

REKAYASA PERANGKAT LUNAK



Tim Penulis:

Siti Maesarah | Dedy Iskandar | Meri Mayang Sari
Erna Astriyani | Martono | Norbertus Tri Suswanto Saptadi
Fifit Alfiah | Mujibbur Rohman | Arif Muhamad Nurdin
Nur Azizah | Ade Setiadi | Rio Wirawan | Oleh Soleh
Euis Nur Fitriani Dewi | Saryani

Editor : Ita Erliyani



REKAYASA PERANGKAT LUNAK

**Siti Maesaroh
Dedy Iskandar
Meri Mayang Sari
Erna Astriyani
Martono
Norbertus Tri Suswanto Saptadi
Fifit Alfiah
Mujibbur Rohman
Arif Muhamad Nurdin
Nur Azizah
Ade Setiadi
Rio Wirawan
Oleh Soleh
Euis Nur Fitriani Dewi
Saryani**



REKAYASA PERANGKAT LUNAK

Tim Penulis:

Siti Maesaroh
Dedy Iskandar
Meri Mayang Sari
Erna Astriyani
Martono
Norbertus Tri Suswanto Saptadi
Fifit Alfiah
Mujibbur Rohman
Arif Muhamad Nurdin
Nur Azizah
Ade Setiadi
Rio Wirawan
Oleh Soleh
Euis Nur Fitriani Dewi
Saryani

Editor : Ita Erliyani, S.Kom., M.M., C.Ed.
Tata Letak : Asep Nugraha, S.Hum.
Desain Cover : Septimike Yourintan Mutiara, S.Gz.
Ukuran : UNESCO 15,5 x 23 cm
Halaman : viii, 240
ISBN : 978-623-8385-50-8
Terbit Pada : Maret 2024
Anggota IKAPI : No. 073/BANTEN/2023

Hak Cipta 2024 @ Sada Kurnia Pustaka dan Penulis

Hak cipta dilindungi undang-undang dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara apapun tanpa izin tertulis dari penerbit dan penulis.

PENERBIT PT SADA KURNIA PUSTAKA

Jl. Warung Selikur Km.6 Sukajaya – Carenang, Kab. Serang-Banten
Email : sadapenerbit@gmail.com
Website : sadapenerbit.com & repository.sadapenerbit.com
Telpon/WA : +62 838 1281 8431

KATA PENGANTAR

Segala puji dan rasa syukur, Tim Penulis panjatkan kehadirat Tuhan Yang Maha Esa yang telah memberikan rahmat dan karunia yang berlimpah tiada henti sehingga tim penulis dapat menyelesaikan buku dengan judul **Rekayasa Perangkat Lunak**. Buku ini merupakan kolaborasi dari berbagai Penulis di bidang IT dari berbagai perguruan tinggi ternama di Indonesia.

Rekayasa Perangkat Lunak (*Software Engineering*) adalah disiplin ilmu yang berkaitan dengan pengembangan, perancangan, pemeliharaan, dan pengelolaan perangkat lunak. Fokus utamanya adalah menciptakan solusi perangkat lunak yang efisien, handal, dan sesuai dengan kebutuhan pengguna. Rekayasa Perangkat Lunak terus berkembang seiring dengan kemajuan teknologi dan perubahan kebutuhan industri. Pemahaman yang mendalam tentang prinsip-prinsip ini menjadi kunci kesuksesan dalam menciptakan solusi perangkat lunak yang inovatif dan berkualitas tinggi. Dengan pengalaman praktis dan wawasan mendalam, penulis membahas tantangan umum yang dihadapi dalam pengembangan perangkat lunak dan memberikan strategi yang terbukti untuk mengatasinya. Selain itu, buku ini mencakup tren terkini dalam industri rekayasa perangkat lunak, seperti pengembangan berbasis cloud, kecerdasan buatan, dan metode pengujian otomatis.

Dengan hadirnya buku ini menjadi panduan esensial bagi para profesional, mahasiswa, dan praktisi Rekayasa Perangkat Lunak, memberikan wawasan mendalam tentang pendekatan sistematis dalam pengembangan perangkat lunak. Penulis, seorang ahli dalam industri IT dengan pengalaman panjang, membimbing pembaca melalui berbagai tahapan siklus hidup pengembangan perangkat lunak dengan penekanan pada praktik terbaik dan metodologi mutakhir.

Kami sebagai Penulis merasa sangat bangga dan berbahagia dengan penerbitan buku ini, kami berharap buku ini dapat

memberikan manfaat dan inspirasi bagi banyak orang. Buku ini perlu untuk dimiliki, dijadikan referensi dan sebagai media pembelajaran dan penambah ilmu pengetahuan bagi para pembacanya. Namun demikian kami tetap berharap mendapatkan Masukan, Saran dan Kritik Membangun untuk Kesempurnaan buku ini. Akhir kata, kami semua mengucapkan Terimakasih kepada semua pihak yang telah membantu dan mengapresiasi buku kami

Tim Penulis

DAFTAR ISI

KATA PENGANTAR	iii
DAFTAR ISI	v
BAB 1 REKAYASA PERANGKAT LUNAK.....	1
Definisi Rekayasa Perangkat Lunak.....	1
Tujuan Rekayasa Perangkat Lunak.....	2
Daftar Pustaka.....	12
Profil Penulis.....	13
BAB 2 PENGENALAN PERANGKAT LUNAK	14
Pengertian Rekayasa Perangkat Lunak.....	14
Tujuan Rekayasa Perangkat Lunak.....	15
Ruang Lingkup Rekayasa Perangkat Lunak.....	15
Rekayasa Perangkat Lunak dan Disiplin Ilmu Lain	20
Profesi dan Sertifikasi.....	21
Rekayasa Perangkat Lunak dan Pemecahan Masalah	22
Metode Rekayasa Perangkat Lunak.....	22
Model Rekayasa Perangkat Lunak.....	23
<i>Prototype</i> yang Berupa Model Lengkap Dari Perangkat Lunak	26
Tahap-Tahap Rekayasa Perangkat Lunak.....	30
Daftar Pustaka.....	35
Profil Penulis.....	36
BAB 3 EVOLUSI PERANGKAT LUNAK	37
Pendahuluan	37
Proses Evolusi	40
Pemeliharaan Perangkat Lunak.....	45
Daftar Pustaka.....	61
Profil Penulis.....	63
BAB 4 PROSES PERANGKAT LUNAK.....	65
Pendahuluan	65
Proses Perangkat Lunak.....	65
Model Proses Pengembangan Perangkat Lunak.....	68
Pengujian Perangkat Lunak.....	77
Komponen Metodologi Pengembangan Aplikasi.....	79
Implementasi Perangkat Lunak.....	80
Tantangan dan Inovasi.....	82

Daftar Pustaka.....	82
Profil Penulis.....	85
BAB 5 PERENCANAAN PERANGKAT LUNAK.....	86
Pendahuluan.....	86
Tahapan Perencanaan.....	87
Penyebab Kegagalan Proses.....	91
Penutup.....	95
Daftar Pustaka.....	95
Profil Penulis.....	96
BAB 6 DASAR ANALISIS KEBUTUHAN.....	97
Pendahuluan.....	97
Persyaratan Kebutuhan.....	101
Analisis Kebutuhan.....	105
Kebutuhan Perangkat Lunak.....	107
Tantangan Penggunaan Perangkat Lunak.....	108
Penutup.....	110
Daftar Pustaka.....	112
Profil Penulis.....	113
BAB 7 MODEL PROSES PENGEMBANGAN PERANGKAT LUNAK	114
.....	
Pendahuluan.....	114
<i>Agile Software Development Model</i>	114
<i>Fase Agile Software Development</i>	115
<i>Karakteristik Agile Software Development</i>	117
<i>Extreme Programming Software Development</i>	117
<i>Karakteristik Extreme Programming</i>	118
Tahap Metodologi <i>Extreme Programming</i>	119
<i>Scrum Software Development Model</i>	120
Cara Kerja <i>Scrum</i>	121
Keuntungan dan Kerugian dari Metodologi <i>Scrum</i>	122
<i>Rapid Application Development Model</i>	123
Fase Pengembangan Aplikasi Cepat.....	124
Daftar Pustaka.....	126
Profil Penulis.....	128
BAB 8 KEBUTUHAN PERANGKAT LUNAK.....	129
Pendahuluan.....	129
Jenis-jenis Kebutuhan Perangkat Lunak.....	129

Identifikasi Kebutuhan Perangkat Lunak	131
Pengumpulan Dokumentasi dan Validasi Kebutuhan Perangkat Lunak	133
Tantangan dan Solusi Dalam Mengelola Kebutuhan Perangkat Lunak	134
Hubungan <i>System Request</i> dan Kebutuhan Perangkat Lunak	135
Hubungan Kebutuhan Perangkat Lunak dan Keberhasilan Pengembangan Perangkat Lunak	137
Daftar Pustaka	138
Profil Penulis	139
BAB 9 ANALISA DAN PERANCANGAN BERORIENTASI OBJEK..	140
Pendahuluan	140
Pengertian Analisa dan Perancangan Berorientasi Objek	141
Konsep Dasar Metodologi Berorientasi Objek.....	141
Dasar-dasar Metodologi Berorientasi Objek	142
Beberapa Bahasa Pemrograman Berorientasi Objek.....	148
Kelebihan Perancangan Berorientasi Objek.....	148
Perbandingan Antara Perancangan Berorientasi Objek dengan Terstruktur	149
Daftar Pustaka.....	150
Profil Penulis.....	151
BAB 10 PEMODELAN SISTEM DENGAN UML.....	152
Pendahuluan	152
Pemodelan (Model)	153
Membangun Model	155
<i>Unified Modeling Language (UML)</i>	156
Konsep Diagram dalam Memodelkan Sistem.....	159
Daftar Pustaka.....	162
Profil Penulis.....	163
BAB 11 PERANCANGAN PIRANTI PERANGKAT LUNAK.....	164
Pendahuluan	164
Model dan Proses Perancangan.....	165
Analisis Piranti Perangkat Lunak.....	166
Desain Piranti Perangkat Lunak.....	168
Bahasa Program Piranti Perangkat Lunak	172
Manfaat Perancangan Piranti Perangkat Lunak	174
Daftar Pustaka.....	175

Profil Penulis.....	176
BAB 12 TEKNIK KEAMANAN REKAYASA PERANGKAT LUNAK	177
Pendahuluan	177
Tujuan	177
Dasar-dasar Keamanan Perangkat Lunak.....	178
Analisis Kerentanan.....	180
Pengembangan Perangkat Lunak yang Aman	184
Daftar Pustaka.....	188
Profil Penulis.....	191
BAB 13 PENGUJIAN PERANGKAT LUNAK.....	192
Tinjauan Pengujian Perangkat Lunak.....	192
Mengapa Pengujian Perangkat Lunak Penting.....	193
Peran Pengujian dalam Rekayasa Perangkat Lunak	194
Jenis Pengujian Perangkat Lunak.....	194
Siklus Hidup Pengujian Perangkat Lunak.....	205
Daftar Pustaka.....	212
Profil Penulis.....	213
BAB 14 IMPLEMENTASI PERANGKAT LUNAK.....	214
Pengertian Perangkat Lunak.....	214
Implementasi Perangkat Lunak.....	214
Pemrograman Fungsional	219
Gaya Pemrograman	220
Pedoman Pengkodean.....	220
<i>Software Documentation</i>	221
<i>Software Implementation Challenges</i>	222
Daftar Pustaka.....	223
Profil Penulis.....	224
BAB 15 PEMELIHARAAN PERANGKAT LUNAK.....	225
Pendahuluan	225
Jenis Pemeliharaan Perangkat Lunak	227
Strategi Pemeliharaan Sistem.....	231
Evolusi Perangkat Lunak.....	234
Daftar Pustaka.....	239
Profil Penulis.....	240

BAB 1

REKAYASA PERANGKAT

LUNAK

Siti Maesaroh, S.Kom., M.T.I.
Universitas Mercu Buana

Definisi Rekayasa Perangkat Lunak

Rekayasa Perangkat Lunak (*Software Engineering*) adalah disiplin ilmu yang berkaitan dengan pengembangan, perancangan, pemeliharaan, pengujian, dan manajemen kualitas perangkat lunak. Tujuan utamanya adalah untuk menciptakan perangkat lunak yang efisien, andal, mudah dimengerti, dan dapat diubah atau diperbaiki dengan mudah.

Beberapa konsep utama dalam rekayasa perangkat lunak melibatkan proses pengembangan perangkat lunak, manajemen proyek, pemodelan perangkat lunak, pengujian perangkat lunak, dan pemeliharaan perangkat lunak. Rekayasa perangkat lunak juga mencakup prinsip-prinsip seperti penggunaan metode formal, dokumentasi yang baik, dan penggunaan alat bantu untuk meningkatkan efisiensi dan kualitas pengembangan perangkat lunak.

Dengan adanya rekayasa perangkat lunak, diharapkan perangkat lunak yang dihasilkan dapat memenuhi kebutuhan pengguna, beroperasi secara efisien, dan memiliki daya tahan terhadap perubahan kebutuhan atau lingkungan. Proses pengembangan perangkat lunak yang terstruktur dan disiplin menjadi kunci untuk mencapai tujuan tersebut.

diandalkan) Perangkat lunak harus dapat dipercaya, *Efficiency* (Daya Guna) Efisien dalam penggunaan sumber daya sistem (memori, *hardware*, listrik, dll), *Acceptability* (Dapat diterima) Perangkat Lunak harus diterima oleh pengguna sebagaimana dirancang. Ini berarti harus dapat dimengerti, dapat digunakan dan kompatibel dengan sistem lain. Tantangan utama yang dihadapi RPL adalah:

- a. *Heterogeneity* (Keberagaman), mengembangkan teknik membangun perangkat lunak yang dapat mengatasi perbedaan *platform* dan lingkungan eksekusi.
- b. *Delivery* (Pengiriman), mengembangkan teknik yang menghasilkan pengiriman perangkat lunak lebih cepat.
- c. *Trust* (Kepercayaan), mengembangkan teknik yang menunjukkan bahwa perangkat lunak dapat dipercaya oleh para penggunanya.

Daftar Pustaka

- Kadir dan Triwahyuni. (2013). *Pengantar Teknologi Informasi*. Yogyakarta: Andi.
- Maxxor. (2018, 07 12). *SCRUM Software Development Process*. Retrieved from www.maxxor.com: <https://www.maxxor.com/software-development-process>.
- Pressman, R.S. (2012). *Rekayasa Perangkat Lunak: Pendekatan Praktisi*. Yogyakarta: Andi.
- Pressman, Roger S., (2001). *Software Engineering A Practitioner's Approach 7th ed*. McGraw-Hill, New York.
- Sukamto, Rosa A., & Shalahuddin, M. (2016). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Object*. Bandung: Informatika. 13.
- Uml-diagrams.org. (2018, 07 20). *State Machine Diagrams*. Retrieved from uml-diagrams.org: <https://www.uml-diagrams.org/state-machine-diagrams.html>.

PROFIL PENULIS



Siti Maesaroh, S.Kom., M.T.I.

Ketertarikan penulis terhadap ilmu komputer dimulai pada tahun 2010 silam. Hal tersebut membuat penulis memilih untuk masuk ke Sekolah Perguruan Tinggi Di bidang Sistem Informasi. Setelah lulus S1, 2 tahun kemudian, penulis menyelesaikan studi S2 di prodi Teknik Informatika Program Pasca Sarjana Universitas Raharja.

Penulis memiliki kepakaran dibidang *Web Technology dan Data Science*. Dan untuk mewujudkan karir sebagai dosen profesional, penulis pun aktif sebagai peneliti dibidang kepakarannya tersebut. Beberapa penelitian yang telah dilakukan didanai oleh internal perguruan tinggi dan juga Kemenristek DIKTI. Selain peneliti, penulis juga aktif menulis buku dengan harapan dapat memberikan kontribusi positif bagi bangsa dan negara yang sangat tercinta ini. Dengan tujuan mengembangkan solusi perangkat lunak inovatif yang memecahkan tantangan bisnis dengan menggunakan keterampilan pemrograman dan pengetahuan teknologi terkini. Terus meningkatkan kemampuan teknis dan kepemimpinan melalui pengalaman praktis dan pembelajaran berkelanjutan.

Email Penulis: mays41946@gmail.com.

BAB 2

PENGENALAN PERANGKAT LUNAK

Dedy Iskandar, S.Kom., M.T.I.
Universitas Raharja Tangerang

Pengertian Rekayasa Perangkat Lunak

Istilah rekayasa perangkat lunak (RPL) secara umum merupakan terjemahan dari istilah *Software Engineering*. Istilah *Software Engineering* dipopulerkan pada *Software Engineering Conference* diselenggarakan oleh NATO tahun 1968. Sebagian orang mengartikan RPL hanya sebatas penulisan program komputer. Meskipun ada perbedaan mendasar antara perangkat lunak dan program komputer.

Perangkat lunak mengacu pada semua perintah yang digunakan untuk memproses informasi. Perangkat lunak dapat berupa program atau prosedur. Program adalah kumpulan perintah yang dapat dipahami oleh komputer, dan prosedur adalah perintah yang dibutuhkan pengguna untuk memproses informasi (O'Brien, 1999). Pengertian RPL sendiri adalah sebagai berikut:

Bidang yang menangani semua aspek produksi perangkat lunak mulai dari tahap awal, yaitu menganalisis kebutuhan pengguna dan menentukan spesifikasi kebutuhan pengguna. disain,

Jelas bahwa RPL tidak hanya berkaitan dengan bagaimana program komputer ditulis. Yang dimaksud dengan “seluruh aspek produksi” dalam pengertian di atas adalah segala sesuatu yang berkaitan dengan proses produksi merupakan bagian dari RPL,

termasuk manajemen proyek, kepegawaian, anggaran, metodologi, jadwal, kualitas, dan pelatihan pengguna.

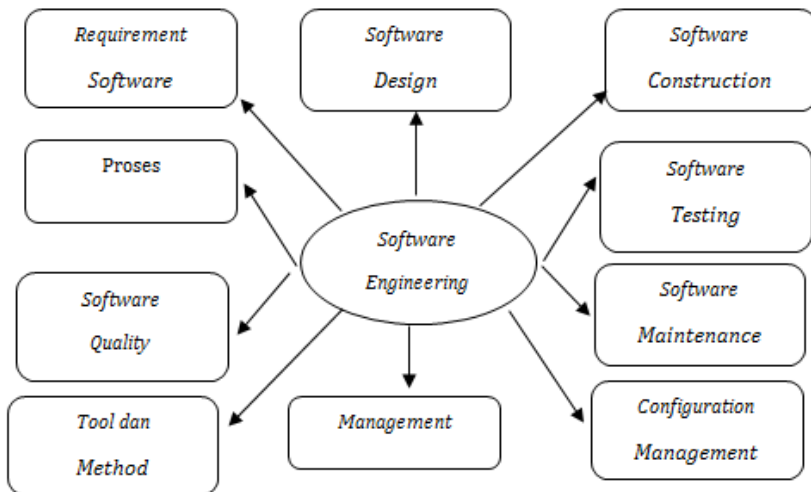
Tujuan Rekayasa Perangkat Lunak

Secara umum tujuan RPL tidak berbeda dengan tujuan disiplin ilmu teknik lainnya. Lebih spesifiknya, tujuan RPL adalah:

1. Membangun perangkat lunak yang kuat, andal, dan tepat waktu.
2. Membangun perangkat lunak dengan biaya pemeliharaan yang rendah.
3. Memperoleh biaya produksi perangkat lunak yang rendah.
4. Membangun perangkat lunak yang berjalan pada berbagai jenis platform.

Ruang Lingkup Rekayasa Perangkat Lunak

Berdasarkan definisi diatas, ruang lingkup RPL dapat digambarkan sebagai berikut:



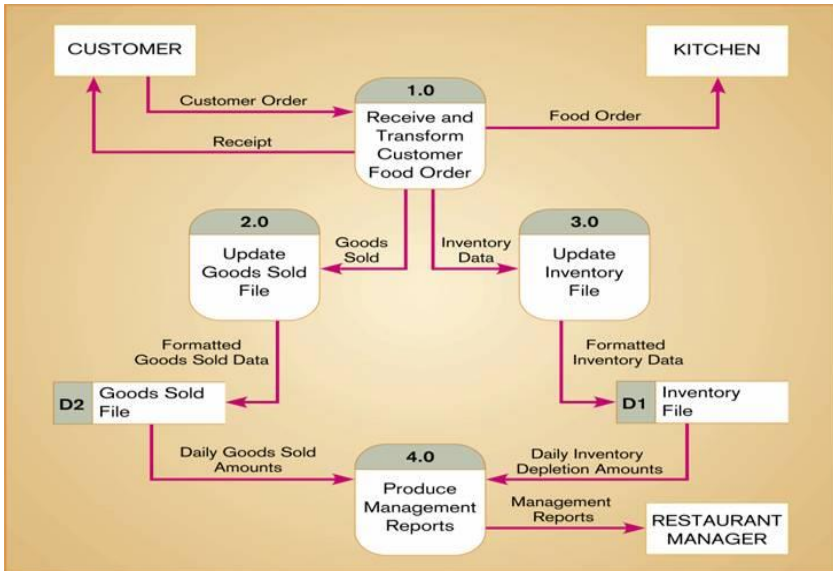
Gambar 2.1: Ruang Lingkup RPL

Sumber: Ahmad Hasyim, 2018.

1. *Software Requirements*

Berkaitan dengan persyaratan perangkat lunak dan spesifikasi kebutuhan.

Langkah selanjutnya adalah menguraikan konteks diagram pada DFD level 0. DFD level 0 merupakan DFD yang merepresentasikan proses, aliran data, dan penyimpanan data utama dalam sistem. DFD level 0 digunakan sebagai dasar untuk membangun DFD level bawah (level 1, 2, 3, dst). Di bawah 19 adalah gambar DFD level 0.



Gambar 2.15: DFD level 0

Sumber: Hoffer et al., 2002.

Daftar Pustaka

- Ahmad Hasyim, (2018). *Makalah Rekayasa Perangkat Lunak*. Universitas Pendidikan Indonesia.
- Nugroho, Adi. (2010). *Rekayasa Perangkat Lunak Berorientasi Objek dengan Metode USDP (Unified Software Development Process)*. Yogyakarta: Andi.
- Pressman, Roger S. (2001). *Software Engineering: A Practitioner's Approach, 5th Edition*. McGraw-Hill International.
- Pressman, Roger S. (2010). *Rekayasa Perangkat Lunak*. Yogyakarta: Andi.
- Sommerville, Ian. (2003). *Software Engineering, 7th Addison Wesley Publishing Company*.
- Tavri D. Mahyuzir. (1997). *Pengantar Rekayasa Perangkat Lunak*, Elexmedia Komputindo,

PROFIL PENULIS



Dedy Iskandar, S.Kom., M.T.I.

Penulis Lulus S1 Program Studi Sistem Informasi di Sekolah Tinggi Manajemen dan Ilmu Komputer (STMIK) Bina Darma Palembang dan menyelesaikan Program Studi Magister Teknik Informatika Sekolah Tinggi Manajemen dan Ilmu Komputer (STMIK) Raharja Tangerang. Saat ini sebagai Dosen Fakultas Sains dan Teknologi Universitas Raharja. Saya juga masih mengajar di SMK Pustek Serpong, dan juga sebagai *Tutor Online* pada Universitas Terbuka. Penulis mengampu matakuliah Rekayasa Sistem, Logika Algoritma dan Pemograman, Sistem Basis Data, Organisasi Komputer, Jaringan Komputer, Struktur Data. Aktif sebagai penulis di beberapa Jurnal nasional yang terindeks Sinta 3 dan Sinta 4. Sekarang saya juga aktif di beberapa organisasi seperti Partai Golkar sebagai Ketua Ranting dan Kosgoro sebagai Ketua Pelatihan dan Pengembangan.

Email Penulis: iskandar@raharja.info.

BAB 3

EVOLUSI PERANGKAT LUNAK

Meri Mayang Sari, S.Kom.
Universitas Raharja

Pendahuluan

Pengembangan perangkat lunak tidak berakhir pada penyerahan sistem, namun berlanjut sepanjang umur sistem. Setelah suatu sistem diimplementasikan, sistem tersebut harus dimodifikasi agar tetap berguna. Perubahan dalam bisnis dan perubahan ekspektasi pengguna menciptakan persyaratan baru untuk perangkat lunak yang ada. Mungkin perlu untuk memodifikasi bagian-bagian Perangkat Lunak untuk memperbaiki kesalahan yang ditemukan selama pengoperasian, beradaptasi dengan perubahan pada platform perangkat keras dan perangkat lunak, dan meningkatkan kinerja atau karakteristik non-fungsional lainnya.

Evolusi perangkat lunak penting karena perusahaan telah menginvestasikan sejumlah besar uang pada perangkat lunak dan sekarang sepenuhnya bergantung pada sistem tersebut. Sistem adalah aset bisnis yang penting, dan mempertahankan nilai aset ini memerlukan investasi dalam perubahan sistem. Akibatnya, sebagian besar perusahaan besar menghabiskan lebih banyak uang untuk memelihara sistem yang ada dibandingkan mengembangkan sistem baru. Berdasarkan survei industri informal, Erlich (2000) menemukan bahwa 85-90% biaya perangkat lunak perusahaan adalah biaya pengembangan. Studi lain menemukan bahwa sekitar dua pertiga dari biaya perangkat lunak adalah biaya pengembangan. Tentu saja, biaya perubahan perangkat lunak merupakan bagian besar dari anggaran TI perusahaan mana pun.

bagian program yang bergantung yang harus diubah untuk mengimplementasikan pemfaktoran ulang.

Refactoring, yang dilakukan selama pengembangan program, merupakan cara efektif untuk mengurangi biaya pemeliharaan jangka panjang suatu program. Namun, jika Anda mengambil alih program untuk pemeliharaan yang strukturnya telah terdegradasi secara signifikan, maka hampir tidak mungkin untuk melakukan refaktorisasi kode saja. Anda mungkin juga harus memikirkan tentang pemfaktoran ulang desain, yang mungkin merupakan masalah yang lebih mahal dan sulit. Pemfaktoran ulang desain melibatkan identifikasi pola desain yang relevan dan mengganti kode yang ada dengan kode yang mengimplementasikan pola desain ini (Kerievsky, 2004).

Daftar Pustaka

- Arthur, L. J. (1988). *Software Evolution*. New York: John Wiley & Sons.
- Banker, R. D., Datar, S. M., Kemerer, C. F. and Zweig, D. (1993). Software Complexity and Maintenance Costs. *Comm. ACM*, 36 (11), 81–94.
- Boehm, B. W., Abts, C., Brown, A. W., Chulani, S., Clark, B. K., Horowitz, E., Madachy, R., Reifer, D. and Steece, B. (2000). *Software Cost Estimation with COCOMO II*. Upper Saddle River, NJ: Prentice Hall.
- Coleman, D., Ash, D., Lowther, B. and Oman, P. (1994). *Using Metrics to Evaluate Software System Maintainability*. *IEEE Computer*, 27 (8), 44–49.
- Erlikh, L. (2000). *Leveraging Legacy System Dollars For E-business*. *IT Professional*, 2 (3), May/June 2000, 17–23.
- Fowler, M., Beck, K., Brant, J., Opdyke, W. and Roberts, D. (1999). *Refactoring: Improving the Design of Existing Code*. Boston: Addison-Wesley.
- Guimaraes, T. (1983). *Managing Application Program Maintenance Expenditures*. *Comm. ACM*, 26 (10), 739–46.
- Hopkins, R. and Jenkins, K. (2008). *Eating the IT Elephant: Moving from Greenfield Development to Brownfield*. Boston: IBM Press.
- Kafura, D. and Reddy, G. R. (1987). The Use Of Software Complexity Metrics In Software Maintenance. *IEEE Trans. on Software Engineering*, SE-13 (3), 335–43.
- Kerievsky, J. (2004). *Refactoring to Patterns*. Boston: Addison Wesley.
- Kozlov, D., Koskinen, J., Sakkinen, M. and Markkula, J. (2008). *Assessing Maintainability Change Over Multiple Software Releases*. *J. of Software Maintenance and Evolution*, 20 (1), 31–58.
- Krogstie, J., Jahr, A. and Sjoberg, D. I. K. (2005). *A Longitudinal Study Of Development And Maintenance In Norway: Report From The 2003 Investigation*. *Information and Software Technology*, 48 (11), 993–1005.
- Lientz, B. P. and Swanson, E. B. (1980). *Software Maintenance Management*. Reading, Mass: Addison-Wesley.
- McCabe, T. J. (1976). A Complexity Measure. *IEEE Trans. on Software Engineering*, SE-2 (4), 308–20.

- Nosek, J. T. and Palvia, P. (1990). *Software Maintenance Management: Changes In The Last Decade*. *Software Maintenance: Research and Practice*, 2 (3), 157–74.
- Opdyke, W. F. and Johnson, R. E. (1990). Refactoring: An Aid in Designing Application Frameworks and Evolving Object-Oriented Systems. 1990 Symposium on Object-Oriented Programming Emphasizing Practical Applications (SOOPPA '90), Poughkeepsie, New York.
- Poole, C. and Huisman, J. W. (2001). Using Extreme Programming in a Maintenance Environment. *IEEE Software*, 18 (6), 42–50.
- Rajlich, V. T. and Bennett, K. H. (2000). A Staged Model for the Software Life Cycle. *IEEE Computer*, 33 (7), 66–71.
- Sousa, M. J. (1998). A Survey on the Software Maintenance Process. *14th IEEE International Conference on Software Maintenance (ICSM '98)*, Washington, D.C.: 265–74.
- Ulrich, W. M. (1990). The Evolutionary Growth of Software Reengineering and the Decade Ahead. *American Programmer*, 3 (10), 14–20.

PROFIL PENULIS



Meri Mayang Sari, S.Kom.

Penulis Lulus Diploma (D3) Program Studi Komputerisasi Akuntansi di AMIK Raharja Informatika Tangerang pada tahun 2015 dan menyelesaikan program studi sarjana (S1) Program Studi Sistem Informasi di Universitas Raharja Tangerang pada tahun 2021 dan saat ini sedang menempuh Pendidikan S2 di Fakultas Sains dan Teknologi Universitas Raharja Tangerang Program Studi Magister Teknik Informatika. Pengalaman kerja penulis, Staff Ketua Program Studi FEB & FST pada Universitas Raharja sejak tahun 2015-sekarang. Capaian publikasi jurnal penulis, sudah lebih dari 19 Jurnal Terindeks Sinta, dan 1 Jurnal Terindeks IEEE Proceeding Scopus. Berikut ini daftar capaian publikasinya:

1. *Frozen Meat Sales Website Design Based on Laravel Framework, CCIT Journal, Vol 16 No 2 (2023).*
2. Pengaruh Gaya Kepemimpinan Dan Motivasi Terhadap Produktivitas Kerja Karyawan Pada PT Arthawenasakti Gemilang Tangerang, *Jurnal Ilmiah Manajemen Ubhara* 5 (1), 20-28, 2023.
3. *Application of RUP (Rational Unified Process) Method in Optimizing Payment Systems and Processing Student Savings Data, CCIT Journal, Vol 16 No 1 (2023).*
4. Rancang Bangun Sistem Informasi Pemasaran Sebagai Penunjang Keputusan Berbasis Web, *Journal CERITA*, 8(2), 2023.
5. Pemanfaatan Aplikasi SLiMS (Senayan *Library Management System*) pada Perpustakaan Universitas Raharja, *Journal CICES*, 9(1), 102-112, 2023.
6. Sistem Informasi Monitoring Pembayaran Piutang Penjualan Berbasis Web, *Indonesian Journal Accounting (IJAcc)*, IJAcc 4(1) 5-13, 2023.
7. Pemanfaatan Media Informasi Dengan Menggunakan Design Souvenir Untuk Mendukung Promosi Pada PT Federal International Finance, Vol 3 NO 1 (2023): URNITY (*Universal Raharja Community*).

8. *Implementation Of Active Learning Using Card Game Model In Increasing Learning Motivation, 2022 IEEE Creative Communication and Innovative Technology (ICCIT)*, 1-5. Terindeks *Proceeding. Scopus*.
9. Analisis Perilaku Pengelolaan Keuangan UMKM Selama Pandemi COVID-19, *Indonesian Journal Accounting (IJAcc)* 3 (2), 73-82, 2022.
10. Rancang Bangun *Website* Pemasaran dan Penjualan Properti (Study Kasus: PT Indah Cemani Raya), *Journal CERITA*, 8(2), 171-176, 2022.
11. Penggunaan Media Digital (Website) Dalam Pengolahan Data Cuti Karyawan, *Technomedia Journal* 7 (1 Juni), 126-135, 2022.
12. Pemanfaatan Framework Codeigniter Dalam Pengembangan Aplikasi *Reporting Research*, *Journal CERITA*, 8(1), 76-85, 2022.
13. Meningkatkan Kedisiplinan Jadwal Kehadiran Guru Menggunakan Sistem Informasi Berbasis Web, *Journal CICES*, 8(2), 2022.
14. Rancang Bangun Sistem Informasi Laporan Penjualan Heart Snack Pada PT. Flavors International, *Journal CERITA*, 7(2), 163-171, 2021.
15. Peran Pengaruh Komite Dan Pengawas Terhadap Pengelolaan Dana Bos Di Sektor Sekolah Dasar Negeri, *Indonesian Journal Accounting (IJAcc)* 2 (2), 175-184, 2021.
16. Analisa Faktur Pajak Yang Tidak Dapat Dikreditkan Terhadap Pajak Pertambahan Nilai Terutang (Studi Kasus PT. Mitra Pemuda Baja Di Tangerang), *Indonesian Journal Accounting (IJAcc)* 2 (1), 1-13, 2021.
17. Perancangan Sistem Informasi Pembayaran SPP Berbasis Web Menggunakan Notifikasi SMS *Gateway* (Studi Kasus: SMP Puspita Tangerang), *Journal CERITA* 6 (1), 106-116, 2020.
18. Rancangan Sistem *Cash flow Statement* Berbasis Web Pada *Clarity Fluency School Of English*, *Journal CICES*, 6(2), 2020.
19. Sistem Informasi Pengendalian Persediaan Barang Menggunakan Metode *Economic Order Quantity* (EOQ), *Indonesian Journal Accounting (IJAcc)* 1 (2), 132-139, 2020.

Email Penulis: meri.mayang@raharja.info.

BAB 4

PROSES PERANGKAT LUNAK

Erna Astriyani, S.Kom., M.T.I.
Universitas Raharja

Pendahuluan

Banyak proses yang digunakan oleh aplikasi untuk memenuhi kebutuhan pengguna. Proses pengembangan aplikasi ini juga dikenal sebagai proses perangkat lunak. Proses perangkat lunak adalah kerangka tindakan, aktivitas, dan tugas yang diperlukan untuk membuat perangkat lunak berkualitas tinggi. (Akil, 2016). Proses perangkat lunak ini tidak hanya berkualitas tinggi, tetapi juga dirancang untuk mengurangi biaya dalam waktu singkat. (Iqbal, 2021).

Proses peningkatan produk adalah rencana yang digunakan untuk mengembangkan item pemrograman yang ditentukan untuk mengembangkan Sistem. Proses ini mencakup hal-hal seperti menerjemahkan kebutuhan pemakai menjadi kebutuhan perangkat lunak, mengubah kebutuhan perangkat lunak menjadi desain, menerapkan desain, dan mengimplementasikan proyek pengembangan sistem sedikit demi sedikit.

Proses Perangkat Lunak

Dalam ilmu komputer, istilah "perangkat lunak" digunakan. Merancang, membuat, mengaplikasikan, dan mendukung atau meningkatkan fungsi perangkat lunak dikenal sebagai perangkat lunak. Dua jenis komponen komputer adalah perangkat lunak dan

Tantangan dan Inovasi

Dalam desain perangkat lunak, terdapat beberapa tantangan dan inovasi, seperti kekurangan sumber daya, kesulitan struktural dalam adaptasi, dan mengadopsi teknologi terbaru, yang pasti akan membutuhkan biaya yang tinggi, dan waktu implementasi yang lama.

1. Tantangan Umum dalam Proses Perangkat Lunak

Tantangan umum ini biasanya terkait pengembangan perangkat lunak dan kompleksitas dalam perangkat lunak, agar dapat bersaing dengan perangkat lunak lainnya.

2. Inovasi Terkini dalam pengembangan Perangkat Lunak

Inovasi terkini dalam pengembangan perangkat lunak biasanya meluncurkan hal-hal baru yang lebih mendalam dari perangkat lunak dengan versi terbaru.

Daftar Pustaka

- Agita, N. (2018). *Apa yang Dimaksud Dengan Model Spiral Dalam Pengembangan Perangkat Lunak?* (2019, September 8).
- Adi Kurniyanti, V., & Murdiani, D. (2022). Perbandingan Model Waterfall Dengan Prototype Pada Pengembangan System Informasi Berbasis Website. *Jurnal Syntax Fusion*, 2(08), 669–675. <https://doi.org/10.54543/fusion.v2i08.210>.
- Akil, I. (2016). Rekayasa Perangkat Lunak Dengan Model Unified Process Studi Kasus: Sistem Informasi Journal. *Jurnal Pilar Nusa Mandiri*, 12(1), 1–11.
- Bolung Maikel & Tampangela Henry Ronald Karunia (2017). Analisa Penggunaan Metodologi Pengembangan Perangkat Lunak: *Journal Eltikom*, 1(1), 1–10.
- Dennis, A, Wixom, B, and Roth, R. (2006). *System Analysis and Design*. John Wiley and Sons, Inc pp. 171-209.

- Febrianto, S., Putra, E. D. P., Rohman, M. A., (2020). *Desain Perangkat Lunak*.
- Gustafson. K. L; Branch. R. M. (2002). *Survey of Instructional Development Models*. Fourth Edition.
- Iqbal, M. (2021). *Software Development*. Telkomuniversity.Ac.Id.
- Ifeyinwa Angela Ajah, John Otozi Ugah. (2013). *Comparative Analysis of Software Development Methodologies*. Volume 3, Issue 6, June.
- Kusuma, W. A., Ramadhani, K. W., & Maulana, A. (2020). Identification of Software Requirements using a Qualitative Study of the Persona User approach (Case Study: The Process of Making a Practicum Module with the Ability of Students to Reduce the Case of Source Code Plagiarism). *Fountain of Informatics Journal*, 5(2), 35.
- Latifatunnisa, H. (2022). *Proses Project Management*.
- Larman. C, Basili. V. R, "Iterative and Incremental Development: A Brief History". (2003). *Computer*, vol. 36, no. 6, pg. 47-56, doi:10.1109/MC.2003.1204375.
- Library.binus.ac.id/eColls/eThesisdoc/Bab1/2014-2-01054-MTIF%20Bab1001.pdf.
- Maikel Bolung, Henry Ronald Karunia Tampangela. (2017). *Analisa Penggunaan Metodologi Pengembangan Perangkat Lunak* Vol. 1 No.1, Juni 2017, Hal 1-10.
- Meliana. (2018, December 6). *Siklus Pengujian Perangkat Lunak*. Socs.Binus.Ac.Id.
- Mihai Liviu Despa. (2014). Comparative study on software development methodologies. *Database Systems Journal*. Vol. V, no. 3.
- Nabil Mohammed Ali Munassar and A. Govardhan. A Comparison Between Five Models Of Software Engineering. (2010). *IJCST International Journal of Computer Science Issues*, Vol. 7, Issue 5, September.
- Pressman, S. Roger. (2005). *Software Engineering: a Practitioner's Approach*. Seventh Edition.
- Sanjana Taya, Shaveta Gupta, (2011). A Comparison Between Five Models Of Software Engineering. *IJCST* Vol. 2, Issue 4, Oct.-Dec.

Setiawan, R. (2021). *Tahap Membuat Software yang Baik*.

Sommerville Ian. (2004). *Software Engineering*. Addison Wesley, 7th Edition.

Susanto Azhar. (2004). Sistem Informasi Manajemen. Bandung: Lingga Jaya. *International Journal of Information Management*, 21 (5), 349-364. 2001.

Suryana, E. (2023, August 25). *Sosialisasi Renstra dan Renop UNIVED Bengkulu*. Unived.Ac.Id.

Syakti, F., & Yani No, J. A. (2019). Metode Pengembangan Perangkat Lunak Berbasis Mobile: A Review. *In Jurnal Bina Komputer JBK* (Vol. 1, Issue 2).

Tantangan Pengembangan Perangkat Lunak/Software. (2017). Se.Ittelkom-Pwt.Ac.Id.

PROFIL PENULIS



Erna Astriyani, S.Kom., M.T.I.

Ketertarikan penulis terhadap ilmu komputer dimulai pada tahun 2010 silam. Hal tersebut membuat penulis memilih untuk melanjutkan pendidikan ke Jenjang Sarjana. Penulis kemudian melanjutkan pendidikan ke Sekolah Tinggi Ilmu Komputer dan berhasil menyelesaikan studi S1 di prodi Sistem Informasi STMIK Raharja pada tahun 2014. Kemudian, penulis menyelesaikan studi S2 di prodi Teknik Informatika Program Pasca Sarjana STMIK Raharja pada tahun 2018. Penulis setelah lulus jenjang Magister Teknik Informatika Penulis mulai dalam bidang pendidikan dengan menjadi Dosen pada sebuah Universitas Raharja di Tangerang. Penulis bekerja sebagai Sekretaris Prodi Sistem Informasi dan juga penulis aktif dalam menulis Penelitian pada sebuah jurnal. Untuk mewujudkan karir sebagai dosen profesional, penulis aktif sebagai peneliti dibidang kepakarannya tersebut.

Email Penulis: erna.astriyani@raharja.info.

BAB 5

PERENCANAAN PERANGKAT LUNAK

Martono, S.Pd.Kim., M.TI.
Universitas Raharja

Pendahuluan

Perencanaan Perangkat Lunak akan mengatur rincian rencana manajemen yang akan diikuti dalam rangka pengembangan perangkat lunak. Ini merupakan bagian dari siklus pengembangan, organisasi, peran spesifik, proyeksi jadwal dan pengujian. Perencanaan Perangkat Lunak akan menyampaikan semua rincian mengenai rencana pembangunan dan siklus pembangunan.

