



# BAB 7

## SDLC

### (SYSTEM DEVELOPMENT LIFE CYCLE)



# BAB 7

## SDLC (Software Development Life Cycle)

### Capaian Pembelajaran

Capaian pembelajaran yang dibebankan pada modul pelatihan ini adalah mahasiswa mampu memahami, menjelaskan dan memvisualisasikan teori SDLC (*Software Development Life Cycle*). Dari capaian pembelajaran tersebut dapat dicapai dari sub pokok bahasan.

### Pokok Bahasan

1. Konsep SDLC
2. Tahapan SDLC
3. Model SDLC
4. Alat Pengembangan Sistem Informasi

### Evaluasi Pembelajaran

Soal Latihan Teori SDLC

---

## Referensi

1. Patricia Wallace, John's Hopkins University, Introduction to Information Systems, 3e, Pearson, 2018.
2. James O' Brien, Introduction to Information Systems, 16e, McGraw-Hill, 2013.
3. R. Kelly Rainer & Brad Prince, Introduction to Information Systems: Supporting and Transforming Business, Willey, 2020.
4. Paul Bocij, Andrew Greasley, Simon Hickie, 5e, Business Information Systems, Prent, Pearson Education Limited, UK, 2015.
5. Ralph Stair, George Reynolds, Principles of Information Systems, 9e, Course Technology Cengage Learning, 2010.
6. Jogiyanto, Sistem Informasi Manajemen, 3e, Universitas Terbuka, 2019.

## Pre Test

### SDLC (Software Development Life Cycle)

1. Apa yang dimaksud dengan SDLC?
2. Proses apa saja yang ada di dalam proses SDLC?
3. Sebutkan macam-macam SDLC yang kamu ketahui?
4. Mengapa penting untuk belajar SDLC?
5. SDLC model apa yang paling sesuai untuk mengembangkan sebuah sistem informasi?

Pada bab tujuh ini akan membahas mengenai SDLC atau *Systems Development Life Cycle* yang merupakan siklus atau tahapan kerja dalam proses pengembangan sistem informasi. Tahapan pengembangan sistem informasi ini yang memiliki tujuan guna menyelesaikan masalah secara efektif dengan hasil yang berkualitas tinggi. SDLC juga berperan dalam memastikan keberhasilan pengimplementasian system dalam pemenuhan tujuan strategis.

## 7.1 Konsep SDLC

Pengertian SDLC adalah kependekan dari Systems Development Life Cycle atau disebut siklus hidup pengembangan sistem. SDLC adalah siklus yang digunakan dalam pembuatan atau pengembangan sistem informasi yang bertujuan untuk menyelesaikan masalah pengembangan sistem informasi secara efektif. Pengertian lain, SDLC adalah tahapan kerja yang bertujuan untuk menghasilkan sistem berkualitas tinggi yang sesuai dengan keinginan pelanggan atau tujuan dibuatnya sistem tersebut. SDLC menjadi kerangka yang berisi langkah-langkah yang harus dilakukan untuk memproses pengembangan suatu perangkat lunak. Dalam proses pengembangan sistem terdapat beberapa hal yang berpotensi menyebabkan tidak dapat tercapainya tujuan utama pengembangan (Jogiyanto, 2008). Sehingga diperlukan tahapan-tahapan dalam proses pengembangan sistem agar tujuan dikembangkannya sistem dapat tercapai.

Proses pengembangan sistem melewati beberapa tahapan (fase), dimulai dari perencanaan sistem sampai dengan sistem tersebut diterapkan, dioperasikan, dan dipelihara. Apabila pengoperasian sistem yang sudah dikembangkan masih timbul permasalahan-permasalahan yang kritis serta tidak dapat diatasi dalam tahap pemeliharaan sistem, maka sistem tersebut perlu ditinjau kembali untuk dikembangkan dengan mengimplementasikan kembali tahap awal, yaitu tahap perencanaan sistem. Siklus ini disebut dengan siklus hidup suatu sistem (system life cycle) yang biasa disebut dengan istilah Siklus Hidup Pengembangan Sistem (System Development Life Cycle/SDLC). SDLC ini berisi rencana lengkap untuk

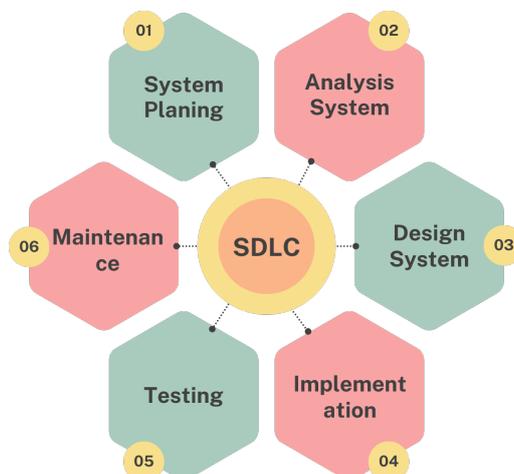
mengembangkan, memelihara, dan menggantikan perangkat lunak tertentu. SDLC menjadi kerangka yang berisi langkah-langkah yang harus dilakukan untuk memproses pengembangan suatu perangkat lunak.

SDLC juga berperan dalam memastikan keberhasilan pengimplementasian sistem dalam pemenuhan tujuan strategis. Adapun fungsi-fungsi yang dimiliki oleh konsep SDLC ini, yaitu:

- a. Sebagai media komunikasi tim develop dengan stakeholder
- b. Sebagai fungsi monitoring dan dokumentasi manajemen tingkat tinggi dalam suatu perusahaan pengembang perangkat lunak
- c. Sebagai penghasil system dengan kualitas yang dapat memenuhi harapan pengguna
- d. Sebagai gambaran input dan output dari tahap ke tahap
- e. Sebagai pembagi tanggung jawab atau peran yang jelas antara pihak yang terlibat seperti manajer proyek, analisis bisnis, developer, serta designer.

