

Survei Metode-Metode *Software Development Life Cycle* Dengan Metode *Systematic Literature Review*

Sayyid Muhammad Uwais Fathanah Assegaf¹, Fadillah Siva², Sheva Athalla Pahlevi³, Muhammad Ainul Yaqin^{4,*}

Jurusan Teknik Informatika, Universitas Islam Maulana Malik Ibrahim, Indonesia

¹19650052@student.uin-malang.ac.id; ²19650055@student.uin-malang.ac.id; ³19650056@student.uin-malang.ac.id;

⁴yaqinov@ti.uin-malang.ac.id

*corresponding author

INFO ARTIKEL

Sejarah Artikel

Diterima: 18 Maret 2022

Direvisi: 20 Mei 2022

Diterbitkan: 30 Agustus 2023

Kata Kunci

Software Development Life Cycle,
Software Development,
Software Engineer,
Systematic Literatur

ABSTRAK

Software development adalah sebuah rangkaian tahap dalam mengembangkan produk berupa sebuah *software*, untuk memiliki *value usability*. Penelitian ini membahas tentang survei metodologi *software development* dengan berbagai macam model. SDLC juga sebuah mekanisme bertujuan memastikan *software* yang dibangun sudah memenuhi persyaratan atau belum. Metodologi yang dibahas di penelitian ini meliputi metode Air terjun (*Waterfall*), metode Prototipe, metode *Rapid Application Development*, model *Evolutionary*, model *Agile*, model *Fountain*, metode *Rational Unified Process*, metode *Build and Fix*, metode *Extreme Programming*, dan metode *V-model*. Survei ini bertujuan untuk mengetahui keefektifan, perbedaan dan tahap pengembangan *Software Development Life Cycle* dalam menyelesaikan masalah dan metode-metode dalam pengembangan SDLC. Metode survei yang digunakan adalah *systematic literature review* terhadap *paper significant* yang membahas isu terkait. Manfaat yang diharapkan, dapat memberikan pembaca tentang keefektifan, perbedaan, dan tahap pengembangan serta metode yang bagus untuk mengembangkan *Software Development Life Cycle* dalam menyelesaikan masalah. Survei ini menghasilkan data keunggulan dan kelemahan metode. Hasil survei didapatkan temuan beberapa rekomendasi pilihan yaitu akan mensurvei dan membuat perbandingan di antara setiap metodologi *software development life cycle* hingga mendapatkan kesimpulan terkait metode mana yang cocok untuk membuat setiap jenis *software*.

PENDAHULUAN

SDLC (*Software Development Life Cycle*) atau bisa juga disebut sebagai siklus dalam sebuah pengembangan *software* dalam rekayasa perangkat lunak merupakan tahapan pembentukan serta modifikasi *software*, model, dan juga metode yang diterapkan dalam pengembangan perangkat lunak. Azhar Susanto memiliki pandangan bahwa SDLC ini sangat populer sebagai metode pengembangan sistem informasi saat pertama kali sistem informasi dibuat [1]. Konsep ini biasanya mengacu pada perangkat lunak maupun sistem informasi. SDLC adalah konsep pengembangan perangkat lunak dan memiliki tahap: perencanaan, analisis, desain, implementasi, dan pengujian. Dalam pengembangan perangkat lunak, konsep SDLC membentuk dasar dari banyak jenis metodologi pengembangan perangkat lunak. Metode SDLC membangun pola kerja pada perancangan dan pengelolaan dalam membuat perangkat lunak, yaitu tahapan dalam mengembangkan perangkat lunak. Masalah yang dirumuskan dalam *paper* survei ini terkait dengan keefektifan *Software Development Life Cycle* dalam menyelesaikan masalah dan apa sajakah metode-metode dalam pengembangan SDLC, tentang perbedaan dari setiap metode-metode pada SDLC yang lebih efisien dalam menyelesaikan masalah serta tentang tahapan pengembangan suatu aplikasi menggunakan *Software*

Development Life Cycle. Akan tetapi belum dapat memberikan kepastian tentang tingkat keefektifan atau metode yang efisien dalam menyelesaikan permasalahan dengan menggunakan SDLC diberbagai kasus tentang implementasi SDLC. Dengan kata lain, masih banyak yang belum diketahui artinya masih banyak pemilihan metode untuk pengembangan *Software Development Life Cycle* yang kurang efektif dan efisien dalam menyelesaikan permasalahan. Perumusan masalah tersebut melatarbelakangi survei *Software Development* ini.

Survei ini bertujuan untuk memberikan tinjauan secara komprehensif mengenai keefektifan *Software Development Life Cycle* serta mengetahui metode apa saja yang terdapat di SDLC. Selain itu, untuk mengetahui efisiensi perbedaan setiap metode dan tahapan-tahapan dalam pengembangan suatu aplikasi dengan menggunakan *Software Development Life Cycle*. Hal tersebut penting, karena jika *Software Development Life Cycle* yang digunakan salah, maka akan banyak membuang waktu dan kerugian secara materi. Jika menggunakan *Software Development Life Cycle* digunakan dengan tepat maka prosesnya akan lebih efektif, efisien dalam menyelesaikan permasalahan. Survei yang dilakukan menggunakan metode *systematic literature review* atau studi literatur sistematis terhadap *paper-paper* penelitian yang membahas mengenai *Software Development Life Cycle*. Dimulai pada pengumpulan data primer dan data sekunder, lalu melakukan *review paper*, menemukan jurnal dan *paper* relevan yang membahas isu terkait.

Solusi yang ditawarkan atas penyelesaian masalah adalah dengan membandingkan satu pengembangan ke pengembangan lainnya dan dapat ditinjau dari keefektifan metode pada SDLC, keefisien metode pada SDLC dalam menyelesaikan masalah. Tahapan metode pengembangan *Software Development Life Cycle* dalam menyelesaikan masalah. Dari analisa perbandingan tersebut, kemudian dapat ditarik kesimpulan mana pengembangan yang lebih efektif, efisien pada *Software Development Life Cycle* dalam menyelesaikan masalah. Hasil dari survei ini berupa ulasan tentang keefektifan, kekurangan, kelebihan, riwayat penelitian masing-masing pengembangan *Software Development Life Cycle* yang ditemukan. Survei ini juga akan menghasilkan rekomendasi pilihan *Software Development Life Cycle* yang dibahas di dalamnya. Juga analisa dalam bentuk tabel dan diagram yang akan memberikan informasi kepada pembaca tentang jumlah penggunaan, nama peneliti, dan tahun penelitian setiap pengembangan *Software Development Life Cycle*.

