BAB 2
TINJAUAN PUSTAKA

2.1 Profil PT. Quicksteps Sarana Solusindo

PT.Quicksteps Sarana Solusindo (QSS) merupakan sebuah perusahaan yang bergerak dibidang Teknologi Informasi(TI) dan Support System. PT.Quicksteps Sarana Solusindo telah berpengalaman di bidang pengembangan perangkat lunak baik *desktop*, *website* maupun *mobile*. PT.QSS sering dipercaya untuk mengerjakan berbagai proyek perangkat lunak baik dari instansi pemerintahan maupun swasta. Berikut profil PT.Quicksteps Sarana Solusindo (QSS):

Nama Perusahaan : PT.Quicksteps Sarana Solusindo

Alamat : Emerald Tower Apartment Lt.12 Jl. Sanggar Kencana XXVII No.50 Soekarno Hatta Bandung

Telepon : 0877 2288 6650

Email : marketing@shopedia.co.id

2.1.1 Visi dan Misi PT. Quicksteps Sarana Solusindo

Visi adalah suatu pandangan jauh tentang program perushaan di masa depan, tujuan – tujuan perusahaan dan apa yang harus dilakukan untuk mencapai tujuan pada masa yang akan datang, sedangkan misi adalah pernyataan tentang apa yang harus dikerjakan oelh perusahaan dalam usahanya mewujudkan visi tersebut. Adapun Visi dan Misi dari PT. Quicksteps Sarana Solusindo adalah sebagai berikut.

1. **Visi**

   "Menjadi Team yang profesional dan mempunyai komitmen,berdaya saing dan mampu menjawab permasalahan global di bidang teknologi informasi".

2. **Misi**

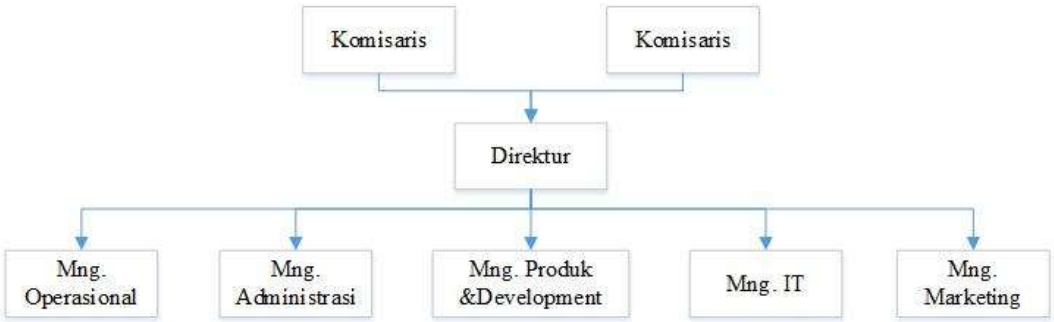
   a. Membangun SDM unggul khususnya dibidang IT yang memiliki kapasitas intelektual,kesiapan profesional dan akhlakul karimah.

   b. Memenuhi kebutuhan dan keinginan klien dengan selalu menyediakan produk dan layanan yang berkualitas tinggi dan anda

c. Menyediakan sarana dan fasilitas berkarya untuk semua team dalam suasana kerja yang profesional, kekeluargaan dan secara individu bermartabat
d. Memasyarakatkan teknologi informasi baik untuk tujuan komersil maupun untuk kepentingan sosial sehingga teknologi informasi akan mudah diakses oleh berbagai kalangan.

2.1.2 Struktur Organisasi dan Deskripsi Kerja

Setiap organisasi mempunyai struktur yang berbeda disesuaikan dengan karakteristik masing-masing organisasi tersebut. Dengan adanya struktur organisasi setiap anggota akan mengetahui dimana posisinya sehingga mereka dapat melakukan aktifitas sesuai dengan tugas, wewenang dan tanggungjawab masing-masing. Berikut ini adalah struktur organisasi yang ada di PT. Quicksteps Sarana Solusindo.



Gambar 0.1 Struktur Organisasi PT. Quicksteps Sarana Solusindo

Adapun tugas dan wewenang masing-masing bagian yang terdapat dalam struktur organisasi PT. Quicksteps Sarana Solusindo adalah sebagai berikut :

1. Komisaris
   a. Memerintah organisasi dengan menetapkan kebijakan-kebijakan dan tujuan luas dari perusahaan.
   b. Melakukan pengawasan dan memberikan nasihat kepada direksi.
   c. Memastikan keberadaan dan kecukupan sumber keuangan.
   d. Mengesahkan anggaran tahunan.
2. Direktur

a. Mengambil keputusan dalam pengambilan proyek atau dalam mendapatkan proyek
b. Mengatur keuangan baik mulai dari modal awal, pemasukan dan pengeluaran.
c. Mengontrol kinerja setiap bagian.

3. Manager Operasinal
   a. Mengelola dan meningkatkan efektivitas dan efisiensi operasi perusahaan.
   b. Memastikan operasional perusahaan berjalan dengan baik.
   c. Melakukan komunikasi dengan klien proyek.
4. Manager Administrasi

   Mengatur, membuat dan melakukan dokumentasi mengenai urusan administrasi perusahaan.
5. *Produk and Development Manager*
   a. Mencari inovasi terbaru tentang produk yang akan di buat.
   b. Melakukan pengembangan produk baik untuk intra perusahaan maupun untuk kebutuhan klien.
   c. Membuat rancangan awal proyek berupa penjadwalan ,pengeluaran serta target proyek yang akan dicapai
   d. Melakukan pengendalian proyek
   e. Melakukan pengawasan tim pelaksana proyek
6. Manager IT
   a. Mengelola teknologi informasi dan sistem komputer perusahaan.
   b. Memberikan solusi teknologi informasi.
   c. Melakukan pengawasan dan perawatan teknologi informasi
7. Manager Marketing
   a. Mengiklankan dan memasarkan produk yang di produksi oleh perusahaan.

2.1.3 Logo Perusahaan

Berikut ini merupakan logo dari PT.QSS yang bisa di lihat pada gambar 2.2.



Gambar 0.2 Logo PT.Quickstpes Sarana Solusindo

2.2 Landasan Teori

Subbab ini berisikan teori – teori pendukung yang digunakan dalam proses analisis dan implementasikan pada permasalahan yang ada di Quicksteps Sarana Solusindo.

2.2.1 Pengenalan Sistem Informasi

Ada beberapa hal yang harus diketahui jika berbicara tentang sistem informasi diantaranya adalah pengertian tentang sistem serta ruang lingkup dari sistem dan pengertian informasi serta ruang lingkupnya . Di bawah ini adalah penjelasan mengenai masing-masing teori yang berhubungan dengan sistem dan informasi.

2.2.1.1 Pengertian Sistem

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu. Sistem dapat didefenisikan sebagai suatu kesatuan yang terdiri dari dua atau lebih komponen atau subsistem yang berinteraksi untuk mencapai suatu tujuan[7].

Sistem kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu

1. Syarat – syarat Sistem

Syarat – syarat yang harus dimiliki oleh suatu sistem, yaitu:

1. Sistem harus dibentuk untuk menyelesaikan suatu tujuan.
2. Elemen sistem harus mempunyai rencana yang ditetapkan.
3. Adanya hubungan diantara elemen system
4. Unsur dasar dari proses (arus informasi, enrgi, dan material) lebih penting dari pada elemen sistem.

2. Karakteristik Sistem

Suatu sistem mempunyai karakteristik tertentu, yaitu mempunyai komponen-komponen (*components*), batas sistem (*boundary)*, lingkungan luar sistem (*environment*), penghubung (*interface*), masukan (*input)*, keluaran (*output*), pengolah (*process*), dan sasaran (*objective*) atau tujuan (*goal*). Di bawah ini merupakan penjelasan dari masing-masing karakteristik tersebut:

1. Komponen Sistem (*components*)

   Sistem terdiri dari sejumlah komponen yang saling berinteraksi , bekerja sama membentuk kesatuan. Komponen-komponen atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap subsistem mempunyai sifat- sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.
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 sistem dipandang sebagai suatu kesatuan. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.
3. Lingkungan Luar Sistem (*environment*)

   Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat

menguntungkan maupun merugikan. Lingkungan yang menguntungkan harus tetap dijaga dan dipelihara karena merupakan energi dari sistem. Sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan, karena jika tidak akan mengganggu kelangsungan sistem.

