BAB II

TINJAUAN PUSTAKA

2.1 Sejarah Perusahaan

Pada tahun 2000, beberapa orang sarjana lulusan ITB dari berbagai disiplin ilmu membentuk suatu studio kerja di Bandung. Perjalanan mereka diawali dengan kegiatan-kegiatan yang bersifat subkontrak yang berujung pada suatu kerjasama dengan beberapa kelompok ahli, serta memperoleh kepercayaan untuk mengerjakan proyek-proyek, baik dari pemerintah maupun swasta. Berbekal kepercayaan serta beberapa pengalaman tersebut, dibentuk PT. INOCREA MEDIA PERSADA.

PT. INOCREA MEDIA PERSADA merupakan suatu perusahaan jasa konsultan yang mendudukkan diri sebagai mediator antar *stakeholders* dalam merencanakan dan melakukan kegiatan pembangunan.

Seiring perkembangan ilmu pengetahuan dan teknologi yang telah membawa dinamika pembangunan yang semakin kompleks, termasuk didalamnya adalah masyarakat serta pergeseran-pergeseran paradigma baru pembangunan, maka PT. INOCREA MEDIA PERSADA senantiasa terus berusaha untuk memperluas bidang keahlian dengan mendalami isu-isu tentang:

> *Pengembangan Sumber Daya Manusia, Pengelolaan Lingkungan, Good Governance, Desentralisasi Pembangunan, Pemberdayaan Masyarakat, Pengembangan Usaha Kecil dan Menengah, Manajemen Perkotaan, Monitoring dan Evaluasi, dan lain-lain.*

PT. INOCREA MEDIA PERSADA terus berupaya untuk melengkapi berbagai *tools,* baik perangkat lunak, perangkat keras maupun SDM sehingga dapat diandalkan dalam memberikan pelayanan jasa konsultasi yang dibutuhkan.

2.2 VISI, MISI DAN STRATEGI PT. INOCREA MEDIA PERSADA

2.2.1 VISI PT. INOCREA MEDIA PERSADA

Menjadi Konsultan Nasional yang turut serta dalam membangun negara dan masyarakat dengan profesionalisme dan dedikasi yang tinggi dalam kerangka bersama mewujudkan kesejahteraan bangsa.

2.2.2 MISI PT. INOCREA MEDIA PERSADA

Memberikan pelayanan jasa konsultasi pembangunan untuk menciptakan hasil yang tepat dan berdayaguna dengan prinsip mekanisme aspiratif, partisipatif, dan komprehensif.

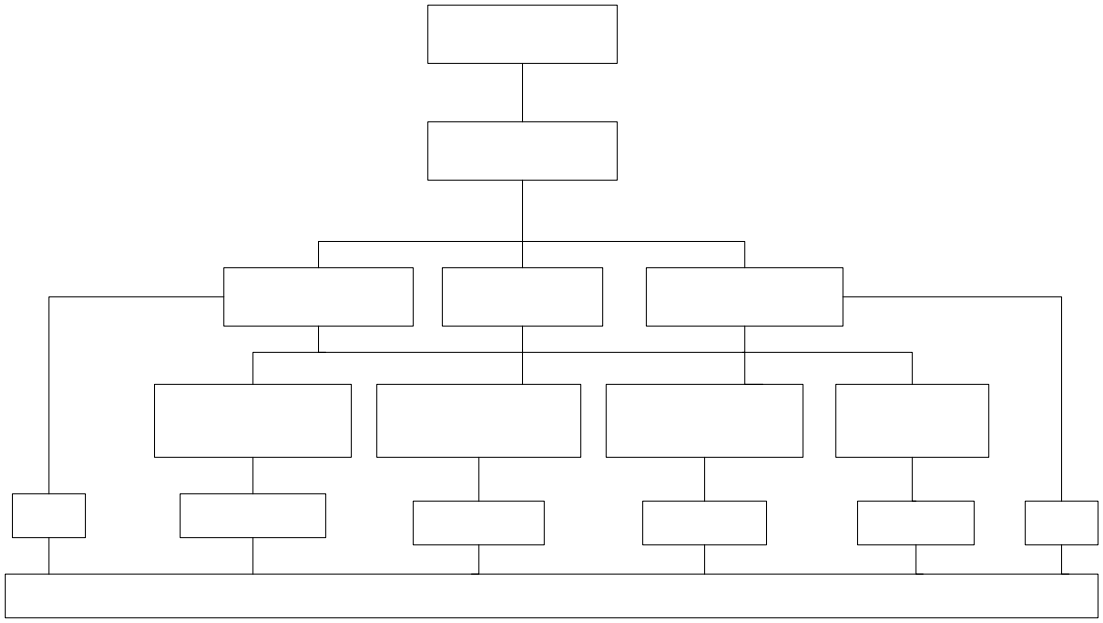
2.2.3 STRATEGI PT. INOCREA MEDIA PERSADA

1. Memberikan bantuan teknis kepada elemen pengguna di bidang penciptaan lingkunagan binaan yang nyaman dan berwawasan lingkungan.
2. Membangkitkan dan mengembangkan kemampuan internal elemen pengguna dalam mengenali dan menyelesaikan permasalahan.

3. Memberikan pemikiran-pemikiran, teknik-teknik, dan pengalaman-pengalaman praktis kepada para elemen pengguna.
4. Menerapkan suatu manajemen kualitas terpadu tehadap setiap jasa layanan konsultasi yang diberikan berorientasi kepada :

   customer- focused product, process approach, system approach management, continual improvement, dan *mutually beneficial partner relationship.*

2.3 STRUKTUR ORGANISASI PT. INOCREA MEDIA PERSADA



Gambar 2.1 Strukur Organisasi PT. INOCREA MEDIA PERSADA

2.4 Lingkup Layanan

Bidang jasa	Layanan konsultasi
Perencanaan Wilayah dan Kota	5. Perencanaan Umum 6. Perencanaan Khusus
Pengembangan Masyarakat	7. Perencanaan Sosial 8. Manajemen SDM 9. Pendampingan Masyarakat
Detail Desain & Supervisi	10. Detail Engineering Desaign 11. Supervisi Pengawasan
Survey & Pemetaan	12. Survei Topografi 13. Pemetaan Wilayah/Kawasan 14. Sistem Informasi Geografis 15. SFAIS & Pengolahan Citra
Manajemen & Pelatihan	16. Pelatihan Pengembangan SDM 17. PBPM

Tabel 2.1 SFAIS : Small Format Aerial Imaging System
PBPM : Pembangunan Bertumpu Pada Masyarakat

2.5 DIVISI PERUSAHAAN

Sebagai suatu perusahaan jasa konsultasi, PT. INOCREA MEDIA PERSADA telah didukung oleh sumberdaya manusia berpengalaman dan profesioanal di bidangnya yang telah memberikan jasa layanan dengan kualitas dan efesiensi kerja yang tinggi, baik secara perorangan maupun kelompok. Hal tersebut yang menjadikan visi dan idealisme profesional PT. INOCREA MEDIA PERSADA sebagai suatu aset pengalaman (track record) perusahaan yang berharga yang mendasari pembentukan badan usaha PT. INOCREA MEDIA PERSADA yang terbagi dalam berbagai divisi pengembangan.

2.5.1 DIVISI ENGINEERING DESIGN

Memberikan layanan jasa konsultasi di bidang :

1. Studi kelayakan program pembangunan.
2. SID (*Survey,Ivestigation,and Design*).
3. Perencanaan teknis dan DED (*Detail Engineering Desain*).

2.5.2 DIVISI STUDI & PERENCANAAN

Memberikan layanan jasa konsultasi di bidang :

1. Perencanaan Program Pembangunan.
2. Manajemen Pelaksanaan Program Pembangunan.
3. Studi dan Rencana Pengembangan Kelembagaan dan Pengembangan Program.
4. Perencanaan Tata Ruang Wilayah dan Kota.
5. Perencanaan Pengembangan Perumahan dan Permukiman.
6. Studi dan Rencana Pengembangan Kelautan, Pesisir dan Pulau-pulau. Kecil.
7. Studi dan Rencana Pengembangan Kawasan Pertanian dan Agroindustri.
8. Studi dan Rencana Pengembangan Kawasan Pariwisata.

2.5.3 DIVISI PENGEMBANGAN MASYARAKAT

Memberikan layanan jasa konsultasi di bidang:

1. Survey Partisipatif.

2. Perencanaan Partisipatif.
3. Manajemen Sumberdaya Partisipatif.
4. Pendampingan & Supervisi.

2.5.4 DIVISI SURVEY & PEMETAAN

Memberikan layanan jasa konsultasi di bidang :

1. Survey Teknik-Topografi & Bathimetri.
2. Pemetaan Kawasan.
3. Small Format Aeril Imaging System (SFAIS)/Foto Udara Media Format Kecil.
4. Pengembangan Sistem Informasi Geografis (SIG).

2.6 Pengertian Perangkat Lunak

Perangkat lunak adalah program komputer yang berkaitan seperti dokumen kebutuhan, rancangan, dan user manual.

Produk perangkat lunak bisa dibangun untuk pengguna umum atau khusus :

- Generic : dibangun untuk dijual ke pengguna yang berbeda-beda, misalnya perangkat lunak untuk PC seperti Excel atau World.
- Bespoke (*custom*) : untuk pengguna khusus/pemesan sesuai kebutuhannya.

Perangkat lunak bisa dibuat dengan membangun program baru, konfigurasi sistem perangkat lunak atau penggunaan kembali (*reuse*) program yang sudah ada.

2.7 Pengertian Rekayasa Perangkat Lunak (RPL)

Rekayasa perangkat lunak adalah disiplin ilmu atau teknik yang berkaitan dengan semua aspek dalam membuat perangkat lunak, dimana rekayasa perangkat lunak harus mengikuti pendekatan yang sistematis dan teratur dan menggunakan alat dan teknik yang cocok atau sesuai dengan masalah yang akan dipecahkan, batasan pembangunan dan *resource* (sumber) yang tersedia.

2.8 Pengertian Proses Perangkat Lunak

Software process adalah serangkaian aktivitas yang tujuannya adalah untuk pembangunan dan evolusi perangkat lunak.

Aktivitas umum dalam semua proses perangkat lunak adalah :

- Spesifikasi : apa yang dilakukan sistem dan batasan pembangunan.
- Pembangunan : produksi dari sistem perangkat lunak.
- Validasi : pemeriksaan apakah perangkat lunak sesuai dengan permintaan pemesan.
- Evolusi : mengubah perangkat lunak untuk menyesuaikan terhadap perubahan permintaan *user*.

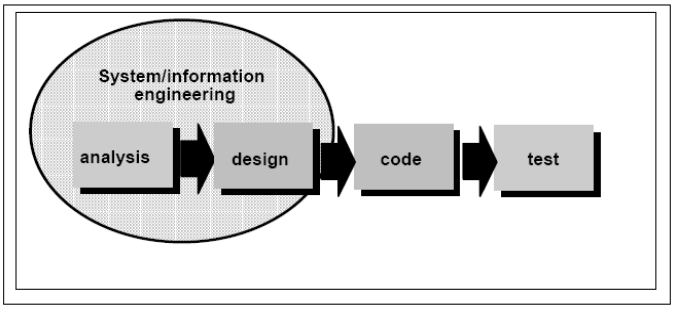
2.9 Pengertian Model Proses Perangkat Lunak

Model proses perangkat lunak adalah pemodelan atau gambaran dari proses perangkat lunak berdasarkan pandangan tertentu, yang biasanya di representasikan ke dalam bentuk diagram.

2.10 Model Waterfall / Linear Sequential Model

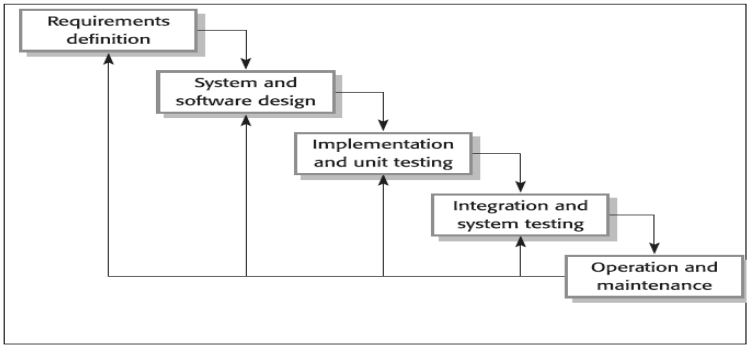
Model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun perangkat lunak. Berikut ini adalah dua gambaran dari model *waterfall*. Sekalipun keduanya menggunakan nama-nama fase yang berbeda, namun sama dalam intinya.

1. Fase-fase dalam model *waterfall* menurut referensi Pressman:



Gambar 2.2 fase-fase dalam model *waterfall* menurut referensi Pressman:

2. Fase-fase dalam model *waterfall* menurut referensi Sommerville:



Gambar 2.3 fase-fase dalam model *waterfall* menurut referensi Sommerville:

1) *Requirements analysis and definition:* Mengumpulkan kebutuhan secara lengkap kemudian kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh program yang akan dibangun. Fase ini harus dikerjakan secara lengkap untuk bisa menghasilkan desain yang lengkap.

2) *System and software design:* Desain dikerjakan setelah kebutuhan selesai dikumpulkan secara lengkap.

3) *Implementation and unit testing:* desain program diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Program yang dibangun langsung diuji baik secara unit.

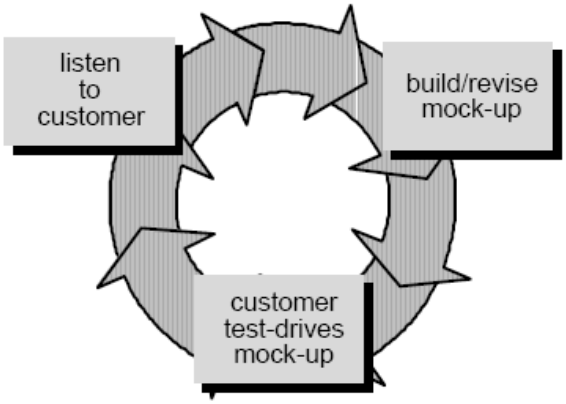
4) *Integration and system testing:* Penyatuan unit-unit program kemudian diuji secara keseluruhan (system testing).

5) *Operation and maintenance:* mengoperasikan program dilingkungannya dan melakukan pemeliharaan, seperti penyesuaian atau perubahan karena adaptasi dengan situasi sebenarnya.

2.11 Model Prototype

Kadang-kadang klien hanya memberikan beberapa kebutuhan umum software tanpa detil input, proses atau detil output. Di lain waktu mungkin dimana tim pembangun (*developer*) tidak yakin terhadap efisiensi dari algoritma yang digunakan, tingkat adaptasi terhadap sistem operasi atau rancangan form user interface. Ketika

situasi seperti ini terjadi model *prototyping* sangat membantu proses pembangunan *software*.



Gambar 2.4 Model Prototype

Proses pada model *prototyping* yang digambarkan pada gambar 2.x di atas, bisa dijelaskan sebagai berikut:

- Pengumpulan kebutuhan: Developer dan klien bertemu dan menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya. Detil kebutuhan mungkin tidak dibicarakan disini, tetapi pada awal pengumpulan kebutuhan.
- Perancangan : Perancangan dilakukan cepat dan rancangan mewakili semua aspek *software* yang diketahui, dan rancangan ini menjadi dasar pembuatan *prototype*.
- Evaluasi *prototype*: Klien mengevaluasi *prototype* yang dibuat dan digunakan untuk memperjelas kebutuhan *software*.