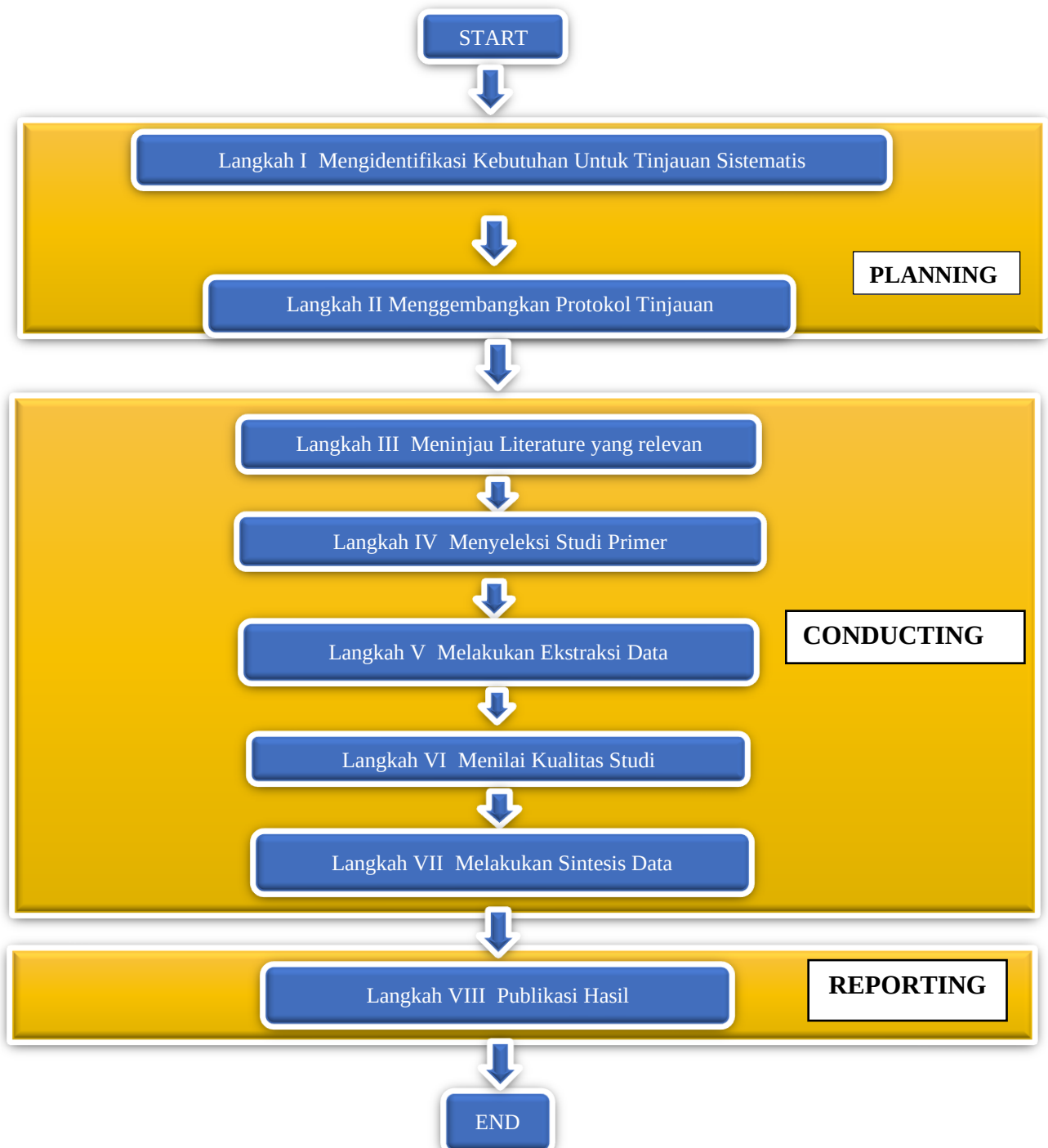
Manfaat yang diharapkan, peneliti selanjutnya dapat merefleksikan contoh kasus dalam *paper* ini dengan kasus pengembangan *Software Development Life Cycle* yang sedang mereka teliti. Sehingga mereka dapat menemukan solusi menggunakan rekomendasi pilihan pengembangan *Software Development Life Cycle* dengan metode yang sesuai dengan kasus mereka. Sedangkan untuk *researchers*, hasil dari survei ini dapat dijadikan dasar untuk menggali *Software Development Life Cycle* perangkat lunak lain atau yang terbaru. Urgensi dari penelitian ini dikarenakan masih sedikit *paper* yang membahas secara jelas perbedaan masing-masing metode SDLC menggunakan metode literatur yang sistematis. Sehingga pembuktian tidak merujuk secara jelas terhadap referensi dari yang digunakan oleh peneliti

Penelitian tentang perbandingan metode SDLC telah dilakukan oleh Titania Pricillia dan Zulfachmi pada tahun 2021 dengan judul “*Survey Paper: Perbandingan Metode Pengembangan Perangkat Lunak*”. Pada jurnalnya, peneliti melakukan perbandingan terhadap tiga metode SDLC yaitu metode *waterfall*, *prototype*, dan *rapid application development*. Penelitian ini menjelaskan tahapan-tahapan pada masing-masing metode. Peneliti membuat tabel perbandingan berupa kelebihan dan kekurangan dari masing-masing metode. Namun pada *paper* ini, penelitian belum menggunakan metode *systematic literature review*, sehingga *review* dari literatur belum sistematis. Referensi yang menyatakan tentang kelebihan dan kekurangan dari masing-masing metode tidak jelas. Dan *paper* ini hanya membahas tiga metode SDLC saja [2].

METODE

Systematic Literature Review

SLR (*Systematic Literature Review*) sebuah tahapan untuk mengidentifikasi, menaksir, hingga menafsirkan data untuk topik penelitian tertentu demi memenuhi pertanyaan penelitian yang telah ditentukan [3]. *Parser* SLR dilakukan melalui tinjauan sistematis dan identifikasi jurnal dengan mengikuti tahapan tertentu. Tujuan penggunaan metode SLR adalah untuk mengembangkan ide yang jelas berdasarkan rumusan masalah yang diteliti.



Gambar 1. Langkah *Systematic Literature Review*

Membuat *Research Question* (RQ)

Research question merupakan suatu pertanyaan yang diajukan kepada sebuah penelitian berupa pertanyaan yang jelas, fokus, dan padat terhadap permasalahan yang dibahas pada penelitian tersebut. *Research Question* memiliki fungsi mengarahkan pencarian dan membatasi pembahasan pada sebuah penelitian dan literasi. RQ harus memiliki formula berdasarkan 5 elemen, yaitu Populasi, Intervensi, Komparasi, Hasil, dan Konteks atau disingkat dengan PICOC.

Tabel 1. PICOC Table

<i>Population</i>	<i>Software Defelopment Life Cycle</i>
<i>Intervention</i>	<i>Classification, software, types of software development life cycle</i>
<i>Comparison</i>	<i>Comparison between software defelopment life cycle methods</i>
<i>Outcomes</i>	<i>Recommendation of Software Defelopment Life Cycle methods that suitable to be used in specific cases</i>
<i>Context</i>	<i>Survey based on literature, papers, journals, and books</i>

Tabel 2. *Research Question*

RQ	Question	Tujuan
RQ1	Jurnal apa saja yang fokus membahas tentang SDLC?	Identifikasi jurnal atau <i>paper</i> yang focus membahas tentang metode SDLC
RQ2	Metode apa saja yang digunakan oleh para peneliti dalam SDLC?	Identifikasi setiap metode yang diimplementasikan pada SDLC
RQ3	Metode apa yang paling banyak digunakan para peneliti pada kurun tahun 1991-2022?	Identifikasi metode SDLC yang paling banyak digunakan
RQ4	Rekomendasi metode apa yang tepat diimplementasikan untuk beberapa contoh kasus?	Identifikasi metode yang tepat diimplementasikan pada contoh kasus tertentu

RQ 1 membantu untuk menyaring literatur. RQ2 membantu untuk menemukan informasi tentang metode pada SDLC. RQ3 memberikan informasi implementasi metode SDLC dalam kurun waktu tertentu. RQ4 memberikan rekomendasi metode SDLC yang tepat digunakan dalam suatu kasus.

