



## Pertemuan 5 (Sesi 5)

CCA220/ANALISA DAN PERANCANGAN SISTEM  
INFORMASI

Oleh :

5165 –Kundang K Juman  
Prodi : Teknik Informatika

[www.esaunggul.ac.id](http://www.esaunggul.ac.id)

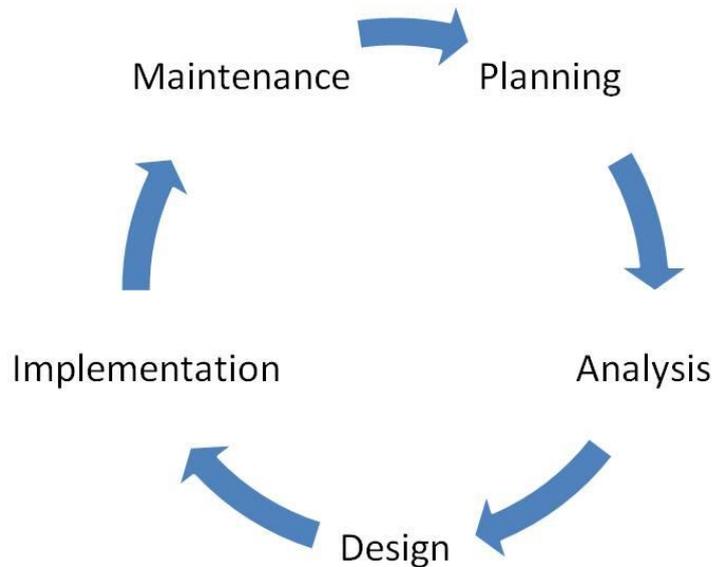
## Methode Analisa system Informasi

Methode Analisa system :

### **Analisis Sistem**

Tahapan dari SDLC(**Sistem Development Life Cycle**) Perencanaan Sistem (Systems Planning yang pertama adalah melakukan analisis terhadap sistem. Analisis sistem merupakan tahap awal dalam sebuah siklus SDLC, dimana analisis sistem akan melakukan berbagai macam analisis terhadap sebuah sistem yang sudah ada, dan bagaimana nantinya sebuah sistem akan berjalan. Hal ini termasuk di dalamnya adalah sebagai bentuk 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 masuk ke dalam tahapan spesifikasi kebutuhan sistem.

Siklus hidup pengembangan sistem ( System development life cycle / SDLC ) adalah tahapan aktivitas yang harus dikerjakan oleh pengembang sistem untuk menghasilkan sebuah sistem yang dapat dioperasikan pada organisasi pemakai sistem.



Tahapan dalam SDLC :

Tahapan-tahapan yang ada pada SDLC secara umum adalah sebagai berikut:

**1. Inisiasi (initiation)**

Tahap ini biasanya ditandai dengan pembuatan proposal proyek perangkat lunak.

**2. Pengembangan konsep sistem (system concept development)**

mendefinisikan lingkup konsep termasuk dokumentasi pengembangan sistem pada analisis lingkup area sistem dan manajemen rencana dan mempelajari cara kerja sistem.

**3. Perencanaan (planning)**

Membuat perencanaan terhadap pekerjaan dan berkas perencanaan yang lainnya. Menyediakan dasar untuk mendapatkan sumber daya (resources) yang dibutuhkan untuk memperoleh solusi.

#### **4. Analisis kebutuhan (requirements analysis)**

Analisa terhadap kebutuhan pengguna sistem perangkat lunak (user) dan mengembangkan kebutuhan user. Membuat dokumen kebutuhan fungsional.

#### **5. Desain (design)**

Mentransformasikan kebutuhan secara terinci, dokumen desain sistem focus pada bagaimana dapat memenuhi fungsi-fungsi yang dibutuhkan oleh sistem.

#### **6. Pengembangan (development)**

Mengubah perancangan ke sistem informasi yang kompleks dan bagaimana mendapatkan dan melakukan penginstalan lingkungan sistem yang diharapkan; membuat basis data dan menyiapkan standar prosedur pada saat pengujian, menyiapkan dokumen atau file coding, testing, compile, repair dan cleaning program.

#### **7. Integrasi dan pengujian (integration and test)**

mepresentasikan sistem perangkat lunak yang telah memenuhi keadaan yang dispesifikasikan pada dokumen kebutuhan fungsional. Dengan diarahkan oleh bagian penjamin mutu (quality assurance) dan user. Menghasilkan pelaporan analisis dari pengujian.

