

## PERTEMUAN 2

### SYSTEMS DEVELOPMENT LIFE CYCLE (SDLC)

#### A. TUJUAN PEMBELAJARAN

Pada pertemuan ini dijelaskan tentang pengertian, sejarah pengembangan, tahapan dan alat pengembangan SDLC (*Systems Development Life Cycle*). Dari pertemuan ini diharapkan mahasiswa mampu memahami apa itu SDLC.

#### B. URAIAN MATERI

##### 1. Pengertian SDLC

Menurut bahasa SDLC bisa diartikan dengan siklus hidup pengembangan sistem. Siklus hidup sistem / produk berfungsi sebagai peta jalan mendasar untuk memahami dan mengkomunikasikan bagaimana sistem alami dan buatan manusia berkembang melalui perkembangan fase siklus hidup yang berurutan. Untuk sistem buatan manusia, peta jalan memberikan dasar untuk menilai kapabilitas dan kinerja sistem yang ada terkait dengan ancaman dan peluang; mendefinisikan, mengadakan, dan mengembangkan sistem baru untuk menanggapi ancaman dan peluang; dan menerapkan sistem baru untuk mencapai tujuan misi yang melawan atau memanfaatkan ancaman dan peluang.

Evolusi sistem apa pun yang dibuat oleh atau diketahui manusia dimulai dari titik pembuahan dan berakhir di pembuangan. Proses ini disebut sebagai siklus hidup sistem. Siklus hidup sistem berfungsi secara struktural sebagai landasan pengembangan sistem. Sistem buatan manusia dikonseptualisasikan, direncanakan, diatur, dijadwalkan, diperkirakan, diadakan, disebarkan, dioperasikan dan didukung, dan dibuang menggunakan struktur ini. Sistem alam mengikuti konstruksi serupa dengan fase kehidupan.

Siklus hidup untuk sistem, produk, atau layanan apa pun terdiri dari serangkaian fase yang dimulai dengan konsepsi sistem dan berlanjut hingga pembuangan akhir. Untuk sistem buatan manusia awalnya dan berakhirnya

setiap fase ditandai dengan titik kontrol yang signifikan atau peristiwa pementasan seperti keputusan kunci pada tinjauan teknis atau acara lapangan yang mengizinkan kemajuan ke fase berikutnya. Siklus hidup sistem yang khas terdiri dari serangkaian fase, antara lain adalah tahap definisi sistem, tahap pengadaan sistem, tahap pengembangan sistem, fase operasi dan dukungan sistem (O&S), tahap produksi sistem, fase pembuangan sistem

Bab ini menyajikan siklus hidup sistem / produk sebagai kerangka kerja tingkat atas dari fase tertanam yang diperlukan untuk mengembangkan kebutuhan operasional Pengguna untuk sistem, produk, atau layanan dari "visi" konseptual melalui pembuangan. Setiap tahapan mewakili sekumpulan aktivitas yang berfokus pada tujuan program dan hasil kerja tertentu. Seperti yang akan segera Anda temukan, beberapa fase ini memiliki akhir yang terdefinisi dengan baik yang ditandai oleh tonggak penting sementara fase lain tumpang tindih dan transisi dari satu fase ke fase lainnya.

## **2. Sejarah Pengembangan SDLC**

Kerangka kerja metodologi pengembangan perangkat lunak (juga dikenal sebagai SDM). Menurut Elliott (2004) siklus hidup pengembangan sistem (SDLC) bisa dianggap sebagai kerangka metodologi formal tertua untuk membangun sistem informasi. Ide awal dari SDLC adalah "untuk mengejar pengembangan sistem informasi dengan cara yang sangat terencana, terstruktur dan metodis, yang membutuhkan setiap tahap siklus hidup dari awal hingga penyampaian sistem akhir dilakukan secara kaku dan berurutan" dalam konteks kerangka yang diterapkan. Sasaran utama dari kerangka metodologi ini pada tahun 1960-an adalah "untuk mengembangkan sistem bisnis fungsional berskala besar. Aktivitas sistem informasi berkisar pada pemrosesan data yang berat dan rutinitas penghitungan angka".

Metodologi, proses, dan kerangka kerja berkisar dari langkah-langkah terlarang khusus yang dapat digunakan secara langsung oleh organisasi dalam pekerjaan sehari-hari, hingga kerangka kerja fleksibel yang digunakan organisasi untuk menghasilkan serangkaian langkah khusus yang disesuaikan dengan kebutuhan proyek tertentu atau kelompok. Dalam beberapa kasus, organisasi "sponsor" atau "pemeliharaan" mendistribusikan sekumpulan dokumen resmi yang menjelaskan proses tersebut. Berikut adalah ringkasan

sejarah pengembangan SDLC dari awal hingga sekarang:

- a. Tahun 1970-an
  - 1) Pemrograman Terstruktur, Sejak 1969
  - 2) Cap Gemini SDM, Sejak 1974
- b. Tahun 1980-an
  - 1) Structured Systems Analysis and Design Method atau disingkat SSADM, Sejak 1980
  - 2) Analisis Kebutuhan Informasi / Metodologi sistem lunak, Sejak 1981
- c. Tahun 1990-an
  - 1) Pemrograman berorientasi objek (OOP) dikembangkan pada awal 1960-an, dan menjadi pendekatan pemrograman yang dominan selama pertengahan 1990-an
  - 2) Pengembangan aplikasi cepat (RAD), sejak 1991
  - 3) Metode pengembangan sistem dinamis (DSDM), sejak 1994
  - 4) Scrum, sejak 1995
  - 5) Proses perangkat lunak tim, sejak 1998
  - 6) Rational Unified Process (RUP), dikelola oleh IBM sejak 1998
  - 7) Pemrograman ekstrim, sejak 1999
- d. Tahun 2000-an
  - 1) Agile Unified Process (AUP) dipertahankan sejak 2005 oleh Scott Ambler
  - 2) Pengiriman tangkas yang disiplin (DAD) Menggantikan AUP
- e. Tahun 2010-an
  - 1) Scaled Agile Framework (SAFe)
  - 2) Scrum Skala Besar (LeSS)
  - 3) DevOps

Ada Lovelace dikenal karena menulis program dasar pertama pada tahun 1843 untuk Mesin Analitik, yang dirancang oleh Charles Babbage. Tetapi hanya setelah bahasa pertama 'short-code' (0 dan 1) muncul pada tahun 1949, dan kompiler pertama ditulis pada tahun 1951; lahirlah program komputer pertama. Pada tahun 1957, Fortran muncul sebagai salah satu bahasa pemrograman besar pertama, yang diikuti oleh Cobol. Dalam 175 tahun sejarah pemrograman hingga saat ini, umat manusia telah mengembangkan lebih dari 700 bahasa pemrograman.

Anehnya, Siklus Hidup Pengembangan Sistem (SDLC) disebut sebagai riwayat Siklus hidup pengembangan *software* yang tidak sedalam program *software*. Kerangka konseptual, "SDLC" yang mempertimbangkan struktur tahapan yang terlibat dalam pengembangan aplikasi dari studi kelayakan awal hingga penerapannya di lapangan dan pemeliharaan, menjadi menonjol dengan model Waterfall. Winston W. Royce mengutip deskripsi formal pertama dari model air terjun dalam sebuah artikel pada tahun 1970.

Model air terjun menyediakan metode yang terorganisir dan terkontrol untuk mengelola proyek; model berkembang secara linier melalui tahapan diskrit, logis dan dapat diinterpretasikan, sehingga mudah untuk dipahami dan diterapkan. Ini memberikan tonggak yang mudah diidentifikasi dalam proses pengembangan. Model air terjun mempertahankan dominasinya selama dua dekade dan masih banyak digunakan oleh banyak organisasi. Perpanjangan formal Waterfall dikembangkan oleh Jerman yang disebut V-model pada tahun 1980 untuk proyek pertahanan, dan sekarang telah ditemukan aplikasi yang luas dalam program komersial serta pertahanan. Model V merangkum langkah-langkah utama yang harus diambil sehubungan dengan kiriman yang sesuai dalam pengembangan siklus hidup proyek.

Saat fokus bergerak ke arah customer centricity, model Waterfall mulai menuai kritik untuk pendekatan bertahap liniernya, yang tidak memungkinkan adanya fleksibilitas untuk perubahan pelanggan di tengah-tengah pengembangan perangkat lunak dan pada banyak kesempatan memiliki garis waktu yang memanjang, yang mengakibatkan waktu yang tinggi untuk -pasar. Pengembangan berulang diciptakan sekitar tahun 1975 sebagai tanggapan atas inefisiensi dan masalah yang ditemukan dalam model Waterfall. Komunitas TI menggembarkan-gemborkannya sebagai terobosan besar, karena model Iteratif berusaha untuk membangun kerangka kerja yang gesit dan adaptif. Hal ini juga membawa revolusi bagi komunitas manajemen proyek. Banyak metode proses pengembangan perangkat lunak baru mulai bermunculan, dan setiap metode telah membawa ide dan konsep baru untuk mengelola proyek, seperti Evolutionary (1976), Incremental (1978), Stage Gate (1983), Spiral (1986), dll. Iterative, Incremental dan Spiral memperoleh apresiasi yang signifikan di antara Manajer Proyek dari berbagai organisasi dan masih digunakan di seluruh dunia oleh banyak orang.