Strategi Pencarian

Pencarian dilakukan dengan menentukan kalimat pencarian yang berhubungan dengan topik permasalahan, lalu memilih Pustaka digital yang tepat dengan melakukan penyaringan dan mengambil daftar awal studi yang relevan [4]. Memilih database dilakukan sebelum melakukan pencarian dengan tujuan untuk menemukan artikel relevan dengan topik yang akan diangkat agar memiliki cakupan literatur luas. Pencarian dilakukan dengan menggunakan *keyword*:

("software development life cylce" **OR** "software development life cycle method")

AND

("systematic literature review" **OR** "SLR" **OR** "survey data"**OR**"Data Collection")

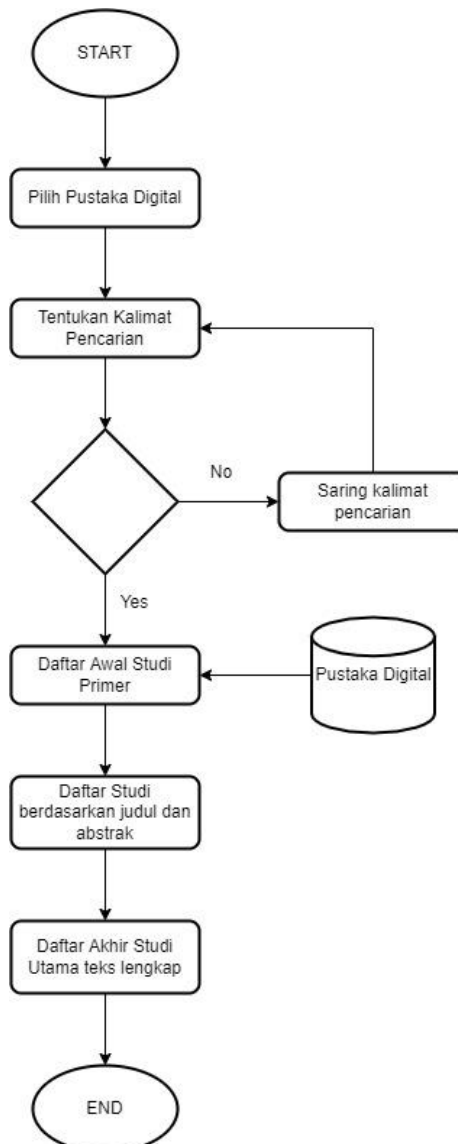
Seleksi Studi

Penyaringan studi primer dilakukan pada fase ini dengan cara *review* judul dan abstrak jurnal yang telah ditemukan. Namun sebelumnya, kriteria *inklusi* dan *eksklusi* dilakukan dahulu agar dapat memilih studi utama yang relevan. Pemilihan studi diberikan kriteria dengan tujuan untuk identifikasi studi utama yang telah memiliki bukti langsung tentang pertanyaan penelitian. Kriteria tersebut harus mendasar kepada pertanyaan pada penelitian tersebut[5].

Tabel 3. *inklusi dan eksklusi*

<i>Inklusi</i>	<i>Eksklusi</i>
Studi berupa buku, jurnal penelitian, dan <i>paper</i>	Studi selain buku, jurnal, dan <i>paper</i>
Studi yang membahas metode SDLC	
Studi yang membahas masalah pada SDLC dan cara mengatasinya	Studi yang membahas SDLC tanpa membahas solusi
Studi yang membahas efektifitas metode SDLC	

Proses rinci pencarian dan jumlah studi yang diidentifikasi pada setiap tahap yang disajikan oleh Gambar 2 dibawah ini:



Gambar 2. Proses Pencarian dan Seleksi Studi

Daftar studi awalnya terdapat 70 (termasuk jurnal, buku, makalah, dan artikel). Lalu studi yang telah didapat diseleksi dengan kriteria *inklusi* maupun *eksklusi*. Relevansi terhadap *research question* juga menjadi pertimbangan dalam seleksi studi. Setelah di seleksi maka akhirnya didapatlah sebanyak 48 studi teks lengkap.

Data Collection

Data Collection merupakan tahap pengumpulan serta memastikan *variable of interest* secara sistematis [4]. Kitcherman berpendapat bahwa dalam melakukan *data collection* memiliki standarisasi berupa ekstraksi data dan pemeriksaan jurnal yang dilakukan oleh semua anggota penelitian[5]. Pada penelitian ini, data primer adalah *paper* yang memiliki judul maupun isi pembahasan yang langsung membahas tentang metode SDLC. Data Sekunder pada penelitian ini adalah buku maupun *paper* yang membahas studi kasus mengenai mimplementasi metode SDLC dan metode penelitian. Tahapan pengumpulan data penelitian adalah sebagai berikut:

1. Observasi ke perpustakaan digital.
2. Pengkajian studi data sesuai metode SLR.
3. Data didokumentasi dan disimpan.

Analisis Data

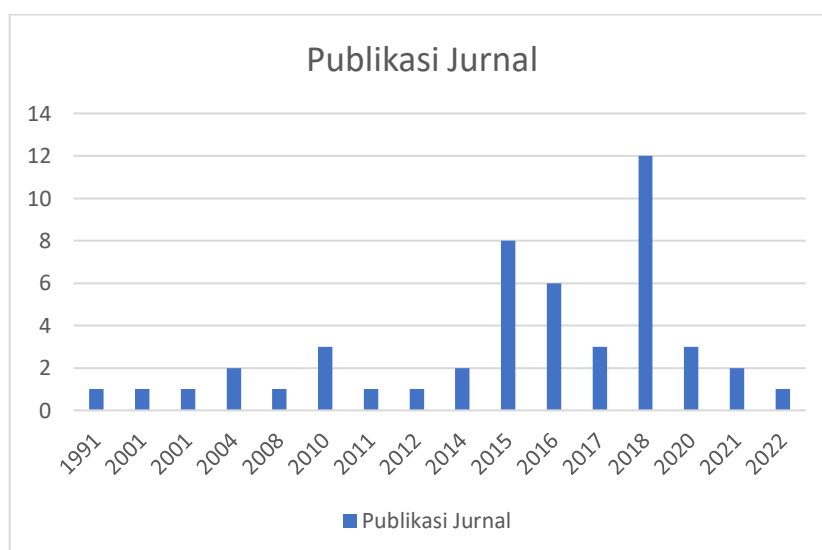
Pada tahap ini, data studi dikumpulkan untuk menjawab pertanyaan penelitian / *Research Question*[4]. Data-data diidentifikasi dengan *research question* dan hasilnya didapatkanlah property yang menjawab *research question*.

Tabel 4. Properti untuk menjawab *Research Question*

Properti	<i>Research Question</i>
Jurnal yang relevan	RQ1
Metode SDLC yang digunakan	RQ2, RQ3, RQ4

Klasifikasi Jurnal

Paper yang telah ditemukan, di klasifikasikan berdasarkan tahun dengan diagram pada Gambar 3.



Gambar 3. Jumlah *Paper* berdasarkan tahun

Jurnal yang Memuat Paper mengenai Metode SDLC (RQ1)

- *Jurnal ESIT*
- *Information System For Educators And Professionals: Journal Of Information System*
- *SNIT*
- *Media Jurnal Informatika*
- *Jurnal Informatika Universitas Pamulang*
- *Creative Information TECHNOLOGY JOURNAL*
- *Jurnal Informatika*
- *Prosiding Semnastek*
- *Jurnal Informa: Jurnal Penelitian Dan Pengabdian Masyarakat*
- *Annales Universitatis Mariae Curie-Sklodowska. Sectio H. Oeconomia*
- *JUSIM (Jurnal Sistem Informasi Musirawas)*
- *Informatica Economica*
- *Jurnal Nasional Informatika Dan Teknologi Jaringan*
- *Ultimatics: Jurnal Teknik Informatika*
- *Seminar Nasional Informatika (SEMNASIF).*
- *Springer Science & Business Media*
- *Insan Pembangunan Sistem Informasi dan Komputer (IPSIKOM)*
- *Jurnal Sisfotek Global*
- *International Journal of Advanced Research in Computer Science and Software Engineering*
- *JURNAL PILAR TEKNOLOGI: Jurnal Penelitian Ilmu-Ilmu Teknik*
- *Fountain of Informatics Journal*
- *Jurnal Fasikom*
- *Jurnal Teknik Informatika*
- *eProceedings of Engineering*
- *PROSISKO: Jurnal Pengembangan Riset dan Observasi Sistem Komputer*
- *INCOSE international symposium*
- *Jurnal Teknik Elektro, Teknologi Informasi dan Komputer*
- *Teknika*
- *PROSIDING SNAST*

Klasifikasi Paper Yang Membahas Metode SDLC

Setelah *paper-paper* yang ditemukan telah disaring, maka ditemukanlah 42 *paper* yang membahas tentang metode SDLC. TKlasifikasi *paper* dibuktikan pada tabel 5 penelitian ini dikelompokkan berdasarkan metode SDLC yang dibahas.

Tabel 5. Klasifikasi Paper yang Membahas Metode SDLC

No	Metode SDLC	Paper
1	<i>Waterfall</i>	[6], [7], [8], [9], [10]
2	<i>Protoype</i>	[11], [12], [13], [14], [15], [16]
3	<i>Rapid Application Development</i>	[17], [18], [19], [20], [21],
4	<i>Evolutionary</i>	[22], [23], [24], [25]
5	<i>Agile</i>	[26], [27], [28], [29], [30], [31], [32], [33]
6	<i>Rational Unified Proccess</i>	[16], [34], [35], [36]
7	<i>Build and Fix</i>	[37], [38], [39], [40], [41]
8	<i>Extreme Programming</i>	[42], [43], [44], [45], [46], [47]

9	<i>V Shape Model</i>	[48], [49], [50]
10	<i>Fountain</i>	[51], [41], [52],

Kebaharuan Ilmiah

Paper ini memuat penelitian dengan contoh kasus yang telah diteliti sebelumnya yang berupa hasil penelitian berupa saran penerapan metode SDLC dalam berbagai kasus

- Saran implementasi metode SDLC dalam kasus Perancangan Aplikasi *Tutorial*
- Saran implementasi metode SDLC dalam kasus Pengembangan *E-Trace*
- Saran implementasi metode SDLC dalam kasus Pembangunan WEB aplikasi

HASIL DAN PEMBAHASAN

Hasil survei Metode-Metode SDLC

Daftar metode-metode SDLC yang dapat ditemukan pada penelitian survei ini adalah sebagai berikut:

<i>Waterfall</i>	<i>Prototype</i>
<i>Rapid Application Development</i>	<i>Evolutionary</i>
<i>Agile</i>	<i>Relational Unified Proccess</i>
<i>Build and Fix</i>	<i>Extreme Programming</i>
<i>V Shape Model</i>	<i>Fountain</i>

Skenario Analisis terhadap Hasil Survei

Untuk mengklarifikasi metode-metode SDLC berdasarkan batasan tertentu, maka dilakukanlah analisis untuk menemukan variable yang dapat menjawab pertanyaan penelitian(*Research Question*). Berikut adalah scenario yang dilakukan:

1. Analisis pertama (untuk menjawab RQ2). Membandingkan kelebihan dan kekurangan metode-metode SDLC berdasarkan studi yang telah dikumpulkan.
2. Analisis kedua (untuk menjawab RQ3). Mengklarifikasi metode-metode SDLC yang sering diteliti pada tahun 1991-2022.
3. Analisis ketiga (untuk menjawab RQ4). Menganalisis saran implementasi metode SDLC ynag digunakan dalam 3 contoh studi kasus: Pengembangan aplikasi tutorial, pengembangan E-Trace, dan Pembangunan web aplikasi.

Hasil Analisis Pertama (RQ2)

Metode-metode SDLC yang telah ditemukan akan dibandingkan dan dijabarkan dalam tabel dengan memperhatikan perbedaan berupa kelebihan dan kekurangan masing-masing metode dan mencantumkan *source* dari penemuan tersebut.

Tabel 6. Perbandingan metode-metode SDLC

No	Metode SDLC	Kelebihan	Kekurangan
1	<i>Waterfall</i>	- Memiliki pendekatan alur hidup yang terurut [6] - Kualitas sistem yang dihasilkan baik karena pelaksanaan dijalankan bertahap [7] [8] - Memudahkan pengembangan sistem karena dokumen yang terorganisir [7] [9]	- Memakan banyak waktu [10] - Memiliki potensi kerugian akibat kesalahan sangat besar [7] - Memerlukan manajemen yang baik [7] - Kesalahan kecil akan menjadi kesalahan besar yang merugikan [7] - <i>Client</i> sulit mengungkapkan kebutuhan eksplisit [7]
2	<i>Prototype</i>	- <i>Client</i> dapat menjelaskan kebutuhan secara rinci [12]	- Tidak optimalnya Teknik dan tools [15]

		- Proyek memiliki peluang yang sangat besar untuk selesai sesuai target [12] - Komunikasi yang baik antara <i>develop</i> dengan <i>client</i> [13] - Pengembangan aplikasi menjadi lebih mudah [13] - Memiliki syarat yang lebih baik [14] - Hasil yang dicapai akurat [14]	- Pemeliharaan yang sulit [15] - Analisis dan perancangan sangat singkat sehingga miskin dokumen [16] - Tidak fleksibel ketika ada perubahan [16] - Sulit mengelola dan mengendalikan tahapan [16]
3	<i>Rapid Application Development</i>	- Dapat menghemat dalam keseluruhan project [20][21] - Pengembangan aplikasi berfokus pada waktu penyelesaian [20] - Cepat mengadopsi perubahan pada sistem [20] [21] - UI dan fungsi sistem yang baik didajikan untuk <i>user</i> [20] - Tercipta rasa kepemilikan yang kuat bagi seluruh pemangku proyek [20]	- Analisis project terburu-buru [20][21] - Hasil tidak menekankan pada permasalahan yang seharusnya diarahkan [20][21] - Tidak detail [20] - Menyulitkan <i>programmer</i> baru dalam pengembangan sistem.[20][21] - Tidak semua aplikasi cocok untuk metode ini [21] - Tidak sesuai untuk sistem dengan resiko tinggi [21]
4	<i>Evolutionary</i>	- Ketika proyek masig berjalan, <i>develope</i> dapat mengembangkan konsep dari sistem tersebut [23] - Tiap iterasi mudah dipelihara [24] - Pembiayaan yang murah [25] - Waktu pengembangan singkat [25]	(Efektif untuk berbagai jenis kasus)
5	<i>Agile</i>	- Fleksibel untuk penyesuaian keadaan apapun [28] - Kepuasan pelanggan adalah prioritas, dan kolaborasi dengan pebisnis [29][30][31] - <i>Software</i> yang dihasilkan baik, serta waktu pengerjaan yang konstan [29][31] - Pengujian dilaksanakan tiap saat [30][32] - <i>Requirement</i> bisa berubah kapan saja [31][32]	- Jarang dipraktekkan langsung, dan juga di project besar [31][32] - Komunikasi terlalu sering dengan <i>client</i> [31][32] - Sulit diterapkan pada proyek berskala besar[32] - Perlu adanya manajemen tim yang terlatih[31][32] - Lemah dalam arsitektur [32][33] - Memiliki waktu yang terbatas [31][32]
6	<i>Rational Unified Proccess</i>	- Memiliki akses yang mudah bagi pengetahuan dasar tim [16][35] - Proses pengulangan dan pengembangan didukung [16][35] - Penambahan fitur pada proses dimungkinkan [16][35] - Tes dimungkinkan saat sedang pengembangan [16][35] - Memungkinkan perubahan yang terjadi dalam pengembangan sistem [36]	- Hanya tersedia dalam perangkat lunak berorientasi objek yang berfokus pada UML [16][35] - Membutuhkan waktu yang lama [16][35]
7	<i>Build and Fix</i>	- Cocok untuk proyek skala kecil [38] - Tidak perlu banyak perencanaan [38]	- Metode yang kasar karena sistem tidak memiliki spesifikasi atau ide rancangan [41]

		- Cocok untuk aplikasi yang membutuhkan waktu singkat [39] - Pembangunan sistem yang cepat [40]	
8	<i>Extreme Programming</i>	- Komunikasi dengan <i>client</i> berjalan baik [45] - Meningkatkan saling menghargai dan komunikasi yang baik antar pengembang [45] - Waktu pengembangan singkat walau kebutuhan berubah [45][46] - Dilakukan hanya dengan 2-10 orang dalam 1 tim [45] - Cocok untuk proyek berskala kecil [47]	- Tidak banyak yang menggunakan, karena tiap individu dituntut dengan spesialis berbeda [46]
9	<i>V Shape Model</i>	- Sederhana [49][50] - <i>Delivery</i> tertentu pada setiap survei [49][50] - Memiliki peluang keberhasilan yang tinggi [49][50] - Cocok untuk proyek kecil [49][50]	- Kaku [49][50] - Tidak terlalu fleksibel [49][50] - Pengembangan yang sulit dan mahal [49][50] - Tidak memberi solusi yang jelas pada suatu masalah [49][50]
10	<i>Fountain</i>	- Dapat melewati tahapan lain kecuali desain [41][51][52] - Waktu membangun sistem singkat [51]	(Efektif untuk berbagai jenis kasus)

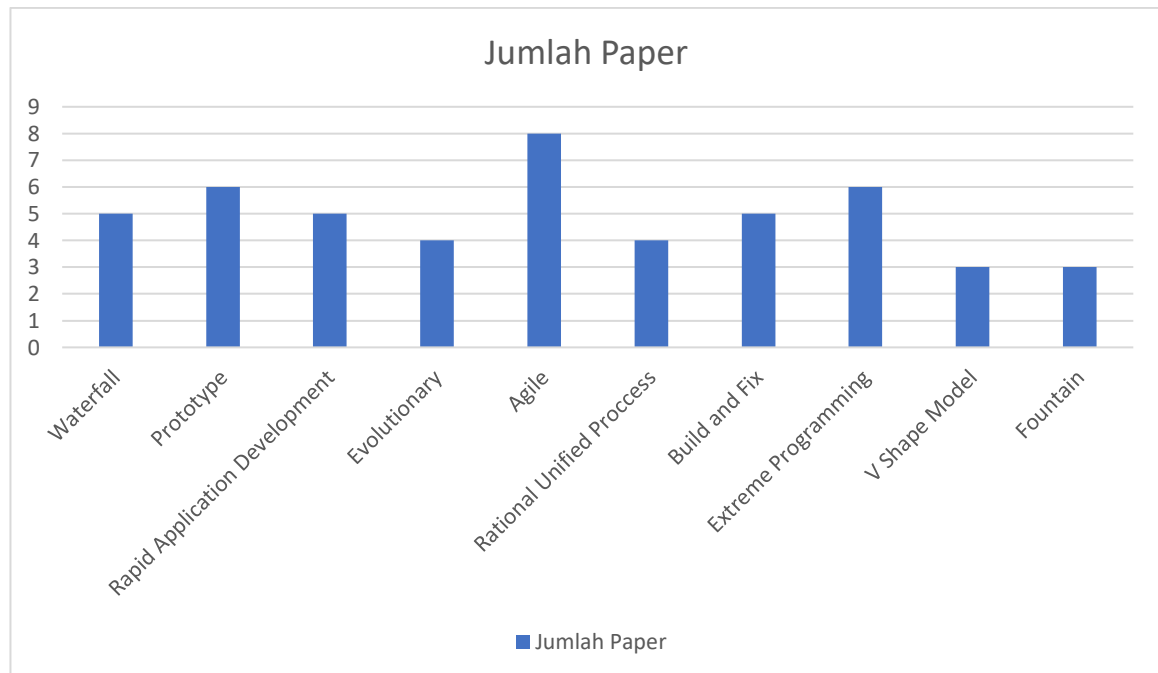
Berdasarkan tabel 6, ditemukan 10 metode SDLC dengan menganalisis kelebihan, kekurangan, serta *source paper* yang menjadi rujukan metode tersebut, lalu dibandingkan dengan metode-metode yang lain. Dari hasil perbandingan, ditemukan beberapa metode dengan tulisan “(efektif untuk berbagai jenis kasus)” yaitu metode *Fountain* dan metode *Evolutionary*. Tulisan tersebut memiliki makna bahwa metode itu dapat digunakan pada seluruh jenis kasus apapun sehingga tidak terklarifikasi kekurangan dari metode tersebut. Selain kedua metode tersebut, maka dapat digunakan hanya pada kondisi kasus tertentu dan akan bermasalah jika diimplementasikan pada jenis kasus lain. Metode-metode tersebut hanya bisa diimplementasikan pada kondisi proyek seperti yang telah dijelaskan pada kekurangan dan kelebihan dari metode-metode tersebut. Agar tepat sasaran dan mengurangi resiko kegagalan.

Hasil Analisis Kedua (RQ3)

Berdasarkan analisis pada tabel 5, dapat diklarifikasikan lebih lanjut untuk dapat menjawab RQ3, yaitu metode mana yang paling banyak digunakan peneliti pada tahun 1991-2022. Klarifikasi dapat dilakukan dengan cara menganalisa diagram dan membandingkan jumlah metode-metode SDLC dengan Batasan sebagai berikut:

- Yang dianalisa adalah metode hasil dari survei pada *paper* ini
- Batasan waktu metode adalah tahun 1992-2022.

Diagram memanfaatkan tahun dari publikasi *paper-paper* yang diambil untuk penelitian ini. Diagram diperlihatkan oleh Gambar 4.

Gambar 4. Jumlah *paper* per metode

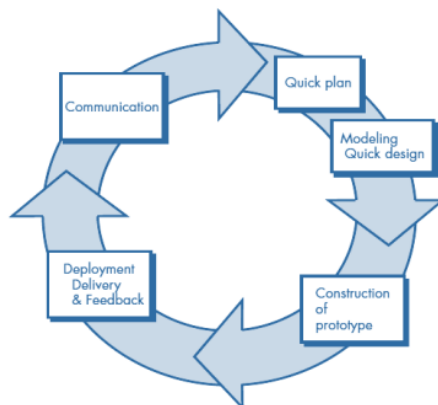
Dari diagram Gambar 4, dapat ditarik kesimpulan metode *Agile* adalah metode yang paling banyak digunakan dengan total 8 publikasi [26], [27], [28], [29], [30], [31], [32], [33]; *Extreme Programming* dengan 6 publikasi [42], [43], [44], [45], [46], [47]; *Protoype* dengan 6 publikasi juga [11], [12], [13], [14], [15], [16]; Ketiga metode SDLC tersebut banyak digunakan pada penelitian dalam studi kasus, *paper*, dan lainnya. Jumlah publikasi *paper* penelitian yang tinggi tentang suatu metode SDLC, menandakan beberapa kemungkinan. Pertama, metode tersebut banyak digunakan karena memiliki tingkat relevansi dan akurasi yang tinggi. Kedua, metode tersebut merupakan asal mula dari metode lain yang lahir setelahnya. Ketiga, metode tersebut juga diimplementasikan, Teknik tersebut kompeten di banyak kasus.

Hasil Analisis Ketiga (RQ4)

Pada analisis ketiga ini, terdapat tiga kasus. Ketiga kasus ini dianalisis dan akan menghasilkan rekomendasi implementasi metode SDLC tertentu pada masing-masing kasus. Berikut hasil analisis ketiga:

Perancangan Aplikasi Tutorial

Kasus ini dijabarkan oleh Dimas Abisono Punkastyo didalam *paper*-nya yang membahas tentang membangun aplikasi yang berisi pembelajaran dasar jurus beladiri cimannda yang dirancang menggunakan metode prototipe yang di publikasikan di Jurnal Informatika Universitas Pamulang pada tahun 2018. Tujuan dari pembuatan aplikasi ini adalah memfasilitasi siswa untuk belajar bela diri dengan interaktif walau tidak bertatap muka. Pada kasus ini, penulis *paper* menggunakan metode *Prototype* dalam membangun sistem aplikasi ini.



Gambar 5. Metode Prototype

Pada Gambar 5, dapat dilihat ada 4 tahapan dalam metode *prototype* menurut Rogger S.Pressman,yang dijelaskan pada Tabel 7.