4. *Interface*

   Interface merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. *Interface* ini memungkinkan satu subsistem untuk mengalirkan sumber daya ke subsistem lainnya.

5. *Input*

   Input merupakan energi yang dimasukkan ke dalam sistem. *Input* dapat berupa *maintenance input* dan *signal input*. *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk menghasilkan *output.*

6. *Output*

   Output merupakan hasil dari energi yang diolah dan diklasifikasikan menjadi *output* yang berguna dan sisa pembuangan. *Output* dapat menjadi *input* untuk subsistem yang lain.

7. Pengolah Sistem (*process*)

   Suatu sistem mempunyai bagian pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran Sistem (*objective*)

   Suatu sistem mempunyai tujuan (*goal*) atau sasaran (*objective)*. Sasaran dari sistem menentukan *input* yang dibutuhkan dan *output* yang akan dihasilkan.

3. Klasifikasi Sistem

Sistem dapat diklasifikasikan ke melalui beberapa sudut pandang, diantaranya:

1. Sistem diklasifikasikan sebagai sistem abstrak (*abstract system*) dan sistem fisik (*physical sistem*). Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, sedangkan sistem fisik merupakan sistem yang ada secara fisik.

2. Sistem diklasifikasikan sebagai sistem alamiah (*natural system*) dan sistem buatan manusia (*human made system*). Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia.
3. Sistem diklasifikasikan sebagai sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*). Sistem tertentu beroperasi dengan tingkah laku yang sudah diprediksi. Interaksi diantara bagian-bagiannya dapat diprediksi dengan pasti, sehingga *output* dari sistem dapat diramalkan. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

4. Sistem diklasifikasikan sebagai sistem tertutup (*closed system*) dan sistem terbuka (*open system*). Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Secara teoritis sistem tertutup ini ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, tetapi yang ada hanyalah *relatively closed system* (secara relatif tertutup, tetapi tidak benar-benar tertutup). Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima input dan menghasilkan *output* untuk lingkungan luar atau subsistem yang lainnya.

2.2.1.2 Pengertian Informasi

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan pada saat sekarang atau masa yang akan datang. Selain itu informasi juga merupakan fakta-fakta atau data yang diproses sedemikian atau mengalami proses transformasi data sehingga berubah bentuk menjadi informasi[7].

1. Umur Informasi

Kapan atau sampai kapan sebuah informasi memiliki nilai/arti bagi penggunanya. Ada *condition informasion* (mengacu pada titik waktu tertentu) dan *operating information* (menyatakan suatu perubahan pada suatu range waktu).

2. Siklus Informasi

Data merupakan bentuk yang masih mentah yang belum berceritra banyak, sehingga perlu diolah lebih lanjut menjadi suatu model untuk dihasilkan informasi. Data yang diolah untuk menghasilkan informasi menggunakan suatu model proses tertentu.

Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, melakukan keputusan berdasarkan informasi tersebut dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap kembali sebagai *input*, diproses kembali melalui suatu model dan seterusnya yang merupakan suatu siklus.

3. Karakteristik dari Informasi yang Baik

Informasi dapat dikatakan baik jika memiliki kriteria dan karakteristik sebagai berikut :

1. *Information must be pertinent*

   Informasi harus berhubungan. Pernyataan informasi harus berhubungan dengan urusan dan masalah yang penting bagi penerima informasi (orang yang membutuhkan informasi tersebut).
2. *Information must be accurate*

   Informasi harus bebas dari kesalahan-kesalahan dan tidak memiliki bias atau menyesatkan. Informasi yang dihasilkan harus mencerminkan maksudnya. Keakuratan informasi seringkali bergantung pada keadaan.
3. *Information must be timely*

   Informasi harus ada ketika dibutuhkan. Informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi karena informasi merupakan landasan di dalam pengambilan keputusan.
4. Relevan

   Informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang yang satu dengan yang lainnya pasti berbeda.

5. Nilai Informasi

   Nilai dari informasi (*value of information*) ditentukan oleh dua hal, yaitu manfaat dan biaya untuk mendapatkan informasi tersebut. Suatu informasi dapat dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya untuk mendapatkannya. Sebagian besar informasi tidak dapat ditaksir secara pasti nilai keuntungannya (dalam satuan uang), tetapi kita dapat menaksir nilai efektifitas dari informasi tersebut. Pengukuran nilai informasi biasanya dihubungkan dengan analisis *cost effectiveness* atau *costbenefit*.

2.2.1.3 Pengertian Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan[7].

Sistem informasi merupakan sebuah sistem terintegrasi atau sistem manusia-mesin, untuk menyediakan informasi untuk mendukung operasi, manajemen dalam suatu organisasi.

a. Kelebihan Sistem Informasi

Tingkat kontrol yang diharapkan pembeli untuk menilai transfer teknologi harus menjadi pertimbangan. Keterpaduan tiap langkah yang diproteksi dari penggunaan yang tidak terautorisasi, kemudian keterpaduan tersebut menjadi pilihan untuk memilih *outsourcing*.

1. Umumnya sistem informasi yang dikembangkan sesuai dengan kebutuhan perusahaan karena karyawan yang ditugaskan mengerti kebutuhan sistem dalam perusahaan.
2. Biaya pengembangannya relatif lebih rendah karena hanya melibatkan pihak perusahaan.
3. Sistem informasi yang dibutuhkan dapat segera direalisasikan dan dapat segera melakukan perbaikan untuk menyempurnakan sistem tersebut.

4. Sistem informasi yang dibangun sesuai dengan spesifikasi yang dibutuhkan dan dokumentasi yang disertakan lebih lengkap.
5. Mudah untuk melakukan modifikasi dan pemeliharaan (*maintenance*) terhadap sistem informasi karena proses pengembangannya dilakukan oleh karyawan perusahaan tersebut.
6. Adanya insentif tambahan bagi karyawan yang diberi tanggung jawab untuk mengembangkan sistem informasi perusahaan tersebut.
7. Lebih mudah melakukan pengawasan (*security access*) dan keamanan data lebih terjamin karena hanya melibatkan pihak perusahaan. Sistem informasi yang dikembangkan dapat diintegrasikan lebih mudah dan lebih baik terhadap sistem yang sudah ada.

b. Kekurangan Sistem Informasi

Kelemahan *insourcing* dihubungkan dengan tingkat kebutuhan investasi yang dibutuhkan ketika keputusan *insourcing* dibuat. Kelemahan lain yang sering terjadi adalah ketika dinas mencoba untuk mengubah atau mengambil alternatif yang sesuai dengan kebutuhan pasar atau permintaan pasar. Kecocokan permintaan terhadap kebutuhan berbagai bagian dari rantai suplai (*supply chain*) sangat berbelit-belit pada proses i*nsourcingoses* internal tidak dengan mudah mencapai kesesuaian.

1. Keterbatasan jumlah dan tingkat kemampuan SDM yang menguasai teknologi informasi. Pengembangan sistem informasi membutuhkan waktu yang lama karena konsentrasi staf harus terbagi dengan pekerjaan rutin sehari-hari sehingga pelaksanaannya menjadi kurang efektif dan efisien.
2. Perubahan dalam teknologi informasi terjadi secara cepat dan belum tentu perusahaan mampu melakukan adaptasi dengan cepat sehingga ada peluang teknologi yang digunakan kurang canggih (tidak *up to date*).
3. Membutuhkan waktu untuk pelatihan bagi operator dan *programmer* sehingga ada konsekuensi biaya yang harus dikeluarkan.
4. Adanya demotivasi dari staf ditugaskan untuk mengembangkan sistem informasi karena bukan merupakan *core competency* pekerjaan mereka.

Kurangnya tenaga ahli (*expert*) di bidang sistem informasi dapat menyebabkan kesalahan persepsi dalam pengembangan distem dan kesalahan/resiko yang terjadi menjadi tanggung jawab dinas (ditanggung sendiri).