## 7.2 Tahapan SDLC

Penerapan SDLC tidak bisa sembarangan, ada beberapa tahapan yang harus dilakukan secara urut. Adapun tahapan-tahapan dalam mengimplementasikan SDLC untuk mengembangkan sebuah sistem informasi dapat dilihat pada Gambar 7.1 berikut.



**Gambar 7. 1** Tahapan SDLC

Tahapan masing-masing tahapan SDLC dari Gambar 7.1 adalah Perencanaan Sistem (*System Planing*), Analisis Sistem (*Analysis System*), Perancangan Sistem (*Design System*), Implementasi Sistem (*Implementation*), Pengujian Sistem (*Testing*), dan Pemeliharaan Sistem (*Maintenance*).

### 7.2.1 Perencanaan Sistem (*System Planing*)

Perencanaan adalah tahap yang paling penting dan mendasar dalam SDLC. Hal ini untuk merencanakan pendekatan proyek dasar dan untuk melakukan studi kelayakan produk dalam bidang ekonomi, operasional, dan teknis. Dalam perencanaan diperlukan adanya spesifikasi kebutuhan sistem, maka setiap teknisi dan juga pengembang harus mampu membuat perencanaan sistem yang sesuai dengan kebutuhan yang ada, serta mampu untuk menjalankan sistem tersebut sesuai dengan peruntukannya. Harapannya agar sistem dapat berjalan pada spesifikasi yang telah direncanakan.

Aktivitas yang dilakukan pada tahap perencanaan sistem yaitu sebagai berikut, a) Pembentukan dan konsolidasi tim pengembang, b) Mendefinisikan tujuan dan ruang lingkup pengembang, c) Mengidentifikasi masalah yang dapat diselesaikan melalui pengembangan sistem, d) Menentukan dan mengevaluasi strategi yang akan digunakan dalam pengembangan sistem dan e) Penentuan prioritas teknologi dan pemilihan aplikasi.

### 7.2.2 Analisis Sistem (*Analysis System*)

Analisis sistem dalam sebuah siklus SDLC adalah proses melakukan berbagai macam analisis terhadap sebuah sistem yang sudah ada, dan bagaimana nantinya sebuah sistem akan berjalan. Proses analisis dilakukan terhadap kelebihan dan kekurangan sistem, fungsi dari sistem, hingga berbagai macam pembaruan yang bisa saja diterapkan pada sebuah sistem. Setelah analisis sistem selesai dilakukan, maka tahapan berikutnya adalah tahapan analisis spesifikasi kebutuhan sistem.

Aktivitas yang dilakukan pada tahap analisis sistem yaitu sebagai berikut, a) Melakukan studi literatur untuk menemukan suatu kasus yang bisa ditangani

oleh sistem, b) Brainstorming dalam tim pengembang mengenai kasus mana yang paling tepat dimodelkan dengan sistem, c) Mengklasifikasikan masalah, peluang, dan solusi yang mungkin diterapkan untuk kasus tersebut, d) Analisa kebutuhan pada sistem dan membuat batasan sistem, dan e) Mendefinisikan kebutuhan sistem.

### 7.2.3 Perancangan Sistem (Design System)

Tahap perancangan sistem merupakan suatu tahap dimana seluruh hasil analisa dan juga hasil pembahasan mengenai perencanaan, spesifikasi sistem dan analisis sistem diterapkan menjadi sebuah rancangan sistem. Tahap perancangan sistem ini disebut juga sebagai cetak biru atau prototype, dimana sistem ini sudah siap untuk dikembangkan. Pada tahap ini, semua persiapan harus dilakukan dengan matang. Persiapan tersebut mulai dari perencanaan, implementasi dari spesifikasi sistem, dan semua analisis terhadap sistem, termasuk berbagai macam tenaga pendukung dari sistem yang akan dikembangkan.

Aktivitas yang dilakukan pada tahap perancangan sistem yaitu sebagai berikut, a) Menganalisa interaksi obyek dan fungsi pada sistem, b) Menganalisa data dan membuat skema database, dan c) Merancang user interface.

### 7.2.4 Implementasi Sistem (Implementation)

Tahapan implementasi sistem ini merupakan tahapan dimana rancangan sistem mulai dikerjakan, dibuat atau diimplementasikan menjadi sebuah sistem yang utuh, dan dapat digunakan. Dalam bahasa teknis proses ini sering disebut dengan coding. Coding adalah proses mewujudkan desain yang telah dibuat kedalam sistem menggunakan bahasa pemrograman yang dipahami oleh komputer. Tahap ini membutuhkan waktu yang cukup lama, karena dalam prakteknya tahap implementasi sistem ini bisa saja terdapat kendala – kendala baru yang menyebabkan proyek menjadi terhambat, sehingga dibutuhkan analisis tambahan, ataupun perancangan tambahan. Bahkan, bukan tidak

mungkin pada tahap ini terjadi perubahan perancangan sistem oleh karena satu dan lain hal.

Aktivitas yang dilakukan pada tahap implementasi sistem yaitu sebagai berikut, a) Mengimplemntasikan desain sistem kedalam bahasa pemrograman (coding), b) Membuat database pada database administrator, dan c) Menempatkan hasil codingan sistem informasi kedalam domain server.

#### 7.2.5 Pengujian Sistem (Testing)

Tahapan setelah sistem selesai diimplementasikan, sistem tersebut tidak akan langsung digunakan secara umum ataupun secara komersil. Tentu saja harus ada proses pengujian terhadap sistem yang sudah dikembangkan tersebut. Tahap pengujian sistem ini merupakan waktu yang tepat untuk mencoba apakah sistem yang sudah berhasil dikembangkan memang dapat bekerja dengan optimal dan juga sempurna. Apabila sistem yang dibuat dapat bekerja dengan baik dan sempurna, maka sistem siap untuk digunakan.

Dalam tahap ini ada banyak hal yang harus diperhitungkan, mulai dari kemudahan penggunaan sistem, hingga pencapaian tujuan dari sistem yang sudah disusun sejak perancangan sistem. Apabila terjadi kesalahan, atau sistem tidak dapat berjalan dengan baik dan sebagaimana mestinya, maka mulai dari tahap perencanaan sistem hingga tahap perancangan dan implementasi sistem harus diperbaharui secara keseluruhan. Bahkan bisa juga seluruh proses diulangi, atau mengalami perombakan total.

