

## PERAN METODOLOGI SIKLUS HIDUP SISTEM DALAM MANAJEMEN PROYEK TEKNOLOGI INFORMASI

NYI RD NINA WANTINA<sup>1</sup>, M. YUNUS YUSPIANI<sup>2</sup>, NABILLAH AGIE AZZAHRA<sup>3</sup>, HADI SUPRATIKTA<sup>4</sup>

<sup>1,2</sup>*Magister Manajemen, Fakultas Ekonomi, Universitas Pamulang  
Jl. Raya Puspitek, Buaran, Kec. Pamulang, Kota Tangerang Selatan, Banten 15310  
\*Email: ninawantina@gmail.com*

### ABSTRAK

Penerapan metodologi Siklus Hidup Sistem (*System Development Life Cycle/SDLC*) menjadi landasan strategis dalam pengembangan sistem informasi yang efektif dan efisien di lingkungan organisasi modern. Penelitian ini bertujuan untuk mengkaji bagaimana tahapan dalam *SDLC* diimplementasikan dalam proyek pengembangan teknologi informasi, serta tantangan yang dihadapi dalam tiap fasenya. Dengan menggunakan pendekatan kualitatif studi kasus di salah satu institusi pendidikan tinggi swasta di Tangerang, data dikumpulkan melalui wawancara, observasi, dan analisis dokumen proyek. Hasil menunjukkan bahwa tahapan *SDLC* yang dilalui — mulai dari perencanaan hingga pemeliharaan — mampu meningkatkan akurasi perancangan sistem, mengurangi risiko kegagalan proyek, dan mempercepat siklus implementasi. Namun demikian, tantangan muncul dalam hal alokasi waktu dan koordinasi antar pemangku kepentingan. Penelitian ini menyarankan perlunya pemetaan ulang peran tim proyek serta pemanfaatan *tools* kolaboratif agar implementasi *SDLC* berjalan optimal.

Kata Kunci: *SDLC*, pengembangan sistem, manajemen proyek TI, studi kasus, perencanaan sistem.

### ABSTRACT

*The application of the System Development Life Cycle (SDLC) methodology is a strategic foundation for the effective development of information systems in modern organizations. This study aims to examine how each stage of SDLC is implemented in information technology projects and the challenges faced during each phase. Using a qualitative case study approach at a private higher education institution in Tangerang, data were collected through interviews, observations, and document analysis. The results show that the SDLC phases—from planning to maintenance—improve system design accuracy, reduce project failure risks, and accelerate implementation cycles. However, challenges remain, especially in time allocation and stakeholder coordination. The study recommends realigning project roles and utilizing collaborative tools to optimize SDLC implementation.*

*Keywords:* *SDLC*, system development, IT project management, case study, system planning.

### PENDAHULUAN

Transformasi digital menuntut organisasi untuk mengembangkan sistem informasi yang tangguh, fleksibel, dan selaras dengan tujuan bisnis. Dalam konteks ini, Siklus Hidup Sistem (*System Development Life Cycle/SDLC*) menjadi kerangka metodologis utama dalam memastikan proses perancangan dan pengembangan sistem berlangsung secara terstruktur, terkontrol, dan terdokumentasi dengan baik (Kendall & Kendall, 2020). *SDLC* tidak hanya menjadi panduan teknis, tetapi juga instrumen manajerial yang membantu tim

proyek mengidentifikasi kebutuhan pengguna, merancang solusi teknologi, serta menguji dan memelihara sistem secara berkelanjutan.

*SDLC* memiliki berbagai model implementasi, seperti *waterfall, spiral, dan agile*. Namun secara umum, setiap model mencakup tahapan inti: perencanaan, analisis sistem, perancangan sistem, implementasi, pengujian, dan pemeliharaan. Setiap fase memiliki tujuan dan *output* spesifik yang harus dicapai agar pengembangan sistem berjalan lancar. Tanpa metodologi yang jelas, proyek teknologi informasi rentan mengalami keterlambatan, pembengkakan biaya, hingga kegagalan implementasi sistem (Pressman, 2021).