2.2.2 Pengertian Manajemen

Manajemen adalah suatu ilmu pengetahuan tentang memimpin organisasi yang terdiri atas kegiatan perencanaan, pengorganisasian, pelaksanaan dan pengendalian terhadap sumber-sumber daya yang terbatas dalam usaha mencapai tujuan dan sasaran yang efektif dan efisien[1]. Kegiatan kegiatan dari manajemen adalah sebagai berikut :

8. Perencanaan dilakukan untuk menetapkan sasaran dan tujuan yang harus dicapai.
9. Pengorganisasian pada tahap ini dilakukan pengelompokan jenis-jenis pekerjaan dan menentukan yang bertanggung jawab atas pekerjaan tersebut.
10. Pelaksanaan merupakan implementasi dari perencanaan yang sebelumnya dilakukan.
11. Pengendalian merupakan tahap untuk mengawasi atau memastikan tahapan pelaksanaan sesuai dengan perencanaan dan terdapat penyimpangan rencana yang seminimal mungkin.

Tujuan dari manajemen adalah untuk mendapatkan metode atau cara teknis yang paling baik agar dengan sumber-sumber daya yang terbatas diperoleh hasil maksimal dalam hal ketepatan, kecepatan, penghematan, dan keselamatan kerja secara komprehensif[1].

2.2.3 Pengertian Proyek

Proyek adalah gabungan dari sumber-sumber daya seperti manusia, material, peralatan,dan modal/biaya yang dihimpun dalam suatu wadah organisasi sementara untuk mencapai sasaran dan tujuan[1]. Adapun sifat-sifat dari proyek yaitu memiliki tujuan yang jelas dan unik, berlangsung sementara, membutuhkan

sumber daya yang beragam, memiliki sponsor atau pelanggan sebagai penyedia sumber dana agar proyek sesuai dengan sasaran dan tujuan yang telah ditetapkan.

2.2.4 Pengertian Manajemen Proyek

Manajemen Proyek adalah penerapan ilmu pengetahuan, keahlian dan keterampilan, cara teknis yang terbaik dan dengan sumber daya yang terbatas, untuk mencapai sasaran dan tujuan yang telah ditentukan agar mendapat hasil yang optimal dalam hal kinerja biaya, mutu dan waktu, serta keselamatan kerja[1]. Manajemen proyek adalah ilmu dan seni yang berkaitan dengan memimpin dan mengkoordinir sumber daya yang terdiri dari manusia dan material dengan menggunakan tehnik pengelolaan modern untuk mencapai sasaran yang telah ditentukan, yaitu lingkup, mutu, jadwal, dan biaya, serta memenuhi keinginan para stake holder.

Dalam manajemen proyek, penentuan waktu penyelesaian kegiatan ini merupakan salah satu kegiatan awal yang sangat penting dalam proses perencanaan karena penentuan waktu tersebut akan menjadi dasar bagi perencanaan yang lain,yaitu :

1. Penyusunan jadwal (*scheduling*), anggaran (*budgeting*), kebutuhan sumber daya manusia (*manpower planning*), dan sumber organisasi yang lain.
2. Proses pengendalian (*controlling*).

   Manajemen Proyek meliputi tiga fase, yaitu:

   a. Perencanaan

      Fase ini mencakup penetapan sasaran, mendefinisikan proyek, dan organisasi timnya.

   b. Penjadwalan

      Fase ini menghubungkan orang, uang, dan bahan untuk kegiatan khusus dan menghubungkan masing-masing kegiatan satu dengan yang lainnya.

   c. Pengendalian

      Perusahaan mengawasi sumber daya, biaya, kualitas, dan anggaran. Perusahaan juga merevisi atau mengubah rencana dan menggeser atau

mengelola kembali sumber daya agar dapat memenuhi kebutuhan waktu dan biaya.

Adapun tujuan dari manajemen proyek seperti yang dinyatakan oleh adalah sebagai berikut:

1. Tepat waktu (*on time*)
   yaitu waktu atau jadwal yang merupakan salah satu sasaran utama proyek, keterlambatan akan mengakibatkan kerugian, seperti penambahan biaya, kehilangan kesempatan produk memasuki pasar.
2. Tepat anggaran (*on budget*)
   yaitu biaya yang harus dikeluarkan sesuai dengan anggaran yang telah ditetapkan.
3. Tepat spesifikasi (*on specification*)
   dimana proyek harus sesuai dengan spesifikasi yang telah ditetapkan.

2.2.5 Metode *Earned Value Management*(EVM)

Metode EVM atau metode Nilai Hasil merupakan suatu metode pengendalian kinerja proyek yang lebih progresif. Metode ini bisa memberikan informasi mengenai posisi kemajuan proyek dalam jangka waktu tertentu serta dapat memperkirakan progres proyek pada periode selanjutnya baik dalam hal biaya maupun waktu penyelesaian proyek [1].

Metode ini menggunakan kurva S sebagai tampilan informasi dengan sumbu X menunjukan durasi proyek dan sumbu Y untuk menyatakan kumulatif biaya. Dimana kelebihan dari metode ini adalah sebagai berikut[1] :

1. Tampilan informasi metode *earned value* lebih progresif dibandingkan kurva S konvensional.
2. Metode ini bisa memprediksi kerugian biaya dan waktu berdasarkan progres kerja yang cenderung lambat, sehingga tambahan durasi proyek dan biaya akhir dapat dihitung dengan pendekatan sistematis.
3. Informasi dari predikisi Biaya penyelesaian akhir dan prediksi waktu penyelesaian akhir dapat digunakan untuk melakukan tindakan koreksi berupa

mempercepat progress proyek dengan pertukaran biaya dan waktu atau dengan penambahan tenaga kerja atau lembur serta penjadwalan kembali sumber daya yang misalnya tenaga kerja, peralatan, serta material

Istilah-istilah yang digunakan dalam metode EVM adalah sebagai berikut :

1. *Planned Value*(PV)

   PV atau yang dahulu di kenal dengan *budgeted Cost for Work Scheduled*(BCWS) merupakan biaya yang dianggarkan untuk pekerjaan yang dijadwalkan dalam suatu periode tertentu dan ditetapkan dalam anggaran. Diperoleh dengan mengalikan persentase progres rencana dengan anggaran total proyek (BAC)[5].

PV = (%progres rencana) x BAC

Rumus 2. 1 *Planned Value*(PV)

2. *Earned Value*(EV)

EV atau *Budgeted Cost of Word Performed* (BCWP) merupakan biaya yang dianggarkan untuk pekerjaan yang telah selesai dilaksanakan.Diperoleh dengan mengalikan antara persentase progress yang telah dilaksanakan dengan anggaran total proyek(BAC)[5].

EV = (%progres aktual) x BAC

Rumus 2. 2 *Earned Value*(EV)

3. *Cost Varience* (CV)

   CV merupakan selisih antara nilai yang diperoleh setelah menyelesaikan paket-paket pekerjaan dengan biaya aktual yang terjadi selama pelaksanaan proyek.

CV = EV – AC

Rumus 2. 3 *Cost Varience*(CV)

4. *Schedule Varience* (SV)

   SV digunakan untuk menghitung penyimpangan antara EV dengan PV

$$SV = EV - PV \qquad (2.4)$$

Rumus 2. 4 *Schedule Varience*(SV)

5. *Cost Performance Index* (CPI)

   Merupakan faktor efisiensi biaya yang telah dikeluarkan dapat diperlihatkan dengan membandingkan nilai pekerjaan yang secara fisik telah iselesaikan(EV) dengan biaya yang telah dikeluarkan dalam periode yang sama (AC)[5].

$$CPI = EV / AC \qquad (2.5)$$

Rumus 2. 5 *Cost Permormance Index*(CPI)

6. *Scedule Performance Index* (SPI)

   Merupakan faktor efisiensi kinerja dalam menyelesaikan pekerjaan yang diperlihatkan dengan perbandingan antara nilai pekerjaan yang secara fisik telah diselesaikan (EV) dengan rencana pengeluaran biaya yang dikeluarkan berdasarkan rencana pekerjaan(PV)[5].

$$CPI = EV / PV \qquad (2.6)$$

Rumus 2. 6 *Schedule Performance Index*(SPI)

7. Prediksi biaya penyelesaian akhir (EAC)

   Predikisi biaya penyelesaian akhir proyek dapat dihitung dengan perbandingan antara Total anggaran(BAC) dengan CPI.

$$EAC = BAC/CP$$

Rumus 2. 7 Prediksi biaya penyelesaian akhir(EAC)

8. Prediksi waktu penyelesaian akhir (ETC)

   Prediksi waktu penyelesaian akhir dapat dihitung dengan perbandingan antara durasi proyek(OD) dengan SPI

$$ETC = OD/SPI$$

Rumus 2. 8 Prediksi waktu penyelesaian akhir (ETC)

2.2.6 Metode *Function Point*

Function Point adalah salah satu pendekatan pengukuran perangkat lunak untuk mengukur ukuran sistem berdasar kebutuhan sistem. *Function Point analysis* (FPA) adalah takaran tidak langsung untuk ukuran fungsional suatu sistem[1].

