



Rekayasa **Perangkat Lunak**

Fingki Marwati
Afrizal Zein



Rekayasa Perangkat Lunak

Tentang Penulis



Fingki Marwati, S.Kom, M.Kom. Lahir di Cianjur Jawa Barat pada tanggal 08 Maret 1995 merupakan dosen tetap di Universitas Pamulang. Telah menamatkan S1 di Universitas Pamulang dan Lulus S2 di STMIK ERESHA pada tahun 2020. Mempunyai pengalaman di bidang industri sebagai Admin di PT RoseWood Living dan Kepala Sekolah di PKBM Aulia. Bidang keilmuan yang diminati adalah Ilmu Komputer.



Drs. Afrizal Zein M.Kom. Lahir di Jakarta 13 Juli 1965 merupakan dosen tetap di Universitas Pamulang. Telah menamatkan S1 di Universitas Padjadjaran dan Lulus S2 di STMIK ERESHA pada tahun 2014 dengan predikat Cumlaude. Berpengalaman sebagai programmer diberbagai project dan membangun Aplikasi Komputer selama 25 tahun bekerja di Konsultan Komputer. Memiliki sertifikasi dalam bidang Pemograman dan Sistem Analisis.

REKAYASA PERANGKAT LUNAK

Fingki Marwati
Afrizal Zein



eureka
media aksara

PENERBIT CV.EUREKA MEDIA AKSARA

REKAYASA PERANGKAT LUNAK

Penulis : Fingki Marwati
Afrizal Zein

Desain Sampul : Eri Setiawan

Tata Letak : Nur Cholifatun Nisa

ISBN : 978-623-120-878-1

Diterbitkan oleh : **EUREKA MEDIA AKSARA, JUNI 2024**
ANGGOTA IKAPI JAWA TENGAH
NO. 225/JTE/2021

Redaksi:

Jalan Banjaran, Desa Banjaran RT 20 RW 10 Kecamatan Bojongsari
Kabupaten Purbalingga Telp. 0858-5343-1992

Surel : eurekaediaaksara@gmail.com

Cetakan Pertama : 2024

All right reserved

Hak Cipta dilindungi undang-undang

Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku ini dalam bentuk apapun dan dengan cara apapun, termasuk memfotokopi, merekam, atau dengan teknik perekaman lainnya tanpa seizin tertulis dari penerbit.

KATA PENGANTAR

Perangkat lunak (*software*) adalah serangkaian instruksi yang memungkinkan komputer untuk melakukan tugas-tugas tertentu. Rekayasa perangkat lunak merupakan disiplin ilmu yang berfokus pada metodologi, proses, dan alat untuk merancang, mengembangkan, memelihara, dan mengelola perangkat lunak.

Buku ini terdiri dari 9 bab yang membahas mengenai:

Bab 1 Perangkat Lunak dan Rekayasa Perangkat Lunak

Bab 2 Proses Model Perangkat Lunak

Bab 3 *Agile Development*

Bab 4 Desain Dalam Konteks Rekayasa Perangkat Lunak

Bab 5 Arsitektur Design

Bab 6 *User Interface Design*

Bab 7 Jaminan Kualitas Perangkat Lunak

Bab 8 Strategi Testing Perangkat Lunak

Bab 9 Penjadwalan Proyek

Buku ini diharapkan mampu memberikan pengetahuan bahwa dalam era digital saat ini, perangkat lunak menjadi fondasi dari banyak sistem dan layanan yang digunakan oleh individu, organisasi, dan pemerintah. Dari aplikasi mobile yang sederhana hingga sistem informasi bisnis yang kompleks, setiap produk perangkat lunak memerlukan pendekatan yang sistematis dan terstruktur untuk memastikan bahwa produk akhir memenuhi kebutuhan dan harapan pengguna.

Kemudian, kami ingin menyatakan apresiasi yang tulus kepada semua pihak yang telah memberikan dukungan dan bantuan mereka dalam penyelesaian buku ini. Semoga buku ini dapat memberikan manfaat yang berarti bagi pembaca dan pemangku kepentingan lainnya.

Hormat kami,

Penulis

DAFTAR ISI

KATA PENGANTAR.....	iii
DAFTAR ISI.....	iv
BAB 1 PERANGKAT LUNAK DAN REKAYASA	
PERANGKAT LUNAK	1
A. Latar Belakang dan Definisi Perangkat Lunak	1
B. Rekayasa Perangkat Lunak	6
C. Aplikasi Perangkat Lunak	13
D. <i>Software Engineering</i>	18
BAB 2 PROSES MODEL PERANGKAT LUNAK.....	22
A. Pendahuluan.....	22
B. Model Proses Preskriptif.....	30
C. <i>Specialized Process Models</i>	37
D. Paradigma Pembuatan Prototipe Dalam RPL	44
BAB 3 AGILE DEVELOPMENT.....	51
A. Apa Itu Agile ?.....	51
B. Agile Proses	52
C. <i>Extreme Programming (Xp)</i>	60
D. <i>XP Proses</i>	65
BAB 4 DESAIN DALAM KONTEKS REKAYASA	
PERANGKAT LUNAK	71
A. Pendahuluan.....	71
B. Definisi Desain Perangkat Lunak	71
C. Prinsip-Prinsip Desain Perangkat Lunak	74
D. Metodologi Design Perangkat Lunak.....	76
E. Pola Design Perangkat Lunak	80
BAB 5 ARSITEKTUR DESIGN.....	89
A. Pendahuluan.....	89
B. Arsitektur dalam Konteks Sejarah	91
C. Prinsip-prinsip Desain Arsitektur.....	92
D. Inovasi dan Teknologi dalam Arsitektur Modern.....	92
E. Software Arsitektur	93
F. Arsitektur Design RPL.....	95
BAB 6 USER INTERFACE DESIGN	99
A. <i>User Interface</i> Analisis dan Design	99
B. Peran dan Tanggung Jawab UI Analisis	101

C. Interface Analisis	103
D. Peran dan Tanggung Jawab User Analisis	105
BAB 7 JAMINAN KUALITAS PERANGKAT LUNAK.....	108
A. Pendahuluan	108
B. Tugas SQA.....	109
C. Peran dan Pentingnya Jaminan Kualitas Perangkat Lunak	111
D. Proses Jaminan Kualitas Perangkat Lunak.....	112
BAB 8 STRATEGI TESTING PERANGKAT LUNAK.....	117
A. Pendahuluan	117
B. Komponen Utama Strategi <i>Testing</i> Perangkat Lunak	118
C. Strategi Uji untuk Perangkat Lunak Konvensional ...	121
D. <i>White-box Testing</i>	126
E. <i>Black-Box Testing</i>	131
BAB 9 PENJADWALAN PROYEK.....	136
A. Konsep Penjadwalan Proyek	136
B. Komponen Utama Penjadwalan Proyek.....	140
C. Teknik Penjadwalan Proyek	141
D. Analisis Nilai Yang Diperoleh	143
DAFTAR PUSTAKA	148
TENTANG PENULIS	150



REKAYASA PERANGKAT LUNAK

**Fingki Marwati
Afrizal Zein**



BAB 1

PERANGKAT LUNAK DAN REKAYASA PERANGKAT LUNAK

A. Latar Belakang dan Definisi Perangkat Lunak

Perangkat lunak, atau lebih dikenal dengan istilah software dalam bahasa Inggris, merupakan elemen fundamental dalam teknologi informasi dan komunikasi modern. Perangkat lunak adalah serangkaian instruksi yang memungkinkan komputer untuk melakukan tugas-tugas tertentu. Dari sudut pandang teknis, perangkat lunak mencakup program komputer, aplikasi, dan sistem operasi yang mengatur dan mengendalikan perangkat keras komputer. Perangkat lunak memainkan peran penting dalam berbagai aspek kehidupan manusia, mulai dari aktivitas sehari-hari hingga operasi bisnis yang kompleks dan pengembangan teknologi canggih.



Gambar 1. 1 Contoh Perangkat Lunak (*Software*)

BAB 2

PROSES MODEL PERANGKAT LUNAK

A. Pendahuluan

Rekayasa Perangkat Lunak (RPL) adalah disiplin ilmu yang berfokus pada metodologi, alat, dan teknik yang digunakan untuk merancang, mengembangkan, menguji, dan memelihara perangkat lunak. Salah satu elemen kunci dalam RPL adalah proses model perangkat lunak, yang menyediakan kerangka kerja terstruktur untuk mengelola siklus hidup perangkat lunak dari awal hingga akhir. Proses model ini mencakup berbagai pendekatan dan metodologi yang dirancang untuk memastikan bahwa perangkat lunak yang dihasilkan memenuhi kebutuhan pengguna, berkualitas tinggi, dan dapat dikelola dengan baik.

1. Pengertian Proses Model Perangkat Lunak

Proses model perangkat lunak adalah sekumpulan langkah atau tahapan yang diikuti selama pengembangan perangkat lunak. Model ini mencakup aktivitas seperti analisis kebutuhan, desain, pengkodean, pengujian, dan pemeliharaan. Proses model berfungsi sebagai panduan bagi tim pengembang untuk memastikan bahwa setiap tahap dalam siklus hidup perangkat lunak dilakukan dengan cara yang terorganisir dan efisien.

BAB

3

AGILE DEVELOPMENT

A. Apa Itu Agile ?

Agility telah menjadi kata kunci saat ini ketika menggambarkan proses perangkat lunak modern. Setiap orang gesit. Tim yang tangkas adalah tim yang gesit dan mampu merespons perubahan dengan tepat. Perubahan adalah inti dari pengembangan perangkat lunak. Perubahan perangkat lunak yang sedang dibangun, perubahan anggota tim, perubahan karena teknologi baru, perubahan segala macam yang mungkin berdampak pada produk yang mereka buat atau proyek yang menciptakannya produk. Dukungan untuk perubahan harus ada di dalam semua yang kita lakukan dalam perangkat lunak, sesuatu kami menerimanya karena ini adalah inti dan jiwa dari perangkat lunak. Tim yang tangkas menyadari hal itu perangkat lunak dikembangkan oleh individu yang bekerja dalam tim dan keterampilan orang-orang ini, kemampuan mereka untuk berkolaborasi adalah inti keberhasilan proyek. Dalam pandangan Jacobson, luasnya perubahan adalah pendorong utama kelincahan. Insinyur perangkat lunak harus cepat bertindak jika ingin mengakomodasi perubahan yang cepat perubahan yang dijelaskan Jacobson.

Namun kelincahan lebih dari sekadar respons efektif terhadap perubahan. Ini juga mencakup filsafat yang dianut dalam manifesto yang disebutkan di awal bab ini. Dia mendorong struktur tim dan sikap yang membuat komunikasi (antar tim anggota, antara ahli teknologi dan pebisnis, antara insinyur perangkat lunak dan manajernya) lebih lancar. Hal ini

BAB 4

DESAIN DALAM KONTEKS REKAYASA PERANGKAT LUNAK

A. Pendahuluan

Desain perangkat lunak adalah fase kritis dalam rekayasa perangkat lunak yang menghubungkan analisis kebutuhan dengan implementasi. Buku ini membahas konsep desain dalam konteks rekayasa perangkat lunak, termasuk prinsip-prinsip dasar, metodologi, pola desain, alat bantu, dan studi kasus implementasi desain perangkat lunak. Tujuan dari buku ini adalah untuk memberikan wawasan mendalam tentang pentingnya desain yang baik dalam menghasilkan perangkat lunak yang berkualitas dan dapat dipelihara.

Rekayasa perangkat lunak adalah disiplin yang menggabungkan prinsip-prinsip teknik dengan praktik pengembangan perangkat lunak untuk menghasilkan produk yang dapat diandalkan, efisien, dan berkualitas tinggi. Salah satu aspek yang paling penting dalam rekayasa perangkat lunak adalah desain perangkat lunak, yang berperan sebagai cetak biru untuk implementasi dan pemeliharaan sistem perangkat lunak.

B. Definisi Desain Perangkat Lunak

Desain perangkat lunak adalah proses mendefinisikan arsitektur, komponen, antarmuka, dan data dari suatu sistem perangkat lunak untuk memenuhi kebutuhan yang telah ditentukan. Ini adalah langkah penting dalam rekayasa perangkat lunak yang menghubungkan hasil analisis kebutuhan dengan implementasi teknis. Desain perangkat lunak berfungsi sebagai cetak biru yang memandu pengembang dalam

BAB

5

ARSITEKTUR DESIGN

A. Pendahuluan

Ketika Anda mempertimbangkan arsitektur suatu bangunan, banyak atribut berbeda yang muncul pikiran. Pada tingkat yang paling sederhana, Anda memikirkan tentang bentuk fisik secara keseluruhan struktur. Namun kenyataannya, arsitektur lebih dari itu. Ini adalah cara di mana berbagai komponen bangunan diintegrasikan untuk membentuk satu kesatuan yang kohesif. Ini adalah cara bangunan tersebut menyesuaikan diri dengan lingkungannya dan menyatu dengan bangunan lain di sekitarnya. Merupakan sejauh mana bangunan memenuhi tujuan yang ditetapkan dan memenuhi kebutuhan pemilikinya. Ini adalah nuansa estetis struktur dampak visual bangunan dan cara tekstur, warna, dan bahan digabungkan untuk membentuk bangunan. menciptakan fasad eksternal dan “lingkungan hidup” internal. Itu adalah detail kecil desain perlengkapan pencahayaan, jenis lantai, penempatan hiasan dinding, daftarnya hampir tidak ada habisnya. Dan yang terakhir adalah seni. Namun arsitektur juga merupakan sesuatu yang lain. Ini adalah “ribuan keputusan, baik besar maupun besar kecil”. Beberapa dari keputusan ini dibuat pada awal desain dan dapat berdampak dampak besar pada semua tindakan desain lainnya. Oleh karena itu, yang lain ditunda sampai nanti Apa saja langkah-langkahnya? Desain arsitektur dimulai dengan desain data dan kemudian melanjutkan ke penurunan satu atau lebih representasi struktur arsitektur sistem. Alternatif gaya atau pola arsitektur dianalisis

BAB 6

USER INTERFACE DESIGN

A. *User Interface Analisis dan Design*

Empat model berbeda ikut berperan ketika antarmuka pengguna dianalisis dan dirancang. Seorang insinyur manusia (atau insinyur perangkat lunak) menetapkan model pengguna, yaitu insinyur perangkat lunak menciptakan model desain, pengguna akhir mengembangkan gambaran mental itu sering disebut model mental pengguna atau persepsi sistem, dan pelaksana sistem membuat model implementasi. Sayangnya, masing-masing model tersebut mungkin berbeda secara signifikan. Peran Anda, sebagai perancang antarmuka, adalah untuk merekonsiliasi perbedaan-perbedaan ini dan memperoleh representasi antarmuka yang konsisten.

Model pengguna menetapkan profil pengguna akhir sistem. Untuk membangun antarmuka pengguna yang efektif, “semua desain harus dimulai dengan pemahaman pengguna yang dituju, termasuk profil usia, jenis kelamin, kemampuan fisik, pendidikan, latar belakang budaya atau etnis, motivasi, tujuan dan kepribadian”. Di dalam Selain itu, pengguna dapat dikategorikan sebagai pemula. Tidak ada pengetahuan sintaksis¹ tentang sistem dan sedikit pengetahuan semantik dari aplikasi atau penggunaan komputer secara umum. Pengguna yang berpengetahuan luas dan terputus-putus. Pengetahuan semantik yang masuk akal tentang aplikasi tetapi daya ingat yang relatif rendah terhadap informasi sintaksis yang diperlukan untuk menggunakan antarmuka. Pengguna yang berpengetahuan luas dan sering. Pengetahuan semantik dan

BAB

7

JAMINAN KUALITAS PERANGKAT LUNAK

A. Pendahuluan

Pengendalian dan jaminan mutu merupakan aktivitas penting bagi setiap bisnis yang memproduksi produk untuk digunakan oleh orang lain. Sebelum abad kedua puluh, pengendalian kualitas adalah hal yang paling penting tanggung jawab sepenuhnya dari pengrajin yang membuat suatu produk. Seiring berjalannya waktu dan massa teknik produksi menjadi hal yang lumrah, pengendalian kualitas menjadi suatu kegiatan dilakukan oleh orang lain selain orang yang membuat produk tersebut.

