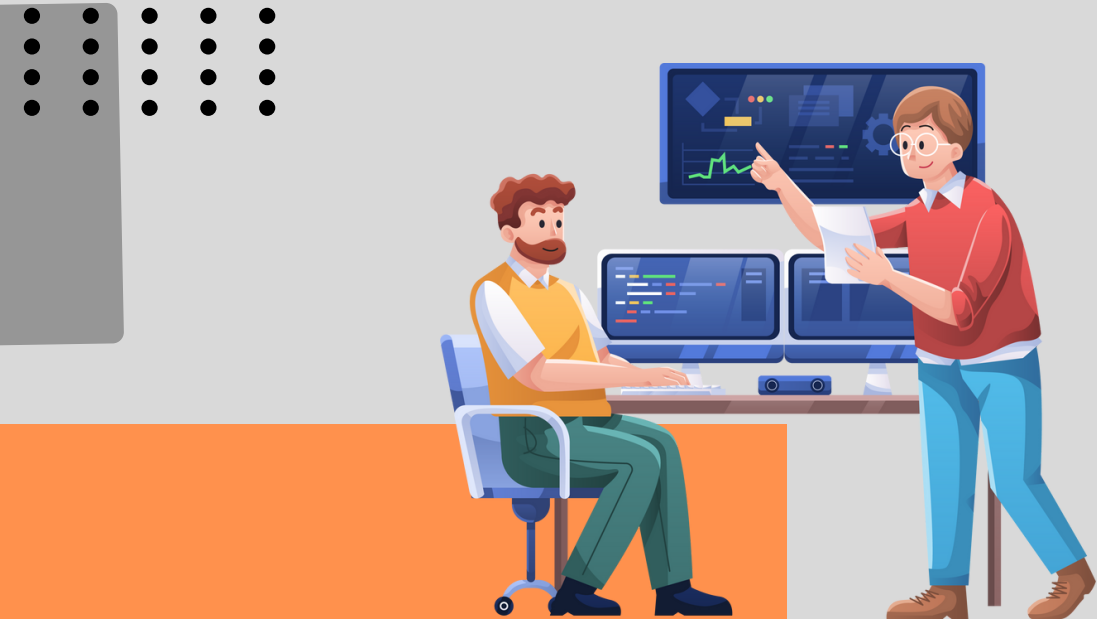


# REKAYASA PERANGKAT LUNAK



Romindo, Reska Mayefis, Tri Yusnanto, Nono Heryana, Jamaludin,  
Allans Prima Aulia, Angga Aditya Permana,  
Sitti Aisa, Johni S Pasaribu, Wahyuddin S, Fredy AH Sihombing

# **REKAYASA PERANGKAT LUNAK**

**Romindo  
Reska Mayefis  
Tri Yusnanto  
Nono Heryana  
Jamaludin  
Allans Prima Aulia  
Angga Aditya Permana  
Sitti Aisa  
Johni S Pasaribu  
Wahyuddin S  
Fredy AH Sihombing**



**PT GLOBAL EKSEKUTIF TEKNOLOGI**

# REKAYASA PERANGKAT LUNAK

**Penulis :**

Romindo  
Reska Mayefis  
Tri Yusnanto  
Nono Heryana  
Jamaludin  
Allans Prima Aulia  
Angga Aditya Permana  
Sitti Aisa  
Johni S Pasaribu  
Wahyuddin S  
Fredy AH Sihombing

**ISBN : 978-623-8102-96-9**

**Editor :** Dr.Safrizal.,ST.,MM.,M.Kom

Diana Purnama Sari, S.E., M.E.

**Penyunting :** Tri Putri Wahyuni, S.Pd

**Desain Sampul dan Tata Letak :** Atyka Trianisa, S.Pd

**Penerbit :** PT GLOBAL EKSEKUTIF TEKNOLOGI

Anggota IKAPI No. 033/SBA/2022

**Redaksi :**

Jl. Pasir Sebelah No. 30 RT 002 RW 001  
Kelurahan Pasie Nan Tigo Kecamatan Koto Tangah  
Padang Sumatera Barat

Website : [www.globaleksekutifteknologi.co.id](http://www.globaleksekutifteknologi.co.id)

Email : [globaleksekutifteknologi@gmail.com](mailto:globaleksekutifteknologi@gmail.com)

Cetakan pertama, 17 Januari 2023

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk  
dan dengan cara apapun tanpa izin tertulis dari penerbit.

## **KATA PENGANTAR**

Segala Puji dan syukur kehadirat Allah SWT dalam segala kesempatan. Sholawat beriring salam dan doa kita sampaikan kepada Nabi Muhammad SAW. Alhamdulillah atas Rahmat dan Karunia-Nya penulis telah menyelesaikan Buku Rekayasa Perangkat Lunak ini.

Proses penulisan buku ini berhasil diselesaikan atas kerjasama tim penulis. Demi kualitas yang lebih baik dan kepuasan para pembaca, saran dan masukan yang membangun dari pembaca sangat kami harapkan.

Penulis ucapkan terima kasih kepada semua pihak yang telah mendukung dalam penyelesaian buku ini. Terutama pihak yang telah membantu terbitnya buku ini dan telah mempercayakan mendorong, dan menginisiasi terbitnya buku ini. Semoga buku ini dapat bermanfaat bagi masyarakat Indonesia.

Padang, 17 Januari 2023

Penulis

## DAFTAR ISI

<b>KATA PENGANTAR .....</b>	<b>i</b>
<b>DAFTAR ISI.....</b>	<b>ii</b>
<b>DAFTAR GAMBAR .....</b>	<b>vi</b>
<b>DAFTAR TABEL .....</b>	<b>viii</b>
<b>BAB 1 PENDAHULUAN .....</b>	<b>1</b>
1.1 Pendahuluan.....	1
1.2 Pengertian Perangkat Lunak Dan Rekayasa Perangkat Lunak.....	1
1.3 Karakteristik Perangkat Lunak.....	5
1.4 Kualitas Perangkat Lunak.....	6
1.5 Konsep Rekayasa Sistem.....	7
DAFTAR PUSTAKA .....	11
<b>BAB 2 PROSES PERANGKAT LUNAK.....</b>	<b>13</b>
2.1 Pendahuluan.....	13
2.2 Pengertian Proses Perangkat Lunak .....	14
2.3 Proses Perangkat Lunak.....	15
2.3.1 Waterfall Model .....	17
2.3.2 Evolutionary Model.....	19
2.3.3 Increment Model.....	20
2.3.4 Spiral Model .....	21
DAFTAR PUSTAKA .....	24
<b>BAB 3 PENGEMBANGAN PERANGKAT LUNAK AGILE.....</b>	<b>25</b>
3.1 Perangkat Lunak .....	25
3.2 Definisi Metode <i>Agile</i> .....	29
3.3 Faktor Manusia pada <i>Agile Process Model</i> .....	31
3.4 Cara Kerja Metode Agile.....	32
3.5 Model Proses <i>Agile</i> .....	34
3.6 Tahapan Metode Agile .....	54
3.7 Kelebihan Metode Agile .....	55
3.8 Kekurangan Metode Agile .....	56
3.9 Kesimpulan .....	56

DAFTAR PUSTAKA .....	58
<b>BAB 4 REKAYASA PERSYARATAN PERANGKAT LUNAK.....</b>	<b>61</b>
4.1 Pendahuluan.....	61
4.2 Definisi .....	65
4.2.1 Definisi Persyaratan.....	65
4.2.1 Definisi Rekayasa Persyaratan.....	66
4.3 Level dan Jenis Persyaratan.....	66
4.4 Manajemen Persyaratan.....	69
4.5 Aktivitas Persyaratan dalam Siklus Hidup Sistem .....	70
DAFTAR PUSTAKA .....	73
<b>BAB 5 DESAIN DAN IMPLEMENTASI .....</b>	<b>75</b>
5.1 Pendahuluan.....	75
5.2 Konsep Dasar Desain.....	75
5.3 Konsep Dasar Implementasi .....	78
5.4 Penerapan Desain dan Implementasi pada Perancangan Sistem Perangkat Lunak.....	81
5.4.1 Desain dan Implementasi Sistem Informasi Pelayanan Kesehatan Masyarakat.....	81
5.4.2 Desain dan Implementasi Sistem Informasi Akademik.....	86
DAFTAR PUSTAKA .....	90
<b>BAB 6 PENGUJIAN PERANGKAT LUNAK.....</b>	<b>93</b>
6.1 Pengujian Perangkat Lunak.....	93
6.2 Level Pengujian Perangkat Lunak.....	95
6.3 Metode Pengujian Perangkat Lunak.....	97
6.4 Pendekatan Pengujian Perangkat Lunak.....	101
DAFTAR PUSTAKA .....	105
<b>BAB 7 EVOLUSI PERANGKAT LUNAK.....</b>	<b>107</b>
7.1 Pendahuluan.....	107
7.2 Prinsip dan Konsep.....	108
7.2.1 Perubahan Korektif.....	108
7.2.2 Perubahan Adaptif .....	109

7.2.3 Perubahan Perfektif .....	109
7.2.4 Perubahan Preventif.....	110
7.3 Proses Evolusi Perangkat Lunak .....	111
7.4 Dinamika Evolusi Perangkat Lunak.....	114
7.5 Pemeliharaan Sistem Perangkat Lunak.....	115
DAFTAR PUSTAKA .....	118
<b>BAB 8 SISTEM YANG DAPAT DIANDALKAN.....</b>	<b>119</b>
8.1 Pendahuluan.....	119
8.2 Pemodelan.....	119
8.3 Defenisi <i>Uml</i> .....	120
8.2.1 Diagram – Diagram <i>UML</i> .....	121
8.4 Tahapan Perancangan Sebuah Sistem.....	127
DAFTAR PUSTAKA .....	
<b>BAB 9 REKAYASA KEANDALAN.....</b>	<b>131</b>
9.1 Pendahuluan.....	131
9.2 Keandalan Perangkat Lunak .....	133
9.3 Kegagalan Perangkat Lunak vs Kegagalan Perangkat Keras .....	134
9.4 Mengapa Perangkat Lunak Gagal?.....	137
9.5 Keandalan Software dan Sistem .....	139
9.6 Teknik, Model dan Metrik.....	142
9.6.1 Model Keandalan Black Box .....	147
9.6.2 Model Keandalan Berbasis Arsitektur (White Box) .....	148
DAFTAR PUSTAKA .....	151
<b>BAB 10 TEKNIK KEAMANAN .....</b>	<b>155</b>
10.1 Keamanan Komputer .....	155
10.2 Segitiga CIA (CIA Triad) .....	155
10.2.1 <i>Confidentiality</i> .....	156
10.2.2 <i>Integrity</i> .....	157
10.2.3 <i>Availability</i> .....	158
10.3 Teknik Keamanan Komputer.....	158
10.3.1 <i>Firewall</i> .....	158

10.3.2 <i>Intrusion Detection System</i> .....	161
10.3.3 <i>Software Antivirus</i> .....	163
10.4 Metode Keamanan Komputer .....	164
DAFTAR PUSTAKA .....	166
<b>BAB 11 MODEL DAN METODE DALAM TEKNIK</b>	
<b>KETAHANAN REKAYASA PERANGKAT LUNAK .....</b>	<b>169</b>
11.1 Pendahuluan.....	169
11.2 Konsep Rekayasa Sistem.....	170
11.3 Pemodelan Sistem.....	172
11.3.1 Analisis.....	172
11.3.2 Berkomunikasi .....	172
11.3.3 Prediksi.....	172
11.3.4 Pengendalian .....	172
11.3.5 Simulasi .....	172
11.4 Metodologi Pengembangan Perangkat Lunak.....	173
11.5 Kelebihan Dan Kelemahan SDLC.....	180
11.5 Pengertian Dan Ragam Metodologi Pengembangan	182
11.6 Pentingnya Metodologi Pengembangan Perangkat Lunak.....	187
DAFTAR PUSTAKA .....	190
<b>BIODATA PENULIS</b>	



## DAFTAR GAMBAR

<b>Gambar 2.1 :</b> Gambaran Umum Proses.....	15
<b>Gambar 2.2 :</b> Gambaran Umum Proses Perangkat Lunak.....	15
<b>Gambar 2.3 :</b> Siklus Waterfall Model .....	18
<b>Gambar 2.4 :</b> Siklus Evolutionary Model .....	19
<b>Gambar 2.5 :</b> Siklus Increment Model.....	21
<b>Gambar 2.6 :</b> Siklus Spiral Model.....	22
<b>Gambar 2.7 :</b> Agile Process .....	23
<b>Gambar 3.1 :</b> Ilustrasi Perangkat Lunak .....	25
<b>Gambar 3.2 :</b> Ilustrasi Kerjasama.....	31
<b>Gambar 3.3 :</b> <i>Extreme Programming</i> .....	34
<b>Gambar 3.4 :</b> Ilustrasi Metode Crystal.....	37
<b>Gambar 3.5 :</b> <i>Adaptive Software Development</i> .....	40
<b>Gambar 3.6 :</b> <i>Dynamic Systems Development Method</i> .....	42
<b>Gambar 3.7 :</b> Ilustrasi Metode SCRUM .....	44
<b>Gambar 3.8 :</b> Ilustrasi Prinsip <i>Lean Software Development</i> .....	47
<b>Gambar 3.9 :</b> Ilustrasi Agile .....	50
<b>Gambar 3.10 :</b> Ilustrasi Tahapan Agile.....	54
<b>Gambar 4.1 :</b> Hubungan antara beberapa jenis informasi persyaratan.....	68
<b>Gambar 5.1 :</b> Metode Waterfall.....	77
<b>Gambar 5.2 :</b> Diagram use case .....	82
<b>Gambar 5.3 :</b> Diagram Activity .....	83
<b>Gambar 5.4 :</b> Diagram Sequence .....	84
<b>Gambar 5.5 :</b> Diagram class.....	85
<b>Gambar 5.6 :</b> Antarmuka ketersediaan kamar dan dokter jaga .....	86
<b>Gambar 5.7 :</b> Diagram use case .....	87
<b>Gambar 5.8 :</b> Activity diagram .....	88
<b>Gambar 5.9 :</b> Gambar tampilan form pengisian KRS .....	89



## DAFTAR TABEL

<b>Tabel 4.1 :</b> Alasan Kegagalan Proyek.....	63
<b>Tabel 4.2 :</b> Faktor Keberhasilan Proyek.....	63
<b>Tabel 4.3 :</b> Faktor Keberhasilan Proyek.....	64
<b>Tabel 4.4 :</b> Beberapa jenis informasi persyaratan .....	67
<b>Tabel 6.1 :</b> Perbedaan Verifikasi dan Validasi .....	100
<b>Tabel 7.1 :</b> Hukum Lehman .....	114
<b>Tabel 8.1 :</b> Simbol – simbol <i>use case diagram</i> .....	122
<b>Tabel 8.2 :</b> Simbol – simbol <i>Class diagram</i> .....	123
<b>Tabel 8.3 :</b> Simbol – simbol <i>Sequence diagram</i> .....	125

# **BAB 4**

## **REKAYASA PERSYARATAN PERANGKAT LUNAK**

*Oleh Nono Heryana*

### **4.1 Pendahuluan**

Persyaratan (*requirement*) adalah atribut yang diperlukan dalam suatu sistem, pernyataan yang mengidentifikasi kemampuan, karakteristik, atau faktor kualitas suatu sistem agar memiliki nilai dan kegunaan bagi pelanggan atau pengguna. Persyaratan penting karena menyediakan dasar untuk semua pekerjaan pengembangan berikutnya. Setelah persyaratan ditetapkan, pengembang memulai pekerjaan teknis lainnya: desain sistem, pengembangan, pengujian, implementasi, dan operasi. Persyaratan ini harus dipenuhi oleh sistem atau komponen sistem untuk memenuhi standar, spesifikasi atau dokumen formal lainnya. Rekayasa persyaratan merupakan bagian penting dari pengembangan perangkat lunak yang bertujuan untuk mengumpulkan persyaratan kualitas, menganalisis dan mendokumentasikannya untuk implementasi selanjutnya baris kode program dengan cara yang tepat untuk mencapai fungsionalitas yang diinginkan dan memenuhi persyaratan pengguna (Rehman, Khan and Riaz, 2013).

Proses ini merupakan tahapan yang kritis dan kompleks untuk pengembangan perangkat lunak karena diversifikasi dalam persyaratan yang diperoleh dan karena perubahan yang cepat dalam persyaratan. Sangat sulit untuk mengembangkan persyaratan yang akurat yang tetap konsisten dalam sistem yang besar dan kompleks, sebagian besar kesalahan dalam

fungsionalitas perangkat lunak terkait dengan kesalahan yang dilakukan pada saat pengumpulan persyaratan (requirement gathering), karena dasar untuk setiap proyek dan teknologi berubah dengan cepat sehingga rekayasa persyaratan yang efektif merupakan jantung pacu yang mendefinisikan apa dibutuhkan stakeholder baik itu pengguna, pelanggan, suplier, pengembang ataupun bisnis dari sistem baru yang potensial dan juga apa yang dibutuhkan sistem untuk memenuhi persyaratan itu serta mengikuti arus kompleksitas sistem yang meningkat.

Proses persyaratan tidak rumit atau mahal (Young, 2004). Namun, persyaratan dibutuhkan untuk apapun ukuran proyeknya, sangat penting bahwa suatu proyek atau organisasi memiliki proses persyaratan yang ditetapkan dan didokumentasikan. Sifat komponen spesifik dari proses yang ditentukan dapat ditingkatkan berdasarkan pengalaman.

Persyaratan dibuat untuk dipahami semua orang dengan berbagai macam latar belakang, umumnya persyaratan dideskripsikan dalam bahasa yang sederhana dan universal, disinilah letak tantangannya. Untuk men-capture persyaratan atau masalah secara lengkap dan tidak ambigu, rekayasa persyaratan merupakan bagian integral dari siklus hidup pengembangan perangkat lunak karena dasar untuk mengembangkan perangkat lunak yang sukses bergantung pada pemahaman persyaratannya sejak awal. Rekayasa persyaratan melibatkan sejumlah proses untuk mengumpulkan persyaratan sesuai dengan persyaratan dan tuntutan pengguna dan pemangku kepentingan produk perangkat lunak. Setelah dikomunikasikan dan disetujui, persyaratan mendorong aktifitas proyek, namun persyaratan para *stakeholder* mungkin banyak dan beragam bahkan bisa menimbulkan berbagai macam konflik. Menurut (Dick, Hull and Jackson, 2017) Persyaratan menjadi dasar untuk:

- Perencanaan Proyek;
- Manajemen Risiko;

- *Trade off*;
- *Acceptance Testing*;
- *Change Control*.

Alasan yang umum untuk kegagalan proyek bukanlah hal teknis. Standish Group telah membuat laporan CHAOS terkait keberhasilan dan kegagalan sebuah proyek sejak tahun 1994. Tabel 4.1 & tabel 4.2 menunjukkan alasan utama kegagalan proyek dan faktor keberhasilan proyek pada tahun 1995.

**Tabel 4.1 : Alasan Kegagalan Proyek (Standish Group, 1995)**

	<b>Alasan Kegagalan</b>	<b>Persentase</b>
**	Persyaratan tidak lengkap	13,1%
**	Kurangnya keterlibatan pengguna	12,4%
	Kurangnya sumberdaya	10,6%
*	Ekspektasi yang tidak realistis	9,9%
**	Kurangnya dukungan eksekutif	9,3%
	Perubahan persyaratan/spesifikasi	8,7%
**	Kurangnya perencanaan	8,1%
**	Tidak membutuhkannya lagi	7,5%

Sumber: Dick et al. 2017

**Tabel 4.2 : Faktor Keberhasilan Proyek (Standish Group, 1995)**

	<b>Faktor Keberhasilan</b>	<b>Persentase</b>
**	Keterlibatan Pengguna	15,9%
	Dukungan Manajemen	13,9%

	<b>Faktor Keberhasilan</b>	<b>Persentase</b>
**	Pernyataan persyaratan yang jelas	13,0%
	Perencanaan yang tepat	9,6%
*	Ekspektasi yang realistis	8,2%
	Milestone yang lebih kecil	7,7%
	Staff yang kompeten	7,2%
*	Kepemilikan	5,3%

Sumber: Dick et al. 2017

Faktor keberhasilan proyek dari periode yang sama tidak berbanding terbalik dengan faktor kegagalan (lihat tabel 4.2). dukungan manajemen dan perencanaan yang tepat, terlihat penting disini. Semakin besar proyek, semakin lama jadwalnya dan semakin tinggi kemungkinan gagalnya.

**Tabel 4.3 : Faktor Keberhasilan Proyek (Standish Group, 1995)**

	<b>Faktor Keberhasilan</b>	<b>Persentase</b>
*	Dukungan manajemen eksekutif	20%
**	Keterlibatan pengguna	15%
	Optimalisasi	15%
	SDM yang terampil	13%
	Keahlian Manajemen Proyek	12%
*	<i>Agile</i> Proses	10%
**	Tujuan bisnis yang jelas	6%
*	Kematangan emosional	5%

	<b>Faktor Keberhasilan</b>	<b>Persentase</b>
	Eksekusi	3%
	Alat dan infratraktur	1%

Sumber: Dick et al. 2017

Tabel 4.3 diatas ini menunjukkan data kriteria keberhasilan proyek tahun 2015. Selama bertahun-tahun cara standish mengkorelasikan data merekamembuahkan hasil karena ada sedikit perubahan, tetapi faktor-faktor yang mendasarinya tetap dan relatif tidak berubah.

## 4.2 Definisi

### 4.2.1 Definisi Persyaratan

Brian Lawrence (1997) dalam (Wiegers and Beatty, 2013) menyarankan persyaratan adalah “segala sesuatu yang mendorong pilihan desain”. Ini menandakan bahwa ada banyak informasi yang terkandung dalam definisi ini. Lagi pula inti dari rekayasa persyaratan adalah membuat pilihan desain yang tepat yang akhirnya akan memenuhi persyaratan pelanggan.

Definisi lain adalah bahwa persyaratan adalah properti yang harus dimiliki suatu produk untuk memberikan nilai tambah kepada pemangku kepentingan (*stakeholders*). Sedangkan menurut (Sommerville, 2016) persyaratan adalah spesifikasi yang harus diimplementasikan yang mendeskripsikan bagaimana sistem harus berperilaku, properti atau atribut sistem yang mungkin jadi kendala dalam proses pengembangan sistem.