Function Point Analysis dikembangkan pertama kali oleh Allan J. Albrecht di pertengahan 1970 dengan tujuan mencoba menyelesaikan kesulitan terkait dengan *Line of Code* (LOC), dan membantu dalam pengembangan sebuah mekanisme untuk meramalkan beban (*effort*) terkait dengan pengembangan perangkat lunak. *Function Point* pertama dipublikasikan pada tahun 1979, kemudian tahun 1983. Tahun 1984, Albrecht memperbaiki metode ini dan sejak 1986, ketika *International Function Point User Group* (IFPUG) dibentuk, beberapa versi *Function Point Counting Practices Manual* diterbitkan oleh IFPUG. (Albrecht, 1983). Dalam metode *Function Points*, ukuran sebuah sistem dapat dihitung dengan 3 komponen, yaitu *information processing size* (*Unadjusted Function Points-UFP*), *Relative complexity adjustment factors* dan *Function Points*.

1. *Unadjusted Function Points*(UFP)

Unadjusted Function Point di dalam sebuah perangkat lunak diidentifikasi dan dikategorikan ke dalam salah satu dari lima tipe fungsi pengguna, yaitu: *External Input* (EI), *External Outputs* (EO), *Internal Logical File* (ILF), *External Interface Files* (EIF) dan *External Inquiry* (EQ), kemudian dinilai kompleksitasnya dan diberi sejumlah nilai *Function Points*.

External Input (EI) adalah proses dasar yang memproses data dan informasi kontrol yang datang dari luar batasan aplikasi. Data mungkin digunakan untuk memelihara satu atau lebih berkas logika internal. External Input dapat untuk mengubah perilaku sistem[11].

External Outputs (EO) merupakan sebuah proses dasar dimana hasil data dilewatkan dari dalam ke keluar dari batasan aplikasi, contohnya aplikasi menghasilkan berkas XML atau CSV. *External Outputs* menampilkan informasi

melalui logika pemrosesan, tidak hanya mengambil data. *External Outputs* dapat berupa data laporan atau output file yang dikirim ke aplikasi lain. Laporan dan file tersebut dibuat dari satu atau lebih *Internal Logical File* (ILF) dan *External Interface Files* (EIF)[11].

Fungsi utama *External Inquiry* (EQ) adalah menyediakan informasi ke user melalui pengambilan atau pemrosesan data atau informasi kontrol dari *Internal Logical File* (ILF) atau *External Interface Files* (EIF). Tampilan beberapa tipe laporan dan pencarian merupakan komponen yang tepat untuk *External Inquiry*. *External Inquiry* tidak mengubah ILF atau EIF, hanya mengambil data untuk ditampilkan.

Internal Logical File (ILF) adalah kelompok data atau kelompok informasi kontrol yang digunakan dalam aplikasi[11]. Peran utama ILF yaitu menyimpan data yang dipelihara oleh satu atau lebih proses dalam aplikasi, contohnya tabel, file dan informasi kontrol seperti *user preferences*. *External Interface File* (EIF) adalah kelompok data berelasi atau informasi kontrol yang dirujuk oleh aplikasi, tapi dipelihara oleh aplikasi lain. Sebuah *External Interface File* yang dihitung untuk sebuah aplikasi harus merupakan *Internal Logical File* di aplikasi lain.

Setiap bagian dari tipe fungsi kemudian diklasifikasikan berdasarkan 3 (tiga) tingkatan kompleksitas (*simple, average, complex*). Tingkat kompleksitas menentukan bobot yang akan diaplikasikan pada jumlah fungsi untuk menentukan kuantitas *Unadjusted Function Points* (UFP). Adapun kriteria yang menentukan ketiga tingkatan kompleksitas, dapat dilihat pada tabel 2.1 dan pada konstanta bobot tiap tingkatan dapat di lihat pada tabel 2.2.

Tabel 0.1 Kriteria Tingkatan Kompleksitas

Tingkat Kompleksitas	Kriteria
Simple	1. Waktu yang diperlukan untuk pengerjaan sebentar 2. Jumlah sub menu sedikit 3. Dapat dikerjakan oleh 1 orang
Average	1. Waktu yang diperlukan untuk pengerjaan rata-rata 2. Jumlah sub menu rata-rata

	3. Dapat dikerjakan oleh 1 orang walau memakan waktu agak lama dari seharusnya
Complex	1. Waktu yang diperlukan untuk pengerjaan lama 2. Jumlah sub menu banyak 3. Harus dikerjakan bersama-sama 4. Menggunakan algoritma yang cukup rumit

Tabel 0.2 konstanta bobot level kompleksitas

Komponen Sistem	**Level Kompleksitas**									**Total CFP**
	Simple			***Average***			***Complex***			
	A	B	AxB=C	D	E	DxE=F	G	H	DxH=I	CFP = C+F+I
Exsternal inputs(EI)	?	3	?	?	4	?	?	6	?	?
Exsternal Output(EO)	?	4	?	?	5	?	?	7	?	?
Internal Logical Files (ILF)	?	7	?	?	10	?	?	15	?	?
External interface Files (EIF)	?	5	?	?	7	?	?	10	?	?
External Inquiry (EI)	?	3	?	?	4	?	?	6	?	?
									Total	?

Keterangan :

- Kolom A , D dan G adalah jumlah inputan pada setiap proses.
- Kolom B, E dan H adalah nilai bobot dari setiap komponen level kompleksitas, nilai tersebut adalah konstanta yang dibuat oleh *Function Point International User Group*(IFPUG).
- Kolom C, F dan H adalah point yang di dapat dari hasil perkalian jumlah inputan dan bobot setiap komponen.

2. *Relative complexity adjustment factors (RCAF)*

Setelah mengakumulasi UFP kemudian menghitung akumulasi RCAF. RCAF dihitung berdasarkan pada keseluruhan kompleksitas sistem dengan menggunakan 14 *General System Characteristics*(GCS), dimana nilai masing-

masing GCS berskala 0(nol) sampai 5(lima). Skala 0(nol) menunjukan tidak adanya pengaruh dan skala 5(lima) penunjukan adanya pengaruh yang luas tehadap keseluhan proyek. Pemberian nilai dari masing-masing GSC dilakukan secara subyektif, maka sari itu untuk menentukan pemberian pada bobot apada proses ini dibutuhkan orang yang sudah berpengalaman atau sudah pernah menangani aplikasi dengan spesifikasi sistem hampir mirip. Tabel 2.3 memperlihatkan 14 *General System Characteristics* yang akan diberikan penilaian. Penilaian Kompleksitas memiliki skala 0 s/d 5 yaitu :

0 = Tidak berpengaruh , 1 = Insidental , 2 = Moderat, 3 = Rata-rata, 4 = Signifikan, 5 = Essential.