Ini akan menetapkan peran dan tugas masing-masing anggota mengenai pengembangannya, dan bagaimana serta metode apa yang akan mereka gunakan untuk menyelesaikannya tugas. Perencanaan akan menetapkan tenggat waktu agar proyek dapat diselesaikan tepat waktu. Ini akan mengikuti spesifikasi yang disepakati.

Estimasi dan perencanaan sangat penting untuk keberhasilan pengembangan perangkat lunak apa pun proyek dengan ukuran atau konsekuensi apa pun. Rencana membantu kita mengetahui siapa yang perlu terlibat dan untuk mengerjakan suatu proyek selama periode tertentu dan membantu kami mengetahui apakah suatu proyek berada pada jalur yang tepat untuk mewujudkannya. Tanpa perencanaan yang baik akan membuat kita memulai proyek dengan sejumlah masalah yang akan timbul.

Penutup

Dalam melakukan perencanaan perangkat lunak pertanyaan tentang 4 W+H meliputi: *What, Who, When, Why dan How*, tanpa *Where*. Berikut ini penjelasan singkat terkait hal tersebut untuk dapat menjadi acuan. Meliputi:

1. *What* atau apa yang akan dicapai dalam perangkat lunak?
2. *Who* untuk menentukan siapa yang akan terlibat dalam pengembangan perangkat lunak?
3. *Why* atau mengapa perangkat lunak ini perlu dibuat?
4. *When* yang berarti kapan perangkat lunak ini harus selesai?
5. *How* atau bagaimana proses itu bisa tercapai dengan baik dan benar?

Untuk mendapatkan perencanaan yang baik. Kegiatan mengumpulkan semua kebutuhan dan fitur yang diharapkan oleh pengguna diharuskan memiliki gambaran yang jelas. Demikian juga persiapan sumber daya apapun yang dibutuhkan, batas waktu dan tata kelola yang baik menjadi sarat utama keberhasilan perencanaan perangkat lunak.

Daftar Pustaka

Cohn, Mike (2005). *Agile Estimating and Planning*, New Jersey, AS: Pearson Education Inc.

<https://www.softkraft.co/software-development-planning/>.

Pressman, Rogger S (2001). *Software Engineering*, New York, AS: McGraw-Hill Companies, Inc.

Villafiorita, Adolfo (2014). *Introduction to Software Project Management*, Boca Raton, AS: Taylor & Francis Group, LLC.

PROFIL PENULIS



Martono, S.Pd.Kim, M.TI.

Ketertarikan penulis pada bidang komputer awalnya hanya karena hobi dan iseng-iseng. Sambil menyelesaikan Pendidikan Kimia jenjang Diploma 3 di IKIP Jakarta pada saat itu penulis juga mengikuti pelatihan elektronika komputer. Belajar tentang jaringan komputer dimulai dari era *Novell Netware* dan *Microsoft NT 4.0* dan saat jaringan LAN yang masih menggunakan *Cable Coaxial* hingga kini era jaringan menggunakan wifi dan *fiber Optics* karena tuntutan pekerjaan.

Pendidikan D3 diselesaikan di IKIP Jakarta dan S1 diselesaikan di Universitas Terbuka pada tahun 2006 masih pada Jurusan yang sama yakni Pendidikan Kimia. Penulis kemudian melanjutkan Pendidikan S2 di STMIK Raharja (sekarang sudah jadi Universitas Raharja) pada program studi Teknik Informatika dan selesai pada tahun 2017. Awalnya penulis bekerja di Perusahaan yang merupakan vendor di bidang Teknologi Informasi yang melayani berbagai organisasi pendidikan dan perusahaan.

Kemudian penulis berlanjut bergabung dengan perusahaan Teknologi Informasi yang memberikan pelatihan-pelatihan kepada sekolah-sekolah dan juga mengembangkan *software-software* Pendidikan. Dari sini penulis akhirnya banyak melakukan eksplorasi *software*, *Mikrokontroler* dan Robotika. Serta terlibat baik secara langsung maupun tidak langsung dalam pengembangan *software*. Beberapa tulisan ringan ditulis dalam bentuk *blog* dan sering menggunakan nickname *martonokita*. Adapun tulisan ilmiahnya telah diterbitkan dalam bentuk buku dan dalam beberapa Jurnal Ilmiah. Penulis juga mengajar di SMKN 4 Depok dan Universitas Raharja.

Email Penulis: martono@raharja.info.

BAB 6

DASAR ANALISIS KEBUTUHAN

Dr. Ir. Norbertus Tri Suswanto Saptadi, S.Kom., M.T., M.M., IPM.
Universitas Atma Jaya Makassar

Pendahuluan

Rekayasa Perangkat Lunak (RPL) memiliki tahapan yang relatif berjenjang dan memberikan tantangan baru dalam menempuh solusi. Pemahaman terhadap persoalan sistem membutuhkan analisis karena mempunyai karakter bervariasi. Setiap sistem mempunyai isi yang memerlukan penjelasan dalam layanan dan batasan sehingga menjadi persyaratan untuk memperoleh hasil maksimal sebuah sistem. Untuk melengkapi persyaratan kualitas harus menjabarkan berbagai kebutuhan sistem. Persyaratan yang dikumpulkan perlu dianalisis untuk menghilangkan persoalan yang terjadi agar menemukan solusi alternatif melalui analisis kebutuhan (Migunani, 2022).

Rekayasa Perangkat Lunak merupakan suatu bidang ilmu yang mendalami dan mempelajari cara peningkatan kinerja perangkat lunak melalui pemeliharaan manajemen, pengembangan perangkat lunak dan peningkatan kualitas. RPL berfokus pada penggunaan *software* komputer, desain grafis, pembuatan aplikasi, dan seluruh aktivitas yang berkaitan dengan *software* komputer (Ristekdikti, 2015). Kunci keberhasilan RPL terkait pada 3 (tiga) poin penting berupa algoritma, desain, dan pemrograman. Terdapat 4 (empat) tahapan rekomendasi dalam RPL (*software engineering*), yaitu:

karena mempunyai karakter bervariasi. Setiap sistem mempunyai isi yang memerlukan penjelasan dalam layanan dan batasan sehingga menjadi persyaratan untuk memperoleh hasil maksimal sebuah sistem.

Pemenuhan persyaratan kebutuhan perangkat lunak dilakukan dengan menetapkan keperluan sistem dan menjabarkan batasan operasi serta implementasi dengan merelasikan seluruh penggunaan fungsi secara sistematis. Analisis persyaratan merupakan sebuah peran RPL dalam mereduksi kesenjangan antara penggunaan perangkat lunak pada *level* sistem dengan desain perangkat lunak yang dilakukan secara terstruktur.

Pendefinisian kebutuhan secara menyeluruh merupakan komponen dasar dalam identifikasi dan evaluasi persoalan dan hambatan sehingga perlu disarankan untuk merevisi. Analisis kebutuhan berguna karena akan membantu setiap organisasi maupun individu dalam memperoleh atau menemukan solusi alternatif sebagai upaya meningkatkan kinerja sistem secara keseluruhan.

Teknologi dan aplikasi merupakan kebutuhan mendasar di era *digital* dan transformasi informasi. Dinamika yang terjadi dalam ranah teknologi komputer telah dipengaruhi oleh kehadiran komputer dan jasa layanan publik, industri, kantor, edukasi, teknik dan dagang. Komputer dan perangkat lunak merupakan sarana dalam pengolahan data yang mempunyai peranan strategis dengan kemampuan yang cepat, handal, dan cermat dalam memproses data.

Penggunaan perangkat lunak secara umum merupakan upaya untuk menyelesaikan berbagai persoalan hidup manusia dengan kemajuan teknologi informasi. Kegiatan analisis kebutuhan akan mampu meningkatkan kinerja dan merintis pengembangan perangkat lunak (*software development*) di masa mendatang. Analisis berawal dari pemahaman masalah (*requirement elicitation*), desain, pemrograman, penerapan, hingga *monitoring* dan evaluasi.

Daftar Pustaka

- Cahyadi, D. (2023). *Metodologi Desain*. Makassar: Penerbit Program Studi Desain Komunikasi Visual Fakultas Seni dan Desain Universitas Negeri Makassar.
- Dyah P.A, N.R. and Arsandy, E.R. (2016). Sistem Informasi Geografis Tempat Praktek Dokter Spesialis Di Provinsi D.I. Yogyakarta Berbasis Web, *Informatika Mulawarman: Jurnal Ilmiah Ilmu Komputer*, 10(1), p. 65. Available at: <https://doi.org/10.30872/jim.v10i1.22>.
- Falah, H. *et al.* (2023). Perancangan dan Pembangunan Sistem Informasi Inventarisasi Barang CV Lintang Ketawang, *JBegaTI*, 4(1), pp. 70–80. Available at: <http://begawe.unram.ac.id/index.php/JBTI/>.
- Migunani. (2022). *Rekayasa Perangkat Lunak*. Semarang: Yayasan Prima Agus Teknik dan Universitas STEKOM.
- Ningsih, K.S., Aruan, N.J. and Siahaan, A.T.A.A. (2022). Aplikasi Buku Tamu Menggunakan Fitur Kamera Dan Ajax Berbasis Website Pada Kantor Dispora Kota Medan, *SITek: Jurnal Sains, Informatika, dan Tekonologi*, 1(3), pp. 94–99.
- Ramadhan, R.S., Voutama, A. and Hannie, H. (2023). Rancang Bangun Sistem Informasi Penjualan Hybrid Berbasis Website (Studi Kasus Toko Rizki Plastik), *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(2), pp. 1227–1235. Available at: <https://doi.org/10.36040/jati.v7i2.6669>.
- Ristekdikti. (2015). Kerangka Kualifikasi Nasional Indonesia', *Direktorat Jendral Pembelajaran dan Kemahasiswaan Kemenristekdikti*, Dokumen 00, pp. 1–9.
- Setiany, A.P. *et al.* (2021). Penggunaan Metode System Development Life Cycle (SDLC) dalam Analisis dan Perancangan Sistem Informasi Penerimaan Kas Sekolah, *Jurnal Teknologi Sistem Informasi dan Aplikasi*, 4(3), pp. 179–186. Available at: <https://doi.org/10.17509/jaset.v4i2.8920>.
- Sudarma, M. (2012). *Dasar-Dasar Manajemen Proyek Teknologi Informasi*. Denpasar: Udayana University Press. Available at: https://www.academia.edu/25095637/Dasar-dasar_Manajemen_Proyek_Teknologi_Informasi.

PROFIL PENULIS



Dr. Ir. Norbertus Tri Suswanto Saptadi, S.Kom., M.T., M.M., IPM.

Lahir di Cirebon Jawa Barat, tanggal 7 Juni 1975. Memiliki Jabatan Fungsional Lektor Kepala, Pembina Tingkat I (IV/b). Berpendidikan Sarjana Komputer (S.Kom.) di Universitas Teknologi Digital Indonesia (UTDI) tahun 1998, Magister Manajemen (M.M.) di Universitas Hasanuddin (UNHAS) tahun 2004, Magister Teknologi Informasi (M.T.) di Universitas Gadjah Mada (UGM) tahun 2007, Insinyur (Ir.) di Pendidikan Profesi Insinyur UNHAS tahun 2020, Insinyur Profesional Madya (IPM.) di Persatuan Insinyur Indonesia (PII) tahun 2021, Doktor (Dr.) di Fakultas Teknik UNHAS tahun 2023, dan Program Pendidikan Reguler Angkatan (PPRA) LX Lemhannas RI tahun 2020.

Menjadi tenaga pengajar (Dosen) pada Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Atma Jaya Makassar (UAJM). Peraih Poster terbaik DPRM Dikti tahun 2016. Dosen berprestasi IKDKI tahun 2020 dan 2021. Pernah menjabat Kepala UPT Komputer, Kepala BAPSI, Wakil Dekan FT, Dekan FT dan FTI, Wakil Rektor III, Ketua Penjaminan Mutu. Tim PAK Dosen dan Asesor BKD UAJM. *Reviewer International Conference* dan Jurnal SINTA. Pemenang Hibah Kemdikbud Penelitian Dosen Pemula, Bersaing, Fundamental, dan Strategi Nasional.

Penulis artikel media massa Tribun Timur, Koinonia, Bisnis Sulawesi, Sesawi.net, Mirifica.net, HidupKatolikCom, OMKNet, KatolikanaTV, Jalan Hidup Katolik, dll. Aktivis organisasi IKA Lemhannas RI LX, IARMI, DPP ISKA, BAPOMI Sulsel, LP3KD Sulsel, IKDKI SulSelTraBar, Komkep KAMS, Komsos KAMS, PUKAT KAMS, TPP KAMS, FMKI KAMS, UPS KAMS, Pengurus Kebun Sawit Laimbo, FDI, PII Makassar, Dewan Keuangan Paroki dan Program Ayo Sekolah Mariso, dll.

Email Penulis: ntsaptadi@gmail.com.

BAB 7

MODEL PROSES PENGEMBANGAN PERANGKAT LUNAK

Fifit Alfiah, M.Kom.
Universitas Raharja

Pendahuluan

Materi pada Bab 7 ini akan membahas 4 jenis model proses pengembangan perangkat lunak berbasis *agile* yang biasa digunakan oleh dosen, mahasiswa untuk keperluan penelitian hingga membuat program dan juga dapat digunakan oleh pengembang program perangkat lunak. Model proses pengembangan perangkat lunak (Mall, 2014) merupakan sebuah metode atau panduan untuk proses tahapan untuk pembuatan sebuah program, karena Pengembangan Perangkat lunak sudah pasti memerlukan proses yang sangat panjang mulai dari analisa hingga pembuatan sampai dengan sistem itu jadi dan dipelihara sehingga dapat digunakan dalam jangka panjang oleh seorang pengembang perangkat lunak. Berikut ini beberapa macam model proses pengembang perangkat lunak yang akan dibahas pada buku ini adalah *Agile Model*, *Extreme Programming*, *Scrum Model* dan *Rapid Application Development (RAD)*.

Agile Software Development Model

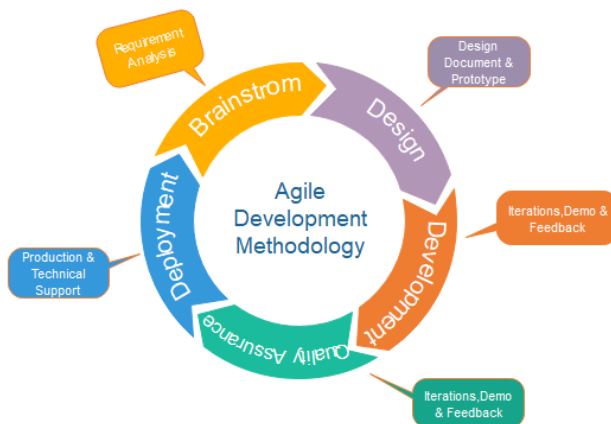
Model *Agile* dimaksudkan untuk membantu proyek menyesuaikan diri dengan perubahan. Oleh karena itu, tujuan utama model *Agile* (Janner Simarmata, 2009) adalah untuk membantu proyek diselesaikan

dengan cepat. Kelincahan adalah kunci untuk menyelesaikan tugas ini. Ini dicapai dengan menyesuaikan proses dengan proyek dan menghapus kegiatan yang mungkin tidak penting untuk proyek tersebut. Juga, hindari hal-hal yang membuang-buang waktu dan usaha dengan menetapkan ruang lingkup dan persyaratan proyek pada awal proses pengembangan. Rencana mengenai jumlah iterasi, durasi, dan ruang lingkup setiap iterasi didefinisikan dengan jelas sebelumnya.

Agile juga melakukan pendekatan manajemen proyek yang melibatkan pemecahan proyek menjadi beberapa fase (Roger S. Pressman, 2010) dan menekankan kolaborasi dan peningkatan berkelanjutan. Tim mengikuti siklus perencanaan, pelaksanaan, dan evaluasi. Pembagian seluruh proyek menjadi bagian-bagian yang lebih kecil membantu meminimalkan risiko proyek dan mengurangi persyaratan waktu pengiriman proyek secara keseluruhan. Setiap iterasi melibatkan tim yang bekerja melalui siklus hidup pengembangan perangkat lunak penuh termasuk perencanaan, analisis persyaratan, desain, pengkodean, dan pengujian sebelum produk kerja ditunjukkan kepada klien.

Fase *Agile Software Development*

Proyek pengembangan perangkat lunak *Agile* biasanya terdiri dari fase-fase (Adenowo dkk., 2013) berikut:



Gambar 7.1: Fase *Agile Software Development*

Sumber: www.javatpoint.com.

Daftar Pustaka

- Adenowo, A., Adenowo, A. A., & Adenowo, B. A. (2013). *Software Engineering Methodologies: A Review of the Waterfall Model and Object-Oriented Approach*. Article in *International Journal of Scientific and Engineering Research*, 4(7). <http://www.ijser.org>.
- Al-Saqqa, S., Sawalha, S., & Abdelnabi, H. (2020). *Agile software development: Methodologies and trends*. *International Journal of Interactive Mobile Technologies*, 14(11), 246-270. <https://doi.org/10.3991/ijim.v14i11.13269>.
- Bambang Hariyanto. (2004). *Rekayasa Sistem Berorientasi Objek*. Informatika.
- Fitria, O., Hasanah, N., Pd, M., & Untari, R. S. . *Buku Ajar Rekayasa Perangkat Lunak Diterbitkan oleh UMSIDA PRESS Universitas Muhammadiyah Sidoarjo 2020*.
- Gomaa, Hassan. (2011). *Software modeling and design: UML, Use Cases, Patterns, And Software Architectures*. Cambridge University Press.
- Hariyanto, D., Sastra, R., Putri, F. E., Informasi, S., Kota Bogor, K., & Komputer, T. (2021). Implementasi Metode Rapid Application Development Pada Sistem Informasi Perpustakaan. Dalam *Jurnal JUPITER* (Vol. 13, Nomor 1).
- Ian Sommerville. (2011). *Software Engineering* (Ninth Edition). Erlangga.
- Janner Simarmata. (2009). *Rekayasa Perangkat Lunak*. Andi.
- Kharisma, B., & Santoso, N. (2020). *Pengembangan Aplikasi Manajemen Proyek Perangkat Lunak Kolaboratif Menggunakan Scrum* (Vol. 4, Nomor 3). <http://j-ptiik.ub.ac.id>.
- Mall, Rajib. (2014). *Fundamentals of software engineering*. Prentice-Hall Of India.
- Rizky, M., & Sugiarti, Y. (2022). Penggunaan Metode Scrum Dalam Pengembangan Perangkat Lunak: Literature Review. *Journal of Computer Science and Engineering (JCSE)*, 3(1), 41-48. <https://doi.org/10.36596/jcse.v3i1.353>.
- Roger S. Pressman. (2010). *Software Engineering: a Practitioner's Approach (Seventh Edition)*. McGraw-Hill.

- Supriyatna, A. (2018). Metode Extreme Programming Pada Pembangunan Web Aplikasi Seleksi Peserta Pelatihan Kerja. *Jurnal Teknik Informatika*, 11(1), 1-18. <https://doi.org/10.15408/jti.v11i1.6628>.
- Usanto, U., Nurlaela, L., Sopian, A., & Alfiah, F. (2022). Umrah Registration System Using Extreme Programming Method Towards Worship Tourism. *International Journal of Cyber and IT Service Management*, 3(1), 22-31. <https://doi.org/10.34306/ijcitsm.v3i1.104>.

PROFIL PENULIS



Fifit Alfiah, M.Kom.

Ketertarikan penulis terhadap ilmu komputer dimulai pada tahun 2011 yaitu semenjak melanjutkan pendidikan ke Perguruan Tinggi dan berhasil menyelesaikan studi S1 di prodi Teknik Informatika Universitas Raharja pada tahun 2015, selama menempuh Pendidikan studi S1 penulis juga membangun komunitas *Software Engineering* pada tahun 2012 bersama rekan-rekan mahasiswa di prodi Teknik Informatika. Setelah rehat beberapa bulan pada tahun 2015 kemudian penulis melanjutkan Pendidikan Pasca Sarjana (S2) pada Januari 2016 dan 2(Dua) tahun kemudian, penulis menyelesaikan studi Pasca Sarjana (S2) di prodi Rekayasa Komputasi Terapan Program Pasca Sarjana Ilmu Komputer Universitas Budi Luhur.

Penulis memiliki ketertarikan dibidang *Decision Support System, Image Processing dan Web Technology*. Dan untuk mewujudkan karir sebagai dosen profesional, penulis pun aktif sebagai peneliti dibidang kepakarannya tersebut dengan telah menghasilkan puluhan artikel ilmiah di berbagai jurnal nasional dan internasional. Selain peneliti, penulis juga aktif menulis buku dengan harapan dapat memberikan kontribusi positif bagi bangsa dan negara yang sangat tercinta ini. Atas dedikasi dan kerja keras Penulis juga telah memiliki Sertifikasi Pendidik sebagai dosen professional yang diakui oleh kemristekdikti pada tahun 2023.

Email Penulis: fifitalfiah@raharja.info.

BAB 8

KEBUTUHAN PERANGKAT LUNAK

Mujibbur Rohman, S.Si.
SMKN 4 Bogor

Pendahuluan

Perangkat lunak telah menjadi bagian integral dalam kehidupan modern, mengendalikan berbagai aspek mulai dari aplikasi *smartphone* hingga sistem industri yang kompleks. Di balik setiap perangkat lunak yang sukses terdapat proses pengembangan yang terstruktur dan terencana, dengan kebutuhan perangkat lunak sebagai fondasi utamanya.

Kebutuhan perangkat lunak mendefinisikan apa yang harus dan tidak boleh dilakukan oleh perangkat lunak. Kebutuhan ini merupakan pernyataan tentang fungsionalitas, kinerja, keamanan, dan karakteristik lain yang diperlukan untuk memenuhi tujuan dan harapan pengguna.

Jenis-jenis Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak dapat dibagi menjadi dua kategori utama yaitu Kebutuhan Fungsional dan Kebutuhan Non Fungsional.

1. Kebutuhan Fungsional

Kebutuhan fungsional menjelaskan fitur-fitur spesifik seperti input, output dan proses yang harus dilakukan perangkat lunak untuk menyelesaikan tugas dan mencapai tujuannya. Kebutuhan fungsional juga merupakan dasar dari apa yang perangkat lunak seharusnya lakukan dan mencakup berbagai aspek, termasuk:

4. Komunikasi yang buruk: Kurangnya komunikasi antara tim developer, pengguna, dan pemangku kepentingan dapat menyebabkan miskomunikasi dan masalah lainnya.
5. Manajemen risiko yang tidak memadai: Kurangnya identifikasi dan mitigasi risiko dapat menyebabkan masalah yang tidak terduga selama pengembangan.

Kurang baiknya pemahaman kebutuhan sangat berpengaruh dalam kegagalan pengembangan perangkat Lunak. Begitu juga sebaliknya, pemahaman yang baik tentang kebutuhan perangkat lunak sangat berpengaruh pada keberhasilan pengembangan perangkat lunak.

Daftar Pustaka

- Dean Leffingwell, Don Widrig (2003). *Managing Software Requirements: A Use Case Approach*. Addison-Wesley Professional.
- Jennifer Lynch. Chaos Report. (2020). Diakses 25-1-2023 dari <https://www.standishgroup.com/news/45>.
- Karl Eugene Wiegers (2003). *Software Requirements*. Microsoft Press.
- Kotonya, Gerald, and Ian Sommerville. (2019). *Requirements Engineering: Processes and Techniques*. John Wiley & Sons.
- Pressman, Roger S. (2019). *Software Engineering: A Practitioner's Approach, 9th Edition*. McGraw-Hill Education.
- Sommerville, Ian. (2016). *Software Engineering, 10th Edition*. Pearson.

PROFIL PENULIS



Mujibbur Rohman, S.Si.

Lulusan Ilmu Komputer Universitas Sanata Dharma Yogyakarta dan menjadi pengajar sejak tahun 2007. Memiliki pengalaman mengajar diantaranya sebagai guru Teknik Komputer dan Jaringan di SMK Perguruan Cikini (2007-2009), guru Teknologi Informasi dan Komunikasi di SMAN 81 Jakarta (2009-2014), dan guru Rekayasa Perangkat Lunak di SMKN 4 Bogor (2014-Sekarang). Selain itu aktif mengajar di *Platform Red Hat Academy* tentang *Redhat System Administration* (RH124, RH134) dan *Redhat Application Development* (AD183). Penulis saat ini sedang menempuh studi S2 di jurusan Manajemen Sistem Informasi Universitas Gunadarma program studi Perangkat Lunak Sistem Informasi. Penulis memiliki ketertarikan dibidang *Data Science*, *IOT*, dan *Cloud Computing*.

Email Penulis: mujibbur.rohman@gmail.com.

BAB 9

ANALISA DAN PERANCANGAN BERORIENTASI OBJEK

Arif Muhamad Nurdin, S.Kom., M.Kom.
Universitas Cipasung Tasikmalaya

Pendahuluan

Analisa dan perancangan berorientasi objek adalah pendekatan pengembangan perangkat lunak berdasarkan konsep objek. Konsep ini memungkinkan pengembang untuk memodelkan suatu sistem sebagai kumpulan objek yang berinteraksi untuk mencapai tujuan tertentu.

Analisa dan perancangan berorientasi objek merupakan cara berpikir baru tentang masalah menggunakan model yang dibangun dari konsep dunia nyata. Dasar pembuatannya adalah objek yang mewakili kombinasi struktur data dan perilaku dalam suatu entitas. Istilah "berorientasi objek" mengacu pada pengorganisasian perangkat lunak sebagai kumpulan objek tertentu yang memiliki struktur data dan perilaku.

Dalam pengembangan perangkat lunak, konsep pendekatan berorientasi objek dapat diterapkan pada tahap analisis, perancangan, pemrograman, dan pengujian perangkat lunak. Pada masing-masing tahap ini dapat menggunakan teknik berbeda dengan menggunakan aturan dan alat bantu pemodelan tertentu.



Gambar 9.4: OOP vs Terstruktur

Sumber:

https://lms.global.ac.id/lms/pluginfile.php/17619/mod_resource/content/1/Konsep%20Object%20Oriented.pdf

Daftar Pustaka

Azis, I. M. F., & Kom, M. (2005). *Object Oriented Programming Php 5*. Elex Media Komputindo.

<https://id.quora.com/Apa-perbedaan-Class-dan-Sub-Class-pada-bahasa-pemrograman-berorientasi-objek-OOP>.

https://lms.global.ac.id/lms/pluginfile.php/17619/mod_resource/content/1/Konsep%20Object%20Oriented.pdf.

<https://medium.com/@muhammadraflyindrawan/enkapsulasi-objek-dan-pewarisan-sifat-di-php-oop-3109c7c53ced>.

Sukamto, R. A (2022). *Analisis dan Desain Perangkat Lunak Rekayasa Perangkat Lunak Untuk Pemrograman Terstruktur, Berorientasi Objek dan Agile*. Bandung: Informatika.

Sukamto, R. A & Salahuddin, R. (2016). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.

PROFIL PENULIS



Arif Muhamad Nurdin, S.Kom., M.Kom.

Penulis mulai tertarik terhadap ilmu komputer dimulai pada saat penulis masuk sekolah SMA. Hal tersebut membuat penulis memilih untuk mengikuti ekstrakurikuler robotika di Sekolah Menengah Atas di SMA Negeri 1 Kota Banjar Jawa Barat. Penulis kemudian melanjutkan pendidikan ke Perguruan Tinggi dan berhasil menyelesaikan studi S1 di prodi Teknik Informatika Universitas Komputer Indonesia pada tahun 2017. Penulis pernah aktif menulis di blog otomotif dari rentang tahun 2010 sampai tahun 2015 dan tergabung dalam berbagai perkumpulan penulis dan komentator blog otomotif di INDOMOTOBLOG, KoBOI (Komunitas Blogger Otomotif Indonesia), KOBOYS (Komunitas Blogger Otomotif Yogyakarta dan Sekitarnya) dan D'GUJUBAR (Paguyuban Juragan Blogger Jawa Barat). Penulis pernah bekerja sebagai *Administrative Analyst System* di PT Kirana Prima Abadi dari tahun 2018 dengan status pegawai dan pada saat ini sebagai *freelance*. Pada tahun 2021 penulis melanjutkan studi S2 di prodi Sistem Informasi Program Pasca Sarjana STMIK LIKMI dan lulus pada tahun 2022. Saat ini penulis aktif sebagai Dosen Tetap di Prodi Sistem Informasi Fakultas Sains dan Teknologi Universitas Cipasing Tasikmalaya.

Email Penulis: arif.mnurdin@uncip.ac.id.

BAB 10

PEMODELAN SISTEM DENGAN UML

Nur Azizah, S.Kom., M.Akt., M.Kom.
Universitas Raharja

Pendahuluan

Perkembangan terkini berdampak pada cara pengembang (developer) mengembangkan sistem. Evolusi seiring berjalannya waktu mengubah cara sistem ini dikembangkan dan oleh karena itu dapat secara langsung mengubah cara berpikir perancang sistem ketika mengembangkan sistem. Perkembangan saat ini tentunya mempengaruhi jenis perangkat lunak yang dikembangkan oleh pengembang. Padahal, pembangunan tersebut tidak menyederhanakan proses pembangunan, melainkan justru mempersulitnya (Pengantar *Unified Modeling Language (UML)*, Sri Dharwiyanti, Romi Satria Wahono, 1).

Oleh karena itu, perkembangan saat ini mungkin meresahkan bagi pengembang sistem dan perangkat lunak yang ingin benar-benar mengimplementasikan suatu sistem dalam satu atau lebih bahasa pemrograman yang digunakan untuk membuatnya. Hal ini juga mempengaruhi kinerja analis sistem yang merancang sistem yang langsung diimplementasikan oleh pengembang (programmer). Hal ini karena pengembang perlu merancang sistem atau perangkat lunak sebelum mereka dapat mengimplementasikannya dibuat oleh analis sistem.

UML (*Unified Modeling Language*) adalah alat yang dapat digunakan atau dipilih oleh analis untuk merancang sistem yang

7. *Component Diagram*

Component Diagram bersifat statis. Diagram ini menunjukkan konfigurasi dan ketergantungan komponen yang ada. Diagram ini mengacu pada diagram kelas, di mana komponen biasanya dipetakan ke satu atau lebih kelas, antarmuka, dan kolaborasi.

8. *Deployment Diagram*

Deployment Diagram bersifat statis. Diagram ini menunjukkan konfigurasi runtime aplikasi. Diagram ini mencakup simpul-simpul (node) dan komponen di dalamnya. Diagram ini sangat berguna ketika aplikasi berfungsi sebagai aplikasi yang berjalan pada mesin (*distributed computing*).

Berbagai jenis grafik di atas dibuat karena alasan berikut:

- a. Sistem yang kompleks selalu paling baik didekati melalui kumpulan pertimbangan-pertimbangan kecil yang sebagian besar *independen* satu sama lain. Sudut pandang tunggal tidak selalu cukup untuk melihat isi objek yang besar dan kompleks.
- b. Diagram yang berbeda ini dapat mewakili tingkat proses rekayasa yang berbeda.
- c. Diagram-diagram ini dibuat agar model yang dibuat lebih mendekati kenyataan. Kombinasi diagram dan fitur dokumentasi ini merupakan artefak utama UML. *Data-flow diagram* dan jenis diagram lain yang bukan bagian dari UML bukan bagian dari paradigma *object-oriented*. *Activity diagram* dan *collaboration diagram* yang disertakan dalam UML merupakan alternatif data-flow diagram. *Activity diagram* juga sangat berguna untuk membuat *workflow*.

Daftar Pustaka

- Dharwiyanti Sri, Wahono Satria Romi. *Pengantar Unified Modeling Language (UML)*. <http://www.ilmukomputer.com>.
- Hartanto Rudi. (2007). *Prinsip Perancangan Berpusat Pada Pengguna*, <http://www.ilmukomputer.com>.
<https://ocw.ui.ac.id/mod/resource/view.php?id=160>.
- Kadir, Abdul. (1989). *Analisa dan Desain Sistem Informasi*. Yogyakarta: Andi Offset.
- Noviana Rina, *Kebutuhan dan Spesifikasi Perangkat Lunak Berorientasi Objek*, <http://www.ilmukomputer.com>.
- Nugroho Adi. (2004). *Analisis dan Perancangan Sistem Informasi dengan Metodologi berorientasi Objek, Informatika*.
- Object Oriented with UML. (2004). <http://www.kakasblog.blogspot.com/materi/oop/oopwithUML.ppt>.
- Pendahuluan Perangkat Lunak. <http://www.petra.ac.id/library/wpcontent/upload/2009/01/02/rpl.pdf>.
- Teguh, Wahyono. (2004). *Computer Based Information System (CBIS)*. <http://www.ilmukomputer.com>.

PROFIL PENULIS



Nur Azizah, S.Kom., M.Akt., M.Kom.

Penulis Lulus S1 Program Studi Sistem Informasi di Sekolah Tinggi Manajemen dan Ilmu Komputer (STMIK) Budi Luhur Jakarta dan menyelesaikan Program Studi magister Akuntansi S2 di Fakultas Ekonomi dan Bisnis Universitas Budi Luhur Jakarta dan Program Studi Magister Komputer S2 di Fakultas Teknologi Universitas Budi Luhur Jakarta.

Saat ini sedang tahap menyelesaikan Pendidikan Program Doktor S3 pada Fakultas Teknologi Informasi Universitas Gunadarma Jakarta. Saat ini sebagai Dekan Fakultas Ekonomi dan Bisnis Universitas Raharja merangkap sebagai Ketua Program Studi Sains Data Jenjang Sarjana. Penulis mengampu matakuliah Dasar Akuntansi, Sistem Informasi Akuntansi, Sistem Informasi Manajemen, Sistem Basis Data, *Business Intelligence Management* dan *IT Risk Management*. Aktif sebagai Pengurus Pusat Asosiasi Perguruan Tinggi Ilmu Komputer (APTIKOM), sebagai pengurus Asosiasi Perguruan Tinggi Swasta (APTISI) Wilayah IVB-Banten dan sebagai Pengurus Asosiasi Dosen Indonesia (ADI) Banten. Aktif menulis di beberapa Jurnal nasional dan International dan Buku yang sudah terbit Manajemen Retail dan Industri Halal di Indonesia. Aktif sebagai pendakwah di Majelis-majelis Ta'lim Muslimah.

Email Penulis: nur.azizah@raharja.info.

BAB 11

PERANCANGAN PIRANTI PERANGKAT LUNAK

Ade Setiadi, M.Kom.
Universitas Raharja

Pendahuluan

Perancangan piranti perangkat lunak merupakan proses yang sangat penting dalam pengembangan teknologi informasi. Hal ini karena piranti perangkat lunak merupakan salah satu komponen utama dalam sistem komputer yang digunakan untuk menjalankan berbagai macam aplikasi dan program (*Software Engineering: A Practitioner's Approach*, n.d.). Perancangan piranti perangkat lunak tidak hanya mencakup aspek teknis, tetapi juga melibatkan aspek-aspek lain seperti kebutuhan pengguna, keamanan, dan efisiensi penggunaan sumber daya.

Proses perancangan piranti perangkat lunak dimulai dari analisis kebutuhan, di mana tim pengembang mengumpulkan informasi mengenai tujuan dan fungsi yang diinginkan dari piranti tersebut. Selanjutnya, tim akan merancang struktur dan desain dari piranti tersebut (Janner Simarmata, 2010), termasuk pemilihan bahasa pemrograman yang sesuai dan algoritma yang efisien (Mall, 2014). Selama proses perancangan, tim juga harus mempertimbangkan kebutuhan pengguna dan memastikan antarmuka yang mudah digunakan dan intuitif.

Keamanan juga merupakan faktor penting dalam perancangan piranti perangkat lunak (Toko et al., n.d.), di mana tim harus memastikan bahwa sistem yang dibangun terlindungi dari serangan

Manfaat Perancangan Piranti Perangkat Lunak

Terdapat beberapa manfaat dari adanya perancangan Piranti Perangkat Lunak, yaitu sebagai berikut:

1. Fleksibilitas

Desain perangkat lunak memungkinkan fleksibilitas yang lebih besar dalam hal penyesuaian dan kemampuan beradaptasi. Dengan perangkat lunak, lebih mudah untuk memperbarui dan memodifikasi desain sesuai perubahan kebutuhan (*Software Engineering: A Practitioner's Approach*, n.d.).

2. Efektivitas Biaya

Desain perangkat lunak dapat menjadi solusi yang lebih hemat biaya dibandingkan dengan desain perangkat keras. Ini menghilangkan kebutuhan akan komponen fisik dan mengurangi biaya produksi dan pemeliharaan.

3. Perkembangan Lebih Cepat

Perangkat lunak dapat dikembangkan dan diuji lebih cepat dibandingkan perangkat keras. Hal ini memungkinkan pembuatan prototipe dan iterasi yang lebih cepat (Mall, 2014), yang pada akhirnya mempercepat waktu pemasaran. Meskipun mungkin ada keterbatasan dan tantangan tertentu yang terkait dengan desain perangkat lunak, kelebihanannya menjadikannya pilihan yang layak bagi banyak industri dan aplikasi.

Daftar Pustaka

- Adenowo, A., Adenowo, A. A., & Adenowo, B. A. (2013). Software Engineering Methodologies: A Review of the Waterfall Model and Object-Oriented Approach. *Article in International Journal of Scientific and Engineering Research*, 4(7). <http://www.ijser.org>.
- Adi Nugroho. (2019). *Rekayasa Perangkat Lunak menggunakan UML dan JAVA*. Andi Offset.
- Budiman, B. (2018). Perancangan Aplikasi Mobile Campus (m-Campus). <https://doi.org/10.17605/OSF.IO/C5FWV>.
- Indah Purnama Sari, S. T., M. K. (2021). *Buku Ajar Rekayasa Perangkat Lunak*. Med.
- Janner Simarmata. (2010). *Rekayasa Perangkat Lunak*. ANDI.
- Mall, Rajib. (2014). *Fundamentals Of Software Engineering*. Prentice-Hall Of India.
- Safii, M., Indrayani, N., Universitas Mulia, D., Letjend TNI Maulani No, J. Z., & Kota Sur-el, B. (2020). Perancangan Piranti Lunak Responsive Untuk Monitoring Ruang Server Menggunakan Nodemcu Esp8266 Berbasis Internet of Things. *Jurnal Ilmiah MATRIK*, 22(3).
- Software Engineering: A Practitioner's Approach. (n.d.). www.mhhe.com/pressman.
- Toko, D., Arizal, M., Rahman, K., Mardiyati, S., & Nugraha, Y. (n.d.). Perancangan Sistem Informasi Penjualan barang Berupa Alat Music. In *Maret* (Vol. 2022, Issue 7).
- UNIT-I: Software Engineering & Process Models. (n.d.).
- Usanto, U., Nurlaela, L., Sopian, A., & Alfiah, F. (2022). Umrah Registration System Using Extreme Programming Method Towards Worship Tourism. *International Journal of Cyber and IT Service Management*, 3(1), 22-31. <https://doi.org/10.34306/ijcitsm.v3i1.104>.

PROFIL PENULIS



Ade Setiadi, M.Kom.

Ketertarikan penulis terhadap ilmu komputer dimulai pada usia 10 tahun. Hal tersebut membuat penulis memilih untuk masuk kuliah Strata 1 ke STMIK Raharja dengan memilih Jurusan Teknik Informatika dengan konsentrasi *Software Engineering* (SE) lulus dengan IPK 3,67 dan menjadi wisudawan terbaik jurusan Teknik Informatika. Penulis kemudian melanjutkan pendidikan ke Strata 2 di Universitas Budi Luhur dengan Jurusan Magister Ilmu Komputer dengan konsentrasi Rekayasa Komputasi Terapan (RKT) dan berhasil lulus dengan IPK 3,44 pada tahun 2018. Penulis mewujudkan karir sebagai dosen pada Universitas Raharja mulai dari 2017.

Kini penulis telah menjadi dosen profesional yaitu telah lulus sertifikasi dosen pada tahun 2023 serta memiliki jabatan fungsional Lektor dalam mata kuliah Pemrograman dan *Web Programming*, penulis pun aktif sebagai peneliti dibidang kepakarannya tersebut dengan menulis jurnal nasional/jurnal internasional setiap semesternya. Aktif menjadi pembicara pada kegiatan pengabdian masyarakat ke Sekolah Menengah Atas Sederajat setiap semesternya. Penulis aktif menjadi *Reviewer* Jurnal Nasional, Selain itu penulis aktif sebagai di anggota profesi yaitu sebagai anggota Asosiasi Dosen Indonesia (ADI) sejak tahun 2022.

Email Penulis: adesetiadi@raharja.info.

BAB 12

TEKNIK KEAMANAN REKAYASA PERANGKAT LUNAK

Rio Wirawan, S.Kom., M.M.S.I.

Universitas Pembangunan Nasional Veteran Jakarta

Pendahuluan

Kebutuhan akan keamanan perangkat lunak menjadi semakin penting seiring dengan perkembangan teknologi informasi dan ketergantungan masyarakat pada perangkat lunak. Keamanan perangkat lunak merujuk pada serangkaian langkah dan praktik yang dirancang untuk melindungi perangkat lunak dari ancaman, kerentanan, dan potensi eksploitasi yang dapat merugikan pengguna atau organisasi yang mengandalkan perangkat lunak tersebut.

Keamanan perangkat lunak adalah suatu aspek yang sangat penting dalam dunia teknologi informasi saat ini. Seiring dengan meningkatnya ketergantungan kita pada perangkat lunak, perlindungan terhadap keamanan perangkat lunak menjadi suatu kebutuhan mendesak. Artikel ini akan membahas pentingnya keamanan perangkat lunak, ancaman yang mungkin dihadapi, dan strategi untuk meningkatkan keamanan.

Tujuan

Keamanan perangkat lunak merupakan aspek kritis dalam era digital yang ditandai oleh ketergantungan kita pada teknologi. Tujuan dari

teknik keamanan perangkat lunak adalah melindungi integritas, kerahasiaan, dan ketersediaan perangkat lunak serta data yang dikelolanya.