#### 7.2.6 Pemeliharaan Sistem (Maintenance)

Tahap ini adalah tahapan final atau tahapan akhir dari satu buah siklus SDLC. Tahapan ini merupakan tahapan dimana sebuah sistem sudah selesai dibuat, sudah diujicoba, dan dapat bekerja dengan baik (optimal). Dalam prakteknya, tahap terakhir ini tidak hanya berhenti pada proses pemeliharaan saja, namun juga melakukan proses pengembangan terhadap sistem yang ada, sehingga dapat menjamin bahwa sistem tersebut akan tetap berfungsi secara normal dan optimal sesuai dengan perkembangan kebutuhan.

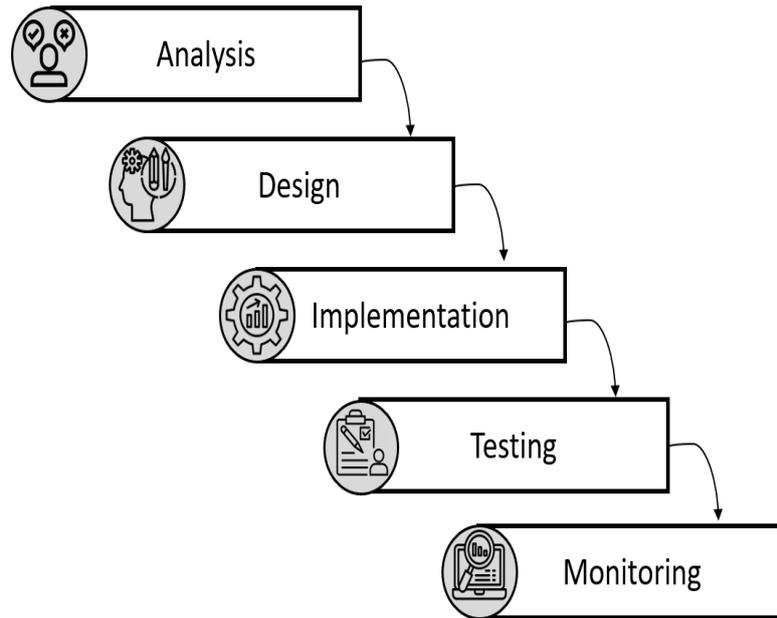
Pemeliharaan sistem dilakukan oleh admin agar sistem informasi tetap mampu beroperasi dan kemampuan sistem dalam mengadaptasikan diri sesuai dengan kebutuhan. Aktivitas yang dilakukan pada tahap pemeliharaan sistem yaitu perbaikan aplikasi, melakukan pengembangan fitur sesuai dengan kebutuhan mendatang dan melakukan upgrade terhadap teknologi yang digunakan oleh sistem sesuai dengan perkembangan teknologi yang ada.

### 7.3 Model SDLC

Model SDLC (*Software Development Life Cycle*) adalah proses pembuatan dan penyusunan tahapan yang digunakan untuk mengembangkan sistem rekayasa perangkat lunak. Model SDLC hadir untuk membantu kamu dalam pengembangan produk. Model ini memiliki banyak jenisnya, tetapi pada modul ini hanya akan fokus membahas 6 model saja. Model tersebut adalah waterfall model, v-model, spiral model, RAD model, agile model dan JAD model.

#### 7.3.1 Waterfall Model

Waterfall model merupakan salah satu metodologi pengembangan software yang paling populer dan paling banyak digunakan karena sangat sederhana. Dalam model ini setiap fase diselesaikan sebelum masuk ke fase berikutnya. Tidak ada pilihan untuk kembali setelah pindah ke fase berikutnya. Pada model waterfall tahap selanjutnya ketergantungan pada hasil fase sebelumnya. Model pengembangan software ini sekuensial, di mana kemajuan terlihat mengalir terus ke bawah seperti air terjun dari mulai fase analisis, desain, implementasi, pengujian, serta monitoring (Auroral 2021). Setiap fasenya digambarkan seperti waterfall atau air terjun yang dapat digambarkan seperti Gambar 7.2 berikut.



**Gambar 7. 2** Waterfall Model

Keuntungan menggunakan *waterfall* model:

- a. Model air terjun sangat sederhana dan mudah dipahami serta prosesnya sistematis,
- b. Ketika kebutuhan sudah jelas maka model ini sangat cocok digunakan sehingga bisa meminimalisir kesalahan,
- c. Pada *waterfall* model ini, terdapat dokumen teknis yang baik dan terorganisir dalam setiap fase, sehingga memudahkan klien untuk mengetahui gambaran oleh software ini,
- d. Pengembangan *software* dengan metode ini biasanya menghasilkan kualitas yang baik,
- e. *Waterfall* model ini cukup menghemat waktu karena setiap fase yang diproses dapat diselesaikan pada waktu tertentu sesuai target yang telah ditetapkan.

Kekurangan menggunakan *waterfall* model:

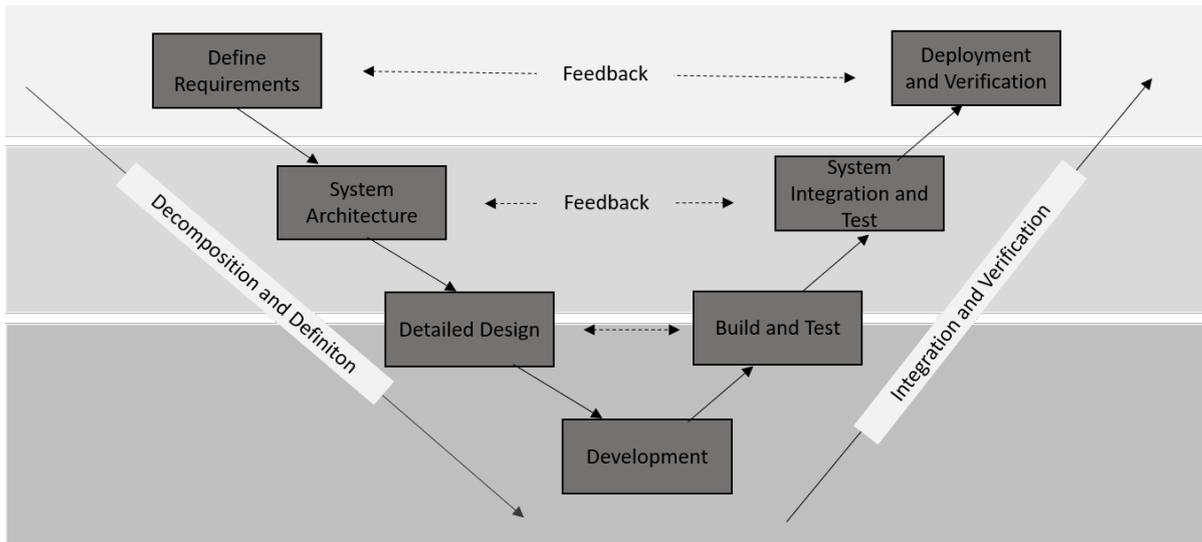
- a. Jika kebutuhan sudah tepat dan sudah tersedia di awal, maka hanya *waterfall* model ini yang dapat digunakan,

- b. *Waterfall* model ini tidak berlaku untuk proyek yang menuntut pemeliharaan berkelanjutan,
- c. Setelah aplikasi dalam tahap pengujian, tidak disarankan untuk kembali dan melakukan perubahan apa pun untuk *software* yang telah selesai, karena hal ini dapat menyebabkan banyak masalah,
- d. Tidak dapat memasukkan *feedback* atau perubahan yang diusulkan oleh klien dalam fase pengembangan yang sedang berlangsung,
- e. Adanya waktu kosong bagi pengembang, karena harus menunggu anggota tim proyek lainnya menuntaskan pekerjaannya di tahap sebelumnya atau sesudahnya.

### 7.3.2 V Model

Model V adalah model lanjutan dan perluasan dari model *waterfall*, dimana pengujian fungsionalitas ditambahkan pada setiap tahap pengembangan proyek alih-alih proyek penyelesaian proyek yang mengarah pada pengembangan proyek yang lebih baik. Pelaksanaan proses ini terjadi secara sistematis dan berurutan dalam bentuk V. Dalam model ini tidak menyimpang dari tujuan proyek karena setiap fase akan mengalami pengujian, maka dari itu model ini dikenal juga sebagai model validasi dan verifikasi.

Dalam model V ini terdapat yang namanya verifikasi dan validasi. Verifikasi melibatkan metode statis yang dilakukan tanpa melakukan eksekusi kode, ini merupakan proses evaluasi pengembangan proyek untuk menemukan apakah kebutuhan yang telah ditentukan dapat terpenuhi. Validasi melibatkan metode analisis dinamis (fungsional dan non fungsional), pengujian dilakukan dengan melakukan eksekusi kode, ini merupakan pengembangan software untuk menentukan apakah software memenuhi harapan dan kebutuhan pelanggan. Penggambaran v model dapat dilihat pada Gambar 7.3 berikut ini.



**Gambar 7. 3** V-Model

Keuntungan menggunakan V-Model:

- Model ini cukup sederhana sehingga mudah dipahami dan digunakan,
- Model sangat sistematis dan disiplin karena setiap tahapan diselesaikan satu persatu dan juga terdapat pengujian,
- Lebih mudah dikelola karena setiap tahapnya memiliki spesifikasi kiriman dan proses review,
- Tingkat keberhasilan lebih tinggi karena setiap tahap dilakukan pengujian.

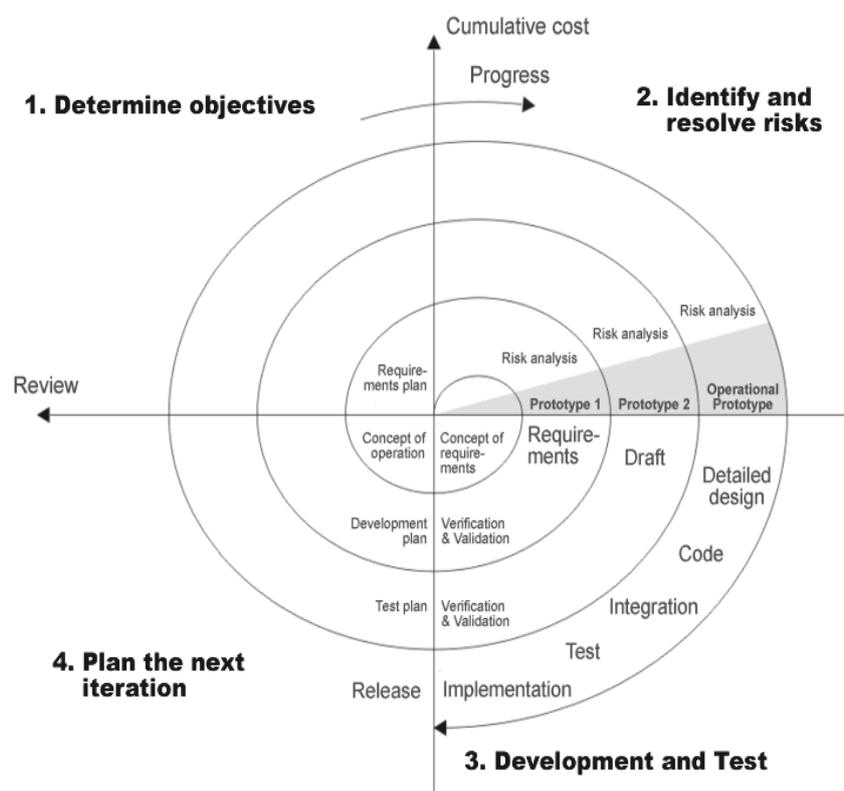
Kekurangan menggunakan V-Model:

- Setelah software dalam tahap pengujian, maka akan sulit jika ada perubahan fungsionalitas,
- Tidak cocok jika diterapkan terhadap proyek yang kompleks, kebutuhan tinggi, dan yang sifatnya berkelanjutan,
- Model ini kurang fleksibel dan cenderung kaku.