Tabel 0.3 14 *General system characteristic*

General System Characteristic	**Bobot**
Tingkat kompleksitas komunikasi data	[0/1/2/3/4/5]
Tingkat kompleksitas pemrosesan data	[0/1/2/3/4/5]
Tingkat kompleksitas performasnce	[0/1/2/3/4/5]
Tingkat kompleksitas konfigurasi	[0/1/2/3/4/5]
Tingkat frekuensi penggunaa software	[0/1/2/3/4/5]
Tingkat frekuensi input data	[0/1/2/3/4/5]
Tingkat kemudahaan penggunaan bagi *user*	[0/1/2/3/4/5]
Tingkat frekuensi update data	[0/1/2/3/4/5]
Tingkat kompleksitas prosesing data	[0/1/2/3/4/5]
Tingkat kemungkinan penggunaan kembali/*reusable* kode program	[0/1/2/3/4/5]
Tingkat kemudahan dalam instalasi	[0/1/2/3/4/5]
Tingkat kemudahan operasional *software*(*backup*,*recovery*,dsb)	[0/1/2/3/4/5]
Tingkat *software* dibuat untuk multi organisasi/perusahaan/klien	[0/1/2/3/4/5]
Tingkat kompleksitas dalam mengikuti perubahan/fleksibel	[0/1/2/3/4/5]
Total RCAF	**?**

Karakteristik diatas merupakan ketetapan atau konstanta yang dibuat oleh *Function Point Internasional User Group* (IFPUG).

3. *Function Point (FP)*

Function Point Adalah proses melakukan perhitungan untuk mendapat nilai *Function point* dari *software* yang akan dibangun. FP didapat dari perkalian CFP dan RCAF dengan rumus :

$$FP = UFP*(0.65+(0.01*RCAF)) \qquad (2.9)$$

Rumus 2. 9 *Function Point* (FP)

Keterangan :

Angka 0.65 dan 0.01 adalah ketetepan atau konstanta yang dibuat oleh *Function Point Internasional User Group* (IFPUG).

2.2.7 Metode *Probability Impact Matrix*(PIM)

Probability Impact Matrix(PIM) adalah sebuah pendekatan yang dikembangkan menggunakan dua kriteria yang penting untuk mengukur risiko[18], yaitu :

1. Kemungkinan (*Probability*), adalah kemungkinan (*Probability*) dari suatu kejadian yang tidak diinginkan.
2. Dampak (*Impact*), adalah tingkat pengaruh atau ukuran dampak (*Impact*) pada aktivitas lain, jika peristiwa yang tidak diinginkan terjadi.

Penilaian resiko pada dasarnya mengacu pada dua faktor, yaitu kuantitas resiko dan kualitas resiko. Kuantitas resiko terkait dengan berapa banyak nilai, atau dampak, yang rentan terhadap resiko sedangkan kualitas resiko terkait dengan kemungkinan suatu resiko muncul. Tujuan penilaian resiko adalah untuk mendapatkan daftar resiko yang telah dinilai berdasarkan tingkat dampak dan kemungkinan terjadinya. Hasil penilaian resiko tersebut kemudian dipetakan untuk mengetahui resiko-resiko utama yang harus menjadi menjadi prioritas untuk ditangani.

Penilaian probabilitas dari setiap resiko dan dampak yang ditimbulkan dibuat dalam suatu skala yaitu 0 sampai 1, dimana skala tersebut menyatakan tingkatan dari rendah, sedang dan tinggi, seperti dijelaskan pada tabel 2.4.

Tabel 0.4 Nilai Skala Resiko

Skala	Nilai Resiko
0 - 0.30	Rendah
0.31 - 0.7	Sedang
0.71 – 1	Tinggi

Tingkat kepentingan resiko merupakan perkalian dari nilai probabilitas dan nilai dampak yang didapat dari responden[18]. Nilai resiko merupakan perkalian dari nilai probabilitas dan nilai dampak, nilai resiko didapat dari responden. Dimana nilai terdiri dari suatu skala yaitu 0 sampai 1 yang menyatakan tingkatan dari rendah, sedang dan tingginya probabilitas dan dampak dari masing-masing resiko. Untuk mengukur tingkat kepentingan resiko dapat menggunakan rumus sebagai berikut :

$$Risk\ Exposure = probality(outcome) * Loss(outcome) \quad (2.10)$$

Rumus 2. 10 *Risk Exposure*

Dimana :

Risk Exposure = Tingkat kepentingan risiko

probality(outcome) = nilai probabilitas resiko

Loss(outcome) = nilai dampak resiko

Tabel 0.5 Contoh hasil perhitungan tingkat kepentingan resiko

Kode Resiko	Probabilitas	Dampak	Tingkat Kepentingan Resiko
R1	0.7	0.8	0.56
R2	0.2	0.7	0.14
R3	0.1	0.7	0.7
R4	0.3	0.5	0.15
R5	0.5	0.5	0.25

Berdasarkan hasil perhitungan tingkat kepentingan resiko, kemudian dibuat matriks resikonya. Tabel 2.6 merupakan matriks resiko yang dihasilkan dari perhitungan kepentingan resiko

Tabel 0.6 Matriks Resiko

Probabilitas / Dampak	Rendah (0 – 0,3)	Sedang (0,31 – 0,7)	Tinggi (0,71 – 1)
Tinggi (0,71 – 1)		R4,	R1
Sedang (0,31-0,7)	R2,R3	R5	
Rendah (0-0,3)			

Proses pengerjaan matriks probabilitas dan dampak adalah dengan cara memplotkan nilai risiko yang telah di dapat ke dalam matriks. Setelah itu di dapat nilai yang dijadikan acuan untuk mengetahui resiko-resiko mana saja yang kemungkinan terjadinya besar dan menimbulkan dampak yang signifikan.

2.2.8 Manajemen Resiko

Konsep manajemen risiko mulai diperkenalkan di bidang keselamatan dan kesehatan kerja pada era tahun 1980-an setelah berkembangnya teori ***accident model*** dari ILCI dan juga semakin maraknya isu lingkungan dan kesehatan. Tujuan dari manajemen risiko adalah minimisasi kerugian dan meningkatkan kesempatan ataupun peluang. Bila dilihat terjadinya kerugian dengan teori ***accident model*** dari ILCI, maka manajemen risiko dapat memotong mata rantai kejadian kerugian tersebut, sehingga efek dominonya tidak akan terjadi. Pada dasarnya manajemen risiko bersifat pencegahan terhadap terjadinya kerugian maupun '*accident*'[8].

Ada beberapa ruang lingkup manajemen resiko Berikut ini adalah langkah-langkah yang dilakukan dalam menajemen resiko ,yaitu :

1. Perencanaan Manajemen Resiko

Perencanaan Manajemen Resiko merupakan proses yang memutuskan tentang pendekatan yang akan dilakukan, dan bagaimana melaksanakan kegiatan manajemen resiko untuk suatu proyek. Pada tahap ini menentukan konteks kegiatan yang akan dikelola resikonya.

Masukan pada ruang lingkup ini adalah sebagai berikut :

1. Faktor-faktor lingkungan organisasi
2. Aset proses organisasi

3. Pernyataan cakupan proyek
4. Rencana manajemen proyek

2. Identifikasi Resiko

identifikasi resiko bertujuan mengidentifikasi serta membuat daftar resiko yang mungkin terjadi. Proses identifikasi kejadian ini dilakukan dengan pendekatan diskusi dan wawancara dengan pihak perusahaan yang menghasilkan daftar lengkap resiko. Identifikasi dikelompokan berdasarkan jenis resikonya.

3. Analisis Kemungkinan dan Konsekuensi Resiko

Pada tahap ini, berdasarkan identifikasi risiko yang telah dilakukan sebelumnya, dilakukan pengidentifikasian mengenai probabilitas terjadinya resiko beserta dampak yang mungkin ditimbulkan jika risiko tersebut terjadi, sehingga akan dihasilkan tingkat kepentingan dari masing-masing resiko. Analisis tersebut menggunakan metode *Probability Impact Matrix*(PIM).