Dasar-dasar Keamanan Perangkat Lunak

Keamanan perangkat lunak menjadi sangat penting untuk menjaga informasi dan data sensitif dari ancaman peretas atau serangan *malware*. Dasar keamanan perangkat lunak mencakup berbagai aspek, mulai dari penggunaan sandi yang kuat hingga instalasi perangkat lunak keamanan yang terbaru. Menurut penelitian yang dilakukan oleh Chaudhry et al. (2018), keamanan perangkat lunak melibatkan langkah-langkah yang bertujuan untuk melindungi sistem komputer dan data dari serangan luar yang bertujuan merusak, mencuri, atau mengganggu operasi normal dari perangkat lunak tersebut. Oleh karena itu, pemahaman yang mendalam mengenai dasar-dasar keamanan perangkat lunak sangat penting bagi setiap individu atau organisasi yang memiliki kepentingan dalam menjaga integritas dan kerahasiaan informasi mereka.

Penerapan dasar-dasar keamanan perangkat lunak juga melibatkan pembahasan mengenai manajemen akses pengguna, monitoring aktivitas pengguna, serta pembaruan perangkat lunak secara berkala untuk mengatasi kerentanan keamanan yang baru ditemukan. Ghasemi et al. (2020) menjelaskan bahwa tindakan preventif seperti penggunaan perangkat lunak anti-virus, firewall, dan enkripsi data juga merupakan bagian yang penting dari strategi keamanan perangkat lunak yang efektif. Dengan memahami dasar-dasar keamanan perangkat lunak dan menerapkan langkah-langkah yang tepat, individu atau organisasi dapat mengurangi risiko terjadinya serangan terhadap perangkat lunak mereka, sehingga dapat memastikan keberlangsungan operasi dan keamanan informasi yang mereka miliki.

4. Ancaman Keamanan Perangkat Lunak

Ancaman keamanan perangkat lunak telah menjadi isu yang semakin meningkat seiring dengan perkembangan teknologi yang pesat. Perangkat lunak yang rentan terhadap serangan dari pihak

d. Uji Perizinan dan Hak Akses

Memastikan bahwa sistem menerapkan prinsip least privilege dengan menguji dan memverifikasi bahwa pengguna atau entitas hanya memiliki akses yang sesuai dengan tugas mereka. Ini mencakup pengujian perizinan dan pembatasan akses yang ketat.

e. Uji Kegagalan Keamanan

Uji ini melibatkan simulasi skenario kegagalan keamanan untuk mengidentifikasi sejauh mana sistem dapat bertahan terhadap serangan atau gangguan keamanan. Hal ini membantu mengukur dan memperbaiki keberlanjutan keamanan.

f. Uji Kestabilan dan Kinerja

Selain pengujian keamanan, uji kestabilan dan kinerja juga penting untuk memastikan bahwa perangkat lunak dapat menangani beban kerja dan tetap beroperasi secara stabil dalam situasi yang ekstrem atau serangan *denial-of-service* (DoS).

Daftar Pustaka

- Anderson, R. J. (2008). *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley.
- Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *Unified Modeling Language User Guide, The (2nd Edition)*. Addison-Wesley.
- Chaudhry, O. N., et al. (2018). A Comprehensive Study of Security Issues and Strategies In Software Defined Networking: A survey. *Journal of Network and Computer Applications*, 103, 43-59.
- Erickson, J. (2008). *Hacking: The Art of Exploitation*. No Starch Press.
- Engelbreton, P. (2014). *The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy*. Elsevier.

- DeMarco, T. (1979). *Structured Analysis and System Specification*. Yourdon Press.
- Ghasemi, F., et al. (2020). Cybersecurity in the Internet of Things: A Systematic Literature Review. *Computers & Security*, 94, 101825.
- Humble, J., & Farley, D. (2010). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Addison-Wesley.
- Jacobson, I., Christerson, M., Jonsson, P., & Övergaard, G. (1992). *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley.
- Jorgensen, P. (2002). *Software Testing: A Craftsman's Approach*. CRC Press.
- Kaspersky Lab. (2021). The Growth Of The Global Threat Landscape. Diakses dari <https://www.kaspersky.com/resource-center/threats/glossary/global-threat-landscape>.
- Nikitina, G. P., & Pavlov, V. V. (2018). Software Vulnerability Analysis Method Based on Scenario. *International Journal of Computer Applications*, 179(21), 32-37.
- Sari, I. P. (2021). *Buku Ajar Rekayasa Perangkat Lunak (Vol. 1)*. UMSU Press.
- Seacord, R. C. (2013). *Secure Coding in C and C++*. Addison-Wesley.
- Tjokrodiharjo, M. (2017). *Analisis Kerentanan Keamanan Sistem Informasi*. Jakarta: Gramedia.
- Tanenbaum, A. S. (2011). *Computer Networks*. Pearson.
- Jones, R. (2014). *Least Privilege in Action*. Manning Publications.
- Jones, R. (2014). *Least Privilege in Action*. Manning Publications.
- Viega, J., & Messier, M. (2002). *Secure Programming Cookbook for C and C++*. O'Reilly Media.
- Wijaya, H. D. (2019). *Pengantar Keamanan Sistem Komputer*. Jakarta: Andi Offset.
- Weber, M. (1947). *The Theory of Social and Economic Organization*. Free Press.

- Sommer, P. M., & Heinlein, P. (2005). *The Art of Software Security Assessment*. Addison-Wesley.
- Seacord, R. C. (2013). *Secure Coding in C and C++*. Addison-Wesley.
- McGraw, G. (2004). *Software Security: Building Security In*. Addison-Wesley.
- Shostack, A. (2014). *Threat Modeling: Designing for Security*. Wiley.
- Sommer, P. M., & Heinlein, P. (2005). *The Art of Software Security Assessment*. Addison-Wesley.
- Veracode. (2021). *The state of software security*. Diakses dari <https://www.veracode.com/state-of-software-security>.
- Viega, J., & Messier, M. (2002). *Secure Programming Cookbook for C and C++*. O'Reilly Media.
- Yourdon, E., & DeMarco, T. (1979). *Structured Analysis: A Language for Communicating Ideas*. Prentice-Hall.

PROFIL PENULIS



Rio Wirawan., S.Kom., M.M.S.I.

Rio Wirawan, S.kom., MMSI Lahir di Jakarta pada tanggal 20 Oktober 1986. Dari orang tua yang bernama Sri widodo dan ibu yang bernama erlisa agus purwatin. Menikah pada tahun 2014 dengan Murniati dan dikaruniai seorang putri dan seorang putra yakni Qaireena Zahrani Wirawan saat ini sedang bersekolah di Sekolah TK dan Abrizam Alhanan Wirawan. Riwayat Pendidikan dimulai dari SDN Depok Baru III (Lulus tahun 1998), kemudian melanjutkan ke SMPN 5 Depok (Lulus tahun 2001), kemudian melanjutkan ke SMA Bintara depok(Lulus Tahun 2004), kemudian melanjutkan kuliah dan meraih gelar Sarjana Komputer di Universitas Gunadarma (Lulus tahun 2008) dan meraih gelar Magister Manajemen Sistem Informasi di Pasca Sarjana Universitas Gunadarma (lulus tahun 2011). Pengalaman mengajar adalah sebagai instruktur sertifikasi IT di Universitas Negeri Jakarta Mulai tahun 2009 sampai Tahun 2012. sebagai Dosen tetap di Universitas Pembangunan Nasional Veteran Jakarta pada program studi sistem informasi mulai tahun 2013 sampai sekarang. Disamping itu juga menjadi dosen tidak tetap di Universitas Gunadarma mulai tahun 2012 sampai sekarang.

BAB 13

PENGUJIAN PERANGKAT LUNAK

Oleh Soleh, S.Kom., M.MSI.
Universitas Raharja

Tinjauan Pengujian Perangkat Lunak

Pengujian perangkat lunak saat ini paling efektif jika dilakukan secara terus-menerus, yang menunjukkan bahwa pengujian dimulai selama desain, berlanjut saat perangkat lunak dibuat, dan bahkan terjadi ketika diterapkan ke dalam produksi. Pengujian berkelanjutan berarti organisasi tidak perlu menunggu seluruh bagian diterapkan sebelum pengujian dapat dimulai. *Shift-left*, yang menjadikan pengujian lebih dekat dengan desain, dan *shift-right*, di mana validasi dilakukan oleh pengguna akhir, juga merupakan filosofi pengujian yang baru-baru ini mendapatkan daya tarik dalam komunitas perangkat lunak. Setelah strategi pengujian dan rencana pengelolaan Anda dipahami, otomatisasi seluruh aspek pengujian menjadi suatu keharusan untuk mendukung kecepatan penyampaian yang diperlukan.

Pengujian perangkat lunak adalah proses memverifikasi kerja sistem perangkat lunak atau aplikasi. Dengan kata lain, ini memastikan aplikasi perangkat lunak bebas bug dan memenuhi kebutuhan teknis dan pengguna. Hal ini tidak hanya berfokus pada menemukan *bug* atau kesalahan dalam perangkat lunak tetapi juga mempertimbangkan langkah-langkah untuk meningkatkan kualitas perangkat lunak dalam hal kegunaan, akurasi, dan efisiensi.

Pengujian perangkat lunak adalah proses menilai fungsionalitas program perangkat lunak. Proses ini memeriksa kesalahan dan kesenjangan dan apakah hasil aplikasi sesuai dengan harapan yang diinginkan sebelum perangkat lunak diinstal dan diluncurkan. (Kinza Yasar, 2022).

Pengujian perangkat lunak adalah proses menilai fungsionalitas aplikasi perangkat lunak. Ia memeriksa kesalahan atau bug dengan mengevaluasi pelaksanaan komponen perangkat lunak untuk memenuhi kebutuhan hasil yang diharapkan. Ini mengidentifikasi keakuratan dan kesesuaian aplikasi dengan mempertimbangkan atribut utamanya. (Bhavya Hada, Harshit Paul, 2020).

Mengapa Pengujian Perangkat Lunak Penting

Pengujian perangkat lunak penting karena dampak perangkat lunak yang belum teruji atau berkinerja buruk dapat menimbulkan efek domino pada ribuan pengguna dan karyawan. Pengujian perangkat lunak adalah puncak dari pengembangan aplikasi di mana pengujian perangkat lunak mengevaluasi kode dengan mempertanyakannya.

Evaluasi ini dapat dilakukan secara singkat atau dilanjutkan hingga seluruh pemangku kepentingan merasa puas. Pengujian perangkat lunak mengidentifikasi bug dan masalah dalam proses pengembangan sehingga diperbaiki sebelum peluncuran produk. Pendekatan ini memastikan bahwa hanya produk berkualitas yang didistribusikan kepada konsumen, yang pada gilirannya meningkatkan kepuasan dan kepercayaan pelanggan.

Meskipun pengujian itu sendiri membutuhkan biaya, perusahaan dapat menghemat jutaan per tahun dalam pengembangan dan dukungan jika mereka memiliki teknik pengujian dan proses QA yang baik. Pengujian perangkat lunak awal mengungkap masalah sebelum produk dipasarkan. Semakin cepat tim pengembangan menerima masukan pengujian, semakin cepat mereka dapat mengatasi masalah seperti:

1. Kelemahan arsitektur.
2. Keputusan desain yang buruk.
3. Fungsionalitas tidak valid atau salah.
4. Kerentanan keamanan.
5. Masalah skalabilitas

TFS), penyimpanan *cloud* (*Google Drive*, *Dropbox*, atau *OneDrive*), atau alat manajemen pengujian (*Jira*, *HP ALM*, atau *TestRail*). (Kiruthika DevarajKiruthika Devaraj, 2024).

2) Menyiapkan Laporan Ringkasan Pengujian

Laporan-laporan ini harus mendokumentasikan aktivitas dan hasil pengujian, termasuk masalah apa pun yang dihadapi dan penyelesaiannya. Hal ini juga harus merangkum ruang lingkup pengujian dan berisi rekomendasi untuk pengujian di masa depan.

h. Evaluasi Siklus Pengujian

Terakhir, langkah terakhir dari siklus hidup pengujian perangkat lunak adalah mengevaluasi hasil pengujian. Informasi dalam ringkasan laporan yang disiapkan pada tahap sebelumnya berfungsi sebagai panduan untuk perbaikan lebih lanjut. Mengevaluasi hasil melibatkan analisis data dan metrik pengujian untuk memahami apakah tujuan pengujian telah terpenuhi dan untuk memahami jenis cacat apa yang paling sering terjadi selama pengujian.

Langkah ini penting untuk menilai efektivitas pengujian dan mengidentifikasi apa yang memerlukan perbaikan lebih lanjut. Setelah pengujian dievaluasi dan semua perbaikan yang diperlukan telah dilakukan dan diperiksa, tim QA menyajikan laporan penutupan kepada pemangku kepentingan dan mendapatkan persetujuan dari mereka.

Daftar Pustaka

- Ahmed Alaa. (2024). *Bagaimana Anda Memvalidasi Dan Mengonfirmasi Persyaratan Menggunakan Kriteria Penerimaan Dan Pengujian Penerimaan Pengguna*.
- Anhelina Velychko, Sofiya Merenysh. (2023). *A Complete Guide To The Software Testing Lifecycle (STLC), Guide To Enhance Your Software Quality Assurance Practices!*. Clockwise Software. September 11, 2023.
- Bhavya Hada, Harshit Paul. (2020). *Apa Itu Pengujian Perangkat Lunak: Panduan Lengkap Dengan Contoh*. Learning Hub: Lamdatest.
- Brita Hammer. (2021). *7 Tahapan Siklus Hidup Pengembangan Perangkat Lunak (SDLC)*, Emergent Software. Agustus 2021.
- Kinza Yasar. (2022). *Definitions of Software Testing, Tech-target What Is? Agustus*.
- Kiruthika Devaraj. (2024). *Test Artifacts Explained With Types*. Testsigma, January 2024.
- Neha Bharati. (2022). *Monkey Testing vs Gorilla Testing, Bworstack*. June 2022.
- NN, (2023). *Test Execution In Software Testing: Exact Process And Plan With Example, Software Testing Help*, June 2023.
- Oliver Moradov. (2022). *Security Testing: Types, Tools, and Best Practices, Automated Security Testing, Bright Resource Center*. May 2022.
- Ramanpreet Singh. (2022). *Mengapa Pengujian Kinerja Penting untuk Membangun Aplikasi pada tahun 2024?*. Software Testing Insight. November 2022.
- Sourojit Das. (2023). *Apa itu Black Box Testing: Jenis, Alat dan Contohnya*. Jun2, 2 2023.
- William Malsam. (2023). *Cara Menulis Kasus Uji (Termasuk Contoh & Template)*. IT & Software Development, Oct 2023.