### 7.3.3 Spiral Model

Model spiral adalah salah satu model SDLC yang paling penting, model spiral menyediakan dukungan dalam penanganan risiko. Dalam representasi diagram, terlihat seperti spiral dengan banyak perulangan. Jumlah pasti putaran spiral tidak diketahui dan dapat bervariasi dari satu proyek ke proyek lainnya.

Setiap perulangan spiral disebut fase dari proses pengembangan software. Jumlah pasti fase yang dibutuhkan untuk mengembangkan produk dapat divariasikan oleh manajer proyek tergantung pada risiko proyek. Karena manajer proyek secara dinamis menentukan jumlah fase, maka manajer proyek memiliki peran penting untuk mengembangkan produk menggunakan model spiral. Jari-jari spiral pada setiap titik mewakili pengeluaran biaya proyek dan dimensi sudut mewakili kemajuan atau progress yang dibuat. Visualisasi model spiral dapat dilihat pada Gambar 7.4 berikut.



**Gambar 7. 4** Spiral Model

Setiap fase model spiral dibagi menjadi empat kuadran seperti yang ditunjukkan pada Gambar 7.4. Fungsi keempat kuadran ini dibahas di bawah ini:

- 1) Penentuan tujuan dan mengidentifikasi solusi alternatif: Informasi mengenai kebutuhan dikumpulkan dari klien, mendefinisikan tujuan, dan dilakukan analisis pada awal setiap fase. Kemudian solusi alternatif yang memungkinkan untuk fase ini akan diusulkan dalam kuadran ini.

- 2) Identifikasi dan penyelesaian risiko: Selama kuadran kedua, semua solusi yang memungkinkan akan dievaluasi untuk memilih solusi terbaik. Kemudian risiko yang terkait dengan solusi tersebut diidentifikasi dan risiko diselesaikan dengan menggunakan strategi terbaik. Di akhir kuadran ini, Prototipe akan dibangun untuk solusi terbaik.
- 3) Pengembangan versi Produk berikutnya: Selama kuadran ketiga, fitur yang diidentifikasi akan dikembangkan dan diverifikasi melalui pengujian. Di akhir kuadran ketiga, versi terbaru dari software berikutnya akan tersedia.
- 4) Peninjauan dan merencanakan untuk fase berikutnya: Di kuadran keempat, klien akan melakukan evaluasi versi perangkat lunak yang dikembangkan sejauh ini. kemudian, perencanaan untuk fase selanjutnya akan dimulai.

Risiko merupakan situasi merugikan yang mungkin mempengaruhi keberhasilan penyelesaian proyek pengembangan software. Fitur paling penting dari model spiral adalah dapat menangani risiko setelah proyek dimulai. Resolusi risiko seperti itu lebih mudah dilakukan dengan mengembangkan prototipe. Model spiral mendukung penyalinan dengan risiko dengan menyediakan ruang lingkup untuk membangun prototipe di setiap fase pengembangan software.

Keuntungan menggunakan Model Spiral:

- a. Proyek dengan banyak risiko yang tidak diketahui terjadi saat pengembangan berlangsung. Dalam hal ini, model spiral adalah model pengembangan terbaik untuk diikuti karena analisis risiko dan penanganan risiko terdapat di setiap fase,
- b. Model spiral sangat bagus diterapkan dalam proyek besar dan kompleks,
- c. Perubahan permintaan dalam kebutuhan pada fase selanjutnya dapat digabungkan secara akurat dengan menggunakan model ini,
- d. Lebih mudah dalam melakukan estimasi biaya karena proses pembuatan prototype yang jelas dan terencana dalam tahapan yang sistematis,
- e. Dapat menampung feedback yang diberikan oleh klien.

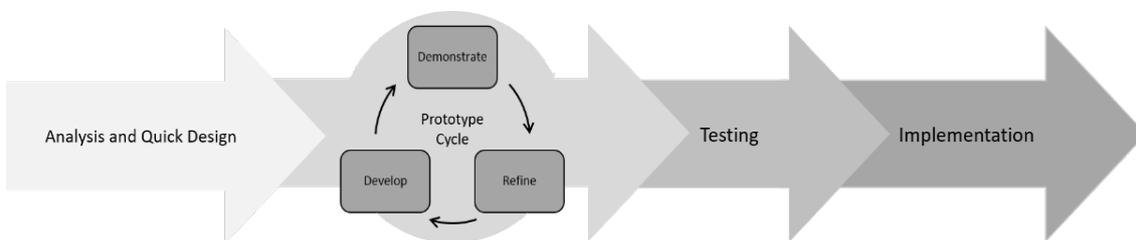
Kekurangan menggunakan Model Spiral:

- a. Model Spiral jauh lebih kompleks daripada model SDLC lainnya,
- b. Model Spiral tidak cocok untuk proyek kecil karena mahal,
- c. Keberhasilan penyelesaian proyek sangat tergantung pada Analisis Risiko,
- d. Tanpa ahli yang sangat berpengalaman, pengembangan proyek menggunakan model ini akan gagal,
- e. Tidak cocok dan sulit diimplementasikan dalam proyek kecil.

#### 7.3.4 RAD (*Rapid Application Development*) Model

Model RAD adalah sebuah adaptasi model yang mengutamakan kecepatan dan fleksibilitas tinggi, jika tiap-tiap kebutuhan dan batasan ruang lingkup proyek telah diketahui dengan baik. Proses RAD memungkinkan tim pengembang untuk menciptakan sebuah sistem yang sangat fungsional dalam jangka waktu yang sangat singkat (Pressman, 2012).

Model RAD memungkinkan penekanan dalam proses pembuatan aplikasi berdasarkan pembuatan prototype, feedback, dan iterasi yang dilakukan secara berulang. Dengan demikian, software bisa dikembangkan dan diperbaiki dengan cepat. Metode ini sangat sesuai dengan kebutuhan dan perkembangan dunia digital yang serba cepat ini. Visualisasi dari model RAD dapat dilihat pada Gambar 7.5 berikut ini.



**Gambar 7.5** RAD Model

Berdasarkan Gambar 7.5 terdapat 4 tahapan dalam model RAD ini, yaitu:

##### 1) Perencanaan Kebutuhan

Tahapan awal dari model ini yaitu perencanaan kebutuhan dimana pada tahap ini dilakukan identifikasi masalah dan pengumpulan data yang diperoleh dari klien dan stakeholder guna menentukan kebutuhan yang akan dipenuhi

dari sebuah proyek. Setelah kebutuhan sudah teridentifikasi, maka tim bisa menentukan hal-hal lain seperti timeline, anggaran, dan tujuan agar sesuai dengan keinginan. Pada tahap ini dibutuhkan keterlibatan baik dari klien maupun stakeholder guna mengembangkan suatu sistem.

## 2) Desain Sistem atau prototype

Dalam tahap ini, diperlukan keaktifan user dalam mencapai tujuan agar ketika ada proses perbaikan mengenai ketidaksesuaian desain yang dilakukan secara berulang bisa segera diperbaiki dan diidentifikasi, jadi dibutuhkan feedback dan testing dari user tersebut. Developer melakukan pembuatan prototype dari software yang diinginkan beserta dengan fungsi dan fitur yang berbeda, serta nantinya memiliki wawasan dalam pembuatan software yang mudah dipakai, desain yang baik, serta stabil. Tools yang digunakan dalam pemodelan sistem biasanya menggunakan unified modeling language (UML). Hasil dari tahapan ini yaitu spesifikasi software yang meliputi struktur data, system secara umum, dan lain-lain.

## 3) Proses pengembangan dan pendefinisian feedback

Dalam tahapan ini developer harus terus-menerus melakukan kegiatan pengembangan dan integrasi dengan bagian-bagian lainnya secara berkelanjutan dengan memperhatikan pertimbangan feedback dari klien. Baik dari segi fungsi, fitur, antarmuka, maupun segala keseluruhan aspek produk yang akan dibuat. Jika proses ini dapat berjalan dengan lancar, maka proses akan dilanjutkan ke tahap berikutnya yaitu implementasi. Jika tidak, maka proses akan dikembalikan ke tahap sebelumnya yaitu desain sistem atau prototype.

## 4) Implementasi atau penyelesaian produk.

Dalam tahap ini dilakukan penerapan desain dari system yang telah ditetapkan pada tahap sebelumnya. system akan dilakukan proses pengujian terlebih dahulu sebelum diterapkan. Dalam tahap ini juga harus sudah harus dipastikan mengenai optimasi agar stabilitas dari software yang dibuat tetap terjaga, memperbaiki antarmuka, melakukan monitoring dan maintenance,

serta melakukan penyusunan dokumen. Setelah semua berjalan dengan baik maka software bisa diterapkan dan diserahkan kepada klien.

Keuntungan menggunakan model RAD:

- a. User atau klien selalu memberikan *feedback* dari tahap awal,
- b. Rentan waktu yang dibutuhkan lebih pendek sehingga lebih mudah dalam mengakomodasi perubahan yang sering terjadi,
- c. Penggunaan komponen dapat dilakukan kembali sehingga waktu pengerjaan proyek juga berkurang cukup signifikan,
- d. Mudah untuk diamati dan melakukan pengukuran perkembangan kemajuan proyek ini karena menggunakan model *prototype*,
- e. Penggunaan *development tools* yang kuat serta meminimalisir kesalahan sehingga dapat memastikan kualitas produk yang lebih baik.

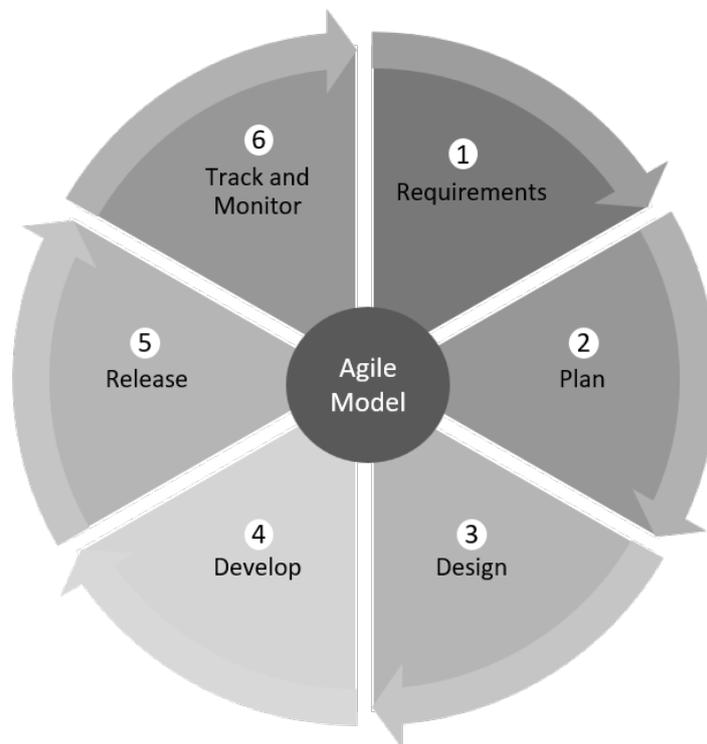
Kekurangan menggunakan model RAD:

- a. Harus melibatkan semua user atau klien terhadap seluruh siklus,
- b. Kesulitan dalam hal pengukuran mengenai kemajuan proses,
- c. Manajer proyek perlu bekerja sama yang baik dengan para user dan developer dalam pemenuhan tenggat waktu,
- d. Fasilitas banyak yang dikurangi karena terbatasnya waktu yang tersedia,
- e. Penggunaan tools yang efektif dan kuat diperlukan untuk para profesional yang sangat terampil,
- f. Hanya cocok untuk proyek yang waktunya singkat.