4. Mitigasi Resiko

Dalam melakukan penanganan terhadap resiko terdapat empat Alternatif tindakan yang dapat dilakukan, yaitu:

a. Menerima Resiko(*Acceptance*)

   Acceptance adalah penerimaan risiko beserta konsekuensinya, yaitu tindakan perusahaan untuk menerima suatu risiko dengan tidak melakukan tindakan berarti yang memerlukan sumber daya yang besar. Tindakan ini biasanya diterapkan pada risiko-risiko yang tingkat risikonya rendah bagi perusahaan, sehingga apabila dilakukan penanganan residual risk menimbulkan biaya yang tidak sebanding dengan keuntungannya.

b. Menghidari Resiko(*Avoidance)*

   Avoidance adalah tindakan perusahaan untuk tidak melakukan usaha tertentu yang mengandung risiko yang tidak diinginkan. Tindakan ini biasanya diterapkan pada risiko-risiko yang tingkat risikonyatidak dapat diterima oleh perusahaan atau berdampak sangat tinggi bagi perusahaan,

dimanapenanganannya akan menimbulkan biaya yang sangat tinggi serta tidak efisien.

c. Mengurangi Resiko(*Mitigation*)

Mitigation adalah tindakan perusahaan dengan menggunakan semua sumber daya yang dimilikinya berusaha untuk dapat meminimalkan risiko tanpa menghilangkan peluang perusahaan untuk meraih keuntungan.

Tindakan ini dapat dilakukan terhadap salah satu dari kedua faktor, yaitu:

a. Mengurangi kemungkinan terjadinya risiko, biasanya dengan melakukan proses perubahan desain dan engineering, prosedur quality assurance atau audit secara periodik.
b. Mengurangi dampak akibat terjadinya suatu risiko, biasanya diterapkan pada resiko yang berdampak tinggi dan kemungkinannya rendah, antaralain dengan membuat rencana kontinjensi atau rencana evakuasi.

d. Membagi Resiko(*Transfer*)

Transfer adalah tindakan perusahaan untuk memindahkan risiko kepada pihak ketiga yang dapat mengelola risiko antara lain melalui kesepakatan kontrak dengan asuransi.

2.3 Internet

Interconnected Network atau yang lebih populer dengan sebutan internet adalah sebuah sistem komunikasi global yang menghubungkan komputer-komputer dan jaringan-jaringan komputer di seluruh dunia. Secara harafiah internet adalah rangkaian komputer yang terhubung ke beberapa jaringan lain. Ketika komputer terhubung secara global dengan menggunakan TCP/IP sebagai protokol pertukaran paket data, Protokol TCP/IP ini memberikan suatu *IP Number* yang unik untuk tiap komputer yang terhubung ke Internet sehingga lalu lintas data di Internet dapat diatur. Demi memudahkan kita mengingat maka dibuatlah *IP Address* sebagai representasi dari nomor IP. [16]

Contoh nomor IP : 192.168.0.1

Contoh alamat IP : namadomain.com

Internet dapat menghubungkan jaringan-jaringan sederhana yang terdiri dari beberapa komputer disuatu ruangan hingga jaringan-jaringan canggih yang merentang antarbenua dan menghubungkan jutaan komputer.[16]

Sebelum internet tercipta, militer Amerika Serikat telah mengembangkan jaringan komunikasi komputer yang dinamai ARPANET atau *Advanced Research Project Agency Network*. Penggunaan jaringan awal ini dibatasi untuk personel-personel militer dan para peneliti yang mengembangkan teknologi ini. Banyak orang menganggap ARPANET sebagai pendahulu dari internet.

Pada rentang waktu antara 1970-an hingga akhir 1980-an, internet merupakan jaringan yang dibiayai oleh pemerintah Amerika Serikat dan digunakan secara ekslusif untuk penggunaan akademis dan militer. Secara umum, ia dikendalikan oleh *National Science Foundation* (NSF), sebuah badan ilmu pengetahuan yang berada di amerika serikat. Pada tingkat universitas hanya peneliti-peneliti tertentu yang memiliki akses ke internet. Akhirnya, demi kemajuan ilmu pengetahuan, pada tahun 1980-an, NSF melonggarkan batasan-batasannya dan mengizinkan fakultas-fakultas di universitas-universitas di seluruh bagian Amerika Serikat menggunakan internet untuk aktifitas-aktifitas penelitian dan perkuliahan. Dalam hal ini kebijakan NSF masih melarang penggunaan internet untuk keperluan komersial.

Pada tahun 1970-an, ARPA menghadapi 2 dilema: masing-masing jaringan memiliki keunggulan teknis untuk beberapa situasi, tetapi tidak saling kompatibel satu dengan yang lainnya. Kemudian pada masa-masa berikutnya, ARPA memfokuskan diri pada bagaimana caranya agar jaringan-jaringan yang tidak saling kompatibel tadi dapat saling dihubungkan, sehingga pada akhirnya lahirlah internet yang diciptakan dengan tujuan untuk menghubungkan jaringan-jaringan tersebut [15].

1. Aplikasi Web

Aplikasi web adalah sebuah sistem informasi yang mendukung interaksi pengguna melalui antarmuka berbasis web[17]. Aplikasi web mempunyai karakteristik-karakteristik sebagai berikut :

1. Aplikasi web akan secara konstan meningkat.
2. Aplikasi web tidak terpisahkan dari perangkat lunak. Konten yang dapat mencankup teks, grafis, citra, audio, dan/atau video terintegrasi dengan pengolahan prosedural.
3. Aplikasi bertujuan untuk digunakan secara luas oleh komunitas penggunan yang variabel, dan sejumlah besar pengguna anonym dengan berbagai kebutuhan, harapan, dan sekumpulan keahlian. Oleh karena itu, antarmuka pengguna dan fitur-fitur usabilitas harus memenuhi kebutuhan yang berbeda.
4. Sistem berbasis web adalah content-driven. Pengembangan sistem berbasis web meliputi penciptaan dan manajemen dari konten, seperti ketentuan manajemen setelah penyebaran dan pengembangan awal pada basis kesinambungan.
5. Secara umum kebanyakan sistem berbasis web menyokong kreativitas visual dan inkorporasi multimedia dalam persentasi dan antarmuka.
6. Aplikasi web mempunyai jadwal pengembangan yang didapatkan dan tekanan waktu yang padat.
7. Pencabangan kegagalan atau ketidakpuasan pengguna IT konvensional.

2. Web Server

Web server adalah sebuah perangkat lunak *server* yang berfungsi menerima permintaan HTTP atau HTTPS dari client yang dikenal dengan *web browser* dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk dokumen HTML[17]. *Web server* yang terkenal di antaranya adalah *Apache* dan *Microsoft Internet Information Service* (IIS). *Apache* merupakan *web server* antar-platform, sedangkan IIS hanya dapat beroperasi di sistem operasi Windows.

Fungsi utama dari sebuah *web server* adalah memberikan halaman *web* untuk *client. Client* dalam hal ini *web browser* memulai komunikasi dengan membuat permintaan untuk suatu sumber daya tertentu menggunakan HTTP dan

server merespon dengan isi dari sumber daya tersebut atau pesan kesalahan jika permintaan tidak dapat direspons oleh *server*.

3. Web Browser

Dalam dunia *web*, perangkat lunak *client*, yaitu *browser web* mempunyai tugas yang sama yaitu menterjemahkan informasi yang diterima oleh *server web* dan menampilkannya pada *layer* komputer pengguna, oleh karena HTTP memungkinkan *server web* mengirimkan beragam data, seperti teks atau gambar, *browser* harus bisa mengenali berbagai macam data yang akan diterimanya, dan selanjutnya harus tahu cara untuk menampilkannya dengan benar. Teks ditampilkan sebagai teks dan gambar ditampilkan sebagai gambar. Umumnya *webbrowser* menerima data dalam bentuk HTML. File HTML sebenarnya adalah *file* teks biasa yang selain berisi informasi yang hendak ditampilkan kepada pengguna, juga mempunyai perintah-perintah untuk mengatur tampilan data tersebut. *Browser*-lah yang memiliki kuasa penuh dalam menerjemahkan perintah-perintah tadi. Meskipun sudah dibuat konsensus untuk menstandarkan format dan elemen-elemen HTML, setiap jenis *browser* bisa menterjemahkan *file* HTML secara berbeda.

2.4 Perancangan *Database*

Merancang *database* merupakan suatu hal yang sangat penting. Kesulitan utama dalam merancang *database* adalah bagaimana merancang dan mengakomodir penyimpanan data yang baik sehingga data tersebut dapat memuaskan keperluan saat ini dan masa mendatang. Konsep *database* merupakan langkah untuk menentukan *database* yang diharapkan dapat mewakili seluruh kebutuhan pengguna.

1. Pengertian *Database*

Database adalah kumpulan file - file yang mempunyai kaitan antara satu file dengan file lain sehingga membentuk satu bangunan data untuk menginformasikan satu perusahaan, instansi dalam batasan tertentu.

Basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti :

a. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
c. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Prinsip utama basis data adalah pengaturan data/arsip. Tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data/arsip. Yang sangat ditonjolkan dalam basis data adalah pengaturan / pemilahan / pengelompokkan /pengorganisasian data yang akan disimpan sesuai fungsi/jenisnya. Pemilahan /pengelompokkan / pengorganisasian Ini dapat berbentuk sejumlah *file* / tabel terpisah atau dalam bentuk pendefinisian kolom-kolom / *field-field* data dalam setiap *file* / tabel. *Database* dapat diciptakan dan dapat pula ditiadakan. Setiap *Database* umumnya dibuat untuk mewakili sebuah semesta data yang spesifik. Misalnya, *Database* akademik, dan dalam basis data akademik kita dapat menempatkan *file* mahasiswa, file matakuliah, *file* dosen, *file* jadwal, *file* kehadiran, *file* nilai, dan sebagainya

2.5 UML

UML singkatan dari *Unified Modeling Language* yang berarti bahasa pemodelan standar. UML memiliki sintaks dan sematik. Ketika kita membuat model menggunakan konsep UML ada aturan-aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat berhubungan satu dengan lainnya harus mengikuti standar yang ada.[13]

UML diaplikasikan untuk maksud tertentu, antara lain untuk :

1. Merancang perangkat lunak
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis

3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

Pada UML 2.3 ada 9 macam diagram yang dikelompokkan menjadi tiga kategori, yaitu :

1. Structure diagram, yaitu kumpulan diagram yang digunakan untuk menggammbarkan suatu struktur statis dalam sistem yang dimodelkan.
2. Behavior diagram, yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. Interaction diagram, yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antarsubsistem pada suatu sistem.

1. Class Diagram

Class Diagram adalah sebuah spesifikasi yang jika diinstansikan akan menghasilkan sebuah obyek dan merupakan inti dari pengembangan dan desain berorientasi objek.[13]

Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi). Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan *object* beserta hubungan satu sama lain seperti *containment*, *pewarisan*, *asosiasi*, dan lain-lain. Sebuah class memiliki tiga area pokok:

1. Nama , merupakan nama dari sebuah kelas.
2. Atribut, merupakan properti dari sebuah kelas. Atribut melambangkan batas nilai yang mungkin ada pada obyek dari *class*.
3. Operasi, adalah sesuatu yang bisa dilakukan oleh sebuah atau yang dapat dilakukan oleh *class* lain terhadap sebuah *class*. Atribut dan metoda dapat memiliki salah sifat berikut:
    a. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan

b. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya.
c. *Public*, dapat dipanggil oleh siapa saja.
d. *Package*, hanya dapat dipanggil oleh instance sebuah *class* pada paket yang sama.

Class juga dapat merupakan implementasi dari sebuah *interface*, yaitu *class abstrak* yang hanya memiliki metode. *Interface* tidak dapat langsung diintansikan, tetapi harus diimplementasikan dahulu menjadi sebuah class. Dengan demikian *interface* mendukung resolusi metoda pada saat *runtime*.

Relasi atau *relationship* merupakan keterhubungan antar kelas yang muncul pada saat sebuah kelas berinteraksi dengan kelas-kelas lainnya.

1. Asosiasi (*Association*)

   Para pemodel menggunakan pemahaman asosiasi adalah pada saat beberapa kelas saling terhubung satu sama lain secara konseptual. Asosiasi juga dapat menjadi lebih kompleks pada saat beberapa kelas terhubung ke satu kelas
2. *Constraint* pada Asosiasi

   Kadangkala sebuah asosiasi diantara dua *class* harus mengikuti sebuah aturan dan aturan ini bias diletakkan dalam sebuah *constraint* pada garis asosiasi dan diletakkan dalam kurung kurawal. Bentuk lain dari tipe *constraint* adalah relasi *OR* yang ditulis dengan {*or*} dalam garis putus-putus yang menghubungkan dua garis asosiasi.
3. Kelas Asosiasi (*Association Class*)

   Sebuah asosiasi dapat memiliki atribut dan operasi sepertinya halnya sebuah *class*. Sebuah *association class* sebenarnya diperlukan apabila salah satu dari kelas yang terhubung mempunya sebuah atau beberapa atribut yang tidak layak dimiliki kelas tersebut, karena secara logis atribut tersebut lebih layak dimiliki oleh asosiasi yang menghubungkan kedua kelas tersebut.
4. Multiplisitas (*Multiplicity*)

Multiplisitas atau kardinalitas adalah jumlah banyaknya objek sebuah *class* yang berelasi dengan sebuah objek lain pada *class* lain yang berasosiasi pada *class* tersebut.

2. Use Case Diagram

Use case adalah deskripsi fungsi dari sebuah sistem dari perspektif / sudut pandang para pengguna sistem. *Use case* mendefenisikan "apa" yang dilakukan oleh sistem dan elemen-elemennya saling berinteraksi [13]. *Use case* bekerja dengan menggunakan "s*cenario*", yaitu deskripsi urutan-urutan langkah yang menerangkan apa yang dilakukan pengguna terhadap sistem maupun sebaliknya.

Fungsi dari diagram *use case* adalah menggambarkan fungsionalitas yang diharapkan dari sebuah sistem, dengan menekankan aspek-aspek apa yang dilakukan sistem bukan bagaimana sistem melakukannya. Sebuah *use case* merepresentasikan sebuah interaksi antara *actor* dengan sistem.

3. Activity Diagram

Activity diagram digunakan untuk mendokumentasikan alur kerja pada sebuah sistem, yang dimulai dari pandangan *business* level hingga ke operational *level* [13]. Pada dasarnya, *activity diagram* merupakan variasi dari *statechart diagram*. *Activity diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku parallel sedangkan flowchart tidak bisa.

4. Sequence Diagram

Sequence diagram menurut Munawar adalah grafik dua dimensi dimana objek ditunjukkan dalam dimensi horizontal, sedangkan *lifeline* ditunjukkan dalam dimensi vertikal[13].

5. Component Diagram

Komponen perangkat lunak adalah bagian fisik dari sebuah sistem yang menetap di komputer. komponen merupakan implementasi *software* dari sebuah

class. Komponen bisa berupa tabel, file data, file exe, file DLL, dokumen dan lain-lain. *Component diagram* mengandung komponen, *interface* dan *relationship*. Komponen diagram ini digunakan pada saat anda ingin memecah sistem menjadi komponen-komponen dan ingin menampilkan hubungan-hubungan mereka dengan antarmuka atau pemecahan komponen menjadi struktur yang lebih rendah.

6. Collaboration Diagram

Collaboration diagram menurut Munawar adalah perluasan dari objek diagram. Objek diagram menunjukkan objek-objek yang hubungannya satu dengan yang lain. *Collaboration diagram* menunnjukkan *message-message* objek yang dikirim satu sama lain [13].

2.6 Pengujian

Pengujian *software* adalah proses verifikasi dan validasi apakah sebuah aplikasi *software* atau program memenuhi persyaratan bisnis dan persyaratan teknis yang mengarahkan desain dan pengembangan dan cara kerjanya seperti yang diharapkan dan juga mengidentifikasi kesalahan yang penting yang digolongkan berdasarkan tingkat *severity* pada aplikasi yang harus diperbaiki[18].

Pengujian *software* adalah teknik yang sering digunakan untuk verifikasi dan validasi kualitas suatu *software*. Pengujian *software* adalah prosedur untuk eksekusi sebuah program atau sistem dengan tujuan untuk menemukan kesalahan.

Kesimpulan yang dapat diambil dari pendapat-pendapat tersebut adalah pengujian *software* merupakan proses verifikasi dan validasi apakah *software* memenuhi *requirement* dan mengidentifikasi kesalahan-kesalahan yang ditemukan saat eksekusi program.

2.6.1 *Behavioral (Black-Box) Tests*

Tester menggunakan *behavioral test* (disebut juga *Black-Box Tests*), sering digunakan untuk menemukan *bug* dalam *high level operations*, pada tingkatan fitur, profil operasional dan skenario *customer*. *Tester* dapat membuat pengujian fungsional *black box* berdasarkan pada apa yang harus sistem

lakukan. *Behavioral testing* melibatkan pemahaman rinci mengenai domain aplikasi, masalah bisnis yang dipecahkan oleh sistem dan misi yang dilakukan sistem. *Behavioral test* paling baik dilakukan oleh penguji yang memahami desain sistem, setidaknya pada tingkat yang tinggi sehingga mereka dapat secara efektif menemukan *bug* umum untuk jenis desain.

Black box testing juga disebut *functional testing*, sebuah teknik pengujian fungsional yang merancang *test case* berdasarkan informasi dari spesifikasi.

2.6.2 Pengujian Beta

Pengujian Beta merupakan pengujian yang dilakukan secara objektif dimana diuji secara langsung ke lapangan atau ke tampat dimana sistem di implementasikan. Pengujian ini bersangkutan mengenai kepuasan pengguna dengan kandungan poin pemenuhan kebutuhan dari tujuan awal pembangunan sistem dan tampilan antarmuka dari sistem tersebut. Pengujian beta dilakukan melalui sebuah teknik pengambilan data, baik itu melalui wawancara atau kuesioner.

2.7 *Tools*

Pada sub bab ini akan menjelaskan peralatan-peralatan yang dibutuhkan dalam membangun perangkat lunak.

1. Notepad++

Notepad++ merupakan sebuah *text editor* yang sangat berguna bagi setiap orang dan khususnya bagi para *developer* dalam membuat program. Notepad++ menggunakan komponen Scintilla untuk dapat menampilkan dan menyuntingan teks dan berkas kode sumber berbagai bahasa pemrograman yang berjalan diatas sistem operasi *Microsoft Windows*.

Selain manfaat dan kemampuannya menangani banyak bahasa pemrograman, Notepad ++ juga dilisensikan sebagai perangkat *free*. Jadi, setiap orang yang menggunakannya tidak perlu mengeluarkan biaya untuk membeli

aplikasi ini karena sourceforge.net sebagai layanan yang memfasilitasi Notepad ++ membebaskannya untuk digunakan.

Bebebapa daftar bahasa program yang didukung oleh Notepad++ adalah C, C++, Java, C#, XML, HTML, PHP, Javascript.

2. PHP

PHP adalah sebuah bahasa pemograman berbasis web yang mempunyai banyak keunggulan dibandingkan dengan bahasa pemograman berbasis web yang lain[17]. PHP merupakan bahasa pemograman yang bersumber Perl. Sedangkan Perl merupakan pengembangan dari bahasa C. Fungsi yang dimiliki oleh PHP sangat lengkap sehingga tidak perlu membuat fungsi sendiri karena daftar fungsi PHP yang lengkap menjadikan baris perintah semakin efisien.

3. HTML

HTML (*Hyper Text Markup Language*) adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman web dan menampilkan berbagai informasi di dalam sebuah browser internet[17]. HTML juga berupa kode – kode tag yang menginstruksikan browser untuk menghasilkan sesuai yang diinginkan. Sebuah file yang merupakan file HTML dapat dibuka dengan menggunakan browser web seperti Mozilla Firefox atau Microsoft Internet Explorer.

4. CSS

Cassading Style Sheet atau yang sering disebut CSS. Berisi rangkain intruksi yang menentukan bagaimana suatu text akan tertampil di halaman web. Perancangan desain text dapat dilakukan dengan mendifinisikan fonts (huruf), color (warna), margins (ukuran), latar belakang (background), ukuran font (font size).

5. Jquery

JQuery adalah salah satu library javascript. Dengan JQuery, kita dapat melakukan banyak hal yang tidak bisa dilakukan oleh HTML maupun CSS[17].

Misalya, menampilkan artikel tanpa me-reupload-halaman, memunculkan pop-up di tengah-tengah halaman, menyembunyikan artikel jika diklik, dan sebagainya.

6. Framework Code Igniter

CodeIgniter sebuah web *application framework* yang bersifat *open source* digunakan untuk membangun aplikasi php dinamis. Tujuan utama pengembangan Codeignter adalah untuk membantu *developer* untuk mengerjakan aplikasi lebih cepat daripada menulis semua code dari awal. Codeigniter menyediakan berbagai macam library yang dapat mempermudah dalam pengembangan. Codeigniter diperkenalkan kepada publik pada tanggal 28 februari 2006.

Codeigniter sendiri dibangun menggunakan konsep *Model-View-Controller development patern* Codeigniter sendiri merupakan salah satu *framework* tercepat dibandingkan dengan *framework* lainnya. Pada acara frOSCon(August 2008), pembuat php Rasmus Lerdorf mengatakan dia menyukai codeigniter karena lebih ringan dan cepat dibandingkan framework lainnya("*Because it is faster, lighter and least like a framework*")[12].

7. Xampp

XAMPP adalah program aplikasi pengembang yang berguna untuk pengembangan website berbasis PHP dan MySQL. Software XAMPP dibuat dan dikembangkan oleh *Apache Friends*[17]. Perangkat lunak komputer ini memiliki kelebihan untuk bisa berperan sebagai *server* web *Apache* untuk simulasi pengembangan *website*. *Tool* pengembangan web ini mendukung teknologi web populer seperti PHP, MySQL, dan Perl. Melalui program ini, programmer web dapat menguji aplikasi web yang dikembangkan dan mempresentasikannya ke pihak lain secara langsung dari komputer, tanpa perlu terkoneksi ke internet. XAMPP juga dilengkapi fitur manajemen *database* PHPMyAdmin seperti pada *server hosting* sungguhan, sehingga pengembang web dapat mengembangkan aplikasi web berbasis *database* secara mudah. Program XAMPP banyak diaplikasikan dan digunakan oleh kalangan pengguna komputer di bidang

pemrograman web. XAMPP merupakan *software* gratis. XAMPP dapat dijalankan di sistem operasi *Windows* 2000/XP/Vista/7 dan sistem operasi lain.

Berikut ini adalah Penjelasan dari bagian-bagian XAMPP :

1. X ,(Sistem Operasi) XAMPP bisa dijalankan di 4 OS besar yang sering digunakan oleh pengguna komputer saat ini. Dan 4 OS tersebut tidak lain dan tidak bukan adalah Windows, Linux, Mac OS dan Solaris.
2. A(Apacahe) merupakan aplikasi web server. Apache ini bersifat opensource yang berarti gratis dan bisa diedit oleh penggunanya. Tugas utama Apache adalah menghasilkan halaman web yang benar kepada user berdasarkan kode PHP yang dituliskan oleh pembuat halaman web. jika diperlukan juga berdasarkan kode PHP yang dituliskan,maka dapat saja suatu database diakses terlebih dahulu (misalnya dalam MySQL) untuk mendukung halaman web yang dihasilkan.
3. M (MySQL), merupakan aplikasi database server. Perkembangannya disebut SQL yang merupakan kepanjangan dari Structured Query Language. SQL merupakan bahasa terstruktur yang digunakan untuk mengolah database. MySQL dapat digunakan untuk membuat dan mengelola database beserta isinya. Kita dapat memanfaatkan MySQL untuk menambahkan, mengubah, dan menghapus data yang berada dalam *database*.
4. P (PHP), bahasa pemrograman web. Bahasa pemrograman PHP merupakan bahasa pemrograman untuk membuat web yang bersifat *server-side scripting*. PHP memungkinkan kita untuk membuat halaman web yang bersifat dinamis. Sistem manajemen basis data yang sering digunakan bersama PHP adalah MySQl. namun PHP juga mendukung sistem manajement database Oracle, Microsoft Access, Interbase, d-base, PostgreSQL, dan sebagainya.

P (Perl), bahasa pemrograman, pertama kali dikembangkan oleh Larry Wall di mesin Unix. Perl pertama kali dirilis pada tanggal 18 Desember 1987 ditandai dengan keluarnya Perl 1. Dua diantara karakteristik utama perl adalah penanganan teks dan berbagai jalan pintas untuk meyelesaikan persoalan-persoalan umum.Perl

sangat populer di gunakan dalam program-program CGI (*Common Gateway Interface*) dan protokol internet lainnya.