

# BAB 7

## PENGEMBANGAN SISTEM INFORMASI

---

### A. Capaian Pembelajaran

Setelah selesai mengikuti materi analisis dan sistem pengembangan informasi manajemen, mahasiswa diharapkan mampu:

1. Menguraikan terkait siklus hidup sistem informasi
2. Menguraikan terkait fungsi siklus hidup sistem informasi
3. Menguraikan terkait tahapan atau fase siklus hidup sistem informasi
4. Menguraikan terkait model pengembangan sebuah sistem informasi

### B. Materi

#### 1. Siklus Hidup Sistem Informasi.

*SDLC* adalah singkatan dari *Systems Development Life Cycle* atau dalam bahasa Indonesia disebut siklus hidup pengembangan sistem. *SDLC* merupakan sebuah siklus yang dipakai dalam pembuatan atau pengembangan sistem informasi yang bertujuan untuk menyelesaikan masalah secara efektif.

Dalam pengertian lain, *SDLC* adalah tahapan kerja yang bertujuan dalam menghasilkan sistem berkualitas tinggi yang sesuai dengan keinginan pelanggan atau tujuan dibuatnya sistem tersebut. *SDLC* adalah kerangka kerja yang berisi langkah-langkah yang harus diikuti untuk pengembangan perangkat lunak. Sistem ini berisi rencana lengkap untuk

pengembangan, pemeliharaan, dan penggantian perangkat lunak tertentu.

*SDLC* adalah suatu kegiatan pelatihan atau pengembangan perangkat lunak (sistem informasi), yang dalam tahapannya juga melalui suatu siklus pengembangan di beberapa departemen. Memang ada pengujian sistem agar perangkat lunak yang dibangun benar-benar memberikan hasil terbaik.

*SDLC* juga merupakan pola yang diambil untuk mengembangkan sistem perangkat lunak, yang terdiri dari tahap-tahap: rencana (*planning*), analisis (*analysis*), desain (*design*), implementasi (*implementation*), uji coba (*testing*) dan pengelolaan (*maintenance*).

Dalam rekayasa perangkat lunak, konsep *SDLC* mendasari berbagai jenis metode pengembangan perangkat lunak. Metode-metode ini membentuk kerangka kerja untuk merencanakan dan mengendalikan pembuatan sistem informasi, khususnya proses pengembangan perangkat lunak.

Terdapat 3 (tiga) jenis metode siklus hidup sistem yang paling banyak digunakan, yakni:

- a. siklus hidup sistem tradisional (*traditional system life cycle*),
- b. siklus hidup menggunakan prototyping (*life cycle using prototyping*),
- c. siklus hidup sistem orientasi objek (*object-oriented system life cycle*).

## 2. Fungsi SDLC

Sulit untuk membuat perangkat lunak tanpa desain yang optimal. Beberapa teknik pengembangan perangkat lunak terus dikembangkan hingga saat ini. Masih banyak perdebatan mengenai metode mana yang terbaik dan paling cocok untuk semua jenis software. Namun, beberapa perencanaan lebih baik daripada tidak ada perencanaan sama sekali.

*SDLC* digunakan untuk membangun suatu sistem informasi agar dapat berfungsi sebagaimana mestinya. *SDLC* (*Systems Development Life Cycle*), dalam rekayasa sistem dan rekayasa perangkat lunak, adalah proses membuat dan memodifikasi sistem serta model dan metode yang digunakan untuk mengembangkan sistem tersebut. Konsep ini sering merujuk pada komputer atau sistem informasi.

Dilihat dari berbagai sudut pandang, *SDLC* memiliki banyak fungsi, yaitu:

- a. sebagai sarana komunikasi antara tim pengembangan dan pemangku kepentingan.
- b. membagi peran dan tanggung jawab antara pengembang, desainer, analis bisnis, dan manajer proyek.
- c. memberikan gambaran yang jelas mengenai input dan output dari satu tahap ke tahap berikutnya.

