

Memahami



Software Development Life Cycle

Angga Aditya Permana | David Agustriawan | Monika Evelin Johan
Melissa Indah Fianty | Samuel Ady Sanjaya | Rudi Sutomo | Wirawan Istiono
Suwito Pomalingo | Jansen Wiratama | Erick fernando | Ahmad Faza
Alexander Waworuntu





Memahami

Software Development Life Cycle

Buku ini adalah panduan praktis untuk memahami cara membuat perangkat lunak. Software Development Life Cycle (SDLC) adalah langkah-langkah yang harus diikuti untuk menciptakan program komputer yang handal. Buku ini menjelaskan SDLC dengan bahasa yang mudah dimengerti, mulai dari merencanakan proyek hingga menghasilkan perangkat lunak yang siap digunakan. Anda akan belajar tentang analisis kebutuhan, desain, pengembangan, pengujian, dan pemeliharaan perangkat lunak. Buku ini cocok untuk pemula yang ingin memahami dunia pengembangan perangkat lunak dan juga berguna bagi mereka yang ingin meningkatkan pemahaman mereka tentang SDLC. Dengan membaca buku ini, Anda akan memiliki dasar yang kuat untuk memulai karir dalam pengembangan perangkat lunak atau mengelola proyek perangkat lunak dengan lebih efisien. buku ini juga memberikan panduan praktis tentang bagaimana mengelola proyek perangkat lunak dengan efektif, termasuk pengaturan anggaran, manajemen waktu, dan pemantauan kemajuan proyek. Anda akan belajar bagaimana mengidentifikasi risiko dan cara mengatasinya selama proses pengembangan perangkat lunak.

Dengan membaca buku ini, Anda akan merasa lebih percaya diri dalam menghadapi tantangan dalam pengembangan perangkat lunak dan dapat berkontribusi lebih efektif dalam tim pengembangan. Ini adalah panduan yang praktis dan mudah dimengerti bagi siapa saja yang tertarik dalam dunia pengembangan perangkat lunak.



☎ 0858 5343 1992
✉ eurekaediaaksara@gmail.com
📍 Jl. Banjaran RT.20 RW.10
Bojongsari - Purbalingga 53362



MEMAHAMI SOFTWARE DEVELOPMENT LIFE CYCLE

Angga Aditya Permana
David Agustriawan
Monika Evelin Johan
Melissa Indah Fianty
Samuel Ady Sanjaya
Rudi Sutomo
Wirawan Istiono
Suwito Pomalingo
Jansen Wiratama
Erick Fernando
Ahmad Faza
Alexander Waworuntu



PENERBIT CV.EUREKA MEDIA AKSARA

MEMAHAMI SOFTWARE DEVELOPMENT LIFE CYCLE

Penulis : Angga Aditya Permana | David Agustriawan |
Monika Evelin Johan | Melissa Indah Fianty |
Samuel Ady Sanjaya | Rudi Sutomo | Wirawan
Istiono | Suwito Pomalingo | Jansen Wiratama |
Erick Fernando | Ahmad Faza | Alexander
Waworuntu

Desain Sampul : Eri Setiawan

Tata Letak : Rizki Rose Mardiana

ISBN : 978-623-151-544-5

Diterbitkan oleh : **EUREKA MEDIA AKSARA, SEPTEMBER 2023**
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 : eurekamediaaksara@gmail.com

Cetakan Pertama : 2023

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

Puji syukur selalu terucap kepada Allah SWT yang sampai saat ini telah memberikan nikmat sehat, sehingga penulis bisa menyelesaikan buku ini walaupun masih terdapat kendala yang masih dapat diselesaikan. Terima kasih juga penulis ucapkan kepada semua yang berkontribusi atas selesainya tulisan ini. Keterbatasan waktu menjadi salah satu hal yang menjadi kesulitan dalam penulisan buku ini. Namun berkat dukungan dari semua pihak, akhirnya tulisan ini dapat selesai tepat waktu. Penulis menyadari masih banyak kekurangan dalam tulisan ini. Oleh karena itu penulis memohon maaf atas kesalahan yang mungkin ada pada buku ini.

Penulis berharap buku yang berjudul “Memahami Software Development Life Cycle” bisa bermanfaat bagi pembaca. Mohon untuk memaklumi jika terdapat penjelasan yang sulit untuk dimengerti. Untuk itu penulis mengharapkan kritik maupun saran, sehingga penulis bisa memperbaikinya dikemudian hari. Terimakasih atas ketertarikan Anda untuk membaca buku yang penulis buat.

DAFTAR ISI

KATA PENGANTAR.....	iii
DAFTAR ISI.....	iv
DAFTAR TABEL	viii
DAFTAR GAMBAR.....	ix
BAB 1 PENDAHULUAN.....	1
A. Pendahuluan	1
B. Perencanaan.....	4
C. Pengembangan.....	6
D. Penerapan	9
E. Manajemen Perubahan Organisasi dan SDLC	11
F. IT Governance dan SDLC.....	14
G. Metode Manajemen Proyek dan SDCL.....	16
H. Mencoba Semuanya	19
I. Harapan	21
DAFTAR PUSTAKA	25
BAB 2 WATERFALL MODEL	26
A. Latar belakang dan Definisi	26
B. Kegunaan Waterfall Model	32
C. Implementasi Waterfall Model dalam Pengembangan Website	34
D. Implementasi Waterfall Model dalam Pengembangan Game	46
DAFTAR PUSTAKA	52
BAB 3 SPIRAL MODEL	55
A. Apa itu Spiral Model?.....	55
B. Tahapan dalam Model Spiral.....	58
C. Kegunaan Model Spiral.....	65
D. Perbandingan dengan Model Lain.....	67
E. Implementasi Model Spiral.....	69
DAFTAR PUSTAKA	78
BAB 4 V-MODEL	79
A. Pendahuluan	79
B. Sejarah Singkat V-Model.....	80
C. Deskripsi Proses yang Komprehensif	82

	D. Pertimbangan Kritis dalam V-Model.....	85
	E. Rekayasa Sistem V-Model	88
	F. V-Model.....	90
	DAFTAR PUSTAKA.....	104
BAB 5	ADAPTIVE SOFTWARE DEVELOPMENT	106
	A. Pendahuluan.....	106
	B. ASD Life Cycle.....	107
	C. Pendekatan ASD.....	109
	D. Perbandingan ASD dengan Metode Lain.....	111
	E. Kelebihan dan Kekurangan Adaptive Software Development.....	116
	F. Contoh Penggunaan ASD.....	118
	G. Kesesuaian Penggunaan ASD	120
	H. Adaptive Software Development dan Transformasi SuperApp.....	121
	I. Kesimpulan	124
	DAFTAR PUSTAKA.....	126
BAB 6	INCREMENTAL MODEL	127
	A. Pendahuluan.....	127
	B. Konsep Dasar Model <i>Incremental</i>	130
	C. Tahapan dalam Model <i>Incremental</i>	138
	D. Metode dan Strategi Incremental.....	142
	E. Keuntungan dan Kelemahan Model Incremental	145
	F. Perbandingan dengan Model Pengembangan Lainnya.....	147
	G. <i>Incrementally Developing Code</i>	149
	DAFTAR PUSTAKA.....	151
BAB 7	RAPID APPLICATION DEVELOPMENT	153
	A. Apa itu Rapid Application Development (RAD).....	153
	B. Langkah-langkah dalam Rapid Application Development.....	155
	C. Kelebihan dan Kekurangan Model Rapid Application Development (RAD)	159

	D. Rapid Application Development Vs Model Pengembangan Perangkat Lunak Lainnya.....	163
	E. Kapan Rapid Application Development dapat Digunakan?.....	165
	F. Apakah RAD Setara dengan Agile?.....	169
	G. Perbandingan Komprehensif antara Metodologi RAD dan Agile	175
	H. Salah Satu Ilustrasi Studi Kasus Rapid Application Development (RAD) adalah Penerapan Metodologi RAD oleh Centric Consulting	185
	DAFTAR PUSTAKA	187
BAB 8	SCRUM.....	188
	A. Pendahuluan	188
	B. Pengenalan Scrum	189
	C. Peran dan Tanggung Jawab dalam Scrum	195
	D. Artefak dan Alat Scrum.....	200
	E. Proses Scrum	205
	F. Implementasi Scrum dalam Proyek Nyata.....	210
	G. Kesimpulan dan Pandangan Masa Depan Scrum	212
	DAFTAR PUSTAKA	215
BAB 9	EXTREME PROGRAMMING (XP) MODEL.....	216
	A. Pendahuluan	216
	B. Apa itu Extreme Programming?.....	219
	C. Sejarah Singkat Model XP	220
	D. Bagaimana Cara Kerja Model XP?.....	221
	E. Nilai-nilai pada Model XP.....	223
	F. Prinsip-prinsip pada Model XP	224
	G. Perbandingan Model XP dengan Model Lainnya	226
	H. Extreme Programming vs Scrum.....	229
	I. Peran Kunci dan Tanggung Jawab.....	229
	J. Metode dan Teknik.....	230
	K. Memulai Penerapan Model XP	233
	L. Kritik pada Model XP	234

	DAFTAR PUSTAKA.....	236
BAB 10	DYNAMIC SYSTEM DEVELOPMENT METHOD (DSDM).....	237
	A. Pendahuluan.....	237
	B. Proses DSDM.....	239
	C. Keuntungan dan Kelemahan dari <i>Dynamic System Development Method</i>	247
	DAFTAR PUSTAKA.....	248
BAB 11	LEAN DEVELOPMENT.....	249
	A. Pendahuluan.....	249
	B. <i>Lean Software Development</i>	251
	C. Prinsip-Prinsip LSD.....	252
	D. <i>Lean vs Agile</i>	258
	E. Keunggulan <i>LSD</i>	260
	F. Kelemahan <i>LSD</i>	261
	G. Manfaat dan Tantangan Penerapan <i>LSD</i>	263
	H. Kanban.....	264
	I. Prinsip-Prinsip Kanban.....	268
	J. Klasifikasi Pemborosan Pengembangan Aplikasi.....	271
	DAFTAR PUSTAKA.....	274
BAB 12	MODEL PROTOTYPING.....	278
	A. Pendahuluan.....	278
	B. Perspektif Sejarah Prototyping.....	279
	C. Tujuan Prototyping Software.....	284
	D. Jenis-Jenis Prototyping.....	292
	E. Proses Prototyping yang Efektif.....	298
	F. Metode Prototyping.....	303
	G. Rangkuman.....	310
	DAFTAR PUSTAKA.....	312
	TENTANG PENULIS.....	313

DAFTAR TABEL

Tabel 2. 1	Perbandingan antara Model SDLC	32
Tabel 2. 2	Age-Specific Incidence Rate (AIR)	36
Tabel 2. 3	Tingkatan Morfologi Terhadap Penyakit Kanker.....	36
Tabel 2. 4	Tingkatan Stadium Penyakit Kanker	36
Tabel 2. 5	Distribusi Regional dari Age-Standardized Rate (ASR) Terhadap Angka Kejadian Kanker pada Populasi Warga Arab Saudi	36
Tabel 2. 6	Age-Standardized Rate (ASR) Terhadap Angka Kejadian Kanker pada Warga Negara Arab Saudi dan Warga Negara Lainnya.....	37
Tabel 3. 1	Perbandingan Model SDLC (Waterfall, Incremental, Spiral)	68
Tabel 3. 2	Perbandingan Berbagai Model Proses	69
Tabel 5. 1	Perbandingan Antara Metode ASD dan Metode Lainnya	111
Tabel 7. 1	Kelebihan dan Kekurangan Model RAD.....	159
Tabel 7. 2	Keuntungan Model RAD dalam Spesifikasi	160
Tabel 7. 3	Kekurangan Model RAD dalam Spesifikasi.....	162
Tabel 7. 4	RAD Berbeda dan Model Lainnya	164
Tabel 7. 5	Perbedaan Prinsip Rapid Application Development dengan Filosofi Agile	169
Tabel 7. 6	Analisis Komparatif dari unsur-Unsur Fundamental	173
Tabel 9. 1	Perbandingan Model XP dengan Model Lain.....	226
Tabel 11. 1	Prinsip-prinsip LSD	253
Tabel 11. 2	Manfaat dan Tantangan LSD.....	263
Tabel 11. 3	Klasifikasi Pemborosan Pengemngangan Software.....	272

DAFTAR GAMBAR

Gambar 1. 1	SDLC.....	2
Gambar 1. 2	Icon Organizational Change Management.....	12
Gambar 1. 3	Icon IT Governance.....	14
Gambar 1. 4	Icon Manajemen Proyek.....	17
Gambar 2. 1	Fase-fase yang dari Waterfall Model.....	28
Gambar 2. 2	Fase-fase yang dari Incremental Model.....	29
Gambar 2. 3	Fase-fase yang dari Incremental Model.....	30
Gambar 2. 4	Fase-fase yang dari V-Shape Model.....	31
Gambar 2. 5	Entity Relationship Diagram (ERD) Dari CSA. ERD Menunjukkan Hubungan Antar Tabel.....	38
Gambar 2. 6	Halaman Utama untuk Admin pada Website CSA.....	40
Gambar 2. 7	Halaman untuk Menyisipkan dan Memodifikasi Data AIR.....	41
Gambar 2. 8	Halaman untuk Menyisipkan dan Memodifikasi Data Histologi (Morfologi)	41
Gambar 2. 9	Halaman untuk Menyisipkan dan Memodifikasi Data Kota (Regional).....	42
Gambar 2. 10	Halaman untuk Menyisipkan dan Memodifikasi Data Negara.....	42
Gambar 2. 11	Halaman untuk Menyisipkan dan Memodifikasi Data Stadium.....	43
Gambar 2. 12	Contoh dalam pengecekan ketepatan dalam menyisipkan data. Sebuah tahapan dalam mengontrol kualitas yang dibuat untuk mengkonfirmasi data yang disisipkan ke dalam website CSA sama dengan data asli SCIR. Figure ini menunjukkan data untuk distribusi regional terhadap NHL pada laki-laki (A: data asli dari SCIR pada tahun 2015) sama dengan data yang disisipkan	

	dalam system CSA (B: data yang di ekspor menggunakan excel)	44
Gambar 2. 13	Alamat URL terhadap Website CSA yang dimulai dengan https.....	45
Gambar 2. 14	Halaman “Contact Us”	45
Gambar 2. 15	Halaman Utama dari CSA.....	46
Gambar 2. 16	Use Case Diagram Pengembangan Game Edukasi Ular Tangga.....	47
Gambar 2. 17	Tampilan Story Board Halaman Awal Pengembangan Game Edukasi Ular Tangga.....	48
Gambar 2. 18	Tampilan Story Board Giliran Pemain 1 Pengembangan Game Edukasi Ular Tangga.....	48
Gambar 2. 19	Tampilan Story Board Halaman Soal Pengembangan Game Edukasi Ular Tangga.....	49
Gambar 2. 20	Tampilan Story Board Jawaban Benar Pengembangan Game Edukasi Ular Tangga.....	49
Gambar 2. 21	Tampilan Story Board Jawaban Salah Pengembangan Game Edukasi Ular Tangga.....	50
Gambar 2. 22	Tampilan Story Board Halaman Keluar Pengembangan Game Edukasi Ular Tangga.....	50
Gambar 3. 1	Diagram Awal Model Pengembangan Spiral.....	56
Gambar 3. 2	Perbedaan Fase pada Model Spiral.....	59
Gambar 3. 3	Knowledge Spiral pada Perusahaan Beijing Future Advertising	62
Gambar 3. 4	Opportunity Spiral pada Perusahaan Beijing Future Advertising	64
Gambar 3. 5	Coorporation Value Spiral pada Perusahaan Beijing Future Advertising	65

Gambar 3. 6	Model Spiral untuk ATMS Berdasarkan Prototipe	73
Gambar 3. 7	Arsitektur ATMS Secara Keseluruhan.....	74
Gambar 3. 8	Model Spiral Mobile Application Development Life Cycle.....	76
Gambar 4. 1	V-Model Dasar.....	80
Gambar 4. 2	V-Model dengan Proses Dasar Pengembangan.....	83
Gambar 4. 3	Model Mempertimbangkan Tingkat Diskrit dan Jumlah Elemen.....	85
Gambar 4. 4	Eksekusi pada Sistem	91
Gambar 5. 1	Adaptive Life Cycle	108
Gambar 6. 1	Pengembangan Inkrementaltaltal	128
Gambar 6. 2	Model Incremental.....	131
Gambar 6. 3	Tahapan Model Incremental.....	138
Gambar 6. 4	Tahapan Model Incremental.....	142
Gambar 6. 5	Incremental Coding Process	149
Gambar 7. 1	Grafik Perkembangan Aplikasi Cepat	154
Gambar 8. 1	Manifesto for Agile Software Development	191
Gambar 8. 2	Trello.....	201
Gambar 8. 3	Perbedaan Product Backlog dan Sprint Backlog.....	203
Gambar 8. 4	Burndown Chart	204
Gambar 9. 1	Extreme Programming Rules.....	218
Gambar 9. 2	Proses Model XP	220
Gambar 9. 3	Proses Penerapan Model XP	222
Gambar 9. 4	Scrum Model Process	229
Gambar 10. 1	Proses DSDM	240
Gambar 11. 1	Papan Kanban	268
Gambar 12. 1	Sketsa Ide Inovatif Karya Leonardo Da Vinci	280
Gambar 12. 2	Ilustrasi Card Sorting Prototyping	303
Gambar 12. 3	Ilustrasi Wireframe Prototyping.....	304
Gambar 12. 4	Ilustrasi Storyboard Prototyping.....	305
Gambar 12. 5	Ilustrasi Paper Prototyping.....	306

Gambar 12. 6 Ilustrasi Digital Prototyping dengan
Figma.....307
Gambar 12. 7 Contoh Blank Model Prototype309

BAB

1

PENDAHULUAN

Angga Aditya Permana

A. Pendahuluan

SDLC (Software Development Life Cycle) adalah suatu pendekatan sistematis yang digunakan dalam pengembangan perangkat lunak. Pendekatan ini membantu tim pengembangan perangkat lunak dalam mengatur proses pembuatan, pengembangan, pengujian, dan pemeliharaan perangkat lunak secara efisien dan terstruktur. Tujuan dari SDLC adalah untuk menghasilkan perangkat lunak berkualitas tinggi yang memenuhi kebutuhan pengguna, mengikuti tenggat waktu, dan berada dalam batas anggaran yang ditentukan. (tutorial point, 2019)

DAFTAR PUSTAKA

- Cluley, E. (2004) 'Research & reports', *Probation Journal*, 51(4), pp. 421–427. Available at: <https://doi.org/10.1177/0264550504048342>.
- Coronato, A. (2018) 'Agile software development life cycles', *Engineering High Quality Medical Software: Regulations, standards, methodologies and tools for certification*, pp. 161–172. Available at: https://doi.org/10.1049/pbhe012e_ch13.
- tutorial point (2019) 'Software Development Life Cycle (SDLC)', <https://Bigwater.Consulting/2019/04/08/Software-Development-Life-Cycle-Sdlc/> [Preprint]. Available at: <https://bigwater.consulting/2019/04/08/software-development-life-cycle-sdlc/>.

BAB 2

WATERFALL MODEL

David Agustriawan

A. Latar belakang dan Definisi

Model *System Development Life Cycle* (SDLC) yang berbeda sering digunakan untuk pengembangan perangkat lunak [Acharya et al, 2020]. Model SDLC menyediakan panduan teori untuk pengembangan software. Model-model SDLC sangat penting untuk evolusi perangkat lunak secara sistematis sehingga dapat dikerjakan dalam batas waktu tertentu dengan kualitas yang baik. Banyak perusahaan mengembangkan program perangkat lunak untuk menyediakan fasilitas bagi kantor. Beberapa masalah terjadi selama tahap awal pengembangan perangkat lunak [Rastogi et al, 2015]. Beberapa alur terorganisir dalam rekayasa perangkat lunak dan program perangkat lunak dibangun untuk menghindari masalah atau hambatan-hambatan yang ada pada tahap awal pengembangan software tersebut. SDLC adalah teknik di mana perangkat lunak dapat dikembangkan dengan sangat detail dan meningkatkan kemungkinan menyelesaikan proyek perangkat lunak dalam batas waktu serta menjaga konsistensi produk perangkat lunak sesuai dengan standar [Arora et el., 2016]. Kerangka kerja (SDLC) memberikan urutan kegiatan yang harus diikuti oleh perancang sistem dan pengembang untuk mengembangkan perangkat lunak. *System Development Life Cycle* (SDLC) merupakan sebuah metodologi untuk merancang, membangun dan memelihara software atau perangkat lunak (Alshamrani et

DAFTAR PUSTAKA

- Acharya, B., & Sahu, P. K. (2020). Software development life cycle models: A review paper. *International Journal of Advanced Research in Engineering and Technology (IJARET)*, 11, 169-176.
- Alsagaby, S.A. and Alharbi, M.T., 2021. Cancer in Saudi Arabia (CSA): Web-Based Application to Study Cancer Data Among Saudis Using Waterfall Model. *Journal of Multidisciplinary Healthcare*, pp.2333-2343.
- Alshamrani, A. and Bahattab, A., 2015. A comparison between three SDLC models: waterfall model, spiral model, and Incremental/Iterative model. *International Journal of Computer Science Issues (IJCSI)*, 12(1), p.106.
- Arumsari, F.T., Maulindar, J. and Pradana, A.I., 2023. Rancang bangun sistem pendeteksi kebakaran berbasis Internet of Things. *Infotech journal*, 9(1), pp.175-182.
- Asrin, F. and Utami, G.V., 2023. Implementing Website-Based School Information Systems in Public Elementary Schools Using Waterfall Model. *Journal of Information Systems and Informatics*, 5(2), pp.590-614.
- Arora, R and Arora, N, "Analysis of SDLC Models," Int. J. Curr. Eng. Technol., 2016.
- Beighley, L. and Morrison, M., 2008. *Head First PHP & MySQL: A Brain-Friendly Guide*. O'Reilly Media.
- Hölldobler, K., Rumpe, B. and Wortmann, A., 2018. Software language engineering in the large: towards composing and deriving languages. *Computer Languages, Systems & Structures*, 54, pp.386-405.
- Ivanovich, J.W., Said, M.A., Rohim, S.A., Wicaksono, M.R. and Yossy, E.H., 2020, August. Development Model of Warteg Online Applications based on Web and Mobile. In *2020 International Conference on Information Management and Technology (ICIMTech)* (pp. 864-869). IEEE.

- Kute, S.S., and Thorat, S., "A Review on Various Software Development Life Cycle (SDLC) Models," *Int. J. Res. Comput. Commun. Technol.*, 2014.
- Meha, I. L., & Talakua, A. C. (2023, August). Game Edukasi Matematika Untuk Minat Belajar Berhitung Berbasis Android: Mathematics Educational Games for an Interest in Learning Calculating Based On Android. In *SENTIMAS: Seminar Nasional Penelitian dan Pengabdian Masyarakat* (pp. 351-359).
- Mishra, A and Dubey, A., "A Comparative Study of Different Software Development Life Cycle Models in Different Scenarios," *Int. J. Adv. Res. Comput. Sci. Manag. Stud.*, 2013.
- Nelson, B., 2018. Getting to know Vue. js. *Apress Media*, August.
- Osetskiy, V., "SDLC Models Explained: Agile, Waterfall, V-Shaped, Iterative, Spiral," *Existek*, v2017.
- Powell, T.A., 1999. *HTML The Complete Reference Second Edition*.
- Praing, R. and Talakua, A.C., 2023, February. Pengembangan Game Edukasi Ular Tangga Sebagai Media Pembelajaran Berbasis Android. In *SEMINAR NASIONAL Sustainable Agricultural Technology Innovation (SATI)* (Vol. 1, No. 1, pp. 65-78).
- Purnomo, A., 2023. Pengembangan Game Belajar Huruf Hijaiyah Untuk Disleksia dengan Metode Waterfall. *JURIKOM (Jurnal Riset Komputer)*, 10(1), pp.68-75.
- Rastogi, V., "Software Development Life Cycle Models- Comparison, Consequences," *Int. J. Comput. Sci. Inf. Technol.*, 2015.
- Rauschmayer, A., 2014. *Speaking JavaScript: an in-depth guide for programmers*. " O'Reilly Media, Inc."
- Rezende, A.V., Silva, L., Britto, A. and Amaral, R., 2019. Software project scheduling problem in the context of search-based software engineering: A systematic review. *Journal of Systems and Software*, 155, pp.43-56.

- Stauffer, M., 2019. *Laravel: Up & running: A framework for building modern php apps*. O'Reilly Media.
- Swara, G.Y., Warman, I. and Putra, D.W.T., 2022. Implementation of The Waterfall Model on Android-Based Travel Ticket Booking Applications. *JISICOM (Journal of Information System, Informatics and Computing)*, 6(1), pp.235-245.
- Trivedi, P and Sharma, A., "A comparative study between iterative waterfall and incremental software development life cycle model for optimizing the resources using computer simulation," 2013.

BAB 3

SPIRAL MODEL

Monika Evelin Johan

A. Apa itu Spiral Model?

1. Definisi Model Spiral

Model Spiral diperkenalkan pertama kali oleh Barry William Boehm, seorang professor asal Amerika pada tahun 1986. Model tersebut diperkenalkan pada sebuah artikel berjudul “A Spiral Model of Software Development and Enhancement”. Model Spiral diperkenalkan sebagai model yang menyediakan kerangka kerja (framework) terbaru untuk memandu proses pengembangan perangkat lunak, dimana model ini menggunakan pendekatan berbasis risiko (risk-driven) dan dapat digunakan dengan menggabungkan kelebihan dari model lainnya. Hal ini menjadikan model Spiral memiliki kelebihan dibandingkan dengan model – model pengembangan perangkat lunak yang sudah ada sebelumnya, yang prosesnya spesifik atau mengikuti prototipe secara ketat. Pendekatan risk-driven pada model Spiral memungkinkan model ini dapat mengakomodasi model lainnya baik yang berorientasi spesifikasi, prototipe, simulasi, transformasi otomatis, atau pendekatan pengembangan perangkat lunak lainnya (Boehm, 1986).

Awalnya, Boehm menyebut Spiral Model sebagai “process model”, yang membuat Spiral Model setara dengan model seperti incremental, waterfall, prototyping, dan pendekatan lainnya. Namun, pada publikasi berikutnya di

DAFTAR PUSTAKA

- Alshamrani, A. and Bahattab, A. (2015) 'A comparison between three SDLC models waterfall model, spiral model, and Incremental/Iterative model', *International Journal of Computer Science Issues (IJCSI)*, 12(1), p. 106.
- Bhosale, S. (2014) 'Spiral model: Applications in web based applications', *IPASJ Int. J. Comput. Sci*, 2, pp. 1-4.
- Boehm, B. (1986) 'A spiral model of software development and enhancement', *ACM SIGSOFT Software engineering notes*, 11(4), pp. 14-24.
- Boehm, B.W. and Hansen, W.J. (2000) 'Spiral development: Experience, principles, and refinements'.
- Boje, D.M. et al. (2017) 'The episodic spiral model: a new approach to organizational processes', *Journal of Organizational Change Management*, 30(5), pp. 683-709.
- Doshi, D., Jain, L. and Gala, K.(2021)'Review of the spiral model and its applications', *Int. J. Eng. Appl. Sci. Technol*, 5, pp. 311-316.
- Khandelwal, A. and Tyagi, G. (2015) 'Review paper on suitability of traditional prototype model and spiral model used for mobile application development life cycle', *Int. J. Eng. Res. Technol*, 3(31), pp. 1-4.
- Raval, R.R. and Rathod, H.M. (2013) 'Comparative study of various process model in software development', *International Journal of Computer Applications*, 82(18).
- Wang, L.(2009)'The Multi-dimension Spiral Model of the Knowledge-based Corporation',*Journal of Software*,4(5),p. 469.
- Xie, J. et al. (2015) 'A Spiral Development Model for an Advanced Traffic Management System (ATMS) Architecture Based on Prototype', *International Journal of Science Technology and Society*, 3(6), pp. 304-308.

BAB

4

V-MODEL

Melissa Indah Fianty

A. Pendahuluan

V-Model, juga dikenal sebagai V-model rekayasa sistem, menekankan pendekatan pemecahan masalah. Dimulai dari pembagian masalah pada tingkat awal menjadi bagian-bagian yang dapat dikelola. Dari tingkat detail, solusi akhir diintegrasikan kembali ke tingkat awal. Pada setiap tingkat, seseorang dapat membandingkan solusi atau bagian solusi dengan masalah atau bagian masalah yang sedang dipecahkan. V Model pada di lihat gambar 4.1 memungkinkan berbagai proyeksi dan menghasilkan banyak interpretasi (Weilkiens et al., 2022).

DAFTAR PUSTAKA

- Childs, P. R. N. (2013). Mechanical Design Engineering Handbook. In *Mechanical Design Engineering Handbook*. <https://doi.org/10.1016/C2011-0-04529-5>
- Habash, R. (2022). *Building as a human-cyber-physical system* (pp. 129–160). <https://doi.org/10.1016/B978-0-323-98826-1.00005-3>
- Handayani, N. P., Fianty, M. I., Shabrina, N. H., & Surendro, K. (2021). Does Implementation of a Human Resource Information System Influence Employee's Turnover Intention in Developing Country? *2021 IEEE International Conference on Technology Management, Operations and Decisions, ICTMOD 2021*. <https://doi.org/10.1109/ICTMOD52902.2021.9739510>
- Hauser, A., & Kuhn, R. (2015). *Cell balancing, battery state estimation, and safety aspects of battery management systems for electric vehicles* (pp. 283–326). <https://doi.org/10.1016/B978-1-78242-377-5.00012-1>
- Karden, E. (2017). Development trends for future automobiles and their demand on the battery. In *Lead-Acid Batteries for Future Automobiles* (pp. 3–25). <https://doi.org/10.1016/B978-0-444-63700-0.00001-5>
- Mondorf, A., & Wimmer, M. A. (2014). Design and adoption of standard specifications using the V-model. *Innovation and the Public Sector*, 21, 141–150. <https://doi.org/10.3233/978-1-61499-429-9-141>
- Oshana, R. (2013). *Software Engineering for Embedded Systems*. <https://doi.org/10.1016/C2011-0-04586-6>
- Pressman, R., & Maxim, B. (2014). *Software Engineering: A Practitioner's Approach, 8th Ed.*
- Sarhadi, P., Naeem, W., Fraser, K., & Wilson, D. (2022). On the Application of Agile Project Management Techniques, V-Model and Recent Software Tools in Postgraduate Theses

Supervision. *IFAC-PapersOnLine*, 55(17), 109–114.
<https://doi.org/https://doi.org/10.1016/j.ifacol.2022.09.233>

V-Modell® XT. (2004). *V-Modell XT, Part 1: Fundamentals of the V-Model*.

Weilkiens, T., Lamm, J. G., Roth, S., & Walker, M. (2022). The V-Model. In *Model-Based System Architecture, Second Edition* (pp. 381–390). Wiley.
<https://doi.org/10.1002/9781119746683.app2>

BAB 5

ADAPTIVE SOFTWARE DEVELOPMENT

Samuel Ady Sanjaya

A. Pendahuluan

Perubahan yang tak terhindarkan sering memengaruhi pengembangan perangkat lunak, yang merupakan proses yang selalu berubah. Adaptive Software Development (ASD) metodologi dibuat pada awal 1990-an oleh John Highsmith dan Sam Bayer, dua manajer proyek. Ini adalah modifikasi dari kerangka kerja Rapid Application Development (RAD), yang terkenal dengan pendekatan ketat untuk proses pengembangan perangkat lunak (J. Highsmith, 2009). Di tengah lingkungan teknologi yang terus berkembang dan tuntutan pasar yang berubah dengan cepat, pendekatan yang kaku dan terlalu terstruktur mungkin tidak lagi cukup efektif. Inilah tempat di mana konsep "adaptive development" atau pengembangan yang adaptif muncul sebagai solusi yang sesuai. Pendekatan ini mengakui kompleksitas lingkungan pengembangan perangkat lunak dan memprioritaskan fleksibilitas serta responsibilitas terhadap perubahan.

Adaptive development, juga dikenal sebagai adaptive methodologies, adalah paradigma dalam siklus hidup pengembangan perangkat lunak yang berfokus pada kemampuan untuk beradaptasi dengan perubahan yang tidak dapat dihindari. Pendekatan ini mengakui bahwa persyaratan dan kebutuhan dalam proyek pengembangan seringkali berubah seiring waktu, baik karena perubahan dalam

DAFTAR PUSTAKA

- J. Highsmith (2009) *Agile Project Management: Creating Innovative Products*. Second Edi. Addison-Wesley Professional.
- Johansen, T. and Gilb, T. (2005) '11.4.1 From Waterfall to Evolutionary Development (Evo): How we rapidly created faster, more user-friendly, and more productive software products for a competitive multi-national market', *INCOSE International Symposium*, 15(1), pp. 1676-1686. Available at: <https://doi.org/10.1002/j.2334-5837.2005.tb00781.x>.

BAB

6

INCREMENTAL MODEL

Rudi Sutomo

A. Pendahuluan

1. Latar belakang

Pengembangan perangkat lunak telah mengalami evolusi penting selama beberapa dekade terakhir, memunculkan banyak pendekatan dan metode berbeda untuk mencapai tujuan ini. Salah satu pendekatan yang mendapat perhatian luas dalam rekayasa perangkat lunak adalah model inkremental. Model ini memberikan pendekatan yang adaptif dan iteratif dalam mengembangkan sistem perangkat lunak, yang memungkinkan tahapan pembangunan yang bertahap dengan peningkatan fungsionalitas di setiap tahapnya. Dengan mengakui kompleksitas dan dinamika yang umumnya terlibat dalam proyek perangkat lunak, model *Incremental* menghadirkan solusi yang memungkinkan para pengembang untuk merespons perubahan dengan lebih efektif sambil memberikan hasil tanggung jawab secara berkelanjutan kepada pihak-pihak terkait. Dalam bab ini, kami akan membahas konsep dasar dari model *Incremental*, tahapan-tahapan yang terlibat, metode dan strategi yang dapat diterapkan, serta analisis keuntungan dan kendalanya dalam konteks pengembangan perangkat lunak modern (Larman & Basili, 2003).