Perulangan ketiga proses ini terus berlangsung hingga semua kebutuhan terpenuhi. *Prototype-prototype* dibuat untuk memuaskan kebutuhan klien dan untuk

memahami kebutuhan klien lebih baik. Prototype yang dibuat dapat dimanfaatkan kembali untuk membangun software lebih cepat, namun tidak semua prototype bisa dimanfaatkan. Sekalipun prototype memudahkan komunikasi antar developer dan klien, membuat klien mendapat gambaran awal dari prototype, membantu mendapatkan kebutuhan detil lebih baik namun demikian prototype juga menimbulkan masalah:

1. Dalam membuat *prototype* banyak hal yang diabaikan seperti efisiensi, kualitas, kemudahan dipelihara/dikembangkan, dan kecocokan dengan lingkungan yang sebenarnya. Jika klien merasa cocok dengan prototype yang disajikan dan berkeras terhadap produk tersebut, maka developer harus kerja keras untuk mewujudkan produk tersebut menjadi lebih baik, sesuai kualitas yang seharusnya.
2. *Developer* biasanya melakukan kompromi dalam beberapa hal karena harus membuat *prototype* dalam waktu singkat. Mungkin sistem operasi yang tidak sesuai, bahasa pemrograman yang berbeda, atau algoritma yang lebih sederhana.

Agar model ini bisa berjalan dengan baik, perlu disepakati bersama oleh klien dan *developer* bahwa *prototype* yang dibangun merupakan alat untuk mendefinisikan kebutuhan *software*.

2.12 Pengertian Sistem

Sistem adalah kumpulan dari unsur atau elemen-elemen yang saling berkaitan atau berinteraksi dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu. Sebagai contoh, sistem komputer terdiri dari: *Software*, *Hardware* dan *Brainware*.

Menurut Jerry FithGerald, Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau mePnyelesaikan suatu sasaran tertentu.

Menurut Ludwig Von Bartalanfy, Sistem merupakan seperangkat unsur yang saling terikat dalam suatu antar relasi diantara unsur-unsur tersebut dengan lingkungan.

Menurut Anatol Raporot, Sistem adalah suatu kumpulan kesatuan dan perangkat hubungan satu sama lain.

Menurut L. Ackof, Sistem adalah setiap kesatuan secara konseptual atau fisik yang terdiri dari bagian-bagian dalam keadaan saling tergantung satu sama lainnya.

Suatu sistem memiliki beberapa syarat – syarat, antara lain:

a. Sistem harus dibentuk untuk menyelesaikan tujuan.
b. Elemen sistem harus mempunyai rencana yang ditetapkan.
c. Adanya hubungan diantara elemen sistem.

d. Unsur dasar dari proses (arus informasi, energi dan material) lebih penting daripada elemen sistem.
e. Tujuan organisasi lebih penting dari pada tujuan elemen.

2.13 Konsep Dasar Sistem

Didalam mendefinisikan sistem, terdapat dua pendekatan yang menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya. Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem sebagai berikut :

> *"Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersam-sama unutk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu".*

Sedangkan pendekatan sistem yang lebih menekankan pada elemen atau komponennya mendefinisikan sistem sebagai berikut :

> "*Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai tujuan tertentu "*.

2.14 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, antara lain akan dijelaskan dibawah ini.

1. Komponen (*Component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem tidak perduli betapapun kecilnya, selalu mengandung komponen-komponen atau ***subsistem-subsistem***. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai suatu sistem yang lebih besar yangdisebut ***supra sistem***, misalnya suatu perusahaan dapat disebut dengan suatu sistem dan industri yang merupakan sistem yang lebih besar dapat disebut dengan supra sistem. Kalaudipandang industri sebagai suatu sistem, maka perusahaan dapat disebut sebagai subsistem. Demikian juga bila perusahaan dipandang sebagai suatu sistem, maka sistem akuntansi adalah subsistemnya.

2. Batas Sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistemdipandang sebagai suatu kesatuan, karena dengan batas sistem ini fungsi dan tugas darisubsistem yang satu dengan lainnya berbeda tetapi tetap saling berinteraksi. Batas suatu sistem menunjukkan ruang lingkup (scope) dari sistem tersebut.

3. Lingkungan Luar Sistem (*Environment*)

Segala sesuatu diluar dari batas sistem yang mempengaruhi operasi dari suatu sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan atau merugikan. Lingkungan luar yang menguntungkan harus dipelihara dan dijaga agar tidak hilang pengaruhnya, sedangkan lingkungan luar yang merugikan harus dimusnahkan dikendalikan agar tidak mengganggu operasi sistem.

4. Penghubung Sistem (*Interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Untuk membentuk satu kesatuan, sehingga sumber-sumber daya mengalir dari subsistem yang satu ke subsistem yang lainnya. Dengan kata lain output dari suatu subsistem akan menjadi input dari subsistem yang lainnya.

5. Masukan Sistem (*Input*)

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa Masukan Perawatan (*Maintenance Input*) adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Masukan Sinyal (*Signal Input*) adalah energi yang diproses untukdidapatkan keluaran. Sebagai contoh didalam sistem komputer, program adalah maintanance input yang digunakan untuk mengoperasikan komputernya dan data adalah signal input untuk diolah menjadi informasi.

6. Keluaran Sistem (*Output*)

Merupakan hasil dari energi yang diolah oleh sistem. Meliputi : Keluaran yang berguna, contohnya Informasi yang dikeluarkan oleh komputer. Dan Keluaran

yang tidak berguna yang dikenal sebagai sisa pembuangan, contohnya panas yang dikeluarkan oleh komputer.

7. Pengolahan Sistem (*Process*)

Merupakan bagian yang memproses masukan untuk menjadi keluaran yang diinginkan. Contoh CPU pada Komputer, Bagian Produksi yang mengubah bahan baku menjadi barang jadi, Bagian akuntansi yang mengolah data transaksi menjadi laporan keuangan.

8. Tujuan Sistem (Goal)

Setiap sistem pasti mempunyai tujuan ataupun sasaran yang mempengaruhi input yang dibutuhkan dan output yang dihasilkan. Dengan kata lain suatu sistem akan dikatakan berhasil kalau pengoperasian sistem itu mengenai sasaran atau tujuannya. Sistem yang tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya.

2.15 Pengertian Informasi

Pengertian informasi menurut Gordon B. Davis dalam bukunya ”Sistem Informasi Manajemen”, menerangkan bahwa : [8]

> *“ Informasi adalah data yang telah diolah menjadi suatu bentuk yang berarti bagi si penerima dan mempunyai nilai nyata atau yang dapat diraskan dalam keputusan-keputusan yang sekarang atau yang akan datang”*.

Kualitas suatu informasi tergantung dari tiga hal berikut :

1. Relevan

   Berarti informasi tersebut mempunyai manfaat bagi pemakainya.

2. Akurat

   Informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan serta informasi harus jelas mencerminkan maksudnya.

3. Tepat pada waktunya

   Berita informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi.

2.16 Sistem Informasi

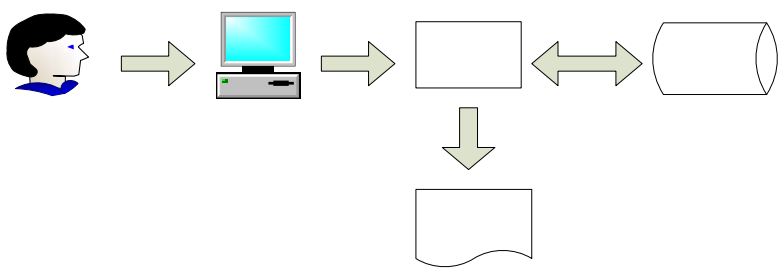
Ada beragam definisi tentang sistem informasi, tetapi pada pembahasan ini hanya akan diambil definisi umum saja. Dari berbagai definisi yang ada, dapat disimpulkan bahwa sistem informasi mencakup sejumlah komponen (Manusia, Komputer, Teknologi informasi, dan Prosedur kerja), ada sesuatu yang di proses (data menjadi informasi) dan dimaksud untuk mencapai suatu sasaran atau tujuan. [3]

Dalam suatu sistem informasi, terdapat komponen-komponen seperti :

1. **Perangkat keras** (*hardware*). Mencakup peranti-peranti fisik seperti komputer dan printer.
2. **Perangkat lunak** (*Software*) atau program. Sekumpulan instruksi yang memungkinkan perangkat keras untuk dapat memproses data.