## 3. Fase – Fase dalam SDLC

*SDLC* juga merupakan pola yang diambil untuk mengembangkan sistem perangkat lunak, yang terdiri dari tahap-tahap sebagai berikut:

- a. **Perencanaan Sistem (*Systems Planning*)**

Lebih menekankan pada aspek studi kelayakan pengembangan sistem (*feasibility study*). Aktivitas-aktivitas yang terdapat pada tahap perencanaan meliputi:

- 1) Pembentukan dan konsolidasi tim pengembang.
- 2) Mendefinisikan tujuan dan ruang lingkup pengembangan.
- 3) Mengidentifikasi apakah masalah-masalah yang ada bisa diselesaikan melalui pengembangan sistem.
- 4) Menentukan dan evaluasi strategi yang akan digunakan dalam pengembangan sistem.
- 5) Penentuan prioritas teknologi dan pemilihan aplikasi.

Pada tahap ini tujuannya adalah mengidentifikasi dan memprioritaskan sistem informasi yang akan dikembangkan, tujuan yang ingin dicapai, jangka waktu pelaksanaan, serta mempertimbangkan pendanaan yang tersedia dan siapa yang akan melaksanakannya.

Pentingnya tahap perencanaan karena:

- 1) Permasalahan didefinisikan dan diidentifikasi secara rinci
- 2) Pembangunan SI harus diarahkan pada peningkatan keunggulan kompetitif
- 3) Perubahan aliran informasi akan terjadi secara besaran-besaran di dalam organisasi
- 4) Implementasi teknologi komputer akan membawa dampak bagi tenaga kerja di dalam organisasi.

Beberapa manfaat dari Perencanaan SI berbasis komputer adalah:

- 1) Meningkatkan komunikasi antara manajer, pemakai dan pembuat
- 2) Meningkatkan efektifitas penggunaan sumber daya organisasi
- 3) Mendukung komunikasi untuk pertanggungjawaban kegiatan yang dilakukan oleh individu maupun departemen
- 4) Mendukung proses evaluasi.
- 5) Memungkinkan para manajer untuk mengelola pembangunan system jangka panjang

**b. Analisis Sistem (*Systems Analysis*)**

Pada tahap ini, sistem akan dianalisis bagaimana akan dijalankan nantinya. Hasil analisis berupa kelebihan dan kekurangan sistem, fungsi sistem, hingga pembaharuan yang dapat diterapkan.

Beberapa aktivitas yang terdapat pada tahap analisa sistem adalah berikut:

- 1) Melakukan studi literatur untuk menemukan suatu kasus yang bisa ditangani oleh sistem.
- 2) Brainstorming dalam tim pengembang mengenai kasus mana yang paling tepat dimodelkan dengan sistem.
- 3) Mengklasifikasikan masalah, peluang, dan solusi yang mungkin diterapkan untuk kasus tersebut.
- 4) Analisa kebutuhan pada sistem dan membuat Batasan-batasan sistem.
- 5) Mendefinisikan kebutuhan sistem.

Pada tahap analisis system pembuat sistem akan menganalisis permasalahan lebih mendalam dengan menyusun suatu studi kelayakan. Menurut Mc. Leod terdapat 6 (enam) dimensi kelayakan:

- 1) Kelayakan teknis, analisis hardware, software, dan organisasi
- 2) Pengembalian Teknis, analisis manfaat dan potensi pengembalian
- 3) Pengembalian Non-ekonomis, analisis manfaat dan pengembalian tidak dapat diukur secara finansial
- 4) Hukum dan Etika, analisis batasan hukum, etika dan kultur
- 5) Operasional, analisis penerapan/implementasi sistem
- 6) Jadwal, analisis batasan waktu penyusunan sistem,

Studi Kelayakan juga harus dilakukan terhadap Faktor-faktor berikut ini agar pemodelan sistem informasi dapat digunakan dalam lingkup yang tepat:

- 1) Kelayakan organisasi
- 2) Memilih kelompok bisnis atau pasar sasaran mana yang akan menjadi tujuan penetrasi produk-produk yang akan dipasarkan
- 3) Melihat kemungkinan-kemungkinan permodalan
- 4) Tingkat kompetisi produk harus dapat dideteksi dengan baik.
- 5) Lingkungan operasional sistem.
- 6) Sistem harga

Menyusun analisis kebutuhan (*requirement analysis*) merupakan lanjutan yang bisa dilakukan sesudah studi

kelayakan, tetapi bisa juga dilakukan setelah proposal disetujui dan kontrak disepakati. Jenis-Jenis DARI Requirement antara lain:

- 1) *Bussiness Requirements*
- 2) *Stakeholder Requirements*
- 3) *End-User Requirements*
- 4) *System Requirements*
- 5) *Software Requirement*

**c. Perancangan Sistem (*Systems Design*)**

Langkah ini akan menghasilkan prototipe dan beberapa keluaran lainnya, termasuk dokumen berisi desain, model, dan komponen yang diperlukan untuk menyelesaikan proyek. Setelah spesifikasi, perancangan sistem akan dilakukan pada langkah tambahan.