Persyaratan juga mencakup pandangan pengguna tentang karakteristik eksternal dari sistem dan pandangan pengembang tentang karakteristik internal dari sistem. Menurut (Koelsch, 2016) Persyaratan bisa berubah secara signifikan dari saat pertama kaliditentukan hingga sistem itu selesai.



#### **4.2.1 Definisi Rekayasa Persyaratan**

Istilah rekayasa persyaratan terlalu sempit jika disamakan dengan analisis kebutuhan yang hanya salah satu kegiatan dalam disiplin ilmu yang lebih luas. Rekayasa persyaratan menggunakan pendekatan yang sistematis dimana pengembang perangkat lunak mengumpulkan persyaratan dari berbagai sumber dan mengimplementasikannya kedalam proses pengembangan perangkat lunak (Pandey, Suman and Ramani, 2010). Kegiatan rekayasa persyaratan mencakup seluruh sistem dan siklus hidup pengembangan perangkat lunak. Jika dilihat lebih detail terkait definisi rekayasa persyaratan yang dikemukakan oleh (Dick, Hull and Jackson, 2017) yang menyatakan bahwa rekayasa persyaratan adalah bagian dari rekayasa sistem yang berkaitan dengan proses menemukan, mengembangkan, melacak, menganalisis, mengkualifikasi, mengkomunikasikan dan mengelola persyaratan yang mendefinisikan sistem pada tingkat abstraksi.

Dari definisi tersebut dapat disimpulkan bahwa Rekayasa persyaratan merupakan bagian penting dari proses rekayasa perangkat lunak, karena hampir semua aktivitas dalam proses pengembangan sistem membutuhkan suatu rekayasa persyaratan untuk mencapai tujuan tertentu. Rekayasa persyaratan (Aurum and Wohlin, 2005) mengacu pada semua aktivitas siklus hidup sistem yang terkait dengan persyaratan yang mencakup pengumpulan, pendokumentasian dan pengelolaan persyaratan.

#### **4.3 Level dan Jenis Persyaratan**

Pada bagian ini disajikan beberapa definisi yang digunakan untuk istilah-istilah yang biasa ditemui dalam domain persyaratan.

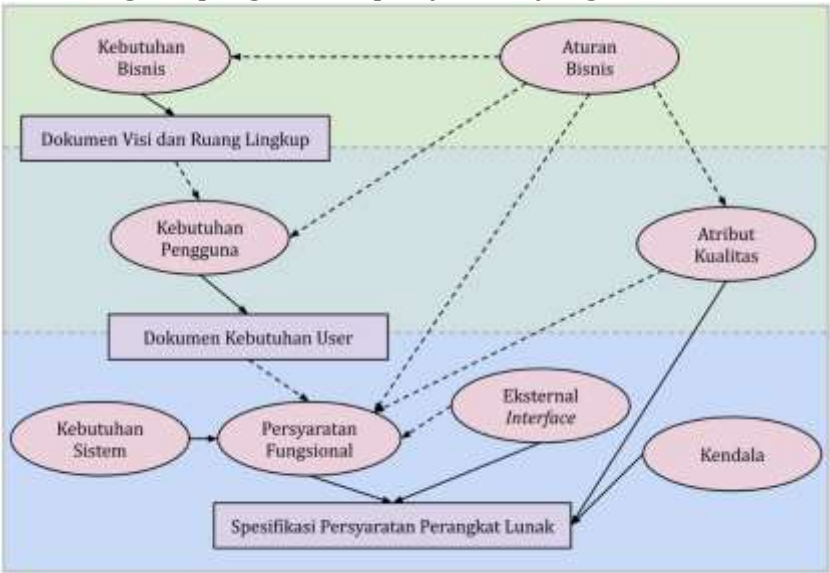
**Tabel 4.4 :** Beberapa jenis informasi persyaratan

No.	Istilah	Definisi
1	Persyaratan Bisnis	Tujuan bisnis tingkat tinggi dari organisasi yang harus dilakukan untuk memenuhi tujuan organisasi
2	Aturan Bisnis	Sebuah kebijakan, pedoman, standar, atau peraturan yang mendefinisikan atau membatasi berbagai aspek bisnis
3	Batasan	Batasan terkait opsi yang tersedia bagi pengembang untuk desain dan konstruksi suatu produk
4	Persyaratan Eksternal Interface	Mendeskripsikan koneksi antar sistem perangkat lunak dan pengguna, sistem lain atau perangkat keras
5	Fitur	Satu atau lebih kemampuan sistem yang terkait secara logis yang memberikan nilai kepada pengguna dan dideskripsikan oleh serangkaian persyaratan fungsional
6	Persyaratan Fungsional	Deskripsi perilaku yang akan ditunjukkan oleh sistem dalam kondisi tertentu
7	Persyaratan Non-fungsional	Deskripsi properti atau karakteristik yang harus ditunjukkan oleh sistem atau batasan yang harus dihormatinya
8	Atribut Kualitas	Semacam persyaratan non-fungsional yang menggambarkan layanan atau karakteristik kinerja dari suatu produk
9	Persyaratan Sistem	Persyaratan tingkat atas untuk produk yang berisi beberapa sub sistem, yang dapat berupa semua perangkat lunak atau perangkat lunak dan perangkat keras