3. **Prosedur**. Sekumpulan aturan yang dipakai untuk memwujudkan pemeroses data dan pembangkitan kelauran yang dikehendaki.
4. **Orang**. Semua pihak yang bertanggung jawab dalam pengembangan sistem informasi, pemeroses, dan penggunaan keluaran sistem informasi.
5. **Basis data** (Database). sekumpulan tabel, hubungan, dan lain-lain yang berkaitan dengan penyimpanan data.
6. **Jaringan komputer dan Komunikasi data**. Sistem penghubung yang memungkinkan *resources* dipakai secara bersama atau diakses oleh sejumlah pemakai.

Pada praktiknya sistem informasi tidak harus selalu berbentuk komplek mencakup keseluruhan komponen-komponen diatas. Gambar 2.2 memperlihatkan sebuah sistem yang sederhana. Sistem tersebut hanya digunakan untuk mencatat transaksi penjualan dan melinatkan satu orang saja. Melalui sebuah komputer, pemakai memasukan data penjualan dan saat setelah toko ditutup, laporan harian penjualan dicetak. Selanjutnya laporan digunakan untuk melakukan analisi tenatng barang-barang yang berlaku, yang berguna untuk pengambilan keputusan pembelian barang.



Gambar 2.5 Sistem Informasi yang Sederhana

Dalam bentuk yang lebih kompleks, sistem informasi melibatkan banyak pemakai dan memerlukan sarana jaringan yang memungkinkan pemakai yang tersebar di berbagai tempat yang berjauhan dapat berbagai informasi.

Hal-hal yang dikerjakan oleh Sistem Informasi tentu saja terkait dengan kemampuan utama yang dapat dilakukannya, yaitu :

1. Melaksanakan komputasi *numerik,* bervolume besar dan dengan kecepatan tinggi.
2. Menyediakan komunikasi dalam oragnisasi atau antara organisasi yang murah, akurat dan cepat.
3. Menyimpan informasi dalam jumlah yang sangat besar dalam ruangan yang kecil tetapi mudah diakses.
4. Memungkikan pengakses infromasi yang sangat banyak diseluruh dunia dengan cepat dan murah.
5. Meningkatkan efektifitas dan efesiensi orang-orang yang bekerja dalam kelompok dalam suatu tempat atau pada beberapa lokasi.
6. Menyajikan informasi dengan jelas yang menggugah pikiran manusia.
7. Mengotomasikan proses-proses bisnis yang semiotomatis dan tugas-tugas yang dikerjakan secara manual.
8. Mempercepat pengetikan dan penyutingan.
9. Pembiayaan yang jauh lebih murah daipada pengerjaan secara manual.

Kemampuan-kemampuan ini mendukung sasaran bisnis yang mencakup :

1. Peningkatan produktivitas
2. Pengurangan biaya
3. Peningkatan pengambilan keputusan
4. Peningkatan layanan ke pelanggan
5. Pengembangan aplikasi-aplikasi strategis yang baru.

Sistem informasi dapat memberikan nilai tambah terhadap proses, produksi, kualitas, manajemen, pengambilan keputusan dan pemecahan masalah. Serta keunggulan kompetitif yang tentu saja sangat berguna bagi kegiatan bisnis.

2.17 *Database Management System*

Database Management System (DBMS) diartikan sebagai suatu program komputer yang digunakan untuk memasukan, mengubah, menghapus, memanipulasi dan memperoleh data atau informasi dengan praktis dan efisien.

Dibandingkan dengan sistem yang berbasis kertas, DBMS memiliki empat keunggulan :

1. Kepraktisan. Sistem yang berbasis kertas akan menggunakan kertas yang sangat banyak untuk menyimpan informasi, sedangkan DBMS menggunakan media penyimpan sekunder yang berukuran kecil tetapi padat informasi.
2. Kecepatan. Mesin dapat mengambil atau mengubah data jauh lebih cepat dari pada manusia.

3. Mengurangi kejemuan. Orang cenderung menjadi bosan kalau melakukan tindakan-tindakan berulang yang menggunakan tangan.
4. Kekinian. Informasi yang tersedia pada DBMS akan bersifat mutakhir dan akurat setiap saat.

Di dalam lingkungan basis data, data lebih mudah digunakan pada beberapa DBMS tersedia fasilitas Query yang memudahkan pengguna untuk memperoleh informasi. Pengguna yang tidak memiliki kemampuan pemograman pun dengan mudah bisa mengguankan fasilitas query tersebut.

2.18 Pengolahan Data

Pengolahan data adalah suatu proses untuk menghasilkan bentuk yang lebih berguna berarti dari suatu kejadian berupa informasi yang diperlukan.

Sedangkan definisi dari Sistem Pengolahan adalah suatu kesatuan yang saling berhubungan yang terdiri dari peralatan, tenaga pelaksana dan prosedur-prosedur yang akan menghasilkan suatu informasi.

2.19 Basis Data

Basis data adalah kumpulan dari data yang saling berhubungan antara satu tabel dengan tabel lainnya yang tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Selain itu Basis data (*database*)

juga merupakan kumpulan data yang disusun atau diatur sedemikian rupa sehingga kita dapat memperoleh informasi yang tepat dengan secapat mungkin.

Tujuan dari perancangan basis data adalah menentukan data-data yang dibutuhkan dalam sistem sehingga kebutuhan informasi dapat terpenuhi dan dapat diperoleh dengan tepat dan secepat mungkin.

2.20 Flowmap

Flowmap adalah representasi grafik dari sistem informasi, proses-proses, aliran-aliran data logis, masukan, keluaran, dan *file-file* serta entitas-entitas sistem operasi yang berhubungan dengan sistem informasi.

2.21 Model Analisis Sistem

Pemodelan (*modelling*) adalah proses merancang piranti lunak sebelum melakukan pengkodean (*coding*). Model proses piranti lunak dapat dianalogikan seperti pembuatan blueprint pada pembangunan gedung. Semakin komplek sebuah sistem, semakin penting pula penggunaan teknik pemodelan yang baik. Dengan menggunakan model diharapkan pengembangan piranti lunak dapat memenuhi semua kebutuhan pengguna dengan lengkap dan tepat, termasuk factor-faktor seperti *scalability, robustness, security,* dan sebagainya. Kesuksesan suatu pemodelan piranti lunak ditentukan oleh tiga unsure, yang kemudian terkenal dengan sebutan segitiga sukses (*the triangle for success*). Ketiga unsur tersebut adalah metode pemodelan (*notation*), proses (*process*) dan *tool* yang digunakan. Memahami notasi pemodelan

tanpa mengetahui cara pemakaian yang sebenarnya (proses) akan membuat proyek gagal. Dan pemahaman terhadap metode pemodelan dan proses disempurnakan dengan penggunaan tool yang tepat.

2.22 UML

Unified Modelling Language (UML) adalah sebuah ‘bahasa’ yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Seperti bahasa-bahasa lainnya, UML mendefenisikan notasi dan *syntax/*semantic. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefenisikan bagaiman bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT (Object Modeling Technique), dan Ivar Jacobson OOSE (Object Oriented Software Engineering).

UML memuat beberapa komponen diagram yang secara umum terdiri dari :

1. **Use Case Diagram**

   Menggambarkan fungsionalitas yang diharapkandari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. *Use case* dapat membantu menyususun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua feature yang ada pada sistem. Sebuah use case dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*.

   Sebuah *use case* dapat juga meng-*extend use case* lain dengan *behavior*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

2. **Class Diagram**

   Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi

objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok:

1) Nama (dan stereotype)
2) Atribut
3) Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut:

- *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan.
- *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya.
- *Public*, dapat dipanggil oleh siapa saja.

Hubungan Antar Class:

1) Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunujukkan arah *query* antar *class.*

2) Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).