Pada tahap ini, features dan operasi-operasi pada sistem dideskripsikan secara detail. Aktivitas-aktivitas yang dilakukan adalah:

- 1) Menganalisa interaksi obyek dan fungsi pada sistem.
- 2) Menganalisa data dan membuat skema database.
- 3) Merancang user interface.

Desain berkonsentrasi pada bagaimana sistem dibangun untuk memenuhi kebutuhan pada fase analisis. Manfaat desain sistem adalah memberikan gambaran rancang bangun (*blue print*) yang lengkap, sebagai penuntun (*guideline*) bagi programmer dalam membuat aplikasi

Selain itu ada beberapa hal yang dilakukan dalam desain sistem adalah:

- 1) Pemodelan sistem
- 2) Desain Basis data
- 3) Desain Aplikasi
- 4) Desain Perangkat Keras/Jaringan
- 5) Desain Antar Muka
- 6) Deskripsi Pengguna

Sistem informasi yang terkomputerisasi setidaknya terdiri dari:

- 1) Hardware terdiri dari komponen input, proses, output, dan jaringan
- 2) Software terdiri dari sistem operasi, utilitas, dan aplikasi
- 3) Data mencakup struktur data, keamanan dan integritas data
- 4) Prosedur seperti dokumentasi, prosedur sistem, buku petunjuk operasional dan teknis
- 5) Manusia pihak yang terlibat dalam penggunaan sistem informasi

Buatlah aplikasi berdasarkan rancangan yang telah dibuat. Selain aplikasi, buatlah juga buku panduan penggunaan aplikasi agar mudah saat melakukan training pada saat implementasi. Lakukan testing aplikasi, diantaranya:

- 1) Testing performa
- 2) Testing program logic / sintaks
- 3) Testing implementasi bisnis rules



- 4) Testing faktor manusia
- 5) Testing bisnis proses / prosedur
- 6) Testing efisiensi input
- 7) Testing output.

**d. Implementasi Sistem (*Systems Implementation*)**

Tahap berikutnya adalah implementasi yaitu mengimplementasikan rancangan dari tahap-tahap sebelumnya dan melakukan uji coba. Dalam implementasi, dilakukan aktivitas-aktivitas sebagai berikut:

- 1) Pembuatan database sesuai skema rancangan.
- 2) Pembuatan aplikasi berdasarkan desain sistem.
- 3) Pengujian dan perbaikan aplikasi (debugging).

Sebelum implementasi, lakukanlah persiapan secara matang mengenai perangkat keras, perangkat lunak, ruangan dan fasilitas pendukung lainnya.

Beberapa hal yang juga penting diperhatikan dalam implementasi sistem adalah:

1) Konversi

Biasanya diperlukan konversi dari sistem lama ke sistem baru, apalagi jika sebelumnya juga telah menggunakan aplikasi terkomputerisasi

2) Pelatihan

Pelatihan secara menyeluruh untuk setiap pihak yang menggunakan. Jangan lupa lakukan sosialisasi kepada pihak-pihak yang terlibat dalam sistem namun tidak menggunakan aplikasi sistem secara langsung.

### 3) Testing Penerimaan

Lakukan testing selama periode tertentu sebagai proses belajar.

Ada empat pendekatan dasar untuk menghentikan sistem lama dan beralih ke sistem baru (Raymon Mc. Leod) antara lain:

- 1) Percontohan,  
pemakain sistem lama melihat dulu sebelum memutuskan beralih ke sistem baru
- 2) Serentak,  
sistem lama dihentikan langsung diganti sistem baru
- 3) Bertahap,  
sistem baru digunakan bagian perbagian dan seterusnya hingga seluruh bagian beralih ke sistem baru
- 4) Paralel,  
sistem lama dipertahankan, sementara sistem baru mulai dioperasikan, jika sistem baru telah beroperasi secara penuh dan teruji kebenarannya, maka sistem lama dilepas.