Iterative dan Incremental sering saling melengkapi dan dalam beberapa proyek, keduanya digunakan bersamaan. Fondasi dasar dari metode ini mendukung pengembangan sistem melalui siklus berulang (Iteratif) dan dalam himpunan bagian yang lebih kecil dalam satu contoh (Inkremental), sehingga memungkinkan tim untuk memanfaatkan pembelajaran dari fase sebelumnya dan berimprovisasi dalam iterasi saat ini. Pembelajaran diperoleh baik melalui proses pengembangan maupun penggunaan sistem sub-set (Incremental) yang digunakan. Proses ini biasanya dimulai dengan implementasi subset dari persyaratan perangkat lunak dan secara berulang meningkatkan versi yang berkembang sampai sistem penuh diimplementasikan. Pada setiap iterasi, modifikasi desain dibuat dan kemampuan fungsional baru ditambahkan.

Evolusi SDLC berlanjut, dengan fokus pada ketangkasan dalam upaya untuk mengurangi waktu ke pasar, membangun produk yang layak minimum sambil menjaga pelanggan di pusat segalanya. Ini membutuhkan paradigma perubahan karena mempromosikan individu dan interaksi melalui proses dan alat, perangkat lunak yang bekerja melalui dokumen yang komprehensif, kolaborasi pelanggan melalui semua fase, dan menanggapi perubahan mengikuti rencana. Kebutuhan saat ini adalah untuk memiliki metode yang lebih fleksibel yang memenuhi permintaan. Metode *agile* muncul sebagai spin-off langsung metode perangkat lunak dari tahun 1980-an, yaitu Joint Application Design atau JAD (1986), Rapid Systems Development atau RSD (1987), dan Rapid Application Development atau RAD (1991).

Metode Agile secara resmi dimulai pada 1990-an dan banyak Metode Pengembangan Perangkat Lunak dikembangkan dalam beberapa dekade mendatang. Ini dimulai dengan Crystal (1991) yang berfokus terutama pada orang dan interaksi, Scrum (1993) yang mencakup empat manifesto tangkas, Pengembangan Sistem Dinamis (1994) muncul ke cakrawala setelah manajer proyek menggunakan RAD (Pengembangan Aplikasi Cepat) mencari lebih banyak tata kelola dan disiplin terhadap cara kerja iteratif baru, Synch-n-Stabilize (1995) menanamkan kerja paralel pada modul aplikasi individu, sering menyinkronkan kode yang dikembangkan dengan yang ada di tim lain, dan secara teratur men-debug (menstabilkan) kode selama proses pengembangan, Feature Driven Development (1996) yang memadukan sejumlah praktik terbaik tangkas yang diakui industri menjadi keseluruhan kohesif yang didorong oleh nilai klien.

Ada berbagai metode agile lainnya seperti Judo Strategy di tahun 1997, Internet Time di tahun 1998, New Development Rhythm di tahun 1989, Adaptive Software Development di tahun 1999, Open Source Software Development di tahun 1999, Lean Development di tahun 2003, dan Agile Proses Terpadu di tahun 2005. Namun, keberhasilan Extreme Programming (1999) menyebabkan adopsi metode *agile* yang belum pernah terjadi sebelumnya pada awal tahun 2000-an.

Extreme Programming (XP) dirancang untuk meningkatkan kualitas *software* dan daya tanggap terhadap perubahan kebutuhan pengguna. Inovasi metode ini mengubah berbagai praktik yang ada hingga ke tingkat yang ekstrim. Misalnya, praktik XP dari Pemrograman Berpasangan mengubah praktik tinjauan kode yang ada ke tingkat yang ekstrem, karena praktik tersebut mendorong kode produksi untuk ditulis oleh dua pengembang di satu mesin (mis. Pengemudi dan navigator). Banyak dari praktik lain dalam Pemrograman Ekstrim membawa perubahan radikal pada metode yang ada, beberapa di antaranya adalah Integrasi Berkelanjutan, permainan perencanaan, pelanggan di lokasi, Refactoring, rilis kecil, dan desain sederhana.