Penelitian ini dilakukan dalam rangka memahami dinamika penerapan *SDLC* dalam proyek sistem informasi di institusi pendidikan tinggi, dengan fokus pada keberhasilan dan hambatan dalam tiap tahapan. Penelitian ini juga mengkaji bagaimana tim proyek beradaptasi dengan tantangan *internal* dan *eksternal* selama implementasi berlangsung.

## METODE

Penelitian ini menggunakan pendekatan kualitatif deskriptif dengan jenis studi kasus, yang dipilih untuk menggambarkan secara mendalam praktik penerapan metodologi *SDLC* di lingkungan organisasi nyata. Lokasi penelitian adalah sebuah institusi pendidikan tinggi swasta di Tangerang yang tengah mengembangkan sistem informasi akademik internal.

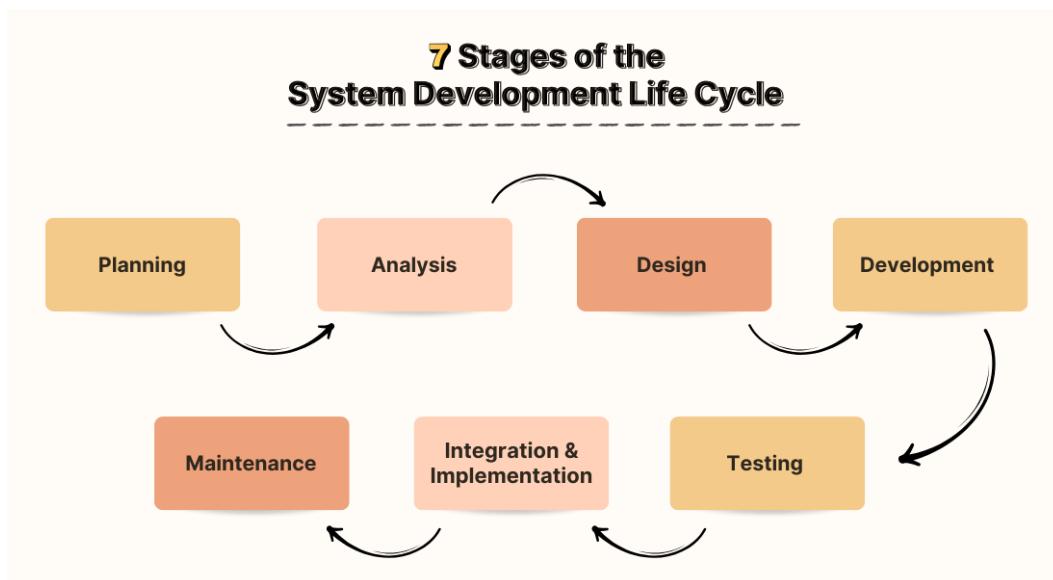
Data dikumpulkan melalui 3 (tiga) teknik utama:

1. Wawancara mendalam dengan tim pengembang (*IT*), *user* dari bagian akademik, dan manajer proyek;
2. Observasi langsung terhadap aktivitas pengembangan sistem menggunakan *tools* seperti *Trello* dan *GitHub*;
3. Studi dokumentasi terhadap *roadmap* pengembangan, laporan *bug*, dan dokumentasi kebutuhan sistem (*SRS*).

Proses analisis data menggunakan analisis tematik, dengan pengkodean berdasarkan 7 (tujuh) tahapan umum *SDLC*, yaitu:

1. *Planning*
2. *System Analysis*
3. *System Design*
4. *Development*
5. *Testing*
6. *Integration/ Implementation*
7. *Maintenance*

Validitas data diperkuat melalui triangulasi sumber dan metode.



Gambar 1. Tahapan Siklus Hidup Sistem (System Development Life Cycle/ SDLC)

## HASIL DAN PEMBAHASAN

### 1. *Planning* (Perencanaan Sistem)

Perencanaan sistem di kampus dimulai ketika pihak rektorat, biro TI (Pusat Data dan Informasi Kampus), atau fakultas menyadari adanya kebutuhan sistem baru, misalnya mahasiswa mengeluh tentang keterlambatan input nilai, dosen kesulitan mengakses data presensi dan jadwal mengajar, penerimaan mahasiswa baru masih manual dan tidak efisien. Aktivitasnya berupa *FGD* dengan pengguna sistem: dosen, staf TU, mahasiswa, ttudi kelayakan: apakah kampus punya infrastruktur server, SDM TI?, menyusun proposal sistem: siapa yang membuat? *Internal* atau *vendor*?, analisis *ROI* dan RAB (Rencana Anggaran Biaya), menyusun jadwal proyek TI, termasuk tenggat waktu untuk implementasi. Hasil akhir dapat berupa dokumen perencanaan proyek sistem (contoh: "Rencana Pengembangan SIAKAD 2025") atau persetujuan manajemen kampus dan anggaran disahkan.