#### e. **Pemeliharaan Sistem (*Systems Maintenance*)**

Dilakukan oleh admin yang ditunjuk untuk menjaga sistem tetap mampu beroperasi secara benar melalui kemampuan sistem dalam mengadaptasikan diri sesuai dengan kebutuhan.

Tahapan pemeliharaan sistem mencakup seluruh proses yang diperlukan untuk menjamin kelangsungan, kelancaran, dan penyempurnaan sistem yang telah dioperasikan.

Beberapa hal yang harus dilakukan:

- 1) Pemantauan pengoperasian
- 2) Antisipasi gangguan kecil (bug)
- 3) Lakukan penyempurnaan
- 4) Antisipasi faktor-faktor luar
- 5) Virus, kerusakan/kehilangan data, atau sistem diakses oleh pihak luar.

#### **4. Metode Pengembangan SDLC**

Model pengembangan *SDLC* penting untuk memudahkan proses pengembangan perangkat lunak. Ada beberapa metode pengembangan *SDLC* yang diadopsi oleh berbagai organisasi, yaitu:

##### **a. Metode *Waterfall***

Metode *waterfall* pertama kali diperkenalkan oleh Windows W. Royce pada tahun 1970. *Waterfall* merupakan model klasik yang sederhana dengan aliran sistem yang biasa disebut linier output atau *sekuensial linier* yaitu dari setiap tahap merupakan input bagi tahap berikutnya (Kristanto, 2004).

Model ini melibatkan penyelesaian satu tahap secara lengkap sebelum melangkah ke tahap berikutnya. Ketika satu tahap selesai langsung dilakukan evaluasi untuk memastikan proyek berjalan sesuai rencana dan layak diteruskan ke tahap berikutnya. *Waterfall Model*, disebut juga model klasik, memiliki beberapa tahap utama, yaitu analisis dan rekayasa sistem, perancangan, penulisan program, pengujian, dan pemeliharaan.

### **Kelebihan Metode *Waterfall Development*.**

- 1) Tahapan proses pengembangannya tetap (pasti), mudah diaplikasikan, dan prosesnya teratur.
- 2) Cocok digunakan untuk produk *software*/program yang sudah jelas kebutuhannya di awal, sehingga minim kesalahannya.
- 3) *Software* yang dikembangkan dengan metode ini biasanya menghasilkan kualitas yang baik.
- 4) Dokumen pengembangan sistem sangat terorganisir, karena setiap fase harus terselesaikan dengan lengkap sebelum melangkah ke fase berikutnya.

### **Kekurangan Metode *Waterfall Development***

- 1) Proyek yang sebenarnya jarang mengikuti alur sekuensial seperti diusulkan, sehingga perubahan yang terjadi dapat menyebabkan hasil yang sudah didapatkan tim pengembang harus diubah kembali/iterasi sering menyebabkan masalah baru.
- 2) Terjadinya pembagian proyek menjadi tahap-tahap yang tidak fleksibel, karena komitmen harus dilakukan pada tahap awal proses.
- 3) Sulit untuk mengalami perubahan kebutuhan yang diinginkan oleh customer/pelanggan.
- 4) Pelanggan harus sabar untuk menanti produk selesai, karena dikerjakan tahap per tahap, dan proses pengerjaannya akan berlanjut ke setiap tahapan bila tahap sebelumnya sudah benar-benar selesai.

- 5) Perubahan ditengah-tengah pengerjaan produk akan membuat bingung tim pengembang yang sedang membuat produk.
- 6) Adanya waktu kosong (menganggur) bagi pengembang, karena harus menunggu anggota tim proyek lainnya menuntaskan pekerjaannya.

**b. Metode *Prototype***

Metode *Prototype* merupakan suatu paradigma baru dalam metode pengembangan *software* atau perangkat lunak dimana metode ini tidak hanya sekedar evolusi dalam dunia pengembangan perangkat lunak, tetapi juga merevolusi metode pengembangan perangkat lunak yang lama yaitu sistem sekuensial yang biasa dikenal dengan nama SDLC atau *waterfall development model*.