3) Pewarisan, yaitu hubungan hirarkis antar *class*. Class dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya

menambahkan fungsionalitas baru, sehingga disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.

4) Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram*.

3. Statechart Diagram

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).

Dalam UML, *state* digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar *state* umumnya memiliki kondisi *guard* yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. *Action* yang dilakukan sebagai akibat dari *event* tertentu dituliskan diawali dengan garis miring.

4. Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, dimana sebagian besar *state* adalah *action* dan

sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

5. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

6. Collaboration Diagram

Collaboration Diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number*, dimana *message* dari level tertinggi memiliki nomor 1. *Message* dari level yang sama memiliki prefiks yang sama.

7. Component Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-kmponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

8. Deployment Diagram

Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, dimana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal. Sebuah *node* adalah server, *workstation*, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefenisikan dalam diagram ini.

2.23 Jaringan Semantik

Model jaringan sematik merupakan grafik, yang terdiri dari simpul-simpul yang merepresentasikan objek fisik atau objek konsep, dan busur-busur yang

menunjukan relasi antara simpul-simpul tersebut. Jaringan semantik merupakan alat efektif untuk merepresentasikan pemetaan data, yang bertujuan mencegah terjadinya duplikasi data. [2]

2.24 Pengertian Proyek

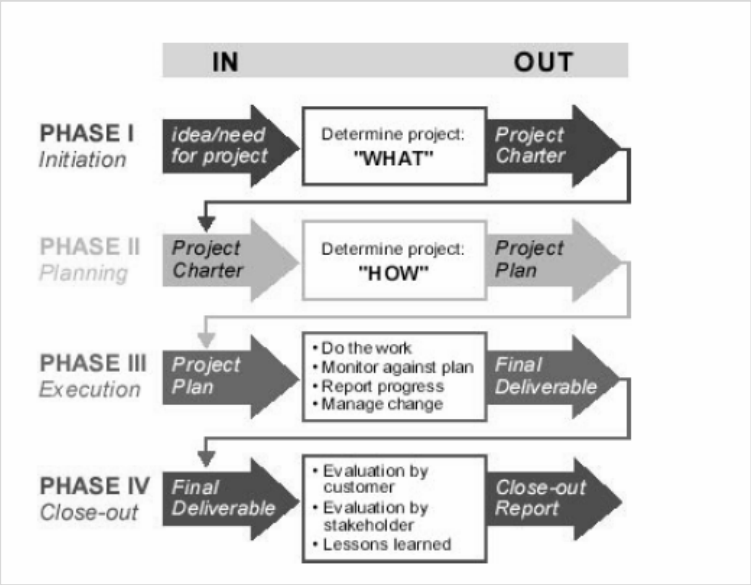
Proyek adalah urutan (sementara) aktivitas-aktivitas unik, kompleks, dan terkoneksi yang memiliki satu tujuan atau sasaran dan harus dilengkapi dengan waktu spesifik, sesuai anggaran, dan sesuai dengan spesifikasi. Sebuah proses atau metodologi pengembangan sistem, seperti *FAST*, mendefenisikan urutan aktivitas-aktivitas bersifat wajib dan opsional. Tiap pengembangan sistem “unik” adalah berbeda dari tiap proyek pengembangan sistem lain yang mendahuluinya. Aktivitas-aktivitas yang mencakup pengembangan sistem relative “kompleks”. Fase-fase dan aktivitas-aktivitas yang membentuk metodologi pengembangan sistem pada umumnya berurutan. Meskipun beberapa tugas saling menggantikan, namun banyak tugas tergantung pada penyelesaian tugas-tugas lain. Meskipun banyak proyek yang memiliki tenggat waktu yang pasti atau waktu yang ditentukan (ada beberapa pengecualian), namun proyek-proyek tersebut ternyata diselesaikan lebih lambat dari yang diperkirakan. Hal itu menjadi kurang diterima oleh manajemen atas yang mendapat tekanan dari seluruh organisasi untuk mengurangi waktu siklus untuk proses-proses produk dan bisnis.

2.25 Pengertian Manajemen Proyek

Manajemen proyek adalah proses melingkupi, merencanakan, menyediakan staf, mengorganisasi, mengarahkan, dan mengontrol pengembangan sebuah sistem yang dapat diterima dengan biaya minimal dan selama jangka waktu tertentu. Untuk sebuah proyek pengembangan sistem, manajemen proyek yang efektif diperlukan unutk memastikan bahwa proyek selesai pada tenggat waktunya, dikembangkan sesuai anggaran yang dapat diteriman dan memenuhi harapan dan spesifikasi para pelanggan. Organisasi-organisasi menggunakan pendekatan yang berbeda pada manajemen proyek. Salah satu pendekatan adalah menunjuk seorang manajer proyek dari tingkatan-tingkatan tim (setelah dibentuk). Alternatifnya, banyak percaya bahwa para manajer proyek yang berhasil menerapkan kumpulan unik pengetahuan dan keterampilan yang harus dipelajari. Organisasi-organisasi ini cenderung memperkerjakan dan atau mengembangkan manajer-manajer proyek professional yang kemudian ditugaskan ke satu atau lebih proyek.

Prasyarat untuk manajemen proyek yang baik adalah proses pengembangan sistem didefenisikan dengan baik. Manajemen proses (*process management*) adalah aktivitas berkesinambungan yang mendokumentasikan, mengelola penggunaan, dan memperbaiki metode terpilih organisasi (“proses”) untuk pengembangan sistem. Manajemen proses berhubungan dengan akrivitas, produk jadi, dan standar kualitas untuk diterapkan pada semua proyek. Lingkup manajemen proses adalah semua proyek, sedangkan lingkup manajemen proyek adalah proyek tunggal.

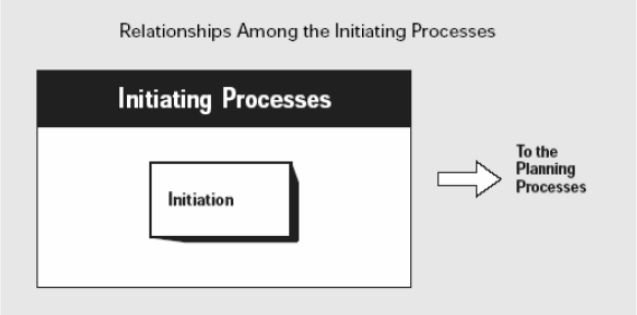
Terdapat beberapa kerangka kerja (framework) yang digunakan didalam merencanakan manajemen proyek. Masing-masing framework tersebut memiliki fase-fase tertentu didalamnya. Setiap fase-fase tersebut digunakan untuk membentuk struktur dan arahan dalam sebuah *life cycle* proyek.



Gambar 2.6 *Project life cycle*

Phase 1: Project Initiation / Inception / Envisioning / Analysis

Didalam metode MartinTate/PMI, fase pertama adalah fase dimana kita harus memutuskan apa yang harus diselesaikan (WHAT). Fase ini menjadi tanggung jawab sponsor, yaitu pihak management, atau dapat juga menjadi tanggung jawab *project leader* yang kemudian di tinjau dan disetujui oleh sponsor.