Mengurangi waktu penyelesaian melalui adopsi yang gesit secara inheren membuat departemen Operasi tertinggal dengan penerapan yang menumpuk lebih cepat daripada yang bisa dirilis. Tren ini pada akhirnya mendorong munculnya DevOps, versi baru dari metodologi Agile yang mencakup segmen Pengembangan (Dev) dan Operasi (Ops) untuk memungkinkan ketangkasan dan kualitas penyampaian layanan. Istilah ini pertama kali diciptakan pada tahun 2009 dalam sebuah konferensi dan sejak saat itu telah mewujudkan siklus pengembangan sistem ujung-ke-ujung yang sering menghadirkan fitur, perbaikan, dan *update* yang sejalan dengan tujuan bisnis. Ini menekankan pada pengiriman berkelanjutan, otomatisasi, dan penerapan berkelanjutan.

Selama lima dekade terakhir pergeseran budaya dari Waterfall ke Agile dan akhirnya, ke DevOps memerintahkan motivasi dan kepemimpinan yang kuat untuk mengatasi tantangan yang terkait dengan perubahan revolusioner dalam kerangka SDLC dan membutuhkan tingkat adaptasi yang tinggi di antara tim yang terlibat dalam pengembangan perangkat lunak. . Evolusi proses membantu mengatasi kekurangan sistem yang berlaku; perubahan semacam itu terkait langsung dengan keuntungan, kesuksesan dan keuntungan

perusahaan, dan organisasi yang menolak untuk berubah hanya berisiko tertinggal dalam perlombaan untuk memberikan produk bernilai tinggi kepada pelanggan.

### **3. Tahapan SDLC**

#### **a. Planning/ Perencanaan Sistem**

Tahap perencanaan sistem adalah proses awal untuk memahami mengapa sistem informasi harus dibangun dan menentukan bagaimana pengembang akan membangunnya. Langkah pertama yang dilakukan adalah melakukan investigasi awal. Dari hasil investigasi awal, survei diperluas menjadi studi kelayakan yang lebih rinci. Studi kelayakan adalah pengujian proposal sistem sesuai dengan kemampuan kerjanya. Dampak pada organisasi, kemampuan untuk memenuhi kebutuhan pengguna, dan penggunaan sumber daya secara efektif. Ini berfokus pada tiga pertanyaan utama:

- 1) Apa kebutuhan pengguna yang dapat dibuktikan dan bagaimana sistem kandidat memenuhinya?
- 2) Sumber daya apa yang tersedia untuk sistem kandidat tertentu? Apakah masalahnya layak dipecahkan?
- 3) Apa kemungkinan dampak dari sistem kandidat pada organisasi? Seberapa cocok itu dengan rencana induk MIS organisasi?

Setiap pertanyaan ini harus dijawab dengan hati-hati. Mereka berputar di sekitar penyelidikan dan evaluasi masalah, identifikasi dan deskripsi sistem kandidat, spesifikasi atau kinerja dan biaya setiap sistem dan pemilihan akhir dari sistem terbaik.

Tujuan dari studi kelayakan bukanlah untuk memecahkan masalah tetapi untuk mengetahui ruang lingkupnya. Selama studi definisi masalah dikristalisasi dan aspek masalah yang akan dimasukkan ke dalam sistem ditentukan. Akibatnya, biaya dan manfaat diperkirakan dengan lebih akurat pada tahap ini.

Hasil studi kelayakan berupa proposal formal. Ini hanyalah sebuah laporan dokumen formal yang merinci sifat dan ruang lingkup solusi yang diusulkan. Proposal tersebut merangkum apa saja yang diketahui dan apa

saja yang perlu dilakukan, terdiri dari:

- 1) Pernyataan masalah yang diucapkan dengan cermat yang mengarah ke analisis.
- 2) Ringkasan temuan dan rekomendasi daftar temuan utama dan rekomendasi penelitian. Ini sangat ideal untuk pengguna yang membutuhkan akses cepat ke hasil analisis sistem yang diteliti. Kesimpulan dinyatakan diikuti dengan daftar rekomendasi dan justifikasi untuk itu.
- 3) Rincian temuan garis besar metode dan prosedur yang dilakukan oleh sistem yang ada diikuti dengan cakupan tujuan dan prosedur sistem kandidat. Termasuk juga diskusi tentang laporan keluaran, struktur file, dan biaya dan manfaat dari sistem kandidat.
- 4) Rekomendasi dan kesimpulan- rekomendasi khusus mengenai sistem kandidat termasuk penugasan personel, biaya, proyek jadwal, dan tanggal target.

Setelah manajemen meninjau proposal, itu menjadi kesepakatan formal yang membuka jalan untuk desain dan implementasi yang sebenarnya. Ini adalah titik keputusan penting dalam siklus hidup. Banyak proyek mati di sini, sedangkan yang lebih menjanjikan terus berlanjut melalui implementasi. Perubahan dalam proposal dibuat secara tertulis, tergantung pada ukuran kompleksitas, dan biaya proyek. Masuk akal untuk memverifikasi perubahan sebelum melakukan desain proyek.

#### b. Analysis/ Analisis Sistem

Langkah berikutnya dalam proses pengembangan sistem adalah analisis sistem. Analisis sistem bertujuan untuk memungkinkan pengembang memiliki pemahaman yang lebih detil tentang masalah dan persyaratan yang memicu proyek. Oleh karena itu, domain bisnis (ruang lingkup proyek yang ditentukan selama perencanaan sistem) dapat diteliti dan dianalisis untuk memahami lebih detail apa yang berfungsi, apa yang tidak berguna, dan apa yang dibutuhkan. Analisis sistem membutuhkan kerja sama dengan pengguna sistem untuk menentukan persyaratan bisnis dan ekspektasi untuk sistem baru apa pun yang akan dibeli atau dikembangkan. Juga, prioritas bisnis mungkin perlu ditetapkan jika jadwal dan anggaran tidak mencukupi untuk diselesaikan sesuai yang diinginkan.



Dalam fase ini, tim akan menyelidiki sistem saat ini, mengidentifikasi peluang peningkatan, dan mengembangkan konsep untuk sistem baru. Tahap ini mencakup tiga langkah:

1. Strategi analitis dibuat untuk memandu tim pengembang. Strategi ini sering melibatkan analisis sistem saat ini beserta masalahnya dan memperkenalkan pendekatan baru untuk desain sistem.
2. Langkah berikutnya adalah pengumpulan persyaratan (misalnya, melalui kuesioner ataupun wawancara). Analisa informasi dan masukan dari manajer proyek dan banyak lainnya mengarah pada pengembangan konsep sistem baru.
3. Hasil analisis, konsep dan model sistem digabungkan menjadi satu dokumen yang disebut proposal sistem, yang akan diserahkan kepada sponsor proyek dan pembuat keputusan utama lainnya (misalnya, anggota komite persetujuan), yang memutuskan apakah akan melanjutkan proyek.

Konsep dan model sistem digabungkan menjadi dokumen yang disebut proposal sistem, yang diajukan ke sponsor proyek. Proposal sistem adalah pernyataan awal yang menjelaskan persyaratan bisnis yang harus dipenuhi oleh sistem baru. Karena ini sebenarnya merupakan langkah pertama dalam merancang sistem baru, beberapa ahli berpendapat bahwa tidak tepat menggunakan istilah "analisis" sebagai nama untuk tahap ini. Beberapa orang berpikir bahwa nama yang lebih baik adalah "analisis dan desain awal". Namun, sebagian besar organisasi terus menggunakan analisis nama pada tahap ini. Harus diingat bahwa hasil dari tahap analisis adalah analisis tingkat tinggi awal dan desain sistem baru, dan keputusan besar lainnya (misalnya, anggota komite persetujuan) menentukan apakah proyek harus dilanjutkan.

#### c. Design/ Perancangan Sistem

Setelah memahami kebutuhan sistem informasi, kita dapat melanjutkan ke perancangan sistem. Selama perancangan sistem, perlu dicari solusi teknis alternatif terlebih dahulu. Ada beberapa solusi untuk masalah apa pun. Misalnya, sebagian besar perusahaan saat ini perlu memilih antara membeli solusi yang masuk akal dan membangun solusi

khusus mereka sendiri.

Setelah alternatif teknis dipilih dan disetujui, dalam fase ini akan dikembangkan *blueprint* teknis dan spesifikasi yang diperlukan untuk mengimplementasikan solusi akhir. *Blueprint* dan spesifikasi akan digunakan untuk mengimplementasikan database, program, antarmuka pengguna, dan jaringan yang diperlukan untuk membuat sebuah sistem.