Dalam *Model Prototype* perangkat lunak yang dihasilkan kemudian dipresentasikan kepada pelanggan, dan pelanggan tersebut diberikan kesempatan untuk memberikan masukan sehingga perangkat lunak yang dihasilkan nantinya betul-betul sesuai dengan keinginan dan kebutuhan pelanggan.

**Kelebihan Metode *Prototype***

- 1) Pelanggan berpartisipasi aktif dalam pengembangan sistem, sehingga hasil produk pengembangan akan semakin mudah disesuaikan dengan keinginan dan kebutuhan pelanggan.
- 2) Penentuan kebutuhan lebih mudah diwujudkan.

- 3) Mempersingkat waktu pengembangan produk perangkat lunak.
- 4) Adanya komunikasi yang baik antara pengembang dan pelanggan.
- 5) Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan pelanggan.
- 6) Lebih menghemat waktu dalam pengembangan sistem.
- 7) Penerapan menjadi lebih mudah karena pelanggan mengetahui apa yang diharapkannya.

#### **Kekurangan Metode *Prototype***

- 1) Proses analisis dan perancangan terlalu singkat.
- 2) Kurang fleksibel dalam menghadapi perubahan.
- 3) Walaupun pemakai melihat berbagai perbaikan dari setiap versi *prototype*, tetapi pemakai mungkin tidak menyadari bahwa versi tersebut dibuat tanpa memperhatikan kualitas dan pemeliharaan jangka panjang.
- 4) Pengembang kadang-kadang membuat kompromi implementasi dengan menggunakan sistem operasi yang tidak relevan dan algoritma yang tidak efisien.

#### **c. Metode *Rapid Application Development (RAD)***

*Rapid Application Development (RAD)* adalah sebuah model proses perkembangan perangkat lunak sekuensial linier yang menekankan siklus perkembangan yang sangat pendek (kira-kira 60 sampai 90 hari). Model RAD ini merupakan sebuah adaptasi “kecepatan tinggi” dari model sekuensial linier dimana perkembangan cepat

dicapai dengan menggunakan pendekatan konstruksi berbasis komponen.

### **Kelebihan Metode RAD**

- 1) Lebih efektif dari *Model waterfall/sequential linear* dalam menghasilkan sistem yang memenuhi kebutuhan langsung dari pelanggan.
- 2) Cocok untuk proyek yang memerlukan waktu yang singkat.
- 3) Model RAD mengikuti tahap pengembangan sistem seperti pada umumnya, tetapi mempunyai kemampuan untuk menggunakan kembali komponen yang ada sehingga pengembang tidak perlu membuatnya dari awal lagi sehingga waktu pengembangan menjadi lebih singkat dan efisien.

### **Kekurangan Metode RAD:**

- 1) Model RAD menuntut pengembangan dan pelanggan memiliki komitmen di dalam aktivitas rapid-fire yang diperlukan untuk melengkapi sebuah sistem, di dalam kerangka waktu yang sangat diperpendek. Jika komitmen tersebut tidak ada, proyek RAD akan gagal.
- 2) Tidak semua aplikasi sesuai untuk RAD, bila system tidak dapat dimodulkan dengan teratur, pembangunan komponen penting pada RAD akan menjadi sangat bermasalah.
- 3) RAD tidak cocok digunakan untuk sistem yang mempunyai resiko teknik yang tinggi.

- 4) Membutuhkan tenaga kerja yang banyak untuk menyelesaikan sebuah proyek dalam skala besar.
- 5) Jika ada perubahan di tengah-tengah pengerjaan maka harus membuat kontrak baru antara pengembang dan pelanggan.

**d. Metode *Evolutionary Development (Evolutionary Software Process Models)***

Model *Evolutionary Development* bersifat iteratif (mengandung perulangan). Hasil prosesnya berupa produk yang makin lama makin lengkap sampai versi terlengkap dihasilkan sebagai produk akhir dari proses.

Model *Evolutionary Development / Evolutionary Software Process* terbagi menjadi 2 (dua), yaitu:

1) *Model Incremental*

- *Model Incremental* merupakan hasil kombinasi elemen-elemen dari model waterfall yang diaplikasikan secara berulang, atau bisa disebut gabungan dari Model linear sekuensial (waterfall) dengan Model Prototype.
- Elemen-elemen tersebut dikerjakan hingga menghasilkan produk dengan spesifikasi tertentu kemudian proses dimulai dari awal kembali hingga muncul hasil yang spesifikasinya lebih lengkap dari sebelumnya dan tentunya memenuhi kebutuhan pemakai.
- Model ini berfokus pada penyampaian produk operasional dalam setiap pertambahannya.



- Pertambahan awal ada di versi stripped down dari produk akhir, tetapi memberikan kemampuan untuk melayani pemakai dan juga menyediakan platform untuk evaluasi oleh pemakai.
- Model ini cocok dipakai untuk proyek kecil dengan anggota tim yang sedikit dan ketersediaan waktu yang terbatas.

## 2) Model Spiral (*Model Boehm*)

- Model ini mengadaptasi dua model perangkat lunak yang ada yaitu model *prototyping* dengan pengulangannya dan model *waterfall* dengan pengendalian dan sistematikanya.
- Model ini dikenal dengan sebutan Spiral Boehm.
- Pengembang dalam model ini menggabungkan beberapa model umum tersebut untuk menghasilkan produk khusus atau untuk menjawab persoalan-persoalan tertentu selama proses pengerjaan proyek.

### e. Metode Agile

Model Agile merupakan model pengembangan jangka pendek yang memerlukan adaptasi cepat dan pengembangan terhadap perubahan dalam bentuk apapun. Dalam agile terdapat beberapa poin penting diantaranya sebagai berikut:

- 1) Interaksi antar personal lebih penting daripada proses dan alat.
- 2) Software yang berfungsi lebih penting daripada dokumentasi yang lengkap

- 3) Kolaborasi dengan klien lebih penting daripada negoisasi kontrak.
- 4) Sikap tanggap lebih penting daripada mengikuti rencana

#### **Kelebihan Metode Agile**

- 1) Functional dapat dibuat dengan cepat dan dilakukan testing
- 2) Perubahan dengan cepat ditangani

#### **Kelemahan Metode Agile**

- 1) Analisis, desain, dan pengembangan sulit diprediksi
- 2) Dapat memunculkan permasalahan dari arsitektur maupun desain.

#### **f. Metode *Fountain* (Air Mancur)**

Model *Fontain* merupakan perbaikan logis dari model waterfall, langkah langkah dan urutan prosedurnya pun masih sama. Namun pada model *Fountain* ini kita dapat mendahulukan sebuah step ataupun melewati step tersebut, akan tetapi ada yang tidak bisa anda lewati stepnya seperti kita memerlukan design sebelum melakukan coding jika itu di lewati maka akan ada tumpang tindih dalam siklus SDLC.

#### **g. Metode *Synchronize and Stabilize* (SDLC)**

Model ini merupakan model yang digunakan oleh Microsoft. Secara garis besar, *Model Synchronize and Stabilize* ini sama dengan model incremental, tetapi oleh CUsamano dan Selby tahun 1997 menyebutnya sebagai *Synchronize and Stabilized Model*, karena ada beberapa proses manajemen yang ditekannya oleh microsoft.

Analisis kebutuhan dilakukan dengan wawancara dengan sejumlah konsumen yang potensial. Kemudian kebutuhan-kebutuhan tersebut dibuat paket dan disusun daftar secara prioritas. Kemudian spesifikasi ditulis. Selanjutnya pekerjaan dibagi dalam tiga atau empat bagian pembangunan software. Bagian pertama menangani hal-hal yang paling kritis, bagian selanjutnya menangani hal-hal yang kritis selanjutnya, dan seterusnya.

### **Kelebihan Metode *Synchronize and Stabilize* (SDLC)**

- 1) Membagi produk yang besar ke dalam bagian-bagian yang lebih kecil (prioritas dari fitur produk yang memiliki tim fitur kecil dapat dibuat dalam beberapa bulan)
- 2) Membuat project bekerja secara sistematis meskipun mereka tidak dapat menggambarkan dan menyelesaikan suatu produk di awal project.
- 3) Mengijinkan tim besar bekerja menjadi tim yang lebih kecil dengan membagi sebuah tim menjadi beberapa bagian, bekerja secara paralel tetapi tetap dapat berkesinambungan dalam men *synchronizing* setiap perubahan, *stabilizing* produk dan menemukan serta memperbaiki kesalahan.
- 4) Memfasilitasi masukan dari customer, fitur produk dan waktu pengembangan yang pendek, yang didukung oleh mekanisme masukan customer, prioritas, menyelesaikan dahulu bagian yang sangat penting dan melakukan perubahan tanpa harus mengurangi fitur yang diperlukan.