DAFTAR PUSTAKA

- Al Kaafi, A., Widiastuti, L., & Arsiadi, F. (2022). Penerapan Incremental Model Pada Sistem Informasi Pendaftaran Peserta Didik Baru (PPDB) SMA Uswatun Hasanah Jakarta. *Reputasi: Jurnal Rekayasa Perangkat Lunak*, 3(1), 32–39. <https://doi.org/10.31294/reputasi.v3i1.1197>
- Ganney, P. S., Pisharody, S., & Claridge, E. (2013). Software Engineering. In *Clinical Engineering: A Handbook for Clinical and Biomedical Engineers*. <https://doi.org/10.1016/B978-0-12-396961-3.00009-3>
- Highsmith, J., & Cockburn, A. (2001). Agile software development: The business of innovation. *Computer*, 34(9), 120–122. <https://doi.org/10.1109/2.947100>
- Hussaini Syed, H., Kumari, B., & Chauhan, N. (2018). a Comparison Between Manual Testing and Automated Testing. *Journal of Emerging Technologies and Innovative Research (JETIR)*, 5(12), 323–331. <https://www.researchgate.net/publication/349636718>
- Jaelani. (2021). Bab 3 Metode Penelitian. *Repository.Upi.Edu*, 02, 2013–2015. http://repository.upi.edu/61268/4/S_JKR_1604261_Chapter3.pdf
- Larman, C., & Basili, V. R. (2003). Iterative and incremental development: A brief history. *Computer*, 36(6), 47–56. <https://doi.org/10.1109/MC.2003.1204375>
- Liu, C. L., Hua, N. T., & Tucker, A. B. (n.d.). *PRACTITIONER ' S*.
- Murdani, M. H., Widhiyanta, N., Priyambudi, S., & Asrori, M. (2022). Rancang Bangun Sistem Informasi Akuntansi menggunakan Metode Incremental – Studi Kasus di Koperasi Karyawan Coca Cola SIER. *Smatika Jurnal*, 12(01), 67–74. <https://doi.org/10.32664/smatika.v12i01.663>

- Rizqiani, K., Ramdhani, M., Rizal, A., & Elektro, D. T. (2009). *Perancangan Perangkat Lunak*. 1-6.
- Rokoyah, K., Chandra, Y. I., & Lukman, S. (2022). Penerapan Model Incremental Dalam Merancang Aplikasi Pengenalan Bentuk Dan Fungsi Gigi Pada Manusia Berbasis Web. *Jurnal SIKOMTEK*, 12(2), 42-47.
- School of Information Systems. (n.d.). Software Development Model: Incremental Model. [online] Available at: <https://sis.binus.ac.id/2019/07/02/software-development-model-incremental-model/> [Accessed 21 08. 2023].
- Sommerville, I. (2011). *Software Engineering (9th ed.; Boston, Ed.)*. Massachusetts: Pearson Education.
- Wahyuni, E. D. (2021). Implementasi Metode Incremental Pada Sistem Informasi Administrasi Desa Jambuwer. *Jurnal Tekno Kompak*, 15(2), 156. <https://doi.org/10.33365/jtk.v15i2.1187>
- Wibowo, E. C., Kusumasari, T. F., & Alam, E. N. (2022). Pengimplementasian Fitur Data Profiling Pada Aplikasi Data Governance Berbasis Open Source Tools dengan Metode Iterative/Incremental. *Journal of Information System Research (JOSH)*, 4(1), 117-126. <https://doi.org/10.47065/josh.v4i1.2315>

BAB

7

RAPID APPLICATION DEVELOPMENT

Wirawan Istiono

A. Apa itu Rapid Application Development (RAD)

Rapid Application Development (RAD) mengacu pada strategi pengembangan perangkat lunak adaptif yang berpusat di sekitar pembuatan prototipe dan umpan balik yang dipercepat, sementara kurang fokus pada desain yang teliti. Teknik RAD dicirikan oleh fokus pada pengembangan dan pembuatan prototipe, berlawanan dengan perencanaan ekstensif, dalam keseluruhan strateginya. Pengembangan aplikasi yang cepat memungkinkan pengembang untuk mengimplementasikan beberapa iterasi dan perubahan secara efisien pada produk tanpa perlu memulai proses pengembangan lagi. Praktik ini membantu menjamin bahwa hasil akhir berpusat pada kualitas dan sesuai dengan spesifikasi pelanggan akhir (Kissflow, 2023).

Salah satu kelemahan signifikan yang terkait dengan model air terjun adalah ketidakmampuan penguji untuk meninjau kembali dan memodifikasi fungsi dan fitur dasar produk setelah beralih ke fase pengujian. Hal ini pada akhirnya menghasilkan tim yang memiliki solusi perangkat lunak yang mungkin selaras atau tidak selaras dengan perubahan kebutuhan pengguna akhir.

Konsep Rapid Application Development (RAD) berasal dari tahun 1980-an, menunjukkan kehadiran historisnya daripada kebaruan. Namun, berbeda dengan model air terjun,

DAFTAR PUSTAKA

- Agarwal, R. *et al.* (2000) 'Risks of Rapid Application Development', *Communications of the ACM*, 43, pp. 177-188. doi: 10.1145/352515.352516.
- Kissflow, T. (2023) *What is Rapid Application Development (RAD)? An Ultimate Guide for 2023*, *kissflow.com*. Available at: <https://kissflow.com/application-development/rad/rapid-application-development/> (Accessed: 21 August 2023).
- Naz, R. and Khan, M. N. A. (2015) 'Rapid applications development techniques: A critical review', *International Journal of Software Engineering and its Applications*, 9(11), pp. 163-176. doi: 10.14257/ijseia.2015.9.11.15.
- Outsystems, T. (2022) *What Is Rapid Application Development?*, *outsystems.com*. Available at: <https://www.outsystems.com/glossary/what-is-rapid-application-development/> (Accessed: 21 August 2023).
- Quick, L. (2023) *RAD vs Agile: Which one is right for you?*, *knowledgehut.com*. Available at: <https://www.knowledgehut.com/blog/agile/rad-vs-agile> (Accessed: 22 August 2023).

BAB

8

SCRUM

Suwito Pomalingo

A. Pendahuluan

Di era digital saat ini, pengembangan perangkat lunak telah menjadi komponen penting dalam kehidupan sehari-hari dan bisnis. Teknologi yang terus berkembang menuntut pendekatan yang fleksibel, adaptif, dan efisien dalam mengelola proyek perangkat lunak. Salah satu metode yang telah menonjol dalam dekade terakhir adalah Scrum.

Scrum adalah kerangka kerja *agile* yang menawarkan pendekatan iteratif dan inkremental dalam pengembangan perangkat lunak. Keunikan dan efektivitas Scrum dalam memenuhi kebutuhan pasar yang cepat berubah menjadikannya pilihan populer di kalangan organisasi modern.

Buku ini ditujukan untuk membantu para profesional, dosen, peneliti, dan siapa saja yang tertarik dalam manajemen proyek perangkat lunak, khususnya dalam menggunakan Scrum. Dengan menyajikan konsep, praktik, dan analisis mendalam, buku ini bertujuan untuk menjadi panduan komprehensif yang menghubungkan teori dan praktik dalam konteks nyata.

Buku ini cocok untuk berbagai kalangan, mulai dari pemula yang ingin memahami dasar-dasar Scrum, praktisi yang ingin meningkatkan keterampilan mereka, hingga pemimpin organisasi yang ingin mengintegrasikan Scrum dalam strategi bisnis mereka.

DAFTAR PUSTAKA

- Beck, K., Beedle, M., Bennekum, A. van, Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). *Manifesto for Agile Software Development*. agilemanifesto.org
- Cohn, M. (2009). *Succeeding with Agile: Software Development Using Scrum* (1st ed.). Addison-Wesley Professional.
- Larman, C., & Vodde, B. (2008). *Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum*. Addison-Wesley Professional.
- Mitchell, I. (2014). *Agile Practice Guide*. ScrumSense.
- Pomalingo, S., & Tobing, F. A. T. (2023). Optimizing Patient Registration Process through Online Admission Application: A Scrum Approach. *Jurnal Bumigora Information Technology (BITe)*, 5(1). <https://doi.org/10.30812/bite.v5i1.3007>
- Schwaber, K., & Sutherland, J. (2017). *The Scrum Guide*. <https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>
- Takeuchi, H., & Nonaka, I. (1986). The New New Product Development Game. *Harvard Business Review*.

BAB 9

EXTREME PROGRAMMING (XP) MODEL

Jansen Wiratama

A. Pendahuluan

Metode pengembangan sistem Agile mungkin terdengar familiar ditelinga kita, namun pembahasan terkait model Extreme Programming (CP) masih cukup asing. Banyak dari kita yang merasa kurang yakin dengan penerapan model XP ini, namun, jangan melewatkan pemahaman terkait model XP. Bab 9 ini akan memberikan kita pemahaman melalui semua yang perlu kita ketahui tentang Model XP sehingga dapat kita manfaatkan secara optimal.

