BAB II
TINJAUAN PUSTAKA

II.1 Profil Perusahaan

Profil Perusahaan adalah untuk mengetahui keadaan di perusahaan diantaranya adalah sejarah perusahaan, struktur organisasi perusahaan, dan deskripsi tugas.

II.1.1 Sejarah Perusahaan

Yayasan Bekam Ruqyah Center (BRC) awalnya dirintis serta dipimpin oleh Bapak Asep Hasan Badri, tanggal 29 Juni 2004 merupakan awal BRC hadir di pentas sejarah kebangkitan Islam. BRC hadir sebagai jawaban atas miskinnya pengobatan islam, hadir pula sebagai jawaban dari kondisi umat islam yang tenggelam dalam pengobatan yang tidak syar'I, terjebak syirik banyak menimbulkan mudharat dan sebagainya.

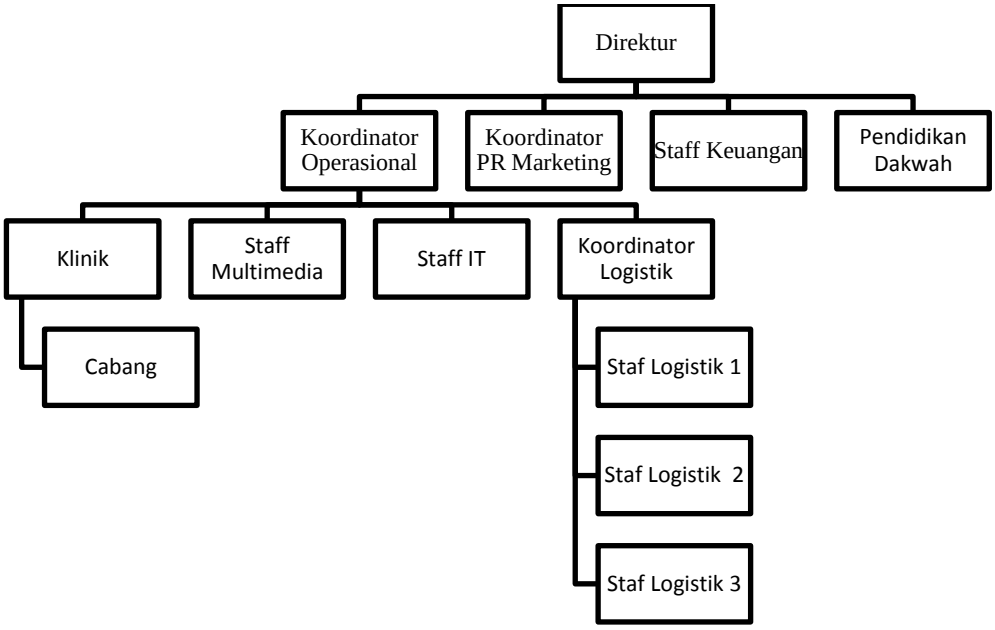
Yayasan Bekam Ruqyah Center pada awalnya menggunakan nama "Bandung Ruqyah Center". Yayasan Bekam Ruqyah Center berkembang sangat pesat, dan mulai membuka cabang di Jabodetabek. Yayasan Bekam Ruqyah Center pada tahun 2008 merubah nama "Bandung Ruqyah Center" menjadi "Bekam Ruqyah Center" karena telah memiliki cabang di seluruh Indonesia.

Dalam kurun waktu perjalanan yang baru berjalan kurang dari 5 tahun BRC telah berhasil mendirikan 42 cabang di Indonesia dan 2 Cabang di luar negeri, yaitu di Malaysia dan Singapura. Kantor pusat BRC saat ini berlokasi di Jl. Pak Gatot Raya No.81, Gerlong Bandung. BRC pun memperluas peta dakwahnya, tidak hanya bergerak di kesehatan saja namun mencoba terjun ke pendidikan yang berbasis Thibbun Nabawi, bisnis, makanan dan training dengan memegang konsep makna sehat yang ternyata bukan fisik saja namun didukung dengan sehat secara finansialnya, sehat lingkungan sosialnya, sehat perekonomiannya dan sebagainya. Telah berdiri pula divisi khusus dlam pengelolaan makanan halal thoyib, BRC Food yang mengeluarkan beberapa produk: air sehat BRC, beras

organic, kue dan sirup, serta ayam organic. BRC juga telah mengawali program ibadah Umrohh plus thibbun nabawi.

II.1.2 Struktur Organisasi

Struktur Organisasi adalah sebuah garis hirarki yang ada dan berlaku pada sebuah perusahaan. Didalam struktur organisasi menggambarkan dengan jelas pemisahan kegiatan pekerjaan antara yang satu dengan yang lain dan bagaimana hubungan aktifitas dan fungsi dibatasi. Struktur organisasi yang ada di Yayasan Bekam Ruqyah Center dapat dilihat pada gambar II.1.



Gambar II.1 Struktur Organisasi Yayasan Bekam Ruqyah Center

II.1.3 Deskripsi Tugas

Deskripsi Tugas digunakan untuk mengetahui tugas, wewenang dan tanggung jawab dari masing-masing bagian. Deskripsi tugas yang ada di Yayasan Bekam Ruqyah Center adalah sebagai berikut.

1. Koordinator Operasional
    a. Merencanakan dan mengimplementasikan strategi pengembangan cabang agar memenuhi target pencapaian omzet perusahaan
    b. Memutuskan, mengimplementasikan dan mengevaluasi SOP terapi dan pengelolaan klinik

c. Memberikan arahan dan kebijakan kepada staf dan kacab untuk dapat melaksanakan tugasnya dengan optimal
d. Melakukan penjajakan, persiapan dan pembukaan cabang baru serta koordinator penyediaan perlengkapannya
e. Mengurusi perizinan dan ketentuan ketentuan yang berkaitan dengan legalitas dan izin perusahaan dan klinik
f. Mengumpulkan informasi dan negosiasi dengan pihak luar untuk keperluan marketing perusahaan

2. Koordinator PR-Marketing
   a. Koordinasi dan peningkatan kerjasama dg pihak Bank,
   b. Merumuskan dan mengkoordinasikan kebijakan yang berkaitan dengan pengelolaan logistik
   c. Melakukan negosiasi dengan pihak supplier dalam hal pengadaan dan penentuan harga herbal
   d. Membuat dan memperpanjang Kontrak Perjanjian Usaha Klinik dengan para Owner Klinik,
   e. Berkonsultasi dengan pihak Notaris tentang legalitas kerjasama Klinik BRC dengan para mitra
   f. Berkomunikasi degan berbagai elemen Mitra Perusahaan baik suplier, Investor, perusahaan lain.
   g. Melakukan Pengawasan dan evauasi divisi logistik agar berjalan dengan lancar
3. Staf Operasional Bandung dsk
   a. Membuat Rekap Laporan Pendapatan dan Perkembangan klinik setiap Bulan
   b. Menerima dan mengolah Informasi dan Melakukan pengawasan Keberjalanan Klinik baik mengenai Santri, dan Sistem (SOP, tata tertib, dll)
   c. Melakukan Survey pada lokasi baru yang diajukan untuk pendidian Cabang BRC

d. Menangani Calon relasi BRC terutama dalam Pengajuan Herbal atau Kerjasama yang sifatnya Teknis
e. Berkoordinasi dan Aktidf dalam membuat kegiatan Spsesial di BRC seperti Kebersamaan, Rihlah, dll
f. Menjadi koordinator dalam Acara Forum Kacab
g. Merumuskan dan aktif dalam kebijakan pengelolaan SDM di BRC termasuk penempatan, rotasi dan penyeselsaian masalah santri.
h. Menjadi admin Milis Forum Kacab

4. Staf Operasional Jabotabek
   a. Menerima dan mengolah Informasi dan Melakukan pengawasan Keberjalanan Klinik baik mengenai Santri, dan Sistem (SOP, tata tertib, dll)
   b. Melakukan Tindakan-tindakan Teknis terkait kebutuhan Santri di lapangan terutama wilayah Jabotabek
   c. Berkoordinasi dan Aktif dalam membuat kegiatan Spsesial di BRC seperti Kebersamaan, Rihlah, dll
   d. Merumuskan dan aktif dalam kebijakan pengelolaan SDM di BRC termasuk rotasi dan penyeselsaian masalah santri.
   e. Menjadi PJ Informasi BRC via telepon
5. Staf IT
   a. Pengadaan dan perakitan Inftrastuktur IT termasuk pencegahan dan pemeliharaan, Trouble Shooting dalam rangka memastikan bekerja dengan baik Memonitor sistem support IT termasuk Instalasi Software, maintenance, perbaikan dalam rangka memastikan bekerja dengan baik
   b. Melakukan pengelolaan dan pemeliharaan website BRC, web IMP, Grup Facebook Internal BRC, update informasi online, dan memastikan kelangsungan dan keamanannya
   c. Melakukan mentoring internal kepada Santri BRC di cabang untuk aplikasi sistem IT dalam bidang administrasi klinik

d. Mengelola program marketing via Gili SMS serta monitoring keberjalanan sms marketing di semua cabang
e. Penyediaan dan monitoring alat-alat elektronik untuk kepentingan internal BRC
f. Membuat dan mengembangkan software dan program aplikasi Internal BRC baik pusat atau Cabang

6. Staf Multimedia
   a. Mengkoordinaskan dan menyiapkan bahan-bahan logistik untuk kebutuhan Operasional dan marketing BRC yang rutin
   b. Membantu staf IT mengelola dan update konten website BRC
   c. Bekerjasama dengan staf operasional dalam penyediaan multimedia agar pelaksanaan Operasional Klinik berjalan baik
   d. Menyiapkan materi dan desain iklan untuk kegiatan marketing di mediacetak
   e. Membuat dan mengelola Buletin Edukasi bagi Konsumen BRC
7. Korlap
   a. Memastikan keberjalanan klinik dengan memberi pegarahan pada para Kacab
   b. Menjadi penyambung informasi dan Konsultasi dari Santri mengenai hal2 yang penting
   c. Koordinasi tentang program-program BRC dengan para Investor Cabang
   d. Melakukan kujungan dan pemantauan rutin ke cabang
8. Koordinator Logistik
   a. Melakukan Stok Opname dan memastikan keberadaan barang logistik
   b. Berkoordinasi dengan pihak Supplier dalam hal pemesanan, pembayaran ataupun komplian
   c. Bekerjasama dengan Staf Keuangan Logistik dalam Pencatatan utang Cabang dalam pengambilan Barang
   d. Membuat Laporan Penjualan dan Stok bulanan

e. Melakukan Pengecekan laporan Bank harian

9. Staf Logistik 1
    a. Menerima Pemesanan Barang logistik dari cabang
    b. Membuatkan Faktur Penjualan Barang
    c. Meminta laporan penjualan barang Konsinyasi ke Cabang
    d. Update Stok dan Penjualan barang logistik setiap hari baik percabang atau perbarang.
    e. Melakukan Stok opname harian
    f. Melakukan pengiriman barang ke Klinik menggunakan Kurir
10. Staf Logistik 2
    a. Melakukan Penyetoran uang Logistik ke Bank
    b. Mempersiapan barang Pesanan Cabang dan mengecek kelengkapannya
    c. Melakukan pengiriman barang untuk klinik (Bandung dan Jabodetabek)
    d. Memeriksa barang yang dikirim supplier baik kuantitas atau kualitasnya
    e. Pengaturan penyimpanan barang di Gudang Logistik
11. Staf Logistik 3
    a. Memasukan data penerimaan Pembayaran barang dari Cabang
    b. Memasukan dan Memastikan pembayaran kepada Supplier tepat waktu dan jumlah
    c. Mengelola Kas logistik untuk kelancaran pembayaran dan meningkatkan cashflownya
    d. Pengecekan dan konfirmasi transaksi logistik via rekening Bank
12. Staf Kepegawaian
    a. Membuat dan mengelola database Seluruh santri karya BRC
    b. Membuat dan pengiriman surat-surat yang berhubungan dengan kepegawaian
    c. Menyediakan dan mengelola penggunaan surat / formulir kepegawaian (izin, cuti, pengajuan)

d. Membuat Update Informasi santri setiap awal bulan
e. Mengumpulkan dan verifikasi data kehadiran santri dalam proses Payroll
f. Mengumpulkan dan verifikasi data kegiatan santri Cabang dan Pusat dalam proses Payroll
g. Berkoordinasi dengan keuangan dalam pengiriman dan Peyediaan Dana untuk sistem Payroll
h. Membuat daftar Gaji dan Slip Gaji Santri setiap bulan
i. Mendistribusikan Slip gaji dan menerima komplain terkait penggajian santri
j. Menerima komplain atau pengajuan santri yang berkaitan dengan kepegawaian
k. Memberikan informasi dalam hal proses Pengelolaan SDM Cabang
l. Melakukan Wawancara baik santri yang masuk atau mengalami permasalahan

13. Staf Keuangan
    a. Mengelola fungsi akuntansi dalam memproses data dan informasi keuangan untuk menghasilkan laporan keuangan yang dibutuhkan perusahaan secara akuratdan tepat waktu
    b. Mengkoordinasikan dan mengontrol perencanaan, pelaporan dan pembayaran gaji santri karya agar akurat dan tepat waktu
    c. Mengkoordinasikan dan mengontrol cashflow perusahaan terutama terkait piutang dan utang
    d. Mengkoordinasikan penyusunan anggaran perusahaan dan mengontrol penggunaannya dalam menunjang kegiatan operasional perusahaan
    e. Melakukan pengembangan sistem dan prosedur keuangan dan akuntansi dan mengontrol pelaksanaannya
    f. Mengkoordinasikan dan melakukan perencanaan dan analisa keuangan untuk memberikan masukan bagi pimpinan dalam

pengambilan keputusan bisnis baik investasi, ekspansi maupun kondisi lainnya

g. Menyusun laporan keuangan bulanan (L/R) dengan data yang didapatkan dari Pusat dan cabang
h. Membuat laporan Keuangan Triwulan untuk penyampaian laporan pada manajemen atau pihak terkait
i. Menyusun lapoan keuangan setengah tahun (semester) ntuk kepentingan pemeriksaan audit, bank, dan stakeholder
j. Melakukan cek dan kontrol dari data pendukung transaksi akuntansi dari divisi2 terkait untuk memastikan transaksi disusun berdsarkan fakta yang ada
k. Melakukan kompilasi/analisa anggaran dari seluruh divisi utuk kepentingan estiasi anggaran keseluruhan perusahaan
l. Membuat jurnal Harian untuk semua aktivitas keuangan perusahaan

14. Staf Rumah Tangga
    a. Mempersiapkan ruang kerja dan peralatan kantor pusat
    b. Melakukan kegiatan surat menyurat, dokumentasi dan perarsipan untuk memastikan dukungan administrasi bagi kelancaran kegiatan perusahaan
    c. Membuat perkiraan biaya bulanan yang berkaitan dengan kegiatan administrasi perkantoran dan bagian umum (supir) sebagai rekomendasi pembuatan anggaran keangan kantor
    d. Melakukan pengadaan ATK, peralatan kantor, alat kebersihan, serta layanan dokumen (potocopy, jilid dll)
    e. Mengawasi pelaksanan kebersihan dan kenyamanan ruang akntor dan keamanannya
    f. Membuat Rekap kehadiran santri karya manajemen dan BU tiap bulan
    g. Membuat Rekap Laporan kehadiran santri klinik via milis Forum Kacab

15. Staf Riset Operasional
    a. PJ Call Center Terapis untu berkonsultasi terkait permalasahan pelayanan Terapi/penyakit
    b. Mengkoordinasi penerapan Terapi Totok Wajah di seluruh Cabang terkait sosialisasi, pelatihan dan penyediaan alat.
    c. Melakukan Evaluasi dan Pengawasan SOP Klinik berdasarkan kebutuhan yang berlaku
    d. Riset dan Pengembangan Terapi Baru untuk menambah nilai pelayanan dan variasi Terapi Klinik

II.2 Landasar Teori

Landasan Teori yang mendukung proses analisis sistem serta mendukung proses perancangan pembangunan Sistem Informasi *Distribution Requirement Planning* adalah Konsep dasar sistem,

II.2.1 Konsep Dasar Sistem

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu [2]. Maka dapat diambil suatu kesimpulan bahwa suatu sistem merupakan elemen yang saling berkaitan dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu.

II.2.1.1 Karakteristik Sistem

Suatu sistem memiliki karakteristik atau sifat-sifat tertentu, yaitu

1. Komponen sistem

   Komponen sistem atau elemen-elemen sistem dapat berupa suatu sub sistem atau bagian-bagian dari sistem. Setiap sistem baik besar maupun kecil, 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 yang lebih besar yang disebut supra system.
2. Batas sistem (*System Boundary*)

Batas sistem merupakan daerah-daerah yang membatasi antara satu sistem dengan sistem lainnya dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batas suatu sistem menujukkan ruang lingkup (scope) dari sistem tersebut.

3. Lingkungan Luar sistem (*System Environment*)

   Lingkungan luar sistem dari suatu sistem adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem yang dapat bersifat menguntungkan dan dapat pula merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem yang harus dijaga dan dipelihara. Sedangkan yang merugikan harus ditahan dan dikendalikan, karena akan mengganggu kelangsungan hidup sistem.

4. Penghubung sistem (*System Interprest*)

   Penghubung sistem merupakan media penghubung antara satu sub sistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Dengan penghubung akan terjadi interaksi antar subsistem, sehingga membentuk satu kesatuan.

5. Masukkan sistem (*System Input*)

   Masukan adalah suatu energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (maintenance input) dan masukan sinyal (signal input). Maintenance input adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Contoh maintenance input di dalam sistem komputer adalah program, yang digunakan untuk mengoperasikan komputer. Sedangkan signal input adalah energi yang diproses untuk mendapatkan keluaran. Contoh signal input di dalam sistem komputer adalah data, yang dapat diolah menjadi Informasi.

6. Keluaran sistem (*System Output*)

   Keluaran (*Output*) merupakan hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna.

7. Pengolah sistem (*System Process*)

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya, yang bertugas untuk merubah masukan menjadi keluaran.

8. Sasaran sistem (*System Objective*)

   Suatu sistem pasti memiliki tujuan (*goal*) atau sasaran (*objective*). Suatu operasi sistem akan berguna dan berhasil apabila mencapai sasaran atau tujuannya. Sasaran sistem sangat menentukan masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

II.2.1.2 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandangan, diantaranya adalah sebagai berikut ini :

1. Sistem diklasifikasikan sebagai sistem abstrak (*abstract system*) dan sistem fisik (*physical system*). Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. 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 manusia. Sistem buatan manusia yang melibatkan interaksi antara manusia dengan mesin disebut dengan *human-machine system* atau ada yang menyebut dengan *man-machine system*.
3. Sistem diklasifikasikan sebagai sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*). Sistem tertentu beroperasi dengan tingkah laku yang sudah diperdiksi. Interaksi diantara bagian-bagiannya dapat dideteksi dengan pasti, sehingga keluaran 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. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak diluarnya. Secara teoritis sistem tertutup ini ada, tetapi kenyataannya tidak

ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system* (secara relatif tertutup, tidak benar-benar tertutup). Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lainnya.

II.2.2 Pengertian Sistem Informasi

Sistem informasi dibentuk dari dua kata yaitu sistem dan informasi. Sistem adalah hubungan satu unit dengan unit-unit lainnya yang saling berhubungan satu sama lain dan tidak dapat dipisahkan serta menuju suatu kesatuan untuk mencapai tujuan tertentu, apabila terdapat gangguan atau kendala maka tujuan yang tercapai pun akan terganggu [3]. Informasi adalah hasil pemprosesan, manipulasi dan pengorganisasian atau penataan dari sekelompok data yang mempunyai nilai pengetahuan (*knowledge*) bagi penggunanya. Namun demikian istilah ini memiliki banyak arti bergantung pada konteksnya, dan secara umum berhubungan erat dengan konsep seperti arti, pengetahuan, komunikasi, kebenaran, representasi, dan rangsangan mental. Kualitas informasi dapat dipengaruhi oleh enam hal, yaitu :

a. Relevan
b. Akurat
c. Tepat waktu
d. Ekonomis
e. Efisien
f. Dapat dipercaya

Setelah mendapatkan pengertian sistem dan informasi maka sistem informasi merupakan sekumpulan subsistem yang saling berhubungan, berkumpul bersama–sama dan membentuk satu kesatuan, saling berinteraksi dan bekerja sama antara bagian satu dengan yang lainnya dengan cara-cara tertentu untuk melakukan fungsi pengolahan data, menerima masukan (*input*) berupa data-data, kemudian mengolahnya (*processing*), dan menghasilkan keluaran (*output*) berupa informasi sebagai dasar bagi pengambilan keputusan yang berguna dan mempunyai nilai yang dapat dirasakan akibatnya baik pada saat itu juga maupun di masa mendatang, mendukung kegiatan operasional, manajerial, dan strategis

organisasi, dengan memanfaatkan beberapa sumber daya yang tersedia bagi fungsi tersebut guna mencapai tujuan [4].

II.2.3 Sistem Informasi Distribusi

Sistem informasi distribusi adalah sistem yang mengumpulkan data-data atau informasi mengenai kegiatan distribusi yang dilakukan oleh suatu perusahaan, kemudian menghasilkan laporan yang dapat digunakan oleh top management dalam pengambilan keputusan sehubungan dengan sistem distribusi perusahaan tersebut.

II.2.4 *Distribution Requirement Planning*

Distribution Resource Planning (DRP) memberikan kerangka kerja untuk menerapkan *centralized push* sistem dalam menejemen distribusi inventori. Istilah DRP memiliki dua pengertian yang berbeda, yaitu: *distribution requirements planning* dan *distribution resource planning*. *Distribution Requirements Planning* berfungsi menentukan kebutuhan-kebutuhan untuk mengisi kembali inventori pada *branch warehouse*. Sedangkan *Distribution Resource Planning* merupakan perluasan dari distribustion requirements planning yang mencakup lebih dari sekedar sistem perencanaan dan pengendalian pengisian kembali inventori, tetapi ditambah dengan perencanaan dan pengendalian dari sumber-sumber yang terkait untuk meningkatkan performansi sistem[8].

Langkah – langkah yang dibutuhkan dalam metode Distribution Requirement Planning terdapat 4 langkah yaitu.

1. *Netting*.

   Netting adalah proses perhitungan untuk menetapkan jumlah kebutuhan bersih yang besarnya merupakan selisih antara kebutuhan kotor dengan keadaan persediaan.

   Rumus NR = (GR + SS) – (SR + POH)

   Untuk tiap periode/ dihitung tiap bulan

2. *Lotting*

Lotting adalah suatu proses untuk menentukan besarnya jumlah pesanan optimal untuk setiap item secara individual didasarkan pada kebutuhan bersih yang telah dilakukan.

3. *Offsetting*

   Langkah ini bertujuan untuk menentukan saat yang tepat untuk melakukan rencana pengiriman dalam rangka memenuhi kebutuhan bersih.

4. *Explosion*

   Proses explosion adalah proses perhitungan kebutuhan kotor untuk tingkat jaringan distribusi yang lebih rendah.

II.2.4.1 *Safety Stock*

Safety Stock adalah *variable* yang digunakan sebagai titik acuan dalam pemesanan ulang untuk memenuhi penjualan berdasarkan peramalan. Dalam menentukan nilai *safety stock*, dilakukan suatu pendekatan dengan menggunakan konsep tingkat layanan (*service level*). Maka untuk menghitung nilai *safety stock* menggunakan rumus *reorder point*. Formulasi titik *reorder point* dapat dilihat pada Tabel II.1.[15]

Tabel II.1 Formulasi Titik *Reorder Point*

Titik Reorder	Tingkat Service Level
$\overline{DL} + 3{,}09\alpha_D\sqrt{L}$	99,90%
$\overline{DL} + 2{,}58\alpha_D\sqrt{L}$	99,50%
$\overline{DL} + 2{,}33\alpha_D\sqrt{L}$	99%
$\overline{DL} + 1{,}96\alpha_D\sqrt{L}$	97,50%
$\overline{DL} + 1{,}64\alpha_D\sqrt{L}$	95%
$\overline{DL} + 1{,}28\alpha_D\sqrt{L}$	90%
$\overline{DL} + 1{,}04\alpha_D\sqrt{L}$	85%
$\overline{DL} + 0{,}85\alpha_D\sqrt{L}$	80%
$\overline{DL} + 0{,}67\alpha_D\sqrt{L}$	75%

Rumus untuk menenentukan safety stock adalah :

$SS = Z \times s \times \sqrt{L}$ ..... (2.1)

Keterangan :

SS = *Safety Stock*

Z = Tingkat *Service Level*

s = standar deviasi

L = *lead time*

rumus untuk menentukan standar deviasi adalah :

$$s = \sqrt{\frac{\sum_{i=1}^{n}(d-\bar{d})^2}{n}} \quad \text{.....(2.3)}$$

Keterangan :

d = penjualan

n = jumlah data penjualan

II.2.4.2 *Lotting*

Lotting adalah tahap untuk menentukan ukuran lot dalam distribusi. Penentuan ukuran lot dipengaruhi oleh beberapa factor seperti frekuensi pengiriman, EOQ, ukuran kapasitas container serta total berat dan luas yang dibutuhkan. Teknik-teknik penentuan ukuran lot diantaranya adalah sebagai berikut :

1. *Lot For Lot* (L-4-L)

   Teknik penetapan ukuran lot dilakukan atas dasar pesanan diskrit. Disamping itu, teknik ini merupakan cara paling sederhana dari semua teknik ukuran lot yang ada. Teknik ini selalu melakukan perhitungan kembali (bersifat dinamis) terutama apabila terjadi perubahan pada kebutuhan bersih. Contoh penetapan ukuran lot dapat dilihat pada Tabel II.2.

Tabel II.2 Contoh Penetapan Ukuran Lot dengan L-4-L

Periode	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Kebutuhan Bersih	20	50	60	80	40	40	40	60
Jumlah Pesan	20	50	60	80	40	40	40	60
Sediaan	0	0	0	0	0	0	0	0

2. *Fix Order Quantity* (FOQ)

   Dalam FOQ ukuran lot ditentukan secara subyektif. Berapa besarnya ditentukan berdasarkan pengalaman produksi atau intuisi. Tidak ada teknik yang dapat dikemukakan untuk menentukan berapa ukuran lot ini. Kapasitas produksi selama lead time produksi dalam hal ini dapat digunakan sebagai dasar untuk menentukan besarnya lot. Sekali ukuran lot ditetapkan, maka lot ini akan digunakan untuk seluruh periode selanjutnya dalam perencanaan. Contoh Penetapan Ukuran Lot Dengan FOQ dapat dilihat pada Tabel II.3.

Tabel II.3 Contoh Penetapan Ukuran Lot dengan FOQ

Periode	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Kebutuhan Bersih	20	50	60	80	40	40	40	60
Jumlah Pesan	100		100	100			100	
Sediaan	80	30	70	90	50	10	70	10

3. *Fixed Periode Requirement*

   Dalam metode FPR penentuan ukuran lot didasarkan pada periode waktu tertentu saja. Besarnya jumlah kebutuhan tidak berdasarkan ramalan, tetapi dengan cara menjumlahkan kebutuhan bersih pada periode yang akan datang. Bila dalam metode FOQ besarnya jumlah lot adalah tetap, sementara selang waktu antar pesanan tidak tetap. Dalam metode FPR ini selang waktu antar pemesanan dibuat tetap dengan ukuran lot sesuai pada kebutuhan bersih. Contoh Penetapan ukuran Lot dengan FPR dapat dilihat pada tabel II.4.

Tabel II.4 Contoh Penetapan Ukuran *Lot* dengan FPR

Periode	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Kebutuhan Bersih	20	50	60	80	40	40	40	60
Jumlah Pesan	70		140		80		100	
Sediaan	50	0	80	0	40	0	60	0

Ukuran lot tidak didasarkan pada minimum biaya penyimpanan dan biaya pemesanan, bila biaya penyimpanan tidak diidentifikasikan baik secara marginal ataupun *incremental*[15].

II.2.5 *Forecasting*

Ramalan adalah situasi atau kondisi yang diperkirakan akan terjadi pada masa yang akan datang. Menurut Assauri (1984) metode peramalan adalah cara memperkirakan secara kuantitatif apa yang akan terjadi pada masa depan, berdasarkan pada data yang relevan pada masa lalu. Oleh karena metode peramalan didasarkan atas data yang relevan pada masa lalu, maka metode peramalan ini dipergunakan dalam peramalan yang objektif.[9]

II.2.5.1 Metode *Exponential Smoothing*

Metode *exponential smoothing* merupakan suatu prosedur perbaikan terus-menerus pada peramalan terhadap objek pengamatan terbaru. Metode peramalan ini menitik beratkan pada penurunan prioritas secara eksponensial pada objek pengamatan yang lebih tua. Dalam pemulusan eksponensial atau *Exponential Smoothing* terdapat satu atau lebih parameter pemulusan yang ditentukan secara eksplisit, dan hasil ini menentukan bobot yang dikenakan pada nilai observasi. Dengan kata lain, observasi terbaru akan diberikan prioritas lebih tinggi bagi peramalan daripada observasi yang lebih lama.[10]

Metode Exponential smoothing dibagi lagi menjadi beberapa metode, yaitu:

1. *Single Exponential Smoothing*

   Single Exponential Smoohting digunakan pada peramalan jangka pendek, biasanya hanya untuk 1 periode ke depan. Model mengasumsikan bahwa data berfluktuasi disekitar nilai mean yang tetap, tanpa trend atau pola pertumbuhan konsisten
2. *Double Exponential Smoothing*.

   Double Exponential Smoothing digunakan apabila model mengasumsikan bahwa data memiliki trend linier namun tidak memiliki variasi musiman.

II.2.5.2 Metode *Mean Square Error* (MSE)

Hasil dalam suatu proses peramalan dapat dievaluasi menggunakan metode *Mean Square Error* (MSE). Dengan menggunakan MSE, error yang ada menunjukan seberapa besar perbedaan hasil estimasi dengan hasil yang

diestimasi. Hal ini membuat berbeda karena adanya keacakan pada data atau karena tidak mengandung estimasi yang lebih akurat.

Root Mean Square Error (RMSE) merupakan pengakaran nilai dari nilai MSE yang sudah dicari sebelumnya. RMSE digunakan untuk mencari keakuratan hasil peramalan dengan data histori dengan menggunakan rumus (makridakis, 1999). Semakin kecil nilai yang dihasilkan semakin bagus pula hasil dari peramalan yang dilakukan[10].

II.2.6 Basis Data

Basis data dapat dipahami sebagai suatu kumpulan data yang terhubung (*interrelated data*) yang disimpan secara bersama–sama pada suatu media, tidak perlu suatu kerangkapan data (walaupun ada harus seminimal mungkin dan terkontrol (*controlled redundancy*)), data disimpan dengan cara–cara tertentu sehingga mudah digunakan atau ditampilkan kembali. Data dapat digunakan oeh satu atau lebih program–program aplikasi secara optimal, data disimpan tanpa mengalami ketergantungan dengan program yang akan menggunakannya, data disimpan sedemikianrupa sehingga proses penambahan, pengambilan dan modifikasi data dapat dilakukan dengan mudah dan terkontrol.

II.2.7 Analisis Sistem

Penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan. Pada model ini analisis terhadap perangkat lunak menggunakan model *Unified Modeling Language* (UML).

II.2.7.1 *Unified Modeling Language* (UML)

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendocumenttasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita 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. Tetapi karena UML juga menggunakan Class dan operation dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasabahasa berorientasi objek seperti C++, Java, C# atau VB.NET [11]. Pada pemodelan UML terdapat beberapa diagram yaitu *Use Case* Diagram, *Sequence* Diagram, *Class* Diagram, *Activity* Diagram.

II.2.7.1.1 *Use Case* Diagram

Use Case Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Sebuah *Use Case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use Case* merupakan sebuah pekerjaan tertentu, misalnya Login ke sistem, mengbuat sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Use Case Diagram dapat sangat membantu bila kita sedang menyusun *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 sama. Sebuah *Use Case* juga dapat meng-*extend Use Case* lain dengan kebutuhannya sendiri. Sementara hubungan generalisasi antar *Use Case* menunjukkan bahwa *Use Case* yang satu merupakan spesialisasi dari yang lain [11].

II.2.7.1.2 *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 atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian 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. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi / metode dari Class. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan *icon* khusus untuk objek *boundary*, *controller* dan *persistent entity* [11].

II.2.7.1.3 *Class* Diagram

Class diagram 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 [11]. Hubungan antar *Class* diagram dapat dilihat dibawah ini :

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* menunjukkan 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 metode *Class* asalnya dan menambahkan fungsionalitas baru, sehingga ia 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.

II.2.7.1.4 *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, di mana 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.

Sebuah aktivitas dapat direalisasikan oleh satu *Use Case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *Use Case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan *behaviour* pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (fork dan join) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa object swimlane untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu [13].

II.2.8 Internet

Internet merupakan jaringan global yang terdiri dari berbagai komputer yang saling berhubungan dan bekerjasama dengan cara berbagai informasi dan data. Media penghubung tersebut bisa melalui kabel, kanal satelit maupun frekuensi radio. Setiap komputer yang terhubung dengan jaringan tersebut, diberikan sebuah nomor yang unik, dan berkomunikasi satu sama

lainnya dengan bahasa komunikasi yang sama. Bahasa komunikasi yang sama ini disebut protokol. Protokol yang digunakan di internet adalah TCP/IP (Transmission Control Protocol / Internet Protocol).

II.2.9 *Word Wide Web* (WWW)

World Wide Web sering disingkat WWW atau *web* saja, yaitu sebuah sistem dimana informasi dapat dinikmati dalam bentuk teks, gambar, suara, dan lain-lain serta ikut dapat diakses oleh perangkat lunak yang disebut *browser*. Informasi di *web* pada umumnya ditulis dalam bentuk HTML. Informasi lainnya disajikan dalam bentuk grafis (dalam format GIF, JPG, PNG), suara (dalam bentuk format AU, WAV), dan objek multimedia lainnya (seperti MIDI, Shockwave, Quicktime Movie, 3D World). WWW dijalankan dalam *server* yang disebut HTTPD.

II.2.10 HTTP (*Hypertext Transfer Protocol*)

Server dan *browser web* berkomunikasi satu sama lain dengan protokol yang memang dibuat khusus untuk ini yaitu HTTP. HTTP bertugas menangani permintaan-permintaan *(request)* dan *browser* untuk mengambil dokumen-dokumen *web*. HTTP bisa dianggap sebagai sistem yang bermodel *client-server*.

II.2.11 *Web Server*

Web server merupakan *software* yang menyediakan layanan data yang berfungsi menerima permintaan HTTP atau HTTPS dari klien yang dikenal dengan browser web dan mengirimkan kembali hasilnya dalam bentuk halaman - halaman web yang umumnya berbentuk dokumen HTML.

Fungsi utama sebuah server web adalah untuk mentransfer berkas atas permintaan pengguna melalui protokol komunikasi yang telah ditentukan. Disebabkan sebuah halaman web dapat terdiri atas berkas teks, gambar, video, dan lainnya pemanfaatan server web berfungsi pula untuk mentransfer seluruh aspek pemberkasan dalam sebuah halaman web yang terkait; termasuk di dalamnya teks, gambar, video, atau lainnya.

Pengguna, biasanya melalui aplikasi pengguna seperti peramban web, meminta layanan atas berkas ataupun halaman web yang terdapat pada sebuah

server web, kemudian server sebagai manajer layanan tersebut akan merespon balik dengan mengirimkan halaman dan berkas-berkas pendukung yang dibutuhkan, atau menolak permintaan tersebut jika halaman yang diminta tidak tersedia.

saat ini umumnya server web telah dilengkapi pula dengan mesin penerjemah bahasa skrip yang memungkinkan server web menyediakan layanan situs web dinamis dengan memanfaatkan pustaka tambahan seperti PHP, ASP.

Pemanfaatan server web saat ini tidak terbatas hanya untuk publikasi situs web dalam World Wide Web, pada prakteknya server web banyak pula digunakan dalam perangkat-perangkat keras lain seperti printer, router, kamera web yang menyediakan akses layanan http dalam jaringan lokal yang ditujukan untuk menyediakan perangkat manajemen serta mempermudah peninjauan atas perangkat keras tersebut.

II.2.12 MVC (*Model View Controller*)

Model-View-Controller atau MVC adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data (*Model*) dari tampilan (*View*) dan cara bagaimana memprosesnya (*Controller*). Dalam implementasinya kebanyakan framework dalam aplikasi website adalah berbasis arsitektur MVC. MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, antarmuka pengguna, dan bagian yang menjadi kontrol dalam sebuah aplikasi web.

Bagian-bagian dari MVC adalah sebagai berikut :

1. *Model*

   Model mewakili struktur data. Biasanya model berisi fungsi-fungsi yang membantu seseorang dalam pengelolaan basis data seperti memasukkan data ke basis data, pembaruan data dan lain-lain.

2. *View*

   View adalah bagian yang mengatur tampilan ke pengguna. Bisa di katakan berupa halaman web.

3. *Controller*

   Controller merupakan bagian yang menjembatani *model* dan *view*. *Controller* berisi perintah-perintah yang berfungsi untuk memproses suatu data dan mengirimkannya ke halaman web.

II.2.13 Bahasa Pemograman

Bahasa Pemograman yang digunakan dalam pembangunan sistem informasi ini adalah sebagai berikut.

II.2.13.1 *Hypertext Markup Language*(HTML)

Dokumen HTML adalah file teks regular (disebut juga ASCII) yang diciptakan dengan menggunakan editor teks (misalnya, Emacs dalam Unix, Gnