

ANALISIS METODE *SYSTEM DEVELOPMENT LIFE CYCLE* (SDLC) DALAM RANCANGAN SISTEM INFORMASI

¹Mugiatno Sumbodo, ²Mustofa

¹Fakultas Informatika, Universitas Persada Indonesia Y.A.I, Jakarta

²Fakultas Ilmu Komunikasi, Universitas Persada Indonesia Y.A.I, Jakarta

Email : mugiatno.sumbodo@upi-yai.ac.id , mustofa@upi-yai.ac.id

Abstrak

Dalam penelitian ini untuk menganalisis *System Development Life Cycle* (SDLC) dalam rancangan suatu system infoemasi. Dalam *System Development Life Cyle* (SDLC) tahapan-tahapan metode pengembangan software dimulai dari fase perencanaan, analisis, perancangan, implementasi hingga pemeliharaan sistem. Dalam penelitian ini untuk menganalisis sistem informasi menggunakan Metode waterfall adalah salah satu model *System Development Life Cycle* (SDLC) dalam pengembangan sistem informasi atau perangkat lunak. Model *waterfall* menggunakan pendekatan sistematis dan berurutan mulai dari tahapan requirement, design, implementation, verification, dan maintenance. Kelebihan menggunakan metode waterfall untuk pengembangan sistem informasi adalah kualitas dari sistem yang dihasilkan akan baik karena pelaksanaannya dilakukan secara bertahap, sementara untuk kekurangannya adalah proses pengembangan sistem membutuhkan waktu yang lama sehingga biaya yang diperlukan juga mahal.

Kata Kunci : SDLC, *waterfall*, *sistem informasi*

1. Pendahuluan
Perkembangan teknologi informasi yang semakin pesat telah mendorong berbagai organisasi dan institusi untuk memanfaatkan sistem informasi dalam mendukung kegiatan operasionalnya. Pengembangan sistem informasi yang baik tidak hanya memerlukan teknologi yang tepat, tetapi juga metode yang terstruktur agar proses perancangan hingga implementasi dapat berjalan secara sistematis dan menghasilkan sistem yang berkualitas. Oleh karena itu, diperlukan suatu pendekatan pengembangan perangkat lunak yang mampu memberikan tahapan kerja yang jelas, terarah, serta terdokumentasi dengan baik.

Salah satu pendekatan yang digunakan dalam pengembangan sistem adalah System Development Life Cycle (SDLC). SDLC merupakan suatu kerangka kerja atau metodologi yang menggambarkan tahapan-tahapan dalam proses pembangunan sistem, mulai dari identifikasi kebutuhan hingga tahap pemeliharaan sistem. Konsep SDLC mulai berkembang pada era 1960-an hingga 1970-an seiring meningkatnya kebutuhan akan pengelolaan proyek perangkat lunak yang lebih terstruktur.

Dalam penerapannya, SDLC memiliki beberapa model atau pendekatan, salah satunya adalah *model Waterfall*. Model Waterfall pertama kali diperkenalkan secara formal oleh Winston W. Royce pada tahun 1970 dalam artikelnya yang berjudul *Managing the Development of Large Software Systems*. Model ini menggambarkan proses pengembangan perangkat lunak yang dilakukan secara berurutan dan sistematis, menyerupai aliran air terjun (*waterfall*), di mana setiap tahapan harus diselesaikan terlebih

dahulu sebelum melanjutkan ke tahap berikutnya.

Pendekatan Waterfall menekankan pentingnya dokumentasi yang lengkap pada setiap tahapan, sehingga memudahkan proses evaluasi dan pengendalian proyek. Model ini sangat sesuai digunakan pada proyek dengan kebutuhan yang telah terdefinisi secara jelas dan tidak banyak mengalami perubahan selama proses pengembangan.

Pemilihan metode SDLC dengan model Waterfall dalam penelitian atau pengembangan sistem ini didasarkan pada pertimbangan bahwa sistem yang akan dibangun memiliki ruang lingkup yang jelas serta kebutuhan pengguna yang dapat diidentifikasi sejak awal. Dengan pendekatan yang sistematis dan bertahap, diharapkan sistem yang dihasilkan memiliki kualitas yang baik, minim kesalahan, serta sesuai dengan tujuan yang telah ditetapkan.

Dengan demikian, penerapan SDLC model Waterfall diharapkan mampu memberikan kerangka kerja yang terstruktur dalam proses pengembangan sistem, sehingga menghasilkan solusi yang efektif,

efisien, dan dapat memenuhi kebutuhan pengguna secara optimal.

2. Landasan Teori

System Development Life Cycle (SDLC) adalah kerangka kerja sistematis yang digunakan dalam proses pengembangan sistem informasi atau perangkat lunak. SDLC memberikan tahapan terstruktur mulai dari identifikasi kebutuhan hingga sistem diimplementasikan dan dipelihara. Winston W. Royce dikenal sebagai tokoh yang mempopulerkan model Waterfall pada tahun 1970 melalui artikelnya *Managing the Development of Large Software Systems*. Model ini menjadi dasar pendekatan SDLC klasik yang bersifat linear dan berurutan. Kemudian dikembangkan oleh Barry Boehm memperkenalkan Spiral Model pada tahun 1986. Model ini menggabungkan pendekatan iteratif dengan analisis risiko, sehingga lebih fleksibel dibandingkan Waterfall. Dalam perjalanannya dikembangkan lagi oleh Roger S. Pressman, (1982), yang banyak membahas konsep SDLC secara sistematis dan menjadi

rujukan utama dalam pendidikan rekayasa perangkat lunak.

Tujuan utama SDLC adalah menghasilkan sistem yang berkualitas, sesuai kebutuhan pengguna, tepat waktu, dan efisien dari segi biaya.

Tahapan – Tahapan utama dalam *System Development Life Cycle* (SDLC)

a. **Perencanaan(Planning)**

Mengidentifikasi kebutuhan sistem, tujuan proyek, studi kelayakan, serta estimasi biaya dan waktu.

b. **Analisis Kebutuhan (Analysis)**

Mengumpulkan dan menganalisis kebutuhan pengguna untuk dirumuskan menjadi spesifikasi sistem.

c. **Perancangan (Design)**

Mendesain arsitektur sistem, basis data, antarmuka pengguna, dan struktur program.

d. **Implementasi (Coding)**

Proses pengembangan sistem dengan menerjemahkan

desain ke dalam bahasa pemrograman.

e. Pengujian (Testing)

Menguji sistem untuk memastikan tidak ada kesalahan (bug) dan sesuai dengan kebutuhan.

f. Pemeliharaan (Maintenance)

Perbaikan, peningkatan, atau penyesuaian sistem setelah digunakan.

Model *Waterfall* adalah salah satu model dalam *System Development Life Cycle (SDLC)* yang menggunakan pendekatan linier dan sistematis, di mana setiap tahap pengembangan harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya. Model ini menggambarkan proses pengembangan perangkat lunak seperti aliran air terjun (waterfall) yang mengalir dari atas ke bawah secara berurutan.

Model Waterfall pertama kali diperkenalkan oleh Winston W. Royce pada tahun 1970 melalui artikelnya yang berjudul *Managing the Development of Large Software Systems*. Meskipun dalam tulisannya

Royce sebenarnya mengkritisi pendekatan linier yang kaku, konsep tersebut kemudian dikenal luas sebagai model Waterfall dan menjadi dasar metodologi pengembangan sistem yang terstruktur pada era awal rekayasa perangkat lunak.

Kelebihan dan Kekurangan waterfall

Kelebihan:

- a. Struktur kerja jelas dan sistematis
- b. Mudah dipahami dan diterapkan
- c. Dokumentasi lengkap memudahkan pemeliharaan

Kekurangan:

- a. Sulit mengakomodasi perubahan kebutuhan
- b. Risiko kesalahan analisis awal berdampak besar
- c. Tidak cocok untuk proyek yang dinamis

Tahapan – tahapan dalam Waterfall

Tahapan Metodologi Waterfall

Berikut adalah tahapan utama dalam metodologi Waterfall:

1. Analisis Kebutuhan (Requirement Analysis)

Pada tahap ini dilakukan pengumpulan dan pendefinisian

kebutuhan sistem secara lengkap. Hasil dari tahap ini adalah dokumen spesifikasi kebutuhan perangkat lunak (*Software Requirement Specification/SRS*).

2. Perancangan Sistem (System Design)

Tahap ini menerjemahkan kebutuhan menjadi desain teknis, meliputi:

- Desain arsitektur sistem
- Desain basis data
- Desain antarmuka pengguna
- Desain struktur program

Dokumen desain menjadi pedoman dalam tahap implementasi.

3. Implementasi (Coding)

Pada tahap ini, desain sistem diwujudkan ke dalam kode program menggunakan bahasa pemrograman tertentu. Setiap modul dikembangkan sesuai spesifikasi yang telah ditentukan.

4. Pengujian (Testing)

Sistem yang telah dikembangkan diuji untuk memastikan bahwa seluruh fungsi berjalan sesuai kebutuhan. Pengujian dapat meliputi:

- Unit Testing
- Integration Testing
- System Testing
- Acceptance Testing

5. Implementasi Sistem (Deployment)

Sistem yang telah lolos pengujian kemudian diimplementasikan dan mulai digunakan oleh pengguna.

6. Pemeliharaan (Maintenance)

Tahap terakhir adalah pemeliharaan sistem untuk memperbaiki kesalahan yang ditemukan setelah implementasi serta melakukan pengembangan tambahan jika diperlukan.