Extreme Programming (XP) adalah salah satu dari beberapa Proses Agile yang populer. Model ini telah terbukti sangat sukses di banyak perusahaan dengan berbagai ukuran dan industri di seluruh dunia. Model XP banyak berhasil diterapkan karena lebih menekankan pada kepuasan pelanggan/pengguna. Keunikan pada model XP juga dapat dilihat pada siklus pengembangannya yang fleksibel. Model XP memungkinkan penggunanya untuk memenuhi kebutuhan pada proses bisnis secara dinamis mengikuti perkembangan tanpa harus menunggu di waktu tertentu. Dalam Extreme Programming, pengembang perangkat lunak diberdayakan untuk menghadapi perubahan permintaan pelanggan secara lebih optimal, bahkan pada akhir siklus pengembangan.

Model XP mendorong kerja sama tim yang kuat. Manajer project, pelanggan, dan pengembang bekerja sama sebagai rekan

DAFTAR PUSTAKA

- Beck, K. (2000) *Extreme Programming Explained: Embrace Change*. Addison-Wesley (An Alan R. Apt Book Series). Available at: <https://books.google.co.id/books?id=G8EL4H4vf7UC>.
- Beck, K. and Andres, C. (2004) *Extreme Programming Explained: Embrace Change*. Pearson Education (XP Series). Available at: <https://books.google.co.id/books?id=-DNcBAAAQBAJ>.
- I Gusti Ngurah Suryantara, S.K.M.K. (2017) *Merancang Aplikasi dengan Metodologi Extreme Programming*. Elex Media Komputindo. Available at: <https://books.google.co.id/books?id=FDBIDwAAQBAJ>.
- Nimble (2021) *What Is Extreme Programming (XP)? & It's Values, Principles, And Practices, 2021*. Available at: <https://www.nimblework.com/agile/extreme-programming-xp/> (Accessed: 21 August 2023).
- Prabowo, M. (2020) *METODOLOGI PENGEMBANGAN SISTEM INFORMASI*. LP2M Press IAIN Salatiga. Available at: <https://books.google.co.id/books?id=UI8dEAAAQBAJ>.
- Rosenberg, D. and Stephens, M. (2008) *Extreme Programming Refactored: The Case Against XP*. Apress (The expert's voice). Available at: <https://books.google.co.id/books?id=Nxi7O7FCdIEC>.
- Wiratama, J., Santoso, H. and Clairence (2023) 'Developing a Class Scheduling Mobile Application for Private Campus in Tangerang with the Extreme Programming (XP) Model', *G-Tech : Jurnal Teknologi Terapan*, 7(2), pp. 484–493. Available at: <https://doi.org/https://doi.org/10.33379/gtech.v7i2.2288>.

BAB

10

DYNAMIC SYSTEM DEVELOPMENT METHOD (DSDM)

Erick Fernando

A. Pendahuluan

Dynamic System Development Method (DSDM) adalah teknik pengembangan yang menekankan pada proses berulang dan peningkatan berkelanjutan, selaras dengan ide dasar metodologi Agile. Ini secara aktif melibatkan pengguna dan pelanggan secara terus menerus sepanjang keseluruhan proses pengembangan (Siau, 2007; Ibrahim and Awny, 2017; Pressman and Maxim, 2020).

Asal-usul DSDM dapat ditelusuri kembali ke tahun 1990-an, periode di mana teknik Rapid Application Development (RAD) mendapatkan popularitas yang signifikan. Ini memfasilitasi penyelesaian masalah yang dipercepat melalui pemanfaatan teknik prototyping. DSDM muncul sebagai tanggapan terhadap keterbatasan yang diamati dalam metodologi sekuensial seperti air terjun dan RAD. Tujuannya adalah untuk mengintegrasikan kekuatan RAD sekaligus menggabungkan pemahaman yang komprehensif tentang pemangku kepentingan dan keterlibatan aktif mereka dalam proses pembangunan (Stapleton, 2003; Coyle and Conboy, 2009; Ibrahim and Awny, 2017).

DSDM awalnya diperkenalkan pada tahun 1994, sehingga menawarkan kerangka kerja sistematis untuk RAD. Meskipun kadang-kadang disorganisasi akibat kekurangan metode standar, tujuan utama Konsorsium DSDM adalah untuk

DAFTAR PUSTAKA

- Coyle, S. and Conboy, K. (2009) 'A study of risk management in DSDM', *Lecture Notes in Business Information Processing*, 31 LNBIP, pp. 142-148. Available at: https://doi.org/10.1007/978-3-642-01853-4_18.
- Ibrahim, A.A.S. and Awny, M.M. (2017) 'A dynamic system development method for startups migrate to c loud', *Lecture Notes on Data Engineering and Communications Technologies*, 1, pp. 813-824. Available at: https://doi.org/10.1007/978-3-319-49109-7_78.
- Pressman, R.S. and Maxim, B.R. (2020) *Software engineering: A Practioner's Approach*.
- Siau, K. (2007) *Research issues in systems analysis and design, databases and software development, Research Issues in Systems Analysis and Design, Databases and Software Development*. Available at: <https://doi.org/10.4018/978-1-59904-927-4>.
- Stapleton, J. (2003) *The DSDM Consortium, DSDM: Business Focused Development*. Great Britain: Pearson Education Limited.

BAB

11

LEAN DEVELOPMENT

Ahmad Faza

A. Pendahuluan

Pendekatan *lean* pada dasarnya berfokus pada pandangan manajemen yang mengedepankan eliminasi proses yang tidak perlu serta mendorong kerjasama yang efektif antara semua pihak yang terlibat dalam setiap tahapan pengembangan, mulai dari perencanaan hingga pengiriman produk akhir kepada pengguna (Janes, 2015). Awalnya, istilah *lean* dikenal sebagai *lean production*. Konsep ini muncul dari penelitian di *Massachusetts Institute of Technology* (MIT) dengan tujuan untuk memahami mengapa perusahaan otomotif Jepang seperti Toyota mampu mengungguli perusahaan-perusahaan Amerika dalam menguasai pangsa pasar (Jonsson et al., 2013; Abdullah and Ward, 2016). Prinsip-prinsip *lean production* sangat berkaitan dengan *Toyota Production System* (TPS), yang dikembangkan di Jepang pada tahun 1940-an (Womack and Jones, 1997). Sistem ini merubah cara kerja industri otomotif dan mendorong Jepang menjadi pemimpin global, meskipun konsep serupa pernah digunakan oleh Ford sebelumnya (Womack and Jones, 1997; Cawley et al., 2013). Kesuksesan penerapan *lean production* dalam TPS untuk industri otomotif mendorong perusahaan-perusahaan, baik di sektor yang sama maupun sektor lainnya, secara cepat mengadopsi *lean production* dalam kegiatan produksinya demi meraih *competitive advantage* (Womack and Jones, 1997; Cawley et al., 2013). Hal ini kemudian menjadikan

DAFTAR PUSTAKA

- Abdullah, F. and Ward, R. (2016) 'Developing a general extended technology acceptance model for e-learning (GETAMEL) by analysing commonly used external factors', *Computers in Human Behavior*, 56, pp. 238-256.
- Ahmad, M.O. et al. (2018) 'Kanban in software engineering: A systematic mapping study', *Journal of Systems and Software*, 137, pp. 96-113.
- Anderson, D.J. and Bozheva, T. (2018) *Kanban Maturity Model: Evolving Fit-for-purpose Organizations*. Lean Kanban University Press.
- Anwaar, Z. et al. (2021) 'Effect of Lean Software Development on the Small Organizations Using LSD Approach', in *2021 International Conference on Innovative Computing (ICIC)*, pp. 1-8.
- Beck, K. (2000) *Extreme programming explained: embrace change*. addison-wesley professional.
- Behroozi, N. and Kamandi, A. (2016) 'Waste elimination of agile methodologies in web engineering', in *2016 Second International Conference on Web Research (ICWR)*, pp. 102-107.
- Cawley, O., Wang, X. and Richardson, I. (2013) 'Lean software development-what exactly are we talking about?', in *Lean Enterprise Software and Systems: 4th International Conference, LESS 2013, Galway, Ireland, December 1-4, 2013, Proceedings*, pp. 16-31.
- Cumbo, D., Kline, D.E. and Bumgardner, M.S. (2006) 'Benchmarking performance measurement and lean manufacturing in the rough mil', *Forest Products Journal*, 56(6), pp. 25-30.
- Dybå, T. and Dingsøy, T. (2008) 'Empirical studies of agile software development: A systematic review', *Information and software technology*, 50(9-10), pp. 833-859.