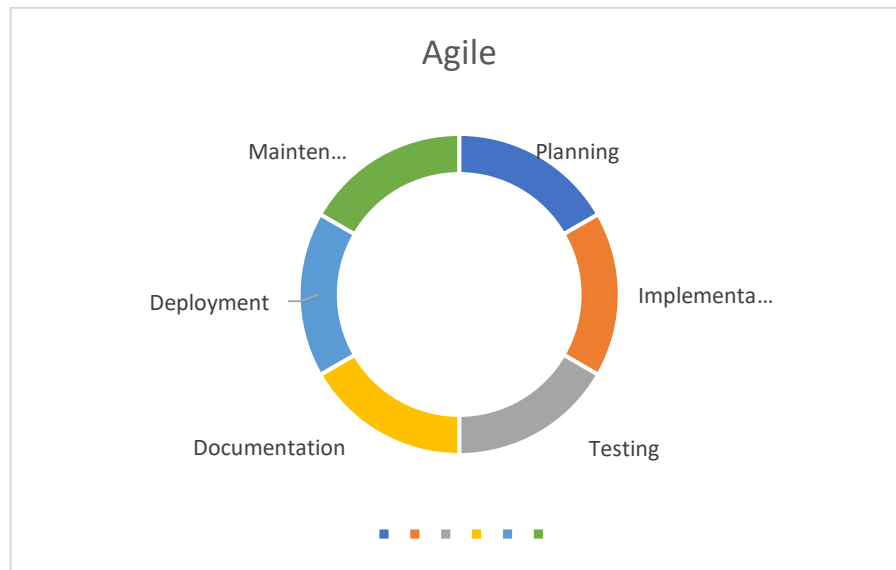
Tabel 7. Penjelasan tahapan metode *Prototype*

No	Tahapan	Penjelasan
1	<i>Communication</i>	Tahap ini merupakan tahap dimana <i>develope</i> dan <i>client</i> berkomunikasi untuk menjabarkan tujuan dari sistem yang ingin dibangun.
2	<i>Quick plan, Modelling, dan Quick Design</i>	Setelah komunikasi telah terjalin, dan gambaran umum sistem telah diketahui, maka pada tahap ini berfokus kepada perancangan <i>User Interface</i> pada sistem yang ingin dibangun.
3	<i>Construction of Prototype</i>	Tahap ini adalah inti dari <i>prototype</i> , yang mana <i>prototype</i> dari sistem akan dibuat pada tahap ini.
4	<i>Deployment Delivery and Feedback</i>	Ketika prototipe telah dibangun, hasilnya akan diberikan ke <i>client</i> dan akan dievaluasi. Jika rancangan sistem belum sesuai, maka akan terjalin lagi komunikasi dan dilanjutkan ke tahap selanjutnya (tahapan Kembali berulang dari tahap 1 hingga 4). Dan jika telah sesuai, maka <i>prototype</i> akan di konstruksi/dibangun

Penulis *paper* tersebut merekomendasikan metode *prototype* pada kasus ini, karena menurut penulis *paper* tersebut, metode ini menghemat waktu *develope*, lalu antara *develope* dengan *client* memiliki komunikasi yang baik, menghasilkan syarat yang baik daripada metode lain. Pengguna bisa melakukan pengeditan ketika dalam *prototype*,dan memberi hasil akhir yang akurat.

Pengembangan *E-Trace*

Kasus ini dijabarkan oleh Usman Ependi didalam *paper*-nya yang membahas tentang mengembangkan *E-Trace* dengan menerapkan metode *Agile* telah di publkasikan di Seminar Nasional Informatika (SEMNASIF). Vol. 1. No. 4 pada 2015. Tujuan dari pembuatan aplikasi ini adalah meningkatkan efektifitas pendataan dan pendistribisian data alumni serta memudahkan pencarian alumni [30]. Pada kasus ini, penulis *paper* menggunakan metode *Agile* dalam membangun sistem aplikasi ini.



Gambar 6. Metode Agile

Pada Gambar 6, dapat dilihat ada 8 tahapan dalam metode *Agile* yang dijelaskan pada Tabel 8.

Tabel 8. Penjelasan tahapan metode *Agile*

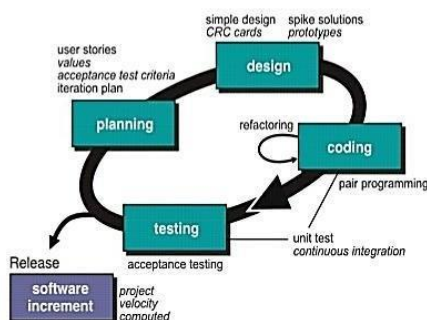
No	Tahapan	Penjelasan
1	<i>Planning</i>	Tahap ini merupakan tahap dimana <i>develope</i> dan <i>client</i> menyusun perencanaan dari sistem yang akan dibangun
2	<i>Implementation</i>	Setelah komunikasi telah terjalin, dan gambaran umum sistem telah diketahui, maka pada tahap ini berfokus kepada perancangan sistem dimulai dari desain dan membangun sistem
3	<i>Testing</i>	Pada tahap ini, sistem yang telah dibuat akan diuji oleh pelanggan dan pengembang untuk menemukan <i>bug</i> pada aplikasi yang dibuat.
4	<i>Documentation</i>	Setelah di uji coba, maka sistem akan di dokumentasi untuk memudahkan proses <i>maintenance</i> nantinya
5	<i>Deployment</i>	Pada tahap ini, kualitas sistem akan diuji apakah telah memenuhi persyaratan
6	<i>Maintenance</i>	Pada tahap ini, sistem akan dilakukan pemeliharaan secara berkala agar sistem bebas dari <i>bug</i>

Penulis *paper* tersebut merekomendasikan metode *Agile* pada kasus ini, karena menurut penulis *paper* tersebut metode ini memiliki prioritas untuk memuaskan *client*, dapat menerima perubahan, *client* dan *develope* bekerja sama dalam membangun sistem, dan pengujian dilakukan tiap saat agar mendapatkan hasil yang maksimal.

Pembangunan Web Aplikasi

Kasus ini dijabarkan oleh Adi Supriyatna didalam *paper*-nya yang membahas tentang pembangunan WEB seleksi peserta pelatihan kerja dengan menerapkan metode *Extreme* yang di publikasikan di Jurnal Teknik Informatika 11.1 pada tahun 2018. Tujuan dari pembuatan aplikasi ini adalah mempermudah masyarakat dalam pendaftaran dan pelaksanaan tes seleksi peserta diklat [43]. Pada kasus ini, penulis *paper* menggunakan metode *Extreme Programming* dalam membangun sistem aplikasi ini.

Extreme Programming (XP)



Gambar 7. Metode *Extreme Programming*

Pada Gambar 7, dapat dilihat ada 5 tahapan dalam metode *Extreme Programming* yang dijelaskan pada Tabel 9.

Tabel 9. Penjelasan tahapan metode *Extreme Programming*

No	Tahapan	Penjelasan
1	<i>Planning</i>	Tahap ini merupakan tahap dimana <i>develope</i> dan <i>client</i> menyusun perencanaan dari sebuah aplikasi yang akan dibuat
2	<i>Design</i>	Ketika komunikasi telah terjalin, dan gambaran umum sistem telah diketahui, maka pada tahap ini berfokus kepada perancangan sistem
3	<i>Coding</i>	Pada tahap ini, sistem akan dibangun menjadi sebuah <i>prototype</i> sistem dan dibangun menggunakan bahasa pemrograman serta <i>Database</i>
4	<i>Testing</i>	Setelah di bangun, maka sistem akan diuji coba untuk mengetahui kekurangan atau <i>bug</i> yang masih ada pada sistem
5	<i>Software Increment</i>	Pada tahap ini, sistem yang dibuat dikembangkan secara bertahap

Penulis *paper* tersebut merekomendasikan metode *Extreme Programming* pada kasus ini, karena menurut penulis *paper* tersebut metode ini menggunakan pendekatan berorientasi objek, tim yang dibentuk dapat berupa tim kecil saja, dan sesuai jika berhadapan dengan perubahan *requirement* yang sangat cepat.

Findings

Bagian ini memuat temuan rekomendasi metode-metode SDLC yang diterapkan pada ketiga contoh kasus yang telah dijelaskan sebelumnya. Tabel 10 adalah ringkasan hasil temuan tersebut.

Tabel 10. Rekomendasi metode *Extreme Programming* pada berbagai kasus

No	Kasus	Metode
1	Perancangan Aplikasi Tutorial	<i>Prototype</i>
2	Pengembangan E-Trace	<i>Agile</i>
3	Pembangunan Web Aplikasi	<i>Extreme Programming</i>

Pada tabel diatas, dapat disimpulkan bahwa pada kasus perancangan aplikasi tutorial, metode yang direkomendasikan adalah metode *Prototype*. Pada kasus pengembangan E-Trace, metode yang direkomendasikan adalah metode *Agile*. Dan pada kasus pembangunan web aplikasi, metode yang direkomendasikan adalah metode *Extreme Programming*. Tiga kasus tersebut telah dijelaskan sebelumnya dan digunakan untuk contoh atau acuan direfleksikan dengan kasus SDLC lainnya.

KESIMPULAN

SDLC (*Software Development Life Cycle*) atau bisa juga disebut sebagai siklus dalam sebuah pengembangan *software* dalam rekayasa perangkat lunak merupakan tahapan pembentukan serta modifikasi *software*, model, dan juga metode yang diterapkan dalam pengembangan perangkat lunak. Masalah yang dirumuskan dalam *paper* survei ini terkait dengan keefektifan *Software Development Life Cycle* dalam menyelesaikan masalah dan apa sajakah metode dalam pengembangan SDLC, tentang perbedaan dari setiap metode pada SDLC yang lebih efisien dalam menyelesaikan masalah serta tentang tahapan pengembangan suatu aplikasi menggunakan *Software Development Life Cycle*. Metode *Systematic Literature Review* digunakan pada survei ini terhadap jurnal-jurnal yang telah ditemukan dengan total 52 *paper* yang membahas tentang metode-metode SDLC dalam penelitian tahun 1991-2022. Survei menjelaskan terdapat 10 metode SDLC yaitu *Waterfall*, *Prototype*, *Rapid Application Development*, *Evolutionary*, *Agile*, *Rational Unified Process*, *Build and Fix*, *Extreme Programming*, *V Shape Model*, dan *Fountain*. Analisis terhadap 10 metode SDLC memberikan hasil bahwa metode yang paling banyak digunakan diantara tahun 1992-2022 adalah metode *Prototype*, *Agile*, dan *Extreme Programming*. Lalu, analisis juga memberikan hasil penerapan metode SDLC dalam contoh tiga kasus dengan memberikan rekomendasi metode SDLC yang cocok digunakan.

- Pada kasus perancangan aplikasi tutorial, metode yang direkomendasikan adalah metode *Prototype*.
- Pada kasus pengembangan E-Trace, metode yang direkomendasikan adalah metode *Agile*.
- Pada kasus pembangunan web aplikasi, metode yang direkomendasikan adalah metode *Extreme Programming*.

REFERENSI

- [1] W. H. DeLone, E. R. McLean, W. S. Indonesia, H. Umar, and P. T. G. P. Utama, "Azhar, Susanto.(2004). Sistem Informasi Manajemen. Bandung: Linggajaya. Alter, Steven.(2002). Information System the Pondation of Business. Pearson Eduacation Inc. Akbar, Mohammad Muzahid dan Noorjahan Parvez.(2009). Impact Of Service Quality, Trust And Customer Satisfaction on Customer Loyalty. ABAC Journal Vol. 29," *Inf. Syst.*, vol. 19, no. 4, pp. 9–30, 2003.
- [2] T. Pricillia and others, "Perbandingan Metode Pengembangan Perangkat Lunak (Waterfall, Prototype, RAD)," *J. Bangkit Indones.*, vol. 10, no. 1, pp. 6–12, 2021.
- [3] I. A. Ajah and J. O. Ugah, "Comparative Analysis of Software Development Methodologies. Volume 3, Issue 6, June." 2013.
- [4] R. S. Wahono, "A systematic literature review of software defect prediction," *J. Softw. Eng.*, vol. 1, no. 1, pp. 1–16, 2015.
- [5] B. P. Lientz, E. B. Swanson, and G. E. Tompkins, "Characteristics of application software maintenance," *Commun. ACM*, vol. 21, no. 6, pp. 466–471, 1978.
- [6] A. S. Rosa, "Rekayasa perangkat lunak terstruktur dan berorientasi objek," 2016.
- [7] C. Trisianto, "Penggunaan metode waterfall untuk pengembangan sistem monitoring dan evaluasi pembangunan pedesaan," *J. ESIT (E-Bisnis, Sist. Informasi, Teknol. Informasi)*, vol. 12, no. 1, 2022.
- [8] N. M. Cahyani, E. Indriyanto, and S. Masripah, "Uji Validitas dan Reabilitas Terhadap Implementasi Aplikasi Penjualan dan Pembelian," *Inf. Syst. Educ. Prof. J. Inf. Syst.*, vol. 1, no. 1, pp. 21–34, 2016.
- [9] A. Abdurrahman and S. Masripah, "Metode Waterfall Untuk Sistem Informasi Penjualan," *Inf. Syst. Educ. Prof. J. Inf. Syst.*, vol. 2, no. 1, pp. 95–104, 2017.
- [10] R. A. Pascapraharastyan, A. Supriyanto, and P. Sudarmaningtyas, "Rancang Bangun Sistem Informasi Manajemen Arsip Berbasis Web Pada Rumah Sakit Bedah Surabaya," Universitas Dinamika, 2014.
- [11] I. Sommerville, "SOFTWARE ENGINEERING Ninth Editon ed," *Marcia Horton. United States Am. Addison-Wesley*, 2011.
- [12] S. Nurajizah, "Sistem Informasi Perpustakaan Berbasis Web Dengan Metode Prototype: Studi Kasus Sekolah Islam Gema Nurani Bekasi," *SNIT 2015*, vol. 1, no. 1, pp. 213–218, 2015.
- [13] H. N. Syaddad, "Rancang Bangun Digital Archiving Di Perguruan Tinggi Menggunakan Metode