3. Metodologi



Metodologi Waterfall lahir pada masa awal perkembangan industri perangkat lunak, ketika proyek-proyek berskala besar membutuhkan kontrol ketat terhadap proses pengembangan. Pada periode tersebut, banyak proyek gagal karena kurangnya perencanaan, analisis kebutuhan

yang tidak lengkap, dan dokumentasi yang minim. Oleh sebab itu, dibutuhkan suatu pendekatan yang terstruktur dan terdokumentasi dengan baik.

Dalam metodologi Waterfall, proses pengembangan mengalir secara bertahap dari satu fase ke fase berikutnya, menyerupai air terjun yang mengalir dari atas ke bawah. Setiap fase memiliki output yang menjadi input bagi fase selanjutnya. Perubahan pada fase sebelumnya umumnya sulit dilakukan setelah masuk ke tahap berikutnya, sehingga perencanaan awal harus dilakukan secara matang.

4. Hasil dan Pembahasan

Dalam pembahasan dalam metode waterfall sebagai berikut :

1. Analisis Kebutuhan (Requirement Analysis)

Tahap pertama adalah mengidentifikasi dan mengumpulkan seluruh kebutuhan pengguna secara detail.

Pada fase ini dilakukan:

- Wawancara dengan pengguna

- Observasi sistem yang berjalan
- Studi dokumen
- Analisis proses bisnis

Hasil dari tahap ini adalah dokumen *Software Requirement Specification (SRS)* yang menjadi dasar seluruh proses berikutnya.

Dalam model Waterfall, kebutuhan harus jelas dan lengkap sejak awal karena perubahan di tahap selanjutnya akan sulit dilakukan.

2. Perancangan Sistem (System Design)

Setelah kebutuhan disepakati, tahap berikutnya adalah merancang sistem berdasarkan spesifikasi tersebut.

Desain mencakup:

- Arsitektur sistem
- Desain database
- Desain antarmuka pengguna (UI)
- Desain alur proses (flowchart, DFD, UML)

Output tahap ini adalah dokumen desain teknis yang akan menjadi pedoman dalam proses pengkodean.

3. Implementasi (Coding)

Pada tahap ini, desain sistem diterjemahkan ke dalam bahasa pemrograman.

Tim pengembang mulai:

- Menulis kode program
- Membuat modul-modul sistem
- Mengintegrasikan komponen

Karena desain sudah dirinci sebelumnya, proses coding mengikuti dokumen desain secara sistematis.

4. Pengujian (Testing)

Setelah sistem selesai dikembangkan, dilakukan pengujian untuk memastikan sistem berjalan sesuai kebutuhan.

Jenis pengujian meliputi:

- **Unit Testing** – Menguji setiap modul
- **Integration Testing** – Menguji integrasi antar modul
- **System Testing** – Menguji sistem secara keseluruhan
- **Acceptance Testing** –

Pengujian oleh pengguna

Jika ditemukan kesalahan, perbaikan dilakukan sebelum sistem diterapkan.

5. Implementasi Sistem (Deployment)

Setelah sistem dinyatakan layak, sistem diinstal dan digunakan oleh pengguna dalam lingkungan operasional.

Tahap ini meliputi:

- Instalasi sistem
- Konfigurasi server
- Pelatihan pengguna
- Serah terima sistem

6. Pemeliharaan (Maintenance)

Tahap terakhir adalah pemeliharaan sistem setelah digunakan.

Pemeliharaan meliputi:

- Perbaikan bug
- Penyesuaian sistem
- Pengembangan fitur tambahan
- Update keamanan

Tahap ini bisa berlangsung dalam jangka waktu lama sesuai kebutuhan organisasi.

5. Kesimpulan

Cara kerja model Waterfall dilakukan secara sistematis, bertahap, dan terdokumentasi mulai dari analisis kebutuhan hingga pemeliharaan sistem. Setiap tahap menjadi fondasi bagi

tahap berikutnya, sehingga ketelitian pada tahap awal sangat menentukan keberhasilan proyek. Meskipun saat ini banyak metode modern seperti Agile yang lebih fleksibel, Waterfall tetap relevan untuk proyek dengan kebutuhan yang jelas dan ruang lingkup yang stabil.

Daftar Pustaka

Barry W. Boehm. (1986). A spiral model of software development and enhancement. *ACM SIGSOFT Software Engineering Notes*, 11(4), 14–24.

Kenneth E. Kendall., & Julie E. Kendall. (2011). *Systems Analysis and Design* (8th ed.). Pearson.

Roger S. Pressman. (1982). *Software Engineering: A Practitioner's Approach*. McGraw-Hill.

Winston W. Royce. (1970). *Managing the development of large software systems*. Proceedings of IEEE WESCON.

Ian Sommerville. (2011). *Software Engineering* (9th ed.). Pearson.