### 7.3.5 Agile Model

Model agile merupakan model pengembangan software yang didasarkan pada proses pengerjaan yang dilakukan berulang dimana solusi dan aturan yang dilakukan dengan kolaborasi antar tiap tim secara terorganisir dan terstruktur. Agile model ini pembaharuan dari metode waterfall dan spiral, diyakini juga dapat mempercepat proses pengembangan suatu proyek karena berlangsung dalam jangka waktu yang pendek dan bertahap.

Penggunaan model agile ini juga memiliki manfaat yang dapat dirasakan baik dari sisi klien, developer, vendor, maupun manajer yang terlibat. Klien nantinya dapat memberikan feedback kepada tim developer untuk penambahan ataupun jika ada perubahan fitur dari software sebelum dirilis. Visualisasi agile model dapat dilihat pada Gambar 7.6 berikut ini.



**Gambar 7. 6** Agile Model

Tujuan dari menggunakan Agile Model yaitu:

*1) High value and working app system*

Tujuan ini dicanangkan untuk menghasilkan software dengan biaya pembuatan yang bisa ditekan, memiliki value yang tinggi, serta produk yang berkualitas.

*2) Iterative, Incremental, Evolutionary*

Model agile ini biasanya dilakukan secara berulang-ulang, cukup fleksibel, serta dapat dilakukan perubahan apabila ada permintaan.

*3) Cost control and value*

Biaya dan waktu biasanya dikontrol oleh developer yang disesuaikan dengan kebutuhan user.

4) *High quality production*

Kualitas dari software yang dihasilkan akan tetap terjaga dengan baik meskipun waktu dan biaya yang dibutuhkan relatif sedikit.

5) *Flexible and risk management*

Pertemuan klien dapat dilakukan dengan waktu yang fleksibel dalam artian bisa dilakukan kapanpun agar fungsionalitas dari software yang dikerjakan dapat terjaga, serta dapat meminimalisir terjadinya kesalahan sebelum dirilis.

6) *Collaboration*

Setiap tim developer selalu berkolaborasi dengan baik dan melakukan diskusi terkait feedback yang diberikan oleh klien.

7) *Self organizing*

Manajer proyek harus bisa menjadi penghubung komunikasi yang baik antara klien dan developer agar tidak terjadi miss communication. Agile model ini juga memiliki tujuan guna melakukan pengembangan dan pemberian akses manajemen dalam hal development.

Keuntungan menggunakan model agile:

- a. Adanya minimalisir risiko akan kegagalan implementasi software dari segi non teknis,
- b. Proses development software ini tidak memerlukan resource dalam skala besar dan waktu yang diperlukan cukup cepat,
- c. Klien akan memberikan feedback kepada tim developer dalam proses pembuatan software sehingga perubahan dapat ditangani dengan cepat apabila terjadi perubahan.

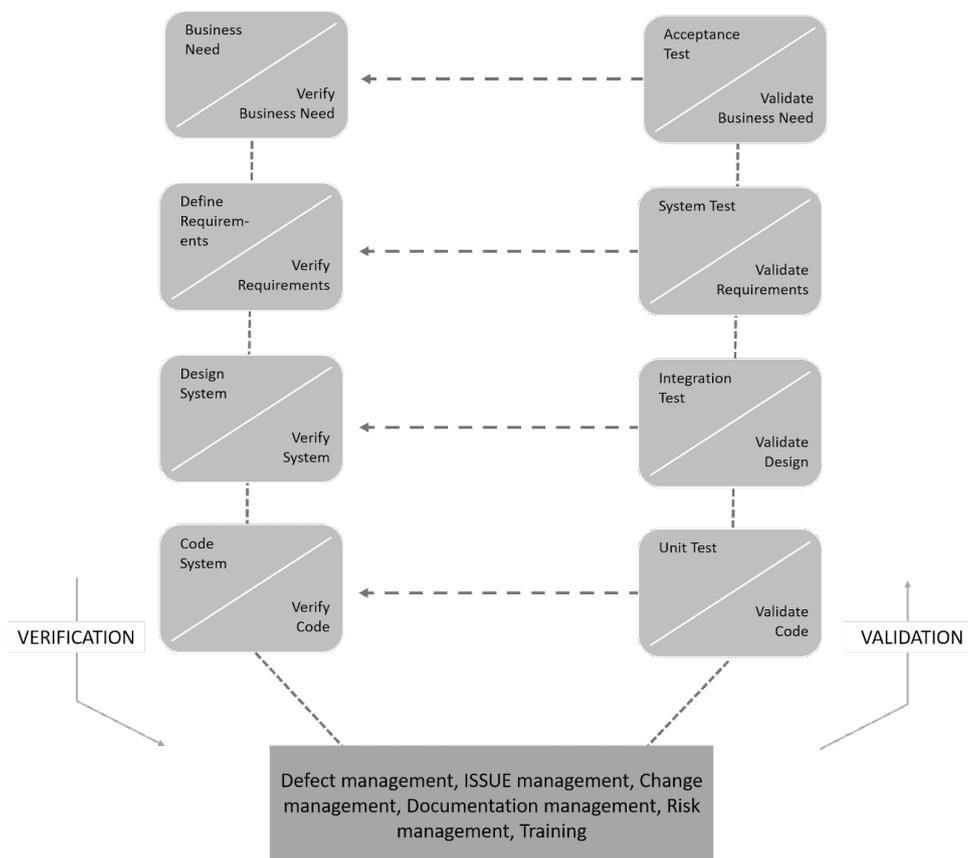
Kekurangan menggunakan model agile:

- a. Tim developer harus selalu siap siaga apabila terjadi perubahan sewaktu-waktu,
- b. Agile model ini kurang cocok jika dikerjakan dengan tim yang berjumlah relatif banyak,

- c. Apabila tim kurang bisa menyesuaikan dengan waktu pengerjaan yang relatif cepat, maka hasilnya akan kurang maksimal.

### 7.3.6 JAD (*Join Application Development*) Model

JAD atau *joint application development* adalah kerjasama yang sistematis dan terstruktur antara manajer proyek, user, maupun developer dalam menentukan permintaan user dan merancang desain terkait proyek yang akan dikerjakan sesuai dengan kebutuhan. Keterlibatan user dalam model ini dirasa cukup penting karena proyek yang dibuat haruslah menyerap kebutuhan user dan user sendirilah yang lebih mengetahui apa kebutuhannya, untuk memperoleh gambaran atau pengetahuan dalam penguasaan kondisi lingkungan kerja user, serta meningkatkan lingkungan yang demokrasi dimana user lah yang mengambil keputusan dan mengetahui dampaknya bagi mereka agar mereka tidak merasa terancam dengan adanya sistem baru. Visualisasi dari JAD model dapat dilihat pada Gambar 7.7 berikut ini.



**Gambar 7. 7** JAD Model

Keuntungan menggunakan JAD model:

- a. Model ini memungkinkan dalam pengumpulan dan konsolidasi terhadap informasi secara simultan serta kolaborasi antara manajer proyek dan klien harus baik untuk meminimalisir segala risiko,
- b. Model ini dapat menghasilkan informasi yang berkualitas tinggi dalam waktu yang singkat, hal ini dapat mengurangi waktu dan biaya yang dibutuhkan dalam pengembangan proyek,
- c. Kebutuhan yang terdefinisi dengan baik dapat meningkatkan kualitas sistem,
- d. Model ini menyediakan forum untuk mengeksplorasi berbagai sudut pandang mengenai suatu topik.

Kekurangan menggunakan JAD model:

- a. JAD model ini membutuhkan blok waktu yang besar untuk perencanaan yang signifikan dan upaya penjadwalan pada bagian dari tim pengembangan proyek,
- b. Membutuhkan komitmen yang baik dan signifikan dalam hal waktu dan usaha,
- c. Setiap tahap tidak akan berjalan dengan baik apabila persiapan dan tindak lanjut laporan tidak lengkap,
- d. Pendekatan ini membutuhkan personil yang terlatih dan berpengalaman untuk pelaksanaan keseluruhan proyek secara efektif,
- e. Pendapat yang berbeda dalam tim membuat sulit untuk menyelaraskan tujuan dan mempertahankan fokus.

#### **7.4 Alat Pengembangan Sistem Informasi**

Terdapat beberapa alat yang digunakan untuk berkomunikasi dengan user guna merancang sistem informasi yang akan dibuat. Alat pengembangan ini dibagi menjadi 2 kategori yaitu alat komunikasi pada saat tahap analisis sistem dan alat komunikasi pada saat perancangan sistem.

#### 7.4.1 Alat Komunikasi pada Tahap Analisis

Pada tahap analisis sistem, dibutuhkan alat yang dapat memfasilitasi komunikasi antara sistem analisis dengan *user*. Alat ini biasanya digunakan untuk memudahkan user memahami alur sistem, alur data dan jalannya bisnis proses sistem informasi yang akan dibangun. Beberapa tools yang biasa digunakan untuk berkomunikasi pada tahap analisis adalah:

- a. Bagan Alir Sistem (*System Flowchart*),  
*System Flowchart* adalah *flowchart* yang menampilkan tahapan atau proses kerja yang sedang berlangsung di dalam sistem secara menyeluruh.
- b. Diagram Arus Data  
Diagram Arus Data adalah suatu diagram yang menggambarkan aliran data dari sebuah proses atau sistem.
- c. Kamus Data  
Kamus data adalah suatu daftar data elemen yang terorganisir dengan definisi yang tetap dan sesuai dengan sistem, sehingga user dan analis sistem mempunyai pengertian yang sama tentang input, output, dan komponen data store.

#### 7.4.2 Alat Komunikasi pada Tahap Perancangan

Pada tahap perancangan sistem, dibutuhkan alat yang dapat memfasilitasi komunikasi antara *programmers* dengan *user*. Alat ini biasanya digunakan untuk memudahkan user memahami alur sistem, alur data dan jalannya bisnis proses sistem informasi yang akan dibangun. Beberapa tools yang biasa digunakan untuk berkomunikasi pada tahap perancangan adalah:

- a. *Entity Relationship Diagram* (ERD)  
ERD (*Entity Relationship Diagram*) adalah diagram yang digunakan untuk perancangan suatu database dan menunjukkan relasi antar objek atau entitas beserta atribut-atributnya secara detail.
- b. *Pseudocode*  
*Pseudocode* atau kode semu dapat diartikan sebagai deskripsi dari algoritma pemrograman yang dituliskan secara sederhana

---

dibandingkan dengan sintaksis bahasa pemrograman (coding). Tujuannya, agar lebih mudah dibaca dan dipahami manusia.

*c. User Interface*

*User Interface* adalah tampilan visual sebuah produk yang menghubungkan sistem dengan pengguna (*user*). Sistem ini bisa berupa website, aplikasi atau lainnya.

---

## Post Test

### SDLC (Software Development Life Cycle)

1. Apa yang dimaksud dengan SDLC?
2. Proses apa saja yang ada di dalam proses SDLC?
3. Sebutkan macam-macam SDLC yang kamu ketahui?
4. Mengapa penting untuk belajar SDLC?
5. SDLC model apa yang paling sesuai untuk mengembangkan sebuah sistem informasi?

---

## Soal Latihan

### SDLC (Software Development Life Cycle)

1. Jelaskan bagaimana konsep dari Systems Development Life Cycle!
2. Jelaskan fungsi-fungsi dari Systems Development Life Cycle!
3. Jelaskan 5 Tahapan Dasar Penerapan Systems Development Life Cycle!
4. Menurut pendapat anda model SDLC mana kah yang paling simple tetapi memiliki banyak keuntungan? Jelaskan!
5. Apakah kita boleh melewati satu tahap dalam metode SDLC? Jelaskan alasannya!