### 2. *System Analysis* (Analisis Sistem)

Tahapan ini bertujuan untuk menggali kebutuhan riil pengguna sistem di lingkungan kampus. Contoh seperti sistem SIAKAD harus bisa mengakomodasi antara lain input nilai dosen, verifikasi KHS oleh Ketua Prodi, cetak transkrip oleh mahasiswa secara mandiri, monitoring IPK dan SKS. Aktivitasnya yaitu melalui wawancara dengan kepala TU, dosen, bagian akademik, dan mahasiswa, observasi proses pengisian KRS manual, identifikasi kebutuhan keamanan data (akses terbatas bagi dosen, mahasiswa, admin), penyusunan *DFD*, *ERD*, atau *Use Case* sesuai kebutuhan akademik. Hasil akhirnya yaitu dokumen Kebutuhan Sistem (*SRS*) untuk konteks perguruan tinggi dan rekomendasi untuk fitur sistem berdasarkan prioritas kebutuhan.

### 3. *System Design* (Desain Sistem)

Merancang sistem sesuai kebutuhan analisis. Dalam konteks kampus, desain sistem harus user-friendly, aksesible dari banyak perangkat, dan terintegrasi antar bagian kampus.

Contoh fitur desain seperti UI mahasiswa untuk KRS dan pengajuan cuti akademik, Panel admin akademik untuk validasi dan approval, integrasi dengan sistem keuangan (untuk blokir jika belum bayar UKT). Beberapa aktivitas seperti mendesain *dashboard* sesuai role: mahasiswa, dosen, prodi, admin, membuat prototipe atau wireframe menggunakan *tools* seperti *Figma*, mendesain basis data akademik: tabel mahasiswa, nilai, jadwal, mata kuliah, merancang keamanan sistem: login terotentikasi, enkripsi data. Hasilnya dapat berupa spesifikasi desain sistem lengkap atau dokumen desain antarmuka dan prototipe sistem untuk divalidasi.

#### 4. *Development* (Pengembangan Sistem)

Tahapan di mana programmer mulai membangun sistem berdasarkan desain. Bisa dikerjakan oleh tim *IT* internal universitas atau oleh pihak ketiga/ *vendor*. Contohnya Universitas Pamulang memiliki tim pengembang internal atau universitas swasta lainnya mungkin menggunakan jasa vendor TI. Salah satunya yaitu pemrograman modul sistem (*backend* dan *frontend*), integrasi dengan *database* kampus, pengembangan API untuk LMS atau ke sistem pembayaran digital (BRIVA, Mandiri VA, dsb), pengujian internal setiap modul (unit testing). Hasil akhir dapat berupa sistem versi awal (beta) untuk diuji coba terbatas dan dokumentasi teknis kode dan struktur basis data.

#### 5. *Testing* (Pengujian Sistem)

Sebelum sistem digunakan secara resmi, harus diuji menyeluruh. Pengujian dilakukan untuk memastikan sistem sesuai harapan dan bebas dari *error*. Contoh di kampus antara lain dosen menguji input nilai di akhir semester, mahasiswa mencoba fitur pengisian KRS, Admin memeriksa laporan rekap nilai, IPK, dan jumlah SKS. Aktivitasnya dapat berupa unit *testing* oleh programmer, *system testing* oleh tim *QA* kampus, *UAT (User Acceptance Testing)* oleh perwakilan fakultas dan biro akademik, simulasi *massal (load testing)* untuk *login* serentak ribuan mahasiswa. Hasil akhirnya yaitu *feedback* dari pengguna uji coba, perbaikan *bug* dan penyesuaian fitur, validasi bahwa sistem siap diimplementasikan penuh.