Gambar 2.7 Phase 1: Project Initiation

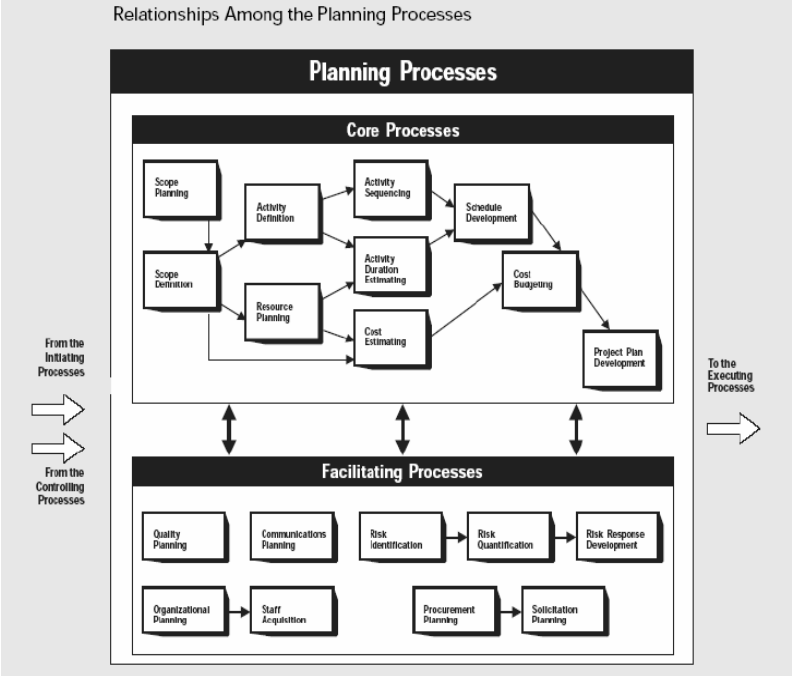
Phase 2: Project Planning / Elaboration / Design / Prototyping

Pada fase ini, tim yang terlibat didalam proyek harus memutuskan bagaimana project akan diselesaikan, dikerjakan oleh siapa, membutuhkan dana berapa dan memakan waktu berapa lama.

Hasil dari fase ini adalah project plan, atau rencana proyek, yang kemudian akan disetujui oleh sponsor, *customer* dan pihak lain yang terlibat dan berpengaruh terhadap jalannya proyek (misal: key stakeholder).

Dalam setiap perencanaan terdapat sedikitnya tiga bagian, yaitu:

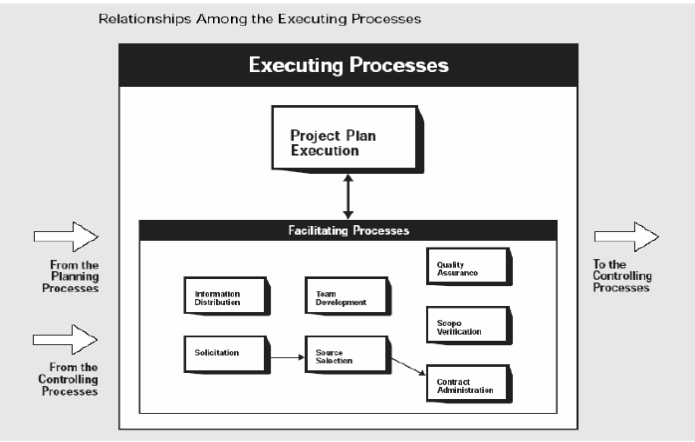
1. *Scope*, yaitu: apa yang akan dihasilkan/dibuat dari proyek tersebut; hal ini kemudian akan dibagi lagi menjadi sub-sub bagian tertentu.
2. *Resources*, yaitu: berapa banyak waktu yang dibutuhkan, effort yang dikeluarkan, dan dana yang dihabiskan.
3. *Assurance*, yaitu: bagaimana mengasumsikan bahwa proyek akan berjalan dengan baik dengan melihat pada resiko, evaluasi dan persetujuan yang dibuat.



Gambar 2.8 Phase 2: Project Planning

Phase 3: Project Execution / Construction / Development / Production

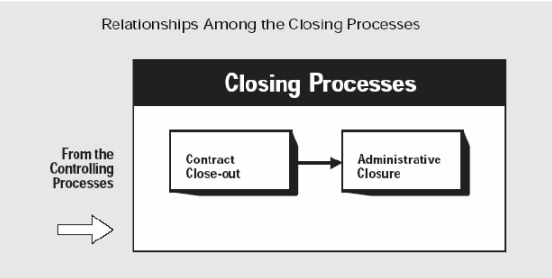
Fase ini adalah fase implementasi yang memiliki yaitu implementasi dari rencana yang telah dibuat pada fase sebelumnya. Hal ini sangat berbeda dengan pengertian instalasi dari produk yang dihasilkan dari proyek.



Gambar 2.9 Phase 3: Project Executing

Phase 4: Project Deployment / Transition / Stabilizing / Close-Out

Fase ini berisikan penyebaran hasil proyek pada end user. Pada akhir proyek sebaiknya juga menyertakan analisa dan laporan hasil proyek yang disebut dengan *post-implementation analysis* and *report*.



Gambar 2.10 Phase 4: Project Close Out

2.26 Fungsi-fungsi Manajer Proyek

Fungsi-fungsi dasar seorang manajer proyek adalah pelingkupan, perencanaan, penyediaan staf, pengorganisasian, penjadwalan, pengarahan, pengontrolan, dan penutupan.

- Pelingkupan (*scoping*) : Lingkup mendefenisikan batas-batas proyek. Harus mampu mencakup harapan-harapan dan batasan-batasan proyek dengan tujuan merencanakan aktivitas, memperkirakan biaya, dan mengelola harapan.
- Perencanaan (*planning*) : Perencanaan mengidentifikasikan tugas-tugas yang diperlukan untuk menyelesaikan proyek. Hal ini didasarkan pada pemahaman manajer terhadap lingkungan proyek dan metodologi yang digunakan untik mencapai proyek.

- Perkiraan (*estimating*) : memperkirakan – waktu untuk menyelesaikan tiap tugas dalam menyelesaikan proyek – berapa banyak orang yang diperlukan – ketrampilan apa saja yang diperlukan – tugas apa saja yang perlu diselesaikan sebelum menyelesaikan tugas lain – berapa biaya yang dibutuhkan.
- Penjadwalan (*sceduling*) : dengan diberikan rencana proyek, manajer proyek bertanggung-jawab atas penjadwalan semua aktivitas proyek. Jadwal proyek tersebut harus dikembangkan dengan pemahaman terhadap tugas yang diperlukan, durasi tugas, dan prasyarat tugas.
- Pengorganisasian (*organizing*) : manajer proyek harus memastikan bahwa para anggota tim proyek memahami peran dan tanggung-jawab masing-masing serta hubungan laporan mereka ke manager proyek.
- Pengarahan (*directing*) : setelah proyek dimulai, manajer proyek harus mengarahkan aktivitas-aktivitas tim. Tiap manajer proyek harus mendemontrasikan ketrampilan manajemen proyek untuk mengkoordinasikan, mendelegasikan, memotivasi, menyarankan, memuji, dan member penghargaan pada para anggota tim.
- Pengontrolan (*controling*) : memonitor dan melaporkan perkembangan tujuan, jadwal, dan biaya dan membuat penyesuaian seperlunya ketika diperlukan.
- Penutupan (*closing*) : manajer proyek yang baik selalu menilai keberhasilan dan kegagalan pada kesimpulan proyek. Mereka belajar dari kesalahan

mereka dan merencanakan perbaikan berkesinambungan pada proses pengembangan sistem.

2.27 Manajemen Personel, Produk dan Proses

Manajemen proyek mengatur 4 hal penting: personel, produk, proses dan proyek. Empat hal ini berurutan mulai dari yang paling penting. Personel merupakan mendapat tempat paling penting karena tanpa personel yang baik dan tepat maka 3 hal lain tidak bisa berjalan dengan baik.

a. Katagori Personel

Proses penyelesaian proyek melibatkan banyak personel. Personel-personel ini digambarkan seperti pemain, dan dikatagorikan dalam 5 katagori pemain :

1. Manajer senior : yang menentukan usaha yang dikerjakan, dan pemegang keputusan dalam proyek.
2. Manajer proyek (teknis) – pemimpin tim : yang membuat rencana, memotivasi, mengatur dan mengendalikan praktisi yang mengerjakan proyek.
3. Praktisi : yang mengerjakan proyek.
4. Klien : yang menentukan kebutuhan proyek dan pihak lain yang berkaitan dengan hasil produk
5. Pengguna hasil proyek : pihak yang menggunakan hasil proyek.