Fungsi jaminan dan pengendalian kualitas formal pertama diperkenalkan di Bell Labs pada tahun 1916 dan menyebar dengan cepat ke seluruh dunia manufaktur. Selama Pada tahun 1940-an, pendekatan pengendalian mutu yang lebih formal diusulkan. Ini bergantung pada pengukuran dan perbaikan proses berkelanjutan sebagai elemen kunci manajemen mutu.

Saat ini, setiap perusahaan memiliki mekanisme untuk menjamin kualitas produknya. Nyatanya, pernyataan eksplisit tentang kepedulian perusahaan terhadap kualitas telah menjadi suatu pemasaran taktik selama beberapa dekade terakhir. Sejarah penjaminan mutu dalam pengembangan perangkat lunak sejajar dengan sejarah kualitas dalam pembuatan perangkat keras. Selama hari-hari awal komputasi (1950an dan 1960an), kualitas adalah tanggung jawab programmer. Standar kualitas jaminan untuk perangkat lunak diperkenalkan dalam

BAB 8

STRATEGI *TESTING* PERANGKAT LUNAK

A. Pendahuluan

Strategi *testing* perangkat lunak adalah rencana menyeluruh yang menetapkan pendekatan dan teknik yang akan digunakan untuk menguji perangkat lunak guna memastikan bahwa perangkat lunak tersebut bebas dari cacat, memenuhi kebutuhan pengguna, dan sesuai dengan spesifikasi yang telah ditentukan. Strategi *testing* yang baik mencakup berbagai jenis pengujian, alat, dan teknik yang akan diterapkan pada berbagai tahap siklus hidup pengembangan perangkat lunak. Pengujian merupakan serangkaian kegiatan yang dapat direncanakan terlebih dahulu dan dilakukan secara sistematis. Oleh karena itu, template untuk pengujian perangkat lunak seperangkat langkah yang harus Anda lakukan dapat menempatkan teknik desain kasus uji dan metode pengujian tertentu seharusnya ditentukan untuk proses perangkat lunak. Sejumlah strategi pengujian perangkat lunak telah diusulkan dalam literatur. Semua memberi Anda templat untuk pengujian dan semuanya memiliki generik berikut karakteristik. Untuk melakukan pengujian yang efektif, Anda harus melakukan tinjauan teknis yang efektif. Dengan melakukan ini, banyak kesalahan akan dihilangkan sebelum pengujian dimulai.

1. Pengujian dimulai pada tingkat komponen dan berjalan “ke luar” menuju ke tingkat komponen integrasi seluruh sistem berbasis komputer.

BAB 9

PENJADWALAN PROYEK

A. Konsep Penjadwalan Proyek

Meskipun ada banyak alasan mengapa perangkat lunak terlambat dikirimkan, sebagian besar dapat ditelusuri pada satu atau lebih akar penyebab berikut:

1. Batas waktu yang tidak realistis yang ditetapkan oleh seseorang di luar tim perangkat lunak dan dipaksakan pada manajer dan praktisi.
2. Mengubah persyaratan pelanggan yang tidak tercermin dalam perubahan jadwal.
3. Benar-benar meremehkan jumlah upaya dan/atau jumlah upaya yang dilakukan sumber daya yang diperlukan untuk melakukan pekerjaan tersebut.
4. Risiko yang dapat diprediksi dan/atau tidak dapat diprediksi yang tidak dipertimbangkan pada saat terjadinya proyek dimulai.
5. Kesulitan teknis yang tidak dapat diperkirakan sebelumnya.
6. Kesulitan manusia yang tidak dapat diperkirakan sebelumnya.
7. Miskomunikasi antar staf proyek yang mengakibatkan penundaan.
8. Kegagalan manajemen proyek untuk menyadari bahwa proyek tersebut gagal
9. Terlambat dari jadwal dan kurangnya tindakan untuk memperbaiki masalah.

DAFTAR PUSTAKA

- Abbott, R., "Program Design by Informal English Descriptions," *CACM*, vol. 26, no. 11, November 1983, pp. 892-894.
- ACM/IEEE-CS Joint Task Force, *Software Engineering Code of Ethics and Professional Practice*, 1998, available at www.acm.org/serving/se/code.htm.
- Adams, D., *Mostly Harmless*, Macmillan, 1993. *Software Risk Abatement*, AFCS/AFLC Pamphlet 800-45, U.S. Air Force, September 30, 1988.
- Airlie Council, "Performance Based Management: The Program Manager's Guide Based on the 16-Point Plan and Related Metrics," Draft Report, march 8, 1999.
- Akao, Y., *Quality Function Deployment*, Productivity Press, 2004.
- Alexander, C., *A Pattern Language*, Oxford University Press, 1977.
- Alexander, C., *The Timeless Way of Building*, Oxford University Press, 1979.
- Ambler, S., "Examining the Cost of Change Curve," in *The Object Primer*, 3d ed., Cambridge University Press, 2004.
- Ambler, S., "UML Component Diagramming Guidelines," available at www.modelingstyle.info/, 2002.
- Ambler, S., "Using Use-Cases," *Software Development*, July 1995, pp. 53-61.
- Ambler, S., "What Is Agile Modeling (AM)?" 2002, www.agilemodeling.com/index.htm.
- Ambler, S., and R. Jeffries, *Agile Modeling*, Wiley, 2002.
- Ambler, S., *Process Patterns: Building Large-Scale Systems Using Object Technology*,
- Ambler, S., *The Object Primer*, 2d ed., Cambridge University Press, 2001.

- Andrews, M., and J. Whittaker, *How to Break Web Software: Functional and Security*
- ANSI/ASQC A3-1987, *Quality Systems Terminology*, 1987.
- Anton, D., and C. Anton, *ISO 9001 Survival Guide*, 3d ed., AEM Consulting Group, 2006.
- AOSD.net (Aspect-Oriented Software Development), glossary, available at <http://aosd.net/wiki/index.php?title=Glossary>.
- Apple Computer, *Accessibility*, 2008, available at www.apple.com/disability/.
- Arlow, J., and I. Neustadt, *UML and the Unified Process*, Addison-Wesley, 2002.
- Arnold, R. S., "Software Restructuring," *Proc. IEEE*, vol. 77, no. 4, April 1989, pp. 607-617.
- Arthur, L. J., "Quantum Improvements in Software System Quality," *CACM*, vol. 40, no. 6, June 1997, pp. 47-52.
- Astels, D., *Test Driven Development: A Practical Guide*, Prentice Hall, 2004.
- Aversan, L., et al., "Managing Coordination and Cooperation in Distributed Software
- Babich, W. A., *Software Configuration Management*, Addison-Wesley, 1986.
- Bach, J., "Good Enough Quality: Beyond the Buzzword," *IEEE Computer*, vol. 30, no. 8, August 1997, pp. 96-98.
- Cambridge University Press/SIGS Books, 1998.
- Processes: The GENESIS Environment," *Software Process Improvement and Practice*, vol. 9, Wiley Interscience, 2004, pp. 239-263.
- Testing of Web Applications and Web Services*, Addison-Wesley, 2006.
- The Agile Alliance Home Page, www.agilealliance.org/home.

TENTANG PENULIS



Fingki Marwati, S.Kom., M.Kom. Lahir di Cianjur Jawa Barat pada tanggal 08 Maret 1995 merupakan dosen tetap di Universitas Pamulang. Telah menamatkan S1 di Universitas Pamulang dan Lulus S2 di STMIK ERESHA pada tahun 2020. Mempunyai pengalaman di bidang industri sebagai Admin di PT RoseWood Living dan Kepala Sekolah di PKBM Aulia. Bidang keilmuan yang diminati adalah Ilmu Komputer.



Drs. Afrizal Zein, M.Kom. Lahir di Jakarta 13 Juli 1965 merupakan dosen tetap di Universitas Pamulang. Telah menamatkan S1 di Universitas Padjadjaran dan Lulus S2 di STMIK ERESHA pada tahun 2014 dengan predikat Cumlaude. Berpengalaman sebagai programmer diberbagai project dan membangun Aplikasi Komputer selama 25 tahun bekerja di Konsultan Komputer. Memiliki sertifikasi dalam bidang Pemograman dan Sistem Analis.