No.	Istilah	Definisi
10	Persyaratan Pengguna	Tujuan atau tugas yang harus dilakukan oleh pengguna tertentu dengan sistem, atau atribut produk yang diinginkan

Sumber: Wiegers et al. 2013

Persyaratan perangkat lunak mencakup tiga level yang berbeda yaitu persyaratan bisnis, persyaratan pengguna, dan persyaratan fungsional. Selain itu, setiap sistem memiliki bermacam-macam persyaratan nonfungsional. Gambar 4.1 mengilustrasikan cara berpikir terkait berbagai jenis persyaratan, model ini tidak mencakup semuanya tetapi memberikan gambaran untuk mengatur pengetahuan persyaratan yang ditemui.



**Gambar 4.1 :** Hubungan antara beberapa jenis informasi persyaratan  
(Sumber : Wiegers et al, 2013)

Pada gambar 4.1 bentuk oval mewakili jenis informasi persyaratan dan persegi Panjang menunjukkan dokumen untuk menyimpan informasi tersebut. Panah solid menunjukkan bahwa jenis informasi tertentu biasanya 'disimpan di' dalam dokumen yang ditunjuk tanda panah. Aturan bisnis dan persyaratan sistem disimpan secara terpisah dari persyaratan perangkat lunak, seperti dalam katalog aturan bisnis atau spesifikasi persyaratan sistem. Panah putus-putus menunjukkan bahwa satu jenis informasi adalah 'berasal dari' atau 'mempengaruhi' persyaratan lainnya. Persyaratan data tidak ditampilkan secara eksplisit dalam diagram diatas.

#### **4.4 Manajemen Persyaratan**

Semua proyek pengembangan memiliki persyaratan yang merupakan dasar untuk desain. Aktifitas manajemen persyaratan mencakup hal-hal berikut :

1. eluruh proyek.
2. Mendefinisikan baseline persyaratan, snapshot dalam waktu yang mewakili seperangkat persyaratan fungsional dan nonfungsional yang disepakati, ditinjau dan disetujui, seringkali untuk rilis produk atau iterasi pengembangan tertentu.
3. Evaluasi dampak perubahan persyaratan yang diusulkan dan memasukkan perubahan yang disetujui kedalam proyek dengan cara yang terkendali.
4. Menjaga rencana proyek terkini sesuai dengan persyaratan.
5. Menegosiasikan komitmen baru berdasarkan perkiraan dampak perubahan persyaratan.
6. Mendefinisikan hubungan dan dependensi yang ada antara persyaratan.
7. Menelusuri persyaratan individu ke desain yang sesuai, kode sumber dan pengujian.

Melacak status persyaratan dan mengubah aktivitas disManajemen persyaratan bukanlah sarana untuk menahan perubahan atau mempersulit perubahan. Tetapi lebih kepada untuk mengantisipasi dan mengakomodir perubahan sesuai dengan harapan untuk meminimalkan dampak gangguan pada proyek pengembangan perangkat lunak.

#### **4.5 Aktivitas Persyaratan dalam Siklus Hidup Sistem**

Aktivitas yang berhubungan dengan persyaratan terutama terkait dengan pengumpulan persyaratan dan mengelola perubahan persyaratan sepanjang siklus hidup pengembangan sistem (Young, 2004). Namun, pada kenyataannya ada beberapa aktivitas terkait kebutuhan lain yang perlu ditangani dalam siklus hidup pengembangan sistem yang terdiri atas aktivitas-aktivitas:

1. Mengidentifikasi pemangku kepentingan (*stakeholder*): aktivitas ini termasuk siapa saja yang memiliki kepentingan dalam sistem atau kualitas yang dimilikinya yang memenuhi kebutuhan tertentu.
2. Mendapatkan pemahaman tentang kebutuhan pelanggan dan pengguna untuk sistem yang direncanakan dan harapan mereka: aktivitas ini sering disebut elisitasi persyaratan.
3. Mengidentifikasi Persyaratan: mendefinisikan kebutuhan bisnis atau persyaratan bisnis merupakan kegiatan penting karena faktor kunci dalam keberhasilan suatu sistem adalah sejauh mana sistem itu mendukung kebutuhan bisnis dan memfasilitasi organisasi dalam mencapainya.
4. Mengklarifikasi dan menyatakan kembali persyaratan: hal ini dilakukan untuk memastikan bahwa persyaratan tersebut menggambarkan kebutuhan nyata pelanggan dan dalam bentuk yang dapat dipahami dan digunakan oleh pengembang sistem.

5. Menganalisis Persyaratan: Hal ini dilakukan untuk memastikan bahwa persyaratan tersebut didefinisikan dengan baik dan sesuai dengan kriteria persyaratan yang baik.
6. Definisi Persyaratan dengan cara yang memiliki arti yang sama bagi semua pemangku kepentingan: perhatikan bahwa setiap kelompok pemangku kepentingan mungkin memiliki perspektif yang sangat berbeda dari sistem dan persyaratan sistem.
7. Menentukan Persyaratan: hal ini diperlukan, termasuk semua detail yang tepat dari setiap persyaratan sehingga dapat dimasukkan kedalam dokumen spesifikasi atau dokumen lainnya, tergantung ukuran proyek.
8. Memprioritaskan persyaratan: semua persyaratan tidak sama pentingnya bagi pelanggan dan pengguna sistem yang direncanakan. Ada yang kritis, ada yang berprioritas relatif tinggi, ada yang berprioritas normal atau rata-rata dan bahkan ada juga yang memiliki prioritas lebih rendah. Memprioritaskan persyaratan begitu penting karena tidak pernah ada cukup waktu ataupun uang untuk melakukan semua hal dalam sistem yang dikembangkan. Memprioritaskan membantu memastikan bahwa jumlah investasi yang tepat dilakukan dalam memenuhi berbagai kebutuhan pelanggan.
9. Persyaratan turunan: ada beberapa persyaratan yang muncul karena desain sistem baru, tetapi tidak memberikan manfaat langsung kepada *end user*.
10. Persyaratan Partisi: mengkategorikan persyaratan sebagai persyaratan yang dapat dipenuhi oleh perangkat keras, perangkat lunak, pelatihan dan dokumentasi.
11. Mengalokasikan persyaratan: mengalokasikan persyaratan ke berbagai subsistem dan komponen sistem.

12. Pelacakan persyaratan: kemampuan untuk melacak setiap persyaratan yang dipenuhi atau tidak, sehingga bisa diverifikasi setiap persyaratan sedang ditangani. Hal ini paling sering dilakukan menggunakan alat persyaratan yang otomatis.
13. Mengelola Persyaratan: hal ini dilakukan untuk dapat menambah, menghapus dan memodifikasi persyaratan selama fase desain sistem, pengembangan, integrasi, pengujian, penerapan, dan operasi. Repositori persyaratan terdiri dari satu set artefak dan database.
14. Pengujian dan verifikasi: proses pemeriksaan persyaratan, desain, kode, rencana pengujian dan produk sistem untuk memastikan bahwa persyaratan terpenuhi.
15. Memvalidasi persyaratan: ini merupakan proses untuk mengonfirmasi bahwa persyaratan nyata diimplementasikan dalam sistem yang di *deliver*. Validasi urutan persyaratan harus diprioritaskan karena dana yang tersedia terbatas.

## DAFTAR PUSTAKA

- Aurum, A. and Wohlin, Claes. 2005. *Engineering and managing software requirements*. Springer.
- Dick, J., Hull, E. and Jackson, K. 2017. *Requirements engineering*. Springer International Publishing. Available at: <https://doi.org/10.1007/978-3-319-61073-3>.
- Koelsch, G. (2016) *Requirements Writing for System Engineering*. Apress. Available at: <https://doi.org/10.1007/978-1-4842-2099-3>.
- Pandey, D., Suman, U. and Ramani, A.K. 2010. 'An effective requirement engineering process model for software development and requirements management', *Proceedings - 2nd International Conference on Advances in Recent Technologies in Communication and Computing, ARTCom 2010*, pp. 287–291. Available at: <https://doi.org/10.1109/ARTCOM.2010.24>.
- Rehman, T. ur, Khan, M.N.A. and Riaz, N. 2013. 'Analysis of Requirement Engineering Processes, Tools/Techniques and Methodologies', *International Journal of Information Technology and Computer Science*, 5(3), pp. 40–48. Available at: <https://doi.org/10.5815/ijitcs.2013.03.05>.
- Sommerville, I. 2016. *Software engineering (10th edition)*, Pearson Education Limited.
- Wieggers, K. and Beatty, J. 2013. *Software Requirements, Third Edition*. Microsoft Press.
- Young, R.R. 2004. *The Requirements Engineering Handbook*. Boston: Artech House.