Fase desain sistem menentukan mode operasi sistem sesuai dengan *hardware*, *software* dan infrastruktur jaringan; *user interface*, formulir dan laporan; dan prosedur khusus, database dan file yang diperlukan. Meskipun keputusan paling strategis tentang sistem dibuat selama pengembangan konsep sistem dalam tahap analisis, langkah-langkah dalam tahap desain dapat secara akurat menentukan bagaimana sistem beroperasi. Fase desain mencakup empat langkah:

- 1) Pertama, strategi dikembangkan untuk menentukan apakah sistem akan dikembangkan oleh programmer mereka sendiri, apakah sistem sedang ditransfer ke perusahaan lain (biasanya perusahaan konsultan), atau apakah perusahaan akan membeli paket *software* yang ada.
- 2) Kedua, pengembangan perancangan desain dasar sistem, yang menjelaskan *hardware*, *software* dan jaringan yang akan digunakan. Dalam beberapa kasus, sistem menambah atau mengubah infrastruktur yang ada. Desain *interface* menentukan bagaimana *user* bisa menggunakan sistem (seperti metode navigasi, menu dan tombol pada layar) serta formulir dan laporan yang akan dipakai sistem.
- 3) Ketiga, pengembangan *database* dan spesifikasi dokumen. Ini menentukan data apa yang akan disimpan dan di mana sistem akan menyimpannya.
- 4) Keempat, mengembangkan desain program yang menentukan program yang akan ditulis dan operasi yang akan dilakukan setiap program.

#### **4. Implementation/ Implementasi Sistem**

Tahap akhir dari siklus hidup sistem adalah implementasi, di mana sistem benar-benar akan dibangun dan dikembangkan. Ini biasanya merupakan tahap yang paling mengkhawatirkan, karena untuk kebanyakan sistem, ini adalah bagian tunggal yang terpanjang dan paling mahal dari proses pengembangan.

Tahap ini mencakup tiga langkah:

- a. Pertama adalah pembangunan sistem. Sistem dibangun dan diuji untuk memastikan berfungsi sebagaimana mestinya. Karena biaya kesalahan bisa jadi tinggi, pengujian merupakan salah satu langkah terpenting dalam proses implementasi. Dibandingkan dengan menulis program dari awal, sebagian besar organisasi mencurahkan lebih banyak waktu dan energi untuk pengujian. Pengujian sistem memeriksa kesiapan dan akurasi sistem untuk mengakses, memperbarui, dan mengambil data dari file baru. Setelah program tersedia, data pengujian dibaca ke komputer dan diproses terhadap file yang disediakan untuk pengujian. Jika berhasil, program akan berjalan dengan data "real-time". Jika tidak, proses diagnostik akan digunakan untuk menemukan dan memperbaiki kesalahan dalam program. Dalam kebanyakan konversi, proses paralel dilakukan dengan sistem baru yang berjalan bersamaan dengan sistem "lama". Meskipun metode ini mahal, namun dapat memberikan jaminan tambahan untuk kesalahan dalam sistem kandidat, dan juga memberikan kesempatan kepada pengguna untuk mendapatkan pengalaman melalui operasi. Namun, dalam beberapa kasus, pemrosesan paralel tidak praktis. Misalnya, tidak masuk akal untuk menjalankan dua sistem titik penjualan (POS) online paralel untuk rantai ritel. Namun, setelah sistem calon terbukti, sistem lama ditinggalkan.
- b. Kedua proses instalasi sistem. Instalasi adalah tahap di mana sistem lama diganti dengan sistem yang baru dibuat. Salah satu hal terpenting dari tahap ini adalah mengembangkan rencana pelatihan untuk memberikan pemahaman kepada *user* cara menggunakan sistem yang baru dan membantu mengelola perubahan yang terjadi. Setelah fase instalasi selesai, pengguna dapat beradaptasi dengan perubahan yang dilakukan pada sistem baru, dan evaluasi serta pemeliharaan dimulai.
- c. Ketiga tim analis membuat rencana pendukung sistem. Rencana tersebut biasanya mencakup tinjauan pasca implementasi, dan pendekatan sistematis untuk mengidentifikasi perubahan yang diperlukan oleh sistem. Seperti sistem lainnya, ada proses penuaan yang membutuhkan perawatan rutin perangkat keras dan perangkat lunak. Jika informasi baru tidak memenuhi spesifikasi desain, maka harus diubah. Perangkat keras juga membutuhkan perawatan rutin untuk memenuhi spesifikasi desain. Pentingnya pemeliharaan adalah untuk menjaga agar sistem baru tetap

sesuai standar.