#### 6. *Integration & Implementation* (Integrasi dan Implementasi)

Tahapan peluncuran sistem ke lingkungan nyata kampus. Ini adalah masa kritis karena perubahan dari sistem lama ke sistem baru bisa menimbulkan resistensi. Salah satunya yaitu sosialisasi dan pelatihan ke dosen dan staf, konversi data: dari sistem *manual* ke sistem *digital*, uji coba sistem pada fakultas tertentu (*pilot project*), Implementasi penuh ke seluruh fakultas/ jenjang studi. Contoh seperti sistem PMB online diterapkan di gelombang 1 untuk uji coba, lalu diluncurkan penuh di gelombang 2. Sistem ini digunakan secara resmi dan pengguna mulai terbiasa menggunakan dashboard masing-masing dan dapat di evaluasi awal dari implementasi.

#### 7. *Maintenance* (Pemeliharaan Sistem)

Setelah sistem digunakan, tim *TI* kampus tetap harus melakukan perawatan dan pengembangan lanjutan. Ini termasuk memperbaiki *bug*, meningkatkan performa, dan menyesuaikan kebutuhan baru. Contohnya seperti penambahan fitur *QR code* presensi dosen, integrasi dengan sistem perpustakaan, perbaikan keamanan login mahasiswa, *update UI/UX* sesuai *feedback*. Hal ini dapat memonitoring kinerja sistem (respon lambat, down server, dsb), melayani laporan masalah dari pengguna (*ticketing system*), peningkatan sistem secara berkala (versi 1.1, 1.2, dst), *backup data* dan audit keamanan. Alhasil sistem tetap berjalan optimal dan aman, tingkat kepuasan pengguna meningkat dan sistem dapat berkembang mengikuti kebutuhan kampus.

Tabel 1. Temuan Tematik Penerapan SDLC

Tahapan SDLC	Temuan Utama	Tantangan Kritis
Perencanaan	Perumusan kebutuhan proyek	Koordinasi awal antar tim
Analisis	Dokumentasi kebutuhan <i>user</i> lengkap	Perubahan kebutuhan di tengah proyek
Perancangan	Desain UI responsif dan modular	Integrasi <i>frontend-backend</i>
Implementasi	Kode dikerjakan paralel melalui <i>GitHub</i>	Konflik kode antar pengembang
Pengujian	<i>Error</i> terdeteksi sejak <i>UAT</i> awal	Perlu revisi beberapa modul penting
Pemeliharaan	Sistem dimonitor dan dievaluasi rutin	Komitmen pembaruan berkala rendah

## KESIMPULAN

Penerapan metodologi Siklus Hidup Sistem (*SDLC*) dalam proyek sistem informasi terbukti mampu meningkatkan efektivitas dan efisiensi pengembangan sistem. Setiap tahapan memiliki peran krusial yang saling berkaitan dan berpengaruh terhadap keberhasilan proyek secara keseluruhan. Studi ini menunjukkan bahwa keberhasilan *SDLC* tidak hanya bergantung pada kemampuan teknis tim pengembang, tetapi juga pada komunikasi yang intensif, partisipasi aktif pengguna, dan dukungan kebijakan institusi.

Disarankan agar organisasi yang menerapkan *SDLC* menyusun mekanisme koordinasi lintas divisi, melakukan pelatihan penggunaan *tools* manajemen proyek, dan memastikan pemantauan jangka panjang pasca implementasi. Penelitian lanjutan diharapkan dapat membandingkan efektivitas model *SDLC* tradisional dan pendekatan *Agile* dalam konteks organisasi publik dan swasta.

## DAFTAR PUSTAKA

- Kendall, K. E., & Kendall, J. E. (2020). *Systems Analysis and Design* (10th ed.). Pearson Education.
- Pressman, R. S. (2021). *Software Engineering: A Practitioner's Approach* (9th ed.). McGraw-Hill Education.
- Dennis, A., Wixom, B. H., & Roth, R. M. (2020). *Systems Analysis and Design* (7th ed.). Wiley.
- Shelly, G. B., & Rosenblatt, H. J. (2019). *Systems Analysis and Design*. Cengage Learning.