PROFIL PENULIS



Oleh Soleh S.Kom., M.MSI.

Ketertarikan penulis terhadap ilmu komputer dimulai pada tahun 1996 silam. Hal tersebut membuat penulis memilih untuk masuk ke Universitas Gunadarma dengan memilih Jurusan Manajemen Informatika (MI) pada fakultas Ilmu Komputer dan berhasil menyelesaikan program studi S1 dengan kelulusan pada tahun 2000. Penulis kemudian melanjutkan pendidikan ke jenjang master yang masih di kampus yang sama pada tahun 2004 berhasil menyelesaikan studi S2 di prodi Ilmu Komputer dengan Jurusan Manajemen Informasi Bisnis. Mulai tahun 2022 sampai saat ini sedang menempuh pendidikan doktoral di Universitas Kristen Satya Wacana Salatiga melalui prodi Ilmu Komputer dengan konsentrasi *Deep Learning & Computer Vision*.

Penulis memiliki kepakaran dibidang Business Intelligence, Data Mining, Manajemen Informasi serta Deep Learning. Dan untuk mewujudkan karir sebagai dosen profesional, penulis pun aktif sebagai peneliti dibidang kepakarannya tersebut. Beberapa penelitian yang telah dilakukan didanai oleh internal perguruan tinggi.

Email Penulis: oleh.soleh@raharja.info.

BAB 14

IMPLEMENTASI PERANGKAT LUNAK

Euis Nur Fitriani Dewi, S.T., M.Kom.
Universitas Siliwangi Tasikmalaya

Pengertian Perangkat Lunak

Perangkat lunak merupakan suatu sistem komputer yang tidak memiliki wujud fisik yang diinstal dalam sebuah komputer atau laptop agar bisa dijalankan. Perangkat lunak juga dapat diartikan sebagai sebuah perintah program dalam sebuah komputer, yang apabila diberi perintah oleh user akan memberikan fungsi dan unjuk kerja seperti yang diharapkan oleh user-nya (Roger, 2002). Beberapa contoh dari perangkat lunak diantaranya yaitu *Microsoft Word*, *Microsoft Excel*, *Microsoft Power Point*, dan lain sebagainya.

Perangkat lunak memiliki beberapa fungsi, diantaranya:

1. Memproses data, memberi perintah, ataupun instruksi khusus.
2. Sebagai sarana interaksi antara user dengan perangkat kerasnya.
3. Mengidentifikasi suatu program.
4. Menyediakan fungsi dasar pengoperasian komputer

Implementasi Perangkat Lunak

Implementasi perangkat lunak adalah proses mengintegrasikan aplikasi ke dalam aplikasi alur kerja organisasi. Prosesnya biasanya dimulai dengan memilih vendor dan menetapkan anggaran. Langkah selanjutnya mungkin melibatkan instalasi aplikasi, migrasi data dan pengujian berbagai fitur. Perusahaan menggunakan implementasi perangkat lunak contohnya saat ada aplikasi baru untuk proyek

manajemen, manajemen rantai pasokan dan perencanaan sumber daya perusahaan. Proses ini juga dapat diterapkan ketika tim pengembangan memperbaharui tim yang sudah menggunakan aplikasi.

1. Mengapa Implementasi Perangkat Lunak itu Penting

Implementasi perangkat lunak penting karena memungkinkan perusahaan untuk mengakses teknologi terkini, dengan mengganti aplikasi lama dengan perangkat lunak baru, karyawan dapat meningkatkan produktivitasnya dan menghasilkan pekerjaan yang lebih berkualitas. Aplikasi baru juga dapat meningkatkan kepuasan pelanggan dengan menyediakan klien dengan pengalaman yang lebih ramah pengguna. Sebuah perusahaan dapat meningkatkan peluangnya menghasilkan manfaat ini dengan menggunakan proses yang efektif untuk perangkat lunak penerapan. Misalnya, perusahaan mungkin memilih aplikasi sesuai anggarannya dan kompatibel dengan sistem yang ada. Penting juga untuk memilih aplikasi yang dapat diinstal oleh organisasi seefisien mungkin meminimalkan waktu henti.



Gambar 15.1: Implementasi Perangkat Lunak

Sumber: (Yayimage, 2017).

2. Standar Keberhasilan Implementasi Perangkat Lunak

Standar suatu implementasi perangkat lunak dapat dikatakan baik, yaitu teknik pemecahan masalah, penyusunan program, perawatan program, dan standar prosedur.

a. Teknik Pemecahan Masalah

2. *Software Design Documentation*

Dokumentasi ini berisi semua informasi yang diperlukan, yang diperlukan untuk membangun perangkat lunak, terdiri dari arsitektur perangkat lunak tingkat tinggi, desain perangkat lunak.

3. *Technical Documentation*

Dokumentasi ini dikelola oleh pengembang dan pembuat kode sebenarnya. Dokumen-dokumen ini, secara keseluruhan, mewakili informasi tentang kode. Saat menulis kode, pemrogram juga menyebutkan tujuan kode tersebut, siapa yang menulisnya, di mana apakah hal tersebut diperlukan, apa fungsinya dan bagaimana cara kerjanya, apa saja sumber daya lainnya kode yang digunakan, dll.

Software Implementation Challenges

Ada beberapa tantangan yang dihadapi oleh tim pengembangan saat ini mengimplementasikan perangkat lunak. Beberapa di antaranya disebutkan di bawah ini:

1. *Code-reuse*

Antarmuka Bahasa pemrograman saat ini sangat beragam dan canggih, serta dilengkapi fungsi *library*. Masalah yang terjadi adalah biasanya para pengguna perangkat lunak, terutama Perusahaan besar lebih memilih untuk menggunakan kembali kode yang telah dibuat sebelumnya untuk beberapa perangkat lunak lain. Hal tersebut yang menjadi masalah besar yang dihadapi programmer untuk proses pemeriksaan kompatibilitas dan memutuskan berapa banyak kode yang akan digunakan kembali.

2. *Version Management*

Setiap kali perangkat lunak baru dikeluarkan untuk pelanggan, pengembang harus mempertahankan versi dan konfigurasi dokumentasi terkait. Dokumentasi ini harus sangat akurat dan tersedia tepat waktu.

3. *Target-Host*

Perangkat lunak yang sedang dikembangkan perlu dirancang untuk mesin host pada *customers end* (pengguna). Namun terkadang, tidak mungkin merancang perangkat lunak yang dapat berfungsi pada perangkat tersebut mesin sasaran.

Daftar Pustaka

- Moad, J. (1990). *Maintainability the Competitive Edge, Datamotion*. Vol 36, No 4, pp.61-66.
- Prashant Palvia, Aaron Ptula and John Nosek. (1995). Problems and Issues in Application Software Maintenance Management. *Journal of Information Technology Management*. Volume VI, Number 3, 1995.
- Syed Sarmad Ali, Muhammad Shoaib Zafar, and Muhammad Tallal Saeed. (2020). Effort Estimation Problems in Software Maintenance – A Survey. *Conference Paper*, Januari 2020.
- Yayimage (2017). Icon Implementasi Perangkat Lunak. https://id.pngtree.com/freepng/editable-software-implementation-icon-for-mobile-app-programming-vector_12491925.html/ diakses tanggal 22 Januari 2024.
- <https://adoc.pub/implementasi-perangkat-lunak.html>/diakses tanggal 22 Januari 2024.

PROFIL PENULIS



Euis Nur Fitriani Dewi, S.T., M.Kom.

Penulis tertarik pada bidang Informatika saat memasuki jenjang Pendidikan SMA, yaitu di SMAN 1 Banjar. Setelah lulus dari SMA, penulis berkeinginan untuk menjadi guru TIK, namun hal itu diurungkan dan akhirnya penulis melanjutkan ke jenjang S1 pada tahun 2007 di Universitas Siliwangi di Kota Tasikmalaya. Lulus S1 kemudian bekerja di sebuah bank swasta, selama 1,5 tahun.

Setelah *resign* dari bank tersebut, penulis mengikuti seleksi beasiswa Pendidikan Profesi Guru (PPG) Pra Jabatan di Universitas Negeri Jakarta selama satu tahun (2013-2014). Lulus PPG mengajar di satu SMK swasta dan satu SMK Negeri di Kota Tasikmalaya. Sembari mengajar di SMK, kemudian melanjutkan Pendidikan jenjang S2 di STMIK Likmi Bandung pada akhir 2016-Juli 2018. Pada akhir 2018, penulis mengikuti seleksi dosen tetap non PNS di Universitas Siliwangi Tasikmalaya dan Alhamdulillah lulus. Penulis resign dari SMK dan fokus menjadi Dosen di Universitas Siliwangi. Selang 3 tahun, penulis mengikuti seleksi Dosen CPNS dan lulus menjadi Dosen PNS Prodi Informatika pada tahun 2022.

Email Penulis: euis.nurfitriani@unsil.ac.id.

BAB 15

PEMELIHARAAN PERANGKAT LUNAK

Saryani, S.Kom., M.TI.
Universitas Tangerang Raya

Pendahuluan

Untuk memulai persepsi mengenai perawatan perangkat lunak terlebih dahulu kita samakan persepsi mengenai apa itu perawatan perangkat lunak? Karena banyak mahasiswa dan insinyur yang berpraktik memiliki prasangka terhadap pekerjaan pemeliharaan perangkat lunak. Penyebutan kata perawatan memunculkan citra seorang tukang obeng, montir yang memegang dengan tangan kotor memegang sekantong penuh suku cadang. Tujuan bab ini adalah untuk menjernihkan istilah yang salah ini, memberikan beberapa pemahaman intuitif tentang proyek pemeliharaan perangkat lunak (Migunani, 2022)

Pentingnya menjernihkan persepsi diatas tergambar dari tujuan untuk memberikan pemahaman intuitif tentang proyek pemeliharaan perangkat lunak. Pemeliharaan perangkat lunak mencakup pemahaman mendalam terhadap struktur dan logika perangkat lunak, identifikasi dan perbaikan bug, peningkatan fungsionalitas, optimasi kinerja, pemeliharaan dokumentasi, dan pemantauan keamanan. Oleh karena itu, perawatan perangkat lunak adalah suatu bidang yang membutuhkan keterampilan analitis, pemahaman bahasa pemrograman, dan kreativitas.

Daftar Pustaka

- Ali, E. (2019). *Rekayasa Perangkat Lunak*. CV MFA.
- Alif, F. (2018). Software Maintenance. *ACM SIGSOFT Software Engineering Notes*, 10(5), 89–98. <https://doi.org/10.1145/382288.382774>.
- Hasanah, F. N., & Utami, R. S. (2020). *Buku Ajar Rekayasa Perangkat Lunak*. UMSIDA Press.
- Migunani. (2022). *Rekayasa*. Yayasan Prima Agus Teknik.
- Parlika, R., Lintang, H., R., I. S., Suriansyah, M., Humania, N., Yulianto, E., & N., R. A. (2018). *Pengaruh Mutu dan Pemeliharaan Perangkat Lunak Pada Kelayakan dan Hasil Validasi Aplikasi Universitas Pembangunan Nasional "Veteran" Jawa Timur*.

PROFIL PENULIS



Saryani, S.Kom., M.TI.

Ketertarikan penulis terhadap ilmu komputer dimulai pada tahun 2003 silam. Hal tersebut membuat penulis memilih untuk masuk ke Universitas Raharja dengan Jurusan Sistem Informasi konsentrasi Sistem Informasi Manajemen dan berhasil lulus pada tahun 2008. Penulis kemudian melanjutkan pendidikan ke Universitas Raharja dan menyelesaikan studi S2 di Magister Teknik Informatika pada tahun 2016 dengan jurusan *Business Intelligence* dan lulus pada tahun 2018. Penulis memiliki kepakaran dibidang Analisis dan Perancangan Sistem Informasi dan Sistem Informasi Manajemen juga pemrograman web, untuk mewujudkan karir sebagai dosen profesional, penulis pun aktif dalam seminar dan sebagai peneliti dibidang kepakarannya tersebut.

Beberapa penelitian yang telah dilakukan didanai oleh internal perguruan tinggi dan juga eksternal Perguruan Tinggi lainnya. Selain peneliti, penulis juga aktif dalam pengabdian kepada masyarakat agar dapat menyeimbangkan diri sehingga bisa bersinergi dengan masyarakat dan menulis buku dengan harapan dapat memberikan kontribusi positif bagi bangsa dan negara yang sangat tercinta ini, besar harapan penulis agar masyarakat Indonesia khususnya dapat meningkatkan minat membaca untuk menambah pengetahuan dan informasi.

Email Penulis: 0709saryani@gmail.com.

REKAYASA PERANGKAT LUNAK

Rekayasa Perangkat Lunak (*Software Engineering*) adalah disiplin ilmu yang berkaitan dengan pengembangan, perancangan, pemeliharaan, dan pengelolaan perangkat lunak. Fokus utamanya adalah menciptakan solusi perangkat lunak yang efisien, handal, dan sesuai dengan kebutuhan pengguna. Rekayasa Perangkat Lunak terus berkembang seiring dengan kemajuan teknologi dan perubahan kebutuhan industri. Pemahaman yang mendalam tentang prinsip-prinsip ini menjadi kunci kesuksesan dalam menciptakan solusi perangkat lunak yang inovatif dan berkualitas tinggi. Dengan pengalaman praktis dan wawasan mendalam, penulis membahas tantangan umum yang dihadapi dalam pengembangan perangkat lunak dan memberikan strategi yang terbukti untuk mengatasinya. Selain itu, buku ini mencakup tren terkini dalam industri rekayasa perangkat lunak, seperti pengembangan berbasis cloud, kecerdasan buatan, dan metode pengujian otomatis.

Dengan hadirnya buku ini menjadi panduan esensial bagi para profesional, mahasiswa, dan praktisi Rekayasa Perangkat Lunak, memberikan wawasan mendalam tentang pendekatan sistematis dalam pengembangan perangkat lunak. Penulis, seorang ahli dalam industri IT dengan pengalaman panjang, membimbing pembaca melalui berbagai tahapan siklus hidup pengembangan perangkat lunak dengan penekanan pada praktik terbaik dan metodologi mutakhir.