Efektifitas kerja masing-masing personel di atas harus diusahakan oleh pemimpin tim. Pemimpin tim ini yang mengatur tim proyek agar dapat memberikan yang terbaik dari masing-masing personel.

b. Pemimpin Tim

Pemimpin tim proyek disini adalah manager proyek. Seorang pemimpin tim diharuskan mempunyai ketrampilan memimpin yang cukup. Seseorang tidak menjadi pemimpin tim secara kebetulan tapi sungguh-sungguh karena punya kemampuan.

Kemampuan yang dibutuhkan dalam kepemimpinan seperti:

- mampu memotivasi
- mampu berorganisasi : mengatur proses yang ada atau membuat yang baru dalam rangka mewujudkan ide/konsep menjadi produk.
- mampu mendorong keluarnya ide-ide baru: memberi dorongan, menciptakan situasi yang kondusif untuk lahirnya ide baru.
- mencari penyelesaian masalah (problem solving): mampu menganalisa masalah-masalah teknis ataupun manajemen/organisasi kemudian mendapatkan jalan keluar atau memotivasi anggota untuk mampu menyelesaikan masalah. Akomodatif terhadap perubahan yang mungkin terjadi.
- mampu menjadi manajer : menggunakan wewenangnya pada saat yang tepat, atau memberikan kebebasan pada anggota timnya jika diperlukan.

- mampu menghargai kerja : menghargai hasil yang dicapai, ide yang dilontarkan dan pendapat yang diajukan oleh anggota timnya.
- mampu mengenali tim: mampu “membaca” dan memahami anggota timnya. Mampu memenuhi kebutuhan tim dan bertahan dalam tekanan yang tinggi.

c. Tim Proyek

Struktur organisasi dalam tim ini bisa mengadaptasi dari banyak struktur organisasi yang sudah ada. Berikut beberapa pilihan pembagian tugas/penugasan yang bisa diterapkan untuk tim proyek yang terdiri dari n personel yang bekerja selama k tahun :

1. n personel ditugaskan untuk sejumlah m tugas yang berbeda dengan sedikit tugas gabungan _ koordinasi adalah tugas dari manajer yang mungkin saja punya m proyek lainnya.
2. n personel di tugaskan untuk sejumlah m tugas yang berbeda dengan m < n sehingga terbentuk tim informal. Pemimpin tim khusus perlu ada _ koordinasi antar tim adalah tanggung jawab manajer.
3. n personel dibagi menjadi sejumlah t tim. Tiap tim ditugaskan mengerjakan satu atau lebih tugas. Tiap tugas mempunyai struktur yang ditentukan sebelumnya bagi semua tim _ koordinasi dikendalikan oleh tim dan manager.

Sekalipun masing-masing pilihan punya argumentasi sendiri-sendiri, namun dari pengamatan yang dilakukan, pilihan no 3 dianggap lebih produktif. Cara atau gaya manajemen, jumlah personel, tingkat kemampuan para personel dan masalah-

masalah yang dihadapi tim menentukan bentuk struktur organisasi yang bisa diterapkan.

Contoh struktur organisasi tim adalah:

1. *Democratic Decentralized (DD) :* Tidak ada pemimpin yang permanen, koordinator ditunjuk untuk jangka waktu yang pendek, keputusan diambil berdasarkan konsensus bersama, komunikasi horizontal antar anggota tim (posisi sejajar semua) _ cocok untuk masalah yang sulit/rumit, cocok untuk proyek besar, tim cenderung awet dan bertahan lama, pekerjaan memuaskan, cocok untuk masalah yang modularitasnya rendah, perlu banyak waktu untuk menyelesaikan proyek.

2. Controlled decentralized (CD) *:* Pemimpin tim ditentukan, ada wakil pemimpin dan mereka berbagi tugas, penyelesaian masalah adalah tugas tim dan implementasinya dibagi di antara beberapa sub-tim oleh pemimpin, komunikasi horisontal di antara sub-tim dan di antara personel, komunikasi vertikal berdasarkan struktur hirarki _ sentralisasi untuk penyelesaian masalah, cocok untuk masalah yang sederhana, cukup cocok untuk proyek besar, masalah dengan modularitas tinggi, menghasilkan sedikit kesalahan.

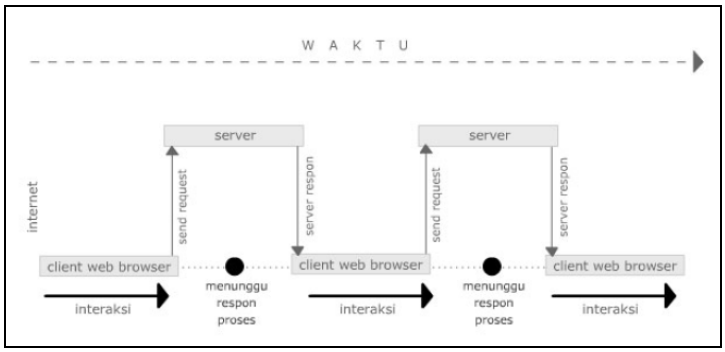
3. *Controlled Centralized (CC) :* penyelesaian masalah dikerjakan oleh pemimpin, pemimpin melakukan koordinasi internal tim, komunikasi lebih banyak vertikal antara pemimpin dan anggota tim _ cocok untuk masalah yang sederhana, melakukan penyelesaian, masalah lebih cepat, masalah dengan modularitas tinggi, menghasilkan sedikit kesalahan.

2.28 AJAX

AJAX bukan merupakan bahasa pemrograman baru, namun hanya sebuah istilah untuk teknik pemanfaatan Javascript dalam mengontrol class object XMLHttpRequest untuk berkomunikasi dengan server kemudian merefresh/mengupdate content yang ada dalam halaman web tanpa melakukan Reload keseluruhan halaman web seperti pada metode Tradisional sebelumnya, AJAX sendiri merupakan singkatan dari "Asynchronous JavaScript And XML".

2.28.1 Asynchronous & Synchronous

SYNCHRONOUS dalam pengembangan web yang telah lama digunakan sebelumnya. Gambar 2.11 dibawah berikut adalah gambar proses secara *synchronous.*

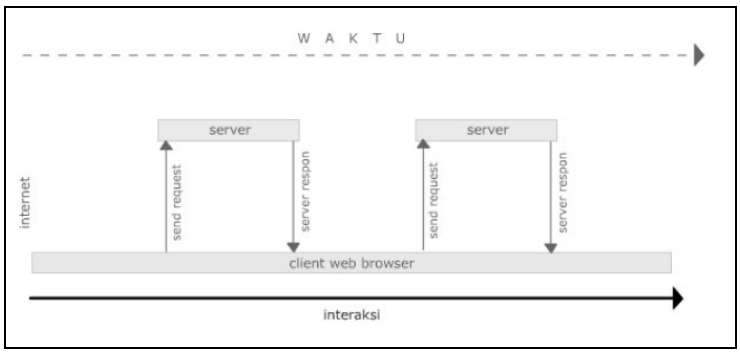


Gambar 2.11 Proses Synchronous

Saat user berinteraksi dengan sebuah halaman web terdapat banyak pilihan link yang dapat ditemukan, dan saat link tersebut di click maka halaman web tersebut akan melakukan komunikasi dengan server melalui internet untuk meminta proses ke server (send Request). Selama server melakukan proses, user akan menunggu hasil

proses tersebut. Setelah server selesai melakukan proses yang diminta maka server akan mengembalikan hasil proses yang akan ditampilkan pada client web-browser. Dalam proses synchronous ini user melakukan permintaan proses (request process) ke server dan menunggu hingga server mengembalikan hasil proses yang di minta, umumnya hasil proses dari server merupakan sebuah halaman web baru yang ditampilkan pada web-browser user.