#### **h. Metode *Rational Unified Process* (RUP)**

Unified Process (UP) adalah metodologi pengembangan sistem berbasis objek. Metode ini sudah menjadi salah satu metode yang banyak digunakan dalam pengembangan sistem berorientasi objek. UP memperkenalkan pendekatan baru untuk siklus hidup pengembangan sistem yang menggabungkan perulangan (iterations) dan tahapan (phases) yang disebut dengan siklus hidup UP (UP life cycle). UP mendefinisikan empat tahapan siklus hidup yaitu inception, elaboration, construction, dan transition.

##### **Kelebihan Metode RUP:**

- 1) Menyediakan akses yang mudah terhadap pengetahuan dasar bagi anggota tim.
- 2) Menyediakan petunjuk bagaimana menggunakan UML secara efektif.
- 3) Mendukung proses pengulangan dalam pengembangan software
- 4) Memungkinkan adanya penambahan-penambahan pada proses.
- 5) Memungkinkan untuk secara sistematis mengontrol perubahan- perubahan yang terjadi pada software selama proses pengembangannya.
- 6) Memungkinkan untuk menjalankan test case dengan menggunakan Rational TestManager Tool.

### **Kekurangan Model RUP:**

- 1) Metode ini hanya dapat digunakan pada pengembangan perangkat lunak yang berorientasi objek dengan berfokus pada UML (*Unified Modeling Language*)

#### **i. Metode *Build & Fix***

*Build & Fix Method* merupakan metode yang paling lemah diantara metode SDLC yang lain tetapi menjadi acuan pengembangan untuk metode SDLC yang lain. *Build & fix* bertujuan untuk memberikan kepercayaan terhadap pelanggan dengan cara memberikan pelayanan perbaikan dan perawatan secara terus menerus terhadap produk yang digunakan oleh user.

#### **Kelebihan *Build & Fix Method*:**

- 1) Build and fix dibuat tanpa melalui tahapan analisis dulu

#### **Kekurangan *Build & Fix Method***

- 1) Tidak cocok ketika di pakai untuk membuat produk dengan kompleksitas tinggi dan dengan ukuran yang besar.
- 2) Biaya yang di butuhkan akan menjadi sangat membengkak dan membesar ketika build and fix di gunakan untuk membuat proyek berskala besar

#### **j. Metode Pengembangan *Extreme Programming***

*Extreme Programming* (XP) merupakan suatu pendekatan yang paling banyak digunakan untuk pengembangan perangkat lunak cepat. Pressman, (2012:88). Alasan menggunakan metode *Extreme Programming* (XP) karena sifat dari aplikasi yang di

kembangkan dengan cepat melalui tahapan-tahapan sebagai berikut:

- 1) *Planning* (Perencanaan),
- 2) *Design* (Perancangan),
- 3) *Coding* (Pengkodean) dan
- 4) *Testing* (Pengujian)

**k. Metode SDLC Big Bang.**

SDLC Big Bang Model adalah dimana tidak mengikuti proses tertentu. Perkembangan hanya dimulai dengan uang dan usaha yang dibutuhkan sebagai masukan, dan hasilnya adalah perangkat lunak yang dikembangkan sesuai dengan kebutuhan pelanggan.

Model Big Bang ini tidak mengikuti dan hanya ada sedikit perencanaan yang diperlukan. Bahkan pelanggan pun tidak yakin dengan apa yang sebenarnya dia inginkan dan persyaratannya diimplementasikan dengan cepat tanpa banyak analisis.

**Keuntungan Metode Big Bang antara lain:**

- 1) Model yang sangat sederhana
- 2) Sedikit atau tidak ada perencanaan yang dibutuhkan
- 3) Mudah untuk dikelola
- 4) Sangat sedikit sumber daya yang dibutuhkan
- 5) Memberikan fleksibilitas kepada pengembang
- 6) Bagus untuk developer yang ingin belajar atau developer pendatang baru.