- Feyh, M. and Petersen, K. (2013) 'Lean software development measures and indicators-a systematic mapping study', in *Lean Enterprise Software and Systems: 4th International Conference, LESS 2013, Galway, Ireland, December 1-4, 2013, Proceedings*, pp. 32-47.
- Flora, H.K. and Chande, S. V (2014) 'A systematic study on agile software development methodologies and practices', *International Journal of Computer Science and Information Technologies*, 5(3), pp. 3626-3637.
- Fowler, F. (2009) *Survey Research Methods (4th ed.)*. CA: Sage.
- Greaves, K. (2011) 'Taming the Customer Support Queue: A Kanban Experience Report', in *2011 Agile Conference*, pp. 154-160.
- Hines, P., Holweg, M. and Rich, N. (2004) 'Learning to evolve: a review of contemporary lean thinking', *International journal of operations & production management*, 24(10), pp. 994-1011.
- Ikonen, M. et al. (2011) 'On the impact of Kanban on software project work: An empirical case study investigation', in *2011 16th IEEE international conference on engineering of complex computer systems*, pp. 305-314.
- Janes, A. (2015) 'A guide to lean software development in action', in *2015 IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pp. 1-2.
- Jonsson, H., Larsson, S. and Punnekkat, S. (2013) 'Synthesizing a comprehensive framework for lean software development', in *2013 39th Euromicro Conference on Software Engineering and Advanced Applications*, pp. 1-8.
- Kniberg, H. (2011) 'Lean from the trenches: Managing large-scale projects with Kanban', *Lean from the Trenches*, pp. 1-178.
- Krijnen, A. (2007) 'The Toyota way: 14 management principles from the world's greatest manufacturer'. Taylor & Francis.

- Middleton, P. (2001) 'Lean software development: two case studies', *Software Quality Journal*, 9, pp. 241–252.
- Mujtaba, S., Feldt, R. and Petersen, K. (2010) 'Waste and lead time reduction in a software product customization process with value stream maps', in 2010 21st Australian software engineering conference, pp. 139–148.
- Ohno, T. (2019) *Toyota production system: beyond large-scale production*. Productivity press.
- Petersen, K. et al. (2014) 'Early identification of bottlenecks in very large scale system of systems software development', *Journal of software: Evolution and Process*, 26(12), pp. 1150–1171.
- Poppendieck, M. and Poppendieck, T. (2003) *Lean software development: an agile toolkit*. Addison-Wesley.
- Power, K. (2014) 'Definition of ready: An experience report from teams at cisco', in *Agile Processes in Software Engineering and Extreme Programming: 15th International Conference, XP 2014, Rome, Italy, May 26-30, 2014. Proceedings 15*, pp. 312–319.
- Reinertsen, D.G. (2009) 'The principles of product development flow: second generation lean product development', (No Title) [Preprint].
- Rodríguez, P. et al. (2019) 'Advances in using agile and lean processes for software development', in *Advances in computers*. Elsevier, pp. 135–224.
- Williams, L. (2012) 'What agile teams think of agile principles', *Communications of the ACM*, 55(4), pp. 71–76.
- Womack, J.P. and Jones, D.T. (1997) 'Lean thinking—banish waste and create wealth in your corporation', *Journal of the Operational Research Society*, 48(11), p. 1148.
- Womack, J.P., Jones, D.T. and Roos, D. (2007) *The machine that changed the world: The story of lean production—Toyota's*

secret weapon in the global car wars that is now revolutionizing world industry. Simon and Schuster.

Zang, J.J. (2011) 'A never ending battle for continuous improvement', in Agile Processes in Software Engineering and Extreme Programming: 12th International Conference, XP 2011, Madrid, Spain, May 10-13, 2011. Proceedings 12, pp. 282-289.

BAB

12

MODEL PROTOTYPING

Alexander Waworuntu

A. Pendahuluan

Dalam dunia pengembangan perangkat lunak, memahami kebutuhan dan ekspektasi pengguna adalah kunci untuk menciptakan produk yang sukses. Model *prototyping* atau pendekatan purwarupa telah lama diakui efektivitasnya dalam menjembatani kesenjangan antara konseptualisasi awal dan implementasi produk akhir. Secara definisi, *prototyping*, yang juga dikenal dengan sebutan prototipe atau purwarupa, adalah proses pembuatan model kerja awal dari sistem, yang kemudian dapat diperbaiki melalui iterasi berdasarkan umpan balik dari pengguna (Sommerville, 2016). Dalam bab ini, ketiga istilah tersebut, yaitu *prototype*, prototipe, dan purwarupa, akan digunakan secara bergantian namun tetap mengarah pada ide yang sama.

Sejarah singkat menunjukkan bahwa pendekatan *prototyping* mulai diterapkan pada dekade 1970-an sebagai respons terhadap tantangan dalam metodologi pengembangan tradisional yang linier. Kesadaran bahwa sistem sering kali dikembangkan tanpa memenuhi kebutuhan pengguna dengan tepat memicu kebutuhan untuk pendekatan yang lebih interaktif dan responsif (Boehm, 1988).

Alasan utama mengadopsi *prototyping* adalah keinginan untuk mengurangi risiko kesalahan interpretasi kebutuhan pengguna. Dengan prototipe, pengembang dan stakeholder

DAFTAR PUSTAKA

- Arnowitz, Jonathan Arent, Michael Berger, N. (2007) *Effective Prototyping for Software Makers*. Elsevier.
- Boehm, B. W. (1988) 'A spiral model of software development and enhancement', *Computer*, 21(5), pp. 61-72. doi: 10.1109/2.59.
- Davis, A. M. (1995) 'Software Prototyping', in, pp. 39-63. doi: 10.1016/S0065-2458(08)60544-6.
- Dreyfuss, H. (1967) 'The measure of man: human factors in design'.
- Larman, C. (2004) *Agile & Iterative Development*. Pearson Education.
- Neubauer, J. *et al.* (2014) 'Prototype-Driven Development of Web Applications with DyWA', in, pp. 56-72. doi: 10.1007/978-3-662-45234-9_5.
- Pfleeger, S. L. and Atlee, J. M. (2009) *Software Engineering: Theory and Practice*. 4th edn. Pearson.
- Pressman, R. S. and Maxim, B. R. (2020) *Software Engineering: A Practitioner's Approach*. 9th edn. New York: McGraw-Hill Education. doi: 10.1049/ic:20040411.
- Snyder, C. (2003) *Paper Prototyping*. Elsevier.
- Sommerville, I. (2016) *Software Engineering*. 10th edn. Pearson.

TENTANG PENULIS



Angga Aditya Permana, M.Kom., MM

Dosen Program Studi Informatika
Fakultas Teknik dan Informatika
Universitas Multimedia Nusantara

Angga Aditya Permana (lahir pada Desember 1989 di Jakarta) seorang dosen full time di bidang Computer Science pada Universitas Multimedia Nusantara sejak tahun 2021, fokus penelitian pada kriptografi, steganografi serta keamanan computer, namun pada tahun 2021 sedang mendalami topik bioinformatika dan network science. Angga juga memiliki hobi yaitu touring dan juga bermain bulutangkis, saat ini sedang menjadi mahasiswa program doctoral pada IPB University. Memulai karir pertama kali sebagai Network Enginner tahun 2011 dan memulai profesinya sebagai dosen pada 2013 di kampus swasta yang ada di Jakarta, pernah juga mengajar di kampus negeri yang berada di Jakarta dan Tangerang, juga pernah di undang menjadi dosen tamu untuk mengenalkan Bioinformatika di kampus negeri di Jakarta. Terimakasih.



David Agustriawan, S.Kom., M.Sc., Ph.D.

Dosen Program Studi Informatika
Fakultas Teknik dan Informatika
Universitas Multimedia Nusantara

David Agustriawan adalah dosen tetap pada Program Studi Informatika, Universitas Multimedia Nusantara sejak tanggal 1 Agustus 2023, sebelumnya menjadi dosen prodi Bioinformatika dan sekaligus bertugas sebagai kepala prodi sejak tahun 2017 sampai tahun 2023 pada Indonesia International Institute for Life Sciences (i3L), dan sebelumnya juga pernah bekerja sebagai dosen tetap di prodi Teknik Informatika pada Universitas Amikom Yogyakarta pada tahun 2008 - 2009. Selama menjadi dosen pada Prodi Informatika dan Bioinformatika David lebih banyak

fokus pada mata kuliah pemrograman dan database dengan riset yang berkaitan dengan data analisis pada dunia kesehatan. David menyelesaikan pendidikan S1 pada Jurusan Teknik Informatika Universitas Amikom Yogyakarta Indonesia dan melanjutkan S2 dan S3 pada Jurusan Bioinformatics and Medical Engineering di Asia University Taiwan, fokus penelitian saat ini yaitu pada topik-topik berikut: bioinformatika, medical informatics, cancer epigenetics, deep learning pada data -omics dan memiliki antusias yang tinggi untuk mempelajari topik-topik baru di dunia IT dan Bioinformatika seperti Big data, AWS, software development dan lain sebagainya.



Monika Evelin Johan, S.Kom., M.M.S.M.

Dosen Program Studi Sistem Informasi
Fakultas Teknik dan Informatika,
Universitas Multimedia Nusantara

Penulis menempuh pendidikan S1 jurusan Sistem Informasi di Universitas Multimedia Nusantara, dan menyelesaikan S2 jurusan Magister Manajemen Sistem Informasi di Universitas Bina Nusantara. Setelah sempat bekerja di industri sebagai *quality assurance engineer*, penulis mengubah haluan kariernya sebagai dosen di Universitas Multimedia Nusantara jurusan Sistem Informasi. Bidang ajar yang ditekuni antara lain Data Modelling, Deep Learning, Web Design and Development, dan Mobile Application Development. Penulis dapat dihubungi melalui e-mail di monika.evelin@umn.ac.id.



Melissa Indah Fianty, S.Kom., MMSI

Dosen Program Studi Sistem Informasi
Fakultas Teknik dan Informatika
Universitas Multimedia Nusantara

Penulis adalah dosen tetap pada Program Studi Sistem Informasi Fakultas Teknik dan Informatika, Universitas Multimedia Nusantara. Menyelesaikan pendidikan S1

pada Teknik Informatika dan melanjutkan S2 pada Manajemen Sistem Informasi di Bina Nusantara University.



Samuel Ady Sanjaya S.T., M.T.

Dosen Program Studi Sistem Informasi
Fakultas Teknik dan Informatika
Universitas Multimedia Nusantara

Penulis lahir di Purworejo pada tanggal 5 April 1994. Penulis adalah dosen tetap pada Program Studi Sistem Informasi Fakultas Teknik dan Informatika, Universitas Multimedia Nusantara. Menyelesaikan S1 pada jurusan Teknik Informatika dan melanjutkan S2 pada jurusan Informatika.



Rudi Sutomo, S.Kom., M.Si., M.Kom.

Dosen Program Studi Sistem Informasi
Fakultas Teknik dan Informatika
Universitas Multimedia Nusantara

Rudi Sutomo memulai karirnya di bidang IT sebagai IT Lecturer di tahun 2002 dan mendalami perkembangan Software Engineering di Kampus dan IT Development di bidang Hospitality (Perhotelan), terlibat dalam beberapa Project Pembangunan Aplikasi untuk menunjang kemajuan dan kelancaran sistem IT di Hotel Novotel di Kota Palembang sebagai Asisten IT Manager. Aktif dan giat sebagai Web Developer yang berpengalaman di pembangunan di bidang Aplikasi Perkantoran dan bidang Pengembangan Website terutama Project Web Desain dan Website Development dengan terlibat dalam beberapa pembangunan project aplikasi seperti Aplikasi Visual Basic “Work Order Multi User” digunakan untuk seluruh Divisi dalam Hotel Novotel Palembang, Sistem Informasi Akademik (SIA) Kampus Tanri Abeng University dan Project berhubungan dengan PUSMENJAR (Pusat Asesmen dan Pembelajaran) Kemendikbud yaitu Project

PUSPENDIK AKSI dan beberapa Project Minor lainnya. Beliau juga aktif berpartisipasi dalam workshop mengenai Pembuatan website baik di sekolah maupun di masyarakat dan beberapa kali dipercaya sebagai Narasumber salah satunya di “Pelatihan Pengenalan dan Pembangunan Website” di Sekolah SMKN 43 Jakarta Tahun 2019. Pada Event Penghargaan Website Tingkat Nasional diberi kepercayaan sebagai Juri pada IWA (Indonesia Website Awards) Penilaian periode Tahun 2018 dan 2019.



Wirawan Istiono, S.Kom., M.Kom.

Dosen Program Studi Informatika
Fakultas Teknik dan Informatika
Universitas Multimedia Nusantara

Penulis lahir di Jakarta tanggal 13 April 1983. Penulis adalah dosen pada Program Studi Informatika Fakultas Teknik dan Informatika, Universitas Multimedia Nusantara.

Menyelesaikan pendidikan S1 pada Jurusan Teknik Informatika dan melanjutkan S2 pada bidang jurusan yang Teknik Informatika. Pada saat ini, penulis menekuni bidang Penelitian terkait komputer sains, sistem komputer dan teknologi informasi.



Suwito Pomalingo, S.Kom., M.Kom

Dosen Program Studi Informatika
Fakultas Teknik dan Informatika
Universitas Multimedia Nusantara

Penulis adalah dosen tetap pada Program Studi Informatika Fakultas Teknik dan Informatika Universitas Multimedia

Nusantara. Menyelesaikan pendidikan S1 pada Jurusan Teknik Informatika dan melanjutkan S2 pada Program Magister Informatika Universitas Islam Indonesia.

Saat ini sementara menyelesaikan studi Doktoral dengan topik penelitian di bidang Security Science dengan pendekatan Deep Learning. Penulis menekuni bidang keilmuan meliputi Cyber

Security, Malware Analytics, Data Science, Machine Learning, dan Deep Learning, selain itu, penulis juga memiliki beberapa serifikasi internasional di bidang Cyber Security, Digital Forensics, Data Science dan Machine Learning.



Jansen Wiratama, S.Kom., M.Kom.

Dosen Program Studi Sistem Informasi
Fakultas Teknik dan Informatika,
Universitas Multimedia Nusantara

Penulis lahir di Tangerang, tanggal 09 Januari 1993. Sejak tahun 2018 Penulis berkarir sebagai Dosen pada bidang ilmu Sistem Informasi. Selain berprofesi sebagai Dosen.

Penulis juga aktif sebagai IT Consultant. Penulis menyelesaikan pendidikan studi Strata-1 pada Jurusan Teknik Informatika tahun 2015 di STMIK Dharma Putra, Tangerang dan menyelesaikan studi Strata-2 pada Jurusan Magister Ilmu Komputer tahun 2017 di Universitas Budi Luhur, Jakarta. Dengan pengetahuan dan pengalaman dalam bidang Sistem Informasi, Penulis juga aktif dalam melakukan publikasi karya ilmiah berupa jurnal yang berkaitan dengan Perancangan Sistem Informasi dan Penerapan Sistem ERP. Penulis dapat dihubungi melalui email: jansen.wiratama@umn.ac.id. Hasil publikasi Penulis dapat dilihat melalui beberapa Portal Publikasi seperti Sinta ID: 6659377 dan Scopus ID: 58080730500.



Dr. Erick Fernando, S.Kom., M.S.I.

Dosen Program Studi Sistem Informasi
Fakultas Teknik dan Informatika
Universitas Multimedia Nusantara

Penulis saat ini merupakan dosen tetap pada Program Studi Sistem Informasi Fakultas Teknik dan Informatika, Universitas Multimedia Nusantara. Menyelesaikan pendidikan S1 pada Jurusan Teknik Informatika dan melanjutkan

S2 pada Jurusan Magister Sistem Informasi. Serta Menyelesaikan S3 pada Doktor of Computer Science. Penulis menekuni bidang Basis data, Teknologi Blockchain, Adopsi Teknologi, Big Data, Sistem Informasi Geografis, *Emmbeded System* DIII.



Ahmad Faza, S.Kom., M.T.I.

Dosen Program Studi Sistem Informasi
Fakultas Teknik dan Informatika
Universitas Multimedia Nusantara

Penulis lahir pada 12 Januari 1995 di Bogor. Dengan latar belakang sebagai dosen tetap di Program Studi Sistem Informasi, Fakultas Teknik dan Informatika, Universitas Multimedia Nusantara. Penulis telah menyelesaikan pendidikan tinggi tingkat S1 di Jurusan Sistem Informasi, Universitas Multimedia Nusantara, serta melanjutkan studi ke jenjang S2 di program Magister Teknologi Informasi, Universitas Indonesia. Komitmen Penulis terhadap bidang Sistem Informasi tercermin dalam minat penelitiannya pada IT Governance. Keahliannya didukung oleh sertifikasi pada ITIL V3, yang memberikan landasan kuat dalam mengelola dan mengatur teknologi informasi dengan efektif. Penulis memiliki aspirasi untuk berperan aktif dalam mengembangkan penelitian di bidang sistem informasi di masa depan.



Alexander Waworuntu, S.Kom., M.T.I.

Dosen Program Studi Informatika
Fakultas Teknik dan Informatika
Universitas Multimedia Nusantara

Penulis lahir di Surabaya pada tanggal 9 Juni. Penulis adalah dosen pada Program Studi Informatika, Fakultas Teknik dan Informatika, Universitas Multimedia Nusantara. Menyelesaikan pendidikan S1 pada jurusan Teknik Informatika pada tahun 2006 di Universitas Bina Nusantara. Selanjutnya penulis menyelesaikan pendidikan S2 pada program

studi Magister Teknologi Informasi di Universitas Indonesia pada tahun 2012.