#### **8. Implementasi (implementation)**

Termasuk pada persiapan implementasi, pelaksanaan perangkat lunak pada area produksi (area pada user) dan menjalankan resolusi dari problem yang terdeteksi dari tahap integrasi dan pengujian.

#### **9. Operasi dan pemeliharaan (operations and maintenance)**

Menjelaskan tentang pekerjaan untuk menjalankan dan maintenance sistem informasi pada area Produksi (lingkungan pada user), termasuk implementasi akhir dan masuk pada proses peninjauan.

#### **10. Disposisi (Disposition)**

Mendeskripsikan aktifitas dari pengembangan sistem dan membangun data yang sesungguhnya sesuai dengan aktifitas user.

## **Spesifikasi Kebutuhan Sistem**

Tahapan kedua pada siklus SDLC adalah tahapan spesifikasi dari kebutuhan sistem. Pada tahap ini, seluruh hasil analisis yang dilakukan pada tahap pertama akan dikaji lebih mendalam untuk mendapatkan spesifikasi yang dibutuhkan dalam pengembangan sebuah sistem. Hal ini nantinya akan sangat bermanfaat terutama apabila kita ingin membangun sebuah sistem yang sangat besar. Dengan adanya spesifikasi kebutuhan sistem, maka setiap teknisi dan juga pengembang akan mampu membuat sebuah sistem yang sesuai dengan kebutuhan yang ada, serta mampu untuk menjalankan sistem tersebut sesuai dengan peruntukannya, dan tentu saja dapat berjalan pada spesifikasi tertentu)

## **Perancangan Sistem**

Tahapan berikutnya dari siklus SDLC pada sebuah sistem adalah perancangan terhadap sistem. Ini merupakan tahapan kelanjutan dari spesifikasi kebutuhan sistem. Tahap ini merupakan tahap dimana seluruh hasil analisa dan juga hasil pembahasan mengenai spesifikasi sistem diterapkan menjadi sebuah rancangan atau cetak biru dari sebuah sistem. Tahap perancangan sistem ini bisa kita sebut sebagai cetak biru, atau bias juga kita sebut sebagai prototype, dimana sistem ini sudah siap untuk dikembangkan. Ibarat sebuah rumah atau gedung, maka perancangan sistem ini merupakan desain dari rumah yang dibuat oleh seorang arsitek. Pada tahap ini, semua persiapan harus dilakukan dengan matang, mulai dari implementasi dari spesifikasi sistem, dan semua analisis terhadap sistem, hingga berbagai macam tenaga pendukung dari sistem yang akan dikembangkan nantinya

## **Pengembangan Sistem**

Tahap berikutnya dari tahapan SDLC ini adalah tahapan pengembangan sistem. Tahapan pengembangan sistem ini merupakan tahapan dimana rancangan atau cetak biru sistem ini mulai dikerjakan dan dibuat atau diimplementasikan menjadi sebuah sistem yang utuh, dan dapat digunakan. Apabila dianalogikan dengan pembangunan gedung, maka tahap ini merupakan tahap dimana gedung atau rumah mulai dibangun, mulai dari pembuatan pondasi, hingga penempatan besi baja ataupun alat konstruksi lainnya. Tahap ini merupakan tahapan yang cukup lama, karena dalam prakteknya tahap pengembangan sistem ini bisa saja menemui 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.

## **Pengujian Sistem**

Setelah sistem selesai dikembangkan dan juga dibuat, maka 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 memang 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 tahap 1 hingga tahap 4 harus diperbaharui, diulangi, atau bahkan bisa saja mengalami perombakan total.

## **Implementasi dan Pemeliharaan Sistem**

Tahap ini bisa dibilang sebagai 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 dan juga optimal. Ketika tahapan sebelumnya sudah berhasil dilewati, maka ini lah saatnya sistem tersebut mulai diimplementasikan dan digunakan secara real oleh user yang membutuhkan. Dalam prakteknya, tahap terakhir ini tidak hanya berhenti pada proses implementasi dan juga penginstallan saja, namun juga melakukan proses pemeliharaan terhadap sistem yang ada, sehingga dapat menjamin bahwa sistem tersebut akan tetap berfungsi secara normal dan juga optimal setiap saat.

### **Metodologi pengembangan sistem terstruktur**

- Dipandang dari cara menentukan kebutuhan dari sistem :  
Pendekatan Bawah Naik (Bottom Up Approach) : dimulai dari bawah yaitu dari ketersediaan data naik sampai ke informasi yang dibutuhkan oleh pemakai.  
Pendekatan Atas Turun (Top Down) : yaitu dimulai dari kebutuhan informasi pemakai dan turun sampai ke data untuk memenuhi kebutuhan ini.
- Dipandang dari Cara Mengembangkannya :  
Pendekatan Sistem Menyeluruh

Pendekatan Modular : memecah sistem yang rumit menjadi beberapa bagian sistem yg disebut modul2 yg lebih sederhana.

Jenis perangkat / tools pemodelan antara lain :

1. Diagram Arus Data (DFD)  
Menunjukkan proses yang dijalankan data dalam sistem
2. Kamus Data  
Definisi elemen data dalam sistem
3. Entity Relationship Diagram (ERD)  
Model penyimpanan data dalam DFD
4. State Transition Diagram (STD)  
Menunjukkan keadaan tertentu dimana suatu sistem dapat ada dan transisi yang menghasilkan keadaan tertentu yang baru. STD digunakan untuk sistem yang real time.
5. Bagan Struktur  
Menggambarkan suatu hierarki modul program perangkat lunak termasuk dokumentasi interface antar modul
6. Diagram Alur Program Terstruktur (Structured Program Flowchart)  
Menggambarkan alur dan logika program
7. Alat Spesifikasi Proses  
Memberikan deskripsi yang lengkap tentang proses-proses yang ditemukan dalam diagram alur data tingkat dasar.

Contoh :

- Bahasa Inggris Terstruktur
- Tabel Keputusan
- Pohon Keputusan

## Tahapan SDLC yang Perlu di Ketahui

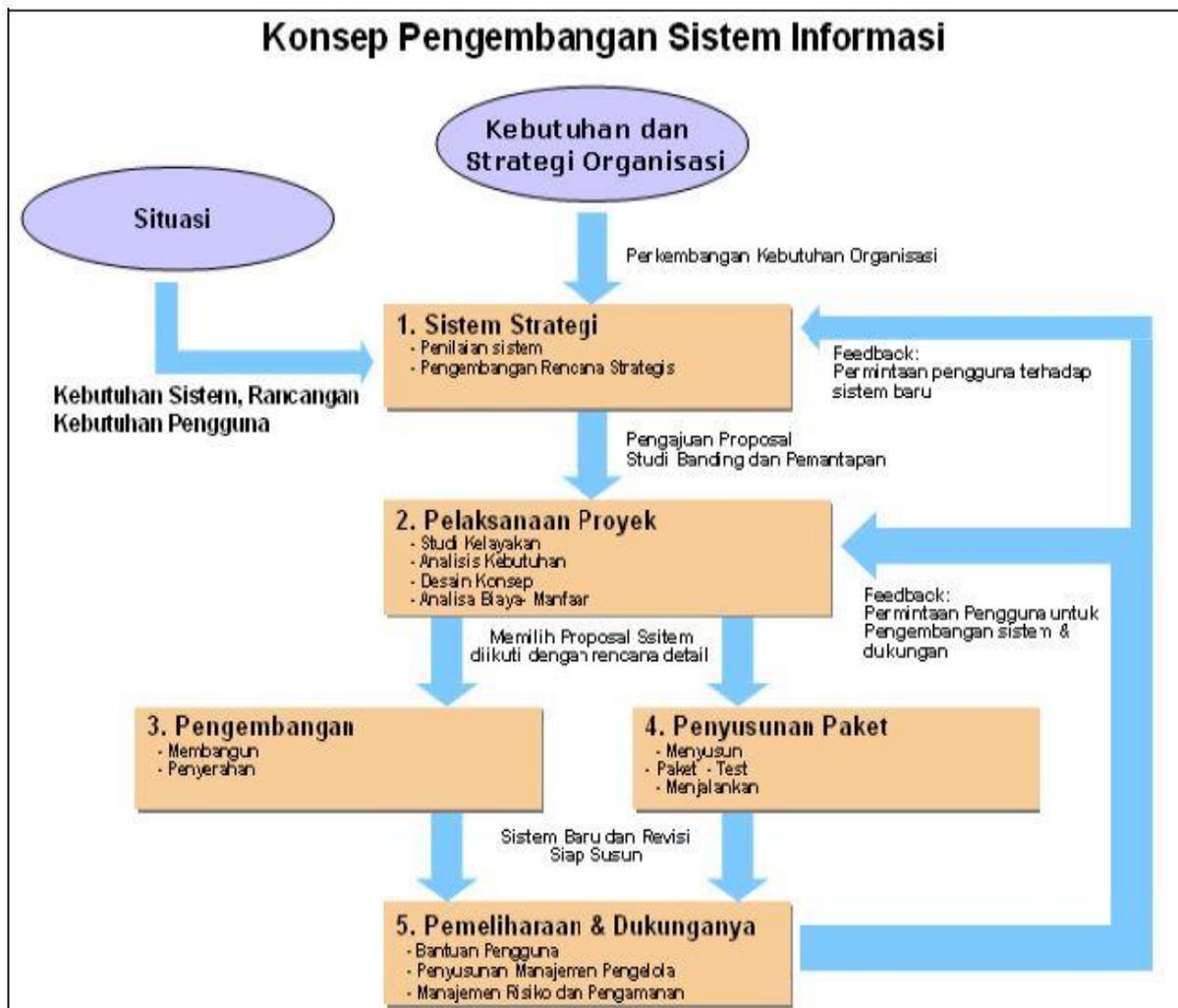
1. Melakukan survey, dan melakukan penilaian terhadap kelayakan pengembangan sistem yang ada
2. Melakukan proses analisis dan juga mempelajari sistem yang sudah ada dan sedang berjalan
3. Mencari solusi dan juga pemecahan masalah yang terbaik dalam pengembangan sistem
4. Menentukan penggunaan hardware atau perangkat keras dan juga software atau perangkat lunak yang akan digunakan
5. Melakukan proses perancangan terhadap sistem yang baru
6. Melakukan proses pembangunan dan juga pengembangan sistem yang baru

