuan :

- Bahan bakar disediakan oleh pelanggan

Biaya Jasa Sopir/hari

- o Jabodetabek = Rp. 100.000
- o Luar Jabodetabek = Rp. 150.000

KWITANSI

No :

Telah Terima dari : Tn/Ny.

Uang Sebesar :

Untuk Pembayaran : Sewa Mobil No.Polisi

Selama hari x = Rp

Sewa Sopir selama hari x= Rp

*

Total Rp.

Jakarta,

Pelanggan

(.....)

"CIPAGANTI"**Laporan Penyewaan**

Periode : dd/mm/yyyy s/d dd/mm/yyyy

No.	No. FSK	Tanggal FSK	Nama Pelanggan	Jenis Kendaraan	Merk	Tujuan	Tanggal Mulai	Tanggal Selesai	Jumlah Hari Sewa	Nama Supir
99	XXXXX	Dd/mm/yyyy	X-25-X	X-30-X	X-12-X	X-20-X	Dd/mm/yyyy	Dd/mm/yyyy	99	X-20-X

Mengetahui, Jakarta, dd/mm/yyyy

Kep. Bagian Pengadaan

Staff Pengadaan

(.....)

(.....)



FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR

Jl. Raya Ciledug, Petukangan Utara, Pesanggrahan

Jakarta Selatan, 12260

Telp: 021-5853753 Fax : 021-5853752

<http://fti.budiluhur.ac.id>