- Prototype Model Studi Kasus: Universitas Suryakencana,” *Media J. Inform.*, vol. 9, no. 1, 2018.
- [14] D. A. Punkastyo, “Perancangan Aplikasi Tutorial Jurusan Dasar Beladiri Cimande Menggunakan Metode Prototype,” *J. Inform. Univ. Pamulang*, vol. 3, no. 2, pp. 87–93, 2018.
- [15] R. S. Pressman, “Software Engineering: A practitioners Ap-proach. McGrawHill,” *New York*, vol. 68, 2010.
- [16] F. Mubarak, H. Harliana, and I. Hadijah, “Perbandingan Antara Metode RUP dan Prototype Dalam Aplikasi Penerimaan Siswa Baru Berbasis Web,” *Creat. Inf. Technol. J.*, vol. 2, no. 2, pp. 114–127, 2015.
- [17] R. S. Pressman and others, “Rekayasa perangkat lunak pendekatan praktisi,” *Yogyakarta Andi*, 2002.
- [18] K. E. Kendall and J. E. Kendall, “Analisis dan Perancangan sistem,” 2018.
- [19] A. Noertjahyana, “Studi Analisis Rapid Application Development Sebagai Salah Satu Alternatif Metode Pengembangan Perangkat Lunak,” *J. Inform.*, vol. 3, no. 2, pp. 64–68, 2002.
- [20] D. Gustina and Y. I. Chandra, “Aplikasi Sistem Pakar Untuk Mendiagnosa Penyakit Paru Pada Anak Menggunakan Metode Rapid Application Development (RAD),” *Pros. Semnastek*, 2015.
- [21] W. W. Widiyanto, “Analisa Metodologi Pengembangan Sistem Dengan Perbandingan Model Perangkat Lunak Sistem Informasi Kepegawaian Menggunakan Waterfall Development Model, Model Prototype, Dan Model Rapid Application Development (Rad),” *J. Inf. J. Penelit. Dan Pengabd. Masy.*, vol. 4, no. 1, pp. 34–40, 2018.
- [22] U. Yanch and L. Wiecheteck, “Task assignment optimization with the use of PESBAT linear programming tool,” *Ann. Univ. Mariae Curie-Sklodowska. Sect. H. Oeconomia*, vol. 52, no. 2, pp. 185–198, 2018.
- [23] G. Lenz and T. Moeller, *NET: a complete development cycle*. Addison-Wesley Professional, 2004.
- [24] S. McConnell, *Rapid development: taming wild software schedules*. Pearson Education, 1996.
- [25] O. Irnawati and I. Darwati, “EVOLUTIONARY PROTOTYPE DALAM PERANCANGAN SISTEM INFORMASI GUNA MENINGKATKAN PENJUALAN,” *JUSIM (Jurnal Sist. Inf. Musirawas)*, vol. 6, no. 1, pp. 1–8, 2021.
- [26] M. Stoica, B. Ghilic-Micu, M. Mircea, and C. Uscatu, “Analyzing agile development-from waterfall style to Scrumban,” *Inform. Econ.*, vol. 20, no. 4, 2016.
- [27] M. Alqudah and R. Razali, “An empirical study of Scrumban formation based on the selection of scrum and Kanban practices,” *Int. J. Adv. Sci. Eng. Inf. Technol*, vol. 8, no. 6, pp. 2315–2322, 2018.
- [28] A. J. Paul and S. K. Rahman, “Study on Agile Management in Construction Project Using Scrumban Methodology,” *Int. Res. J. Eng. Technol*, vol. 774, pp. 774–777, 2018.
- [29] N. Lutfiani, E. P. Harahap, Q. Aini, A. Ahmad, and U. Rahardja, “Inovasi Manajemen Proyek I-Learning Menggunakan Metode Agile Scrumban,” *InfoTekJar J. Nas. Inform. Dan Teknol. Jar.*, vol. 5, no. 1, pp. 96–101, 2020.
- [30] U. Ependi, “Pengembangan E-Trace Alumni Dengan Menggunakan Pendekatan Metode Agile,” in *Seminar Nasional Informatika (SEMNASIF)*, 2015, vol. 1, no. 4.
- [31] P. Utomo, S. Setiawan, and F. W. Prayitno, “Perancangan Dashboard Sistem Informasi Untuk Agile Manajemen Proyek dengan Menggunakan JIRA--Studi Kasus di PT. FLASHiZ Indonesia,” *J. Sisfotek Glob.*, vol. 5, no. 2, 2015.
- [32] Z. R. Fathiah, H. Nukman, and J. A. N. D. Tibang, “PERBANDINGAN METODOLOGY KLASIK DAN AGILE DALAM PENGEMBANGAN SISTEM INFORMASI.”
- [33] N. M. S. Iswari, “Tinjauan Proses Elisitasi Kebutuhan Perangkat Lunak Menggunakan Metode Agile,” *Ultim. J. Tek. Inform.*, vol. 4, no. 1, pp. 33–36, 2012.
- [34] P. Jalote, *A concise introduction to software engineering*. Springer Science & Business Media, 2008.
- [35] I. F. MR, “IMPLEMENTASI METODE RATIONAL UNIFIED PROCESS PADA PERANCANGAN APLIKASI PEMBAGIAN HARTA WARISAN BERBASIS ANDROID,” *Insa. Pembang. Sist. Inf. dan Komput.*, vol. 5, no. 2, 2018.
- [36] A. D. Kinasih, F. Amalia, and B. Priyambadha, “Pengembangan Sistem Evaluasi Pembelajaran PAUD (Studi Kasus di PAUD Seruni 05 Kota Malang),” *J. Pengemb. Teknol. Inf. dan Ilmu Komput. e-ISSN*, vol. 2548, p. 964x, 2018.
- [37] A. Suhartanto and R. Pamungkas, “Implementasi Sistem Informasi Dalam Perbaikan Kualitas Laporan Keuangan Pada CV. Tri Agri,” *Res. Comput. Inf. Syst. Technol. Manag*, vol. 1, no. 02, pp. 46–49, 2018.
- [38] H. Dhani, “Comparative Study and Analysis of Software Process Models on Various Merits,” *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 6, no. 9, pp. 234–243, 2016.
- [39] A. Budiman and P. Utomo, “Pengembangan Website Informasi Hama Wereng,” *J. PILAR Teknol. J. Penelit. Ilmu-Ilmu Tek.*, vol. 3, no. 2, 2018.
- [40] P. Utomo and D. N. Amadi, “Pemilihan Tanaman Pangan Untuk Ketahanan Pangan Masa Pandemi Dengan Simple Additive Weighting Di Kabupaten Madiun,” *Fountain Informatics J.*, vol. 5, no. 3, pp. 12–16, 2020.
- [41] I. J. Dewanto, “System Development Life Cycle Dengan Beberapa Pendekatan,” *J. Fasilkom*, vol. 2, no. 1, pp. 39–47, 2004.

- [42] U. Ependi and Q. Widayati, "Extreme Programming Study Method Case Study on Designing of Accounting Term Dictionary," 2014.
- [43] A. Supriyatna and M. Informatika, "Metode Extreme Programming pada pembangunan WEB aplikasi seleksi peserta pelatihan kerja," *J. Tek. Inform.*, vol. 11, no. 1, pp. 1–18, 2018.
- [44] B. O. Lubis, "Penerapan Global Extreme Programming Pada Sistem Informasi Workshop, Seminar Dan Pelatihan Di Lembaga Edukasi," *J. Inform.*, vol. 3, no. 2, 2016.
- [45] F. N. Azizah, N. Ambarsari, and R. Hanafi, "Membangun Web Service Pada Sistem Informasi Perizinan Angkutan Umum Dengan Metode Extreme Programming (studi Kasus Dinas Perhubungan Kabupaten Bandung Barat)," *eProceedings Eng.*, vol. 2, no. 2, 2015.
- [46] A. Fatoni and D. Dwi, "Rancang bangun sistem extreme programming sebagai metodologi pengembangan sistem," *PROSISKO J. Pengemb. Ris. dan Obs. Sist. Komput.*, vol. 3, no. 1, 2016.
- [47] A. D. Aipassa, I. Darmawan, and R. Andreswari, "Pembuatan Dashboard Pada Sistem Informasi Pengadaan Barang dan Jasa dengan Metode Extreme Programming (Studi Kasus Logistik Universitas Telkom)," *eProceedings Eng.*, vol. 4, no. 1, 2017.
- [48] K. Forsberg and H. Mooz, "The relationship of system engineering to the project cycle," in *INCOSE international symposium*, 1991, vol. 1, no. 1, pp. 57–65.
- [49] M. Bolung and H. R. K. Tampangela, "Analisa penggunaan metodologi pengembangan perangkat lunak," *J. ELTIKOM J. Tek. Elektro, Teknol. Inf. dan Komput.*, vol. 1, no. 1, pp. 1–10, 2017.
- [50] D. S. Budi and H. Abijono, "Analisis Pemilihan Penerapan Proyek Metodologi Pengembangan Rekayasa Perangkat Lunak," *Teknika*, vol. 5, no. 1, pp. 24–31, 2016.
- [51] N. PUTRI and F. KURNIA, "Sistem Informasi Cuti Tahunan Pegawai Dinas Sosial Provinsi Riau," *J. Ilm. Rekayasa dan Manaj. Sist. Inf.*, vol. 6, no. 2, pp. 189–191, 2020.
- [52] S. Ernawati, R. Wati, and I. Maulana, "Penerapan Model Fountain Untuk Pengembangan Aplikasi Text Recognition dan Text To Speech Berbasis Android Menggunakan Flutter," *Pros. SNAST*, pp. 178–186, 2021.