### METODOLOGI PENGEMBANGAN SOFTWARE

Software Process adalah Pendekatan sistematis yang digunakan dalam rekayasa perangkat lunak serta merupakan sebuah aktifitas terurut yang menuju kepada produksi dari pembuatan produk rekayasa perangkat lunak.(Ian Somovile)

Software process adalah serangkaian langkah-langkah yang terurut untuk membuat sebuah produk rekayasa perangkat lunak. Software process adalah suatu hirarki yang tiap langkah dapat memiliki sub-sub langkah dalam proses rekayasa perangkat lunak(M Deploy)

**Software Development Life Cycle (SDLC)** adalah proses yang terdiri dari serangkaian kegiatan yang direncanakan untuk mengembangkan atau mengubah produk perangkat lunak. SDLC digunakan oleh industri perangkat lunak (*software*) untuk merancang, mengembangkan dan menguji serta bertujuan untuk menghasilkan perangkat lunak berkualitas tinggi yang memenuhi harapan *User/Customer* dalam waktu dan perkiraan biaya yang telah ditentukan. Standar Internasional untuk proses siklus hidup perangkat lunak adalah ISO / IEC 12207 yang mendefinisikan semua tugas yang dibutuhkan untuk mengembangkan dan memelihara perangkat lunak.



## 1. Perencanaan dan Kebutuhan Analisis

Tahap ini merupakan tahapan yang paling penting dan mendasar dalam SDLC. Hal ini dilakukan oleh anggota senior tim dengan input dari User, Divisi Product, Divisi Sales dan Divisi Marketing. Informasi ini kemudian digunakan untuk merencanakan pendekatan *baseline project* dan untuk melakukan studi kelayakan produk dalam bidang ekonomi, operasional, dan teknis.

Perencanaan untuk kualitas dan identifikasi risiko yang terkait dalam SDLC juga dilakukan dalam tahap perencanaan. Hasil dari studi kelayakan teknis digunakan untuk menentukan berbagai pendekatan yang dapat diterapkan untuk kesuksesan proyek tersebut dengan risiko yang minimal.

## **2. Mendefinisikan Requirements**

Setelah tahap *Defining Requirements* selesai dilakukan, maka langkah berikutnya adalah mendefinisikan dan mendokumentasikan *product requirement* secara jelas dan harus mendapatkan *approval* dari Customer atau Market Analyst. Hal ini dilakukan melalui *Software Requirement Specification (SRS)* yang terdiri dari semua *Product Requirement* yang harus didisain dibuat selama siklus pengembangan project.

## **3. Merancang Arsitektur Produk**

*Software Requirement Specification (SRS)* adalah referensi untuk pengembangan arsitektur produk . Berdasarkan persyaratan yang ditentukan dalam SRS, biasanya lebih dari satu pendekatan desain yang diusulkan untuk arsitektur produk didokumentasikan dalam *Design Document Specification (DDS)*.

## **4. Mengembangkan Produk**

Pada tahap ini, pengembangan yang sebenarnya dimulai dan produk mulai dibangun. Kode pemrograman yang dihasilkan sesuai dengan DDS. Jika desain dilakukan dengan cara yang rinci dan terorganisir, pembuatan kode dapat dicapai tanpa banyak kesulitan.