### **Kekurangan Metode *Big Bang* antara lain:**

- 1) Beresiko tinggi dan kepastian dari requirement yang tidak jelas
- 2) Tidak cocok untuk project skala besar dan berorientasi objek
- 3) Model yang buruk untuk proyek yang panjang dan sedang berlangsung.
- 4) Bisa berubah menjadi sangat mahal jika persyaratan disalahpahami

### **I. Metode the V**

The V-Model merupakan model SDLC dimana proses yang terjadi secara berurutan dalam bentuk-V. Dikenal juga sebagai model *Verifikasi* dan *Validasi*. The V-Model merupakan perluasan dari *waterfall* model dan didasarkan pada asosiasi dari fase pengujian untuk setiap tahap pengembangan yang sesuai.

Ini berarti bahwa untuk setiap fase tunggal dalam siklus pengembangan, ada tahap pengujian terkait langsung. Ini adalah model yang sangat disiplin dan tahap berikutnya dimulai setelah selesainya tahap sebelumnya.

#### **Kelebihan dari *V-Model* SDLC**

- 1) Model ini yang sangat-disiplin dan tahapan selesai satu per satu.
- 2) Untuk proyek-proyek yang lebih kecil dimana persyaratan dipahami dengan baik akan bekerja lebih baik
- 3) Sederhana, mudah dimengerti dan digunakan.

- 4) Karena setiap fase memiliki spesifik kiriman dan proses review sehingga mudah dikelola

### **Kekurangan dari V-Model SDLC**

- 1) Berisiko tinggi dan ketidakpastian.
- 2) Tidak cocok untuk proyek-proyek yang kompleks dan berorientasi objek.
- 3) Tidak cocok untuk proyek-proyek dimana persyaratan berisiko tinggi
- 4) Tidak cocok untuk proyek-proyek yang lama dan berkelanjutan.
- 5) Setelah aplikasi dalam tahap pengujian, sulit untuk kembali dan mengubah fungsionalitas.

### **C. Latihan**

1. Apa yang saudara mengenai siklus hidup sistem informasi? Uraikan secara singkat dan jelas!
2. Dalam siklus hidup sistem informasi terdapat beberapa fungsi. Sebutkan dan jelaskan secara jelas?
3. Dalam tahapan siklus hidup sistem informasi terdapat tahap perencanaan. Kegiatan apa saja yang terdapat pada tahap perencanaan? Sebutkan dan jelaskan secara singkat dan tepat?
4. Dimensi kelayakan apa saja yang terdapat pada siklus hidup sistem informasi tahap analisis?
5. Sebutkan dan jelaskan kelebihan dari model Model Prototype yang terdapat pada model pengembangan sistem informasi!



## D. Referensi.

- Azhar Susanto, 2002, Sistem Informasi Manajemen: Konsep dan Pengembangannya, Lingga Jaya, Bandung
- Cegielski, R. P. (2015). Introduction to Information System. John Wiley and Sons.
- George M. Scott. 2002. Prinsip-prinsip Sistem Informasi Manajemen. Jakarta : Raja Grafindo Persada
- Ismail, M. (2004). Konsep Sistem Informasi Manajemen. Kertas Kerja. Fakultas Ekonomi. Universitas Sumatera Utara. Medan.
- Kenneth C. Laudon, 2012, *Sistem Informasi Manajemen*, Jakarta , Salemba Empat
- Laudon, Kenneth C, dan Jane P. Loudon, 2014, *Management Informaton System: Managing the Digital Firm, 13<sup>th</sup> edition*, Upper Sadle River, New Jersey: Prentice Hall
- O'Brien, J.A, 2005. *Introduction To Information Systems*, 12<sup>th</sup> ed. McGrawHill Companies, Inc.America.
- O'Brien, J.A, 2006, *Pengantar Sistem Informasi*, Jakarta, Salemba Empat
- O' Brien, J.A., & Markas, G. (2011). *Management Information System* (Vol. 10th).
- Porter. M.E. 1985. *Competitive Advantage: Creating and Sustaining Superior Performance*. Free Press, New York
- Raymond McLeod, Jr.George P. Schell 2012. *Sistem Informasi Manajemen*. Jakarta, Salemba Empat