## 5. Alat Pengembangan

Selama menjalankan metodologi pengembangan sistem dari mulai perencanaan hingga implementasi diperlukan alat dan teknik untuk melakukannya. Alat yang digunakan biasanya berupa gambar, bagan atau grafik. Selain itu digunakan pula alat non grafik, seperti kamus data, bahasa Inggris terstruktur, pseudo-code, dan formulir untuk merekam dan menyajikan data.

### a. Alat-Alat Yang Berupa Grafik

*Tools* pengembangan sistem grafis meliputi:

#### 1) Diagram HIPO

HIPO singkatan dari Hierarchy Plus Input Process Output, merupakan alat dokumentasi program yang basisnya fungsi, setiap modul dalam sebuah sistem dijelaskan melalui fungsi utamanya.

#### 2) DFD

DFD singkatan dari *Data Flow Diagram*, adalah alat yang digunakan untuk menjelaskan secara terperinci sistem lama ataupun sistem baru yang akan dikembangkan secara logis tanpa melihat lingkungan fisik tempat data diekstraksi atau lingkungan fisik tempat data akan disimpan.

#### 3) *Structured Chart*

*Structured Chart* merupakan alat yang digunakan untuk mendefinisikan dan menjelaskan struktur sistem informasi secara bertahap dalam bentuk modul dan sub modul dengan menampilkan hubungan antar elemen data dan elemen kontrol antar modul, sehingga dapat memberikan gambaran yang lengkap tentang sistem.

#### 4) SADT

SADT singkatan dari *Structure Analysis and Design Technique* atau Teknik analisis dan desain terstruktur merupakan alat yang memperlakukan sistem sebagai dua bagian yakni objek dan peristiwa. Dua jenis diagram yang digunakan yaitu diagram aktivitas (*activity diagram* disebut action diagram) dan diagram data (diagram data disebut *datagram*).

#### 5) Diagram Jackson's

*Jackson's System Development* yang disingkat JSD menetapkan model dunia nyata (*real world*) yang menyediakan subjek masalah sistem. Selain alat grafis yang digunakan dalam metode tertentu, ada beberapa alat grafis yang bersifat universal dan dapat digunakan di semua metode yang ada. Alat tersebut berbentuk bagan dan dapat diklasifikasikan sebagai berikut:

- a) Bagan yang menggambarkan aktivitas (*activity charting*)
  - (1) *Flowchart* sistem
  - (2) *Flowchart* program yang terdiri dari :
    - (a) *Flowchart* logika program
    - (b) *Flowchart* program computer terperinci
  - (3) *Flowchart* dokumen
  - (4) *Flowchart* proses
  - (5) Gantt chart
- b) Bagan yang menggambarkan tata letak atau *layout charting*
- c) Bagan yang menggambarkan hubungan personil atau *personil relationship charting*
  - (1) *Working distribution chart*
  - (2) *Organization chart*

b. Teknik Pengembangan Sistem

Berikut adalah teknik yang digunakan dalam pengembangan sistem :

- 1) Teknik manajemen proyek yang dipakai untuk tahap perencanaan. Antara lain CPM (*critical path method*) dan PERT (*prosedur evaluasi dan teknik review*).
- 2) Teknik pencarian fakta, yaitu teknik yang dipakai untuk mengumpulkan data dan fakta dalam kegiatan analisa sistem. Teknik tersebut antara lain meliputi:
  - a) Wawancara atau *Interview*
    - Persiapan yang dilakukan :
      - ✓ Membuat janji pertemuan
      - ✓ Memastikan orang yang akan diwawancarai
      - ✓ Menentukan pokok permasalahan
      - ✓ Perhatikan siapa yang diwawancarai
      - ✓ Mencatat tanggapan
      - ✓ Membuat jadwal kapan akan bertemu kembali

- b) Observasi atau *Observation*
  - c) Daftar pertanyaan atau *Questionnaires*
  - d) Pengumpulan Sampel atau *Sampling*
- 3) Teknik analisis biaya/ manfaat atau *Cost Effectiveness Analysis* atau *Cost Benefit Analysis* adalah Suatu teknik yang digunakan untuk menghitung biaya yang diperlukan dalam pengembangan sistem informasi.
- 4) Teknik menjalankan rapat

Tujuan diadakannya rapat dalam pengembangan sistem meliputi:

- ✓ Mendefinisikan permasalahan
- ✓ Mengumpulkan gagasan-gagasan
- ✓ Memecahkan setiap permasalahan
- ✓ Menyelesaikan setiap konflik yang terjadi
- ✓ Menganalisa perkembangan proyek
- ✓ Mengumpulkan data ataupun fakta
- ✓ Melakukan perundingan-perundingan

Tahapannya antara lain:

- ✓ Membuat Perencanaan rapat
- ✓ Menjalankan rapat sesuai rencana
- ✓ Menindaklanjuti semua hasil rapat

- 5) Teknik Inspeksi atau *Walkthrough*

Proses analisis sistem dan desain sistem perlu dipantau dan diawasi. Pengawasannya bisa dilakukan dengan melakukan verifikasi setiap hasil dari tahap pengembangan sistem yang telah selesai dilakukan. Verifikasi formal atas hasil kerja disebut inspeksi, dan secara informal disebut *walkthrough*.

### C. SOAL LATIHAN/TUGAS

1. Apa yang dimaksud dengan SDLC? Jelaskan!
2. Pada tahun berapa SDLC mulai dikembangkan? Ceritakan secara singkat awal mula SDLC!
3. Dari sekian banyak metode pengembangan yang ada, mana yang menurut kalian metode yang paling baik dalam pengembangan sistem? Jelaskan!
4. Ada berapa tahapan dalam pengembangan SDLC? Jelaskan!
5. Sebutkan beberapa alat yang diperlukan dalam pengembangan sistem dan jelaskan fungsinya dalam tahapan pengembangan sistem!

### D. REFERENSI

Alan Dennis, Barbara Haley Wixom, David Tegarden (2015). *Systems Analysis & Design: an object-oriented approach with UML*. 5th edition. New Jersey: John Wiley & Sons, Inc.

Charles S. Wasson (2006). *System Analysis, Design, and Development: Concepts, Principles, and Practices*. New Jersey: John Wiley & Sons, Inc.

Jeffrey L. Whitten, Lonnie D. Bentley (2007). *Systems Analysis and Design Methods*. New York: The McGraw-Hill Companies, inc.

Shantanu Choudhary (2018). *Evolution of System Development Life Cycle (SDLC)*. Diakses dari: <https://www.linkedin.com/pulse/evolution-system-development-life-cycle-sdlc-shantanu-choudhary>

Wikipedia (2020). *Software Development Process*. Diakses dari : [https://en.wikipedia.org/wiki/Software\\_development\\_process#:~:text=The%20software%20development%20methodology%20\(also,framework%20for%20building%20information%20systems](https://en.wikipedia.org/wiki/Software_development_process#:~:text=The%20software%20development%20methodology%20(also,framework%20for%20building%20information%20systems).

## GLOSARIUM

**Blueprint atau Cetak Biru** adalah kerangka rinci, sebagai dasar perumusan kebijakan, meliputi penetapan tujuan dan sasaran, perumusan strategi, rencana pelaksanaan dan prioritas kegiatan, serta langkah-langkah atau implementasi yang harus dilakukan setiap departemen di lingkungan kerja.

**Data** merupakan kumpulan angka dan karakter yang tidak berarti. Dari data tersebut dapat diolah untuk mendapatkan informasi.

**Database** adalah kumpulan file yang saling terkait dan membentuk struktur data. Database minimal terdiri dari file yang cukup bagi komputer untuk beroperasi dengan cara ini.

**Flowchart** adalah suatu diagram yang di situ terdapat simbol-simbol tertentu dan memiliki penjelasan yang berbeda. Simbol tersebut dapat mengilustrasikan tentang urutan proses secara detail.

**Gate** adalah Sirkuit digital yang hanya menghasilkan dalam kondisi input tertentu. Setiap fungsi logika (seperti DAN atau ATAU) dapat ditampilkan melalui gerbang yang sesuai. Fungsi logika yang lebih kompleks dapat diwujudkan dengan menghubungkan beberapa gerbang secara seri.

**Internet** adalah istilah umum yang digunakan untuk menunjuk jaringan kelas dunia yang terdiri dari komputer dan layanan atau sekitar 30-50 juta pengguna komputer dan lusinan sistem informasi termasuk e-mail, Gopher, FTP dan *World Wide Web*.