Pengembang harus mengikuti pedoman pengkodean yang didefinisikan oleh organisasi dan *tool* pemrograman mereka seperti *compiler*, *interpreter*, *debugger*, dan lain sebagainya, yang digunakan untuk menghasilkan kode. Bahasa pemrograman tingkat tinggi yang berbeda seperti C, C ++, Pascal, Java, dan PHP digunakan untuk *coding*.

## **5. Pengujian Produk**

Tahap ini biasanya bagian dari semua tahapan pada SDLC. Kegiatan pengujian sebagian besar terlibat dalam semua tahapan SDLC. Namun tahap ini mengacu pada pengujian produk yang rusak atau cacat, dilacak, tetap diuji ulang, sampai produk mencapai standar kualitas yang ditetapkan di SRS.

## **6. Market Deployment and Maintenance**

Setelah produk diuji dan siap, maka produk akan di-*release* secara resmi di-*market* yang tepat. Waktu penyebaran produk terjadi secara bertahap sesuai dengan organisasi dan strategi

bisnis. Produk ini akan dirilis dalam segmen terbatas dan diuji dalam lingkungan bisnis yang *real* sebagai *User Acceptance Test (UAT)*

Kemudian berdasarkan *feedback*, produk dapat di-*release* kembali dengan perangkat tambahan yang disarankan dan ditargetkan di segmen pasar. Setelah produk dirilis di pasar, *maintenance* dilakukan untuk pelanggan yang sudah ada.

### Bagaimana memilih metodologi pengembangan sistem yang tepat?

Pertanyaannya kemudian, metodologi apa yang sebaiknya digunakan? Memilih sebuah metodologi bukanlah hal yang mudah dilakukan karena tidak satupun metodologi yang bisa dikatakan terbaik. Setiap organisasi biasanya memiliki standarisasi tertentu. Memilih sebuah metodologi bukanlah hal yang mudah dilakukan karena tidak satupun metodologi yang bisa dikatakan terbaik. Setiap organisasi biasanya memiliki standarisasi tertentu. Sebenarnya banyak hal yang bisa dijadikan pertimbangan dalam pemilihan sebuah metodologi. Beberapa pertimbangan pemilihan metodologi meliputi; kejelasan kebutuhan pengguna (*clarity user requirement*), penguasaan teknologi (*familiarity with technology*), tingkat kerumitan sistem (*system complexity*), tingkat kehandalan sistem (*system reliability*), waktu pelaksanaan (*short time schedules*) dan visibilitas jadwal pelaksanaan (*schedule visibility*)



- **Kejelasan kebutuhan pengguna**

Jika pada suatu saat kita dihadapkan pada kondisi ketidakjelasan kebutuhan pengguna, maka metodologi RAD berbasis prototipe dan prototipe sekali pakai (throwaway prototyping) merupakan salah satu metodologi yang tepat untuk digunakan.

- **Penguasaan teknologi**

Penguasaan teknologi merupakan satu bagian yang vital untuk dipertimbangkan dalam menentukan sebuah metodologi. Familiaritas terhadap teknologi dasar yang tidak memadai akan menimbulkan pembengkakan waktu dan biaya.

- **Tingkat kerumitan sistem yang akan dibangun**

Sistem yang kompleks membutuhkan analisis dan desain yang sangat hati-hati. Oleh karena itu metodologi agile dan prototyping dipandang kurang begitu baik diterapkan jika tingkat kerumitan sistem sangat tinggi.

- **Tingkat kehandalan sistem**

Kehandalan sistem biasanya merupakan faktor penting dalam pengembangan sistem. Metodologi berbasis prototipe umumnya bukan pilihan yang baik karena mereka kurang berhati-hati tahap analisis dan desain.

- **Waktu pelaksanaan pengembangan**

Metodologi berbasis RAD cocok untuk proyek-proyek dengan jadwal waktu singkat yang membutuhkan kecepatan *deliverables*. metodologi berbasis *waterfall* adalah pilihan terburuk ketika waktu adalah penting karena tidak memungkinkan untuk memudahkan perubahan jadwal.

- **Visibilitas jadwal pelaksanaan**

Metodologi berbasis RAD banyak bergerak dari keputusan penting sehingga metodologi ini paling cocok diterapkan jika manager proyek mengenali dan memberikan perhatian lebih bagi tahapan yang mempunyai faktor resiko dan ekspektasi yang tinggi.

Referensi:

- Systems Analysis and Design with UML Version 2.0: An Object-Oriented Approach, 2nd Edition, Alan Dennis, Barbara Wixom, and David Tegarden, 2005  
Kendal and Kendal ,design analysis