Pemanfaatan teknik Asynchronous dengan Javascript ini dalam pengembangan web ini lebih dikenal dengan istilah WEB 2.0. Teknik proses Asynchronous dapat membuat sebuah aplikasi web lebih kaya atau banyak yang menulis dengan istilah *Rich internet aplication (RIA)*. User dapat tetap berinteraksi dengan halaman web yang ada selama proses request dilakukan dan selama server belum mengembalikan hasil proses yang diminta. Dan saat hasil proses telah selesai, kemudian mengupdate data halaman web yang telah ada, berikut gambaran dari proses kerja Asynchronous tersebut:



Gambar 2.12 Proses Asynchronous

User berinteraksi dengan link-link yang ada untuk meminta proses ke server, proses yang diminta akan dikirimkan secara background ke server, selama server belum mengembalikan data hasil proses, user dapat tetap bekerja dengan halaman web yang ada sebelumnya. Setelah server selesai melakukan proses, hasil proses tersebut akan dikirmkan kembali kepada web-browser, saat data yang dikirimkan server telah diterima oleh web-browser user maka data tersebut ditampilkan pada halaman web yang telah ada sebelumnya.

Disini terlihat semua proses komunikasi dengan server dilakukan secara background atau mungkin bisa dikatakan semua proses dilakukan Behind the Scene.

2.28.2 XMLHttpRequest Object

Untuk dapat mengembangkan aplikasi web dengan tehnik AJAX, maka perlu mengkombinasikan beberapa hal berikut:

1. Javascript untuk membuat object XMLHttpRequest yang kita gunakan untuk berkomunikasi dengan server secara behind the scene.
2. DOM (Document Object Model), hasil proses yang diterima akan ditampilkan dengan memanipulasi object DOM yang telah dipersiapkan sebelumnya untuk menampilkan data hasil proses yang diberikan server.
3. XML (eXtensible Markup Language) format data yang dikembalikan oleh server, data XML ini siap dibaca dan ditampilkan untuk mengupdate content pada halaman web.

2.28.3 XML DATA FORMAT

Setelah berkenalan dengan AJAX dan mengetahui proses kerjanya dalam melakukan request Asychronous dan juga menerima respon dari server, sekarang kita akanmembahas tentang XML data format yang dijadikan salah satu bagian dalam singkatan AJAX.

Sedikit penjelasan singkat XML

- XML merupakan singkatan dari eXtensible Markup Language.
- XML merupakan Tag base sintaks.
- Masing-masing tag dalam XML diawali dengan tag pembuka dan diakhiri dengan tag penutup, beberapa artikel atau tulisan juga biasa menyebut TAG sebagai ELEMENT. Namun dalam artikel ini saya akan mengunakan istilah TAG dalam menyebutkannya.

  contoh <data> ... </data>
- Single line tag dapat dibuat dengan memberikan closing tag pada bagian akhir “/>”

  contoh: <data atrr=”value” />
- Attribut harus dituliskan pada bagian tag pembuka dan diapit oleh tanda “ .. “ atau '.. '

  contoh: <tagName attr=”value”>
- Comment dalam XML dbuat didalam tanda <!-- komentar disini -->, bagian comment

Untuk membaca data dalam format XML tersebut mengunakan bantuan DOM (Document Object Model), dengan mengetahui tentang struktur dari sebuah Document maka akan lebih mudah membacanya.

2.29 PHP

PHP adalah salah satu bahasa *Server-side* yang didesain khusus untuk aplikasi web. PHP dapat disisipkan diantara bahasa HTML dan karena bahasa *Server side*, *m*aka bahasa PHP akan dieksekusi di server, sehingga yang dikirimkan ke browser adalah “hasil jadi” dalam bentuk
HTML, dan kode PHP anda tidak akan terlihat.

PHP dahulunya merupakan proyek pribadi dari Rasmus Lerdorf (dengan dikeluarkannya PHP versi 1) yang digunakan untuk membuat home page pribadinya. Versi pertama ini berupa kumpulan script PERL. Untuk versi keduanya, Rasmus menulis ulang script-script PERL tersebut menggunakan bahasa C, kemudian menambahkan fasilitas untuk Form HTML dan koneksi MySQL. Adapun PHP didapat dari singkatan Personal Home Pages.

PHP telah menampakkan keunggulannya sebagai salah satu bahasa server scripting yang handal. Melalui perkembangan yang pesat ini banyak fasilitas yang ditambahkan dan oleh kelompok ini PHP disebut sebagai "PHP: Hypertext Preprocessor" . Sintak yang digunakan berasal dari bahasa C , Java maupun Perl.

PHP merupakan bahasa script yang digunakan untuk membuat halaman web yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh client. Mekanisme ini menyebabkan informasi yang diterima client selalu yang terbaru. Semua script PHP dieksekusi pada server dimana script tersebut dijalankan. Oleh karena itu, spesifikasi server lebih berpengaruh pada eksekusi dari script php daripada spesifikasi client.

Namun tetap diperhatikan bahwa halaman web yang dihasilkan tentunya harus dapat dibuka oleh browser pada client. Dalam hal ini versi dari html yang digunakan harus didukung oleh browser client.

PHP termasuk dalam *Open Source Product*. Jadi anda dapat merubah *source code* dan mendistribusikannya secara bebas. Tanpa mengurangi kemampuan-kemampuan yang sudah ada, PHP 5 hadir dengan fitur yang lebih kompleks, dan merupakan rilis terbaru yang dikeluarkan oleh pengembangnya. Secara besar-besaran pengembang PHP merombak bug yang sering dijumpai dalam PHP 4, hal ini akan sangat terasa ketika akan membuat aplikasi berbasis objek.

Dengan tujuan untuk lebih meningkatkan kinerja aplikasi tanpa bermaksud membuang fitur yang ada, PHP 5 menambahkan fitur-fitur baru seperti dukungan terhadap MySQL. Dukungan ini diwujudkan dalam bentuk fungsi-fungsi lama dan ditulis ulang.

2.30 Javascript

Javascript mulanya bernama LiveScript, dikembangkan pertama kali tahun 1995 di Netscape Communication. Pada akhir tahun 1995 Netscape Communication dan Sun Microsystems berkolaborasi dan mengganti nama LiveScript menjadi Javascript.

Javascript adalah bahasa skrip yang ditempatkan pada kode HTML dan diproses pada sisi klien. Dengan adanya bahasa ini maka kemampuan dokumen HTML menjadi lebih luas. Sebagai contoh, digunakan untuk validasi masukan pada formulir sebelum diproses ke tahap selanjutnya. Bias untuk membuat permainan interaktif dan juga bisa menambah desain web.

2.31 WAMP 5

Wamp 5 adalah sebuah installer untuk Windows. Wamp 5 akan menginstal secara lengkap WAMP solution pada komputer. WAMP adalah untuk : Windows – Apache – MySQL – PHP 5. WAMP secara otomatis mengatur keempat software tersebut untuk bekerja sama, dimanapun software-software tersebut di instal pada drive local.

Adapun software yang terinstal adalah :

- Apache 1.3.x
- PHP 5.x.x
- MySQL 4.x.x
- PHPmyadmin

- SQLitemanager
- Wampserver service manager

WAMP 5 tidak dirancang untuk untuk produksi tetapi sebagai lingkungan pengembangan. Dengan WAMP 5, dapat digunakan untuk membuat naskah secara lokal pada komputer, dengan mencoba dan menguploadnya pada server produksi.

Apache dan MySQL di atur dengan file konfigurasi default agar memperoleh platform standar yang dapat disesuaikan dengan kebutuhan pribadi.

Selama instalasi, sebuah direktori ‘www’ akan dihasilkan pada direktori WAMP 5. Dengan hanya menaruh skrip file (file php) pada direktori tersebut dan mengakses localhost melalui sebuah Internet Windows atau service manager (icon tray).

2.32 Macromedia Dreamweaver

Macromedia Dreamweaver adalah sebuah HTML editor professional untuk mendesain secara visual dan mengelola situs Web maupun halaman Web. *Macromedia Dreamweaver* merupakan *software* utama yang digunakan oleh *Web Designer* maupun *Web Progremer* guna mengembangkan situs Web. Ruang kerja, fasilitas dan kemampuan *Macromedia Dreamweaver* mampu meningkatkan produktivitas dan efektivitas dalam desain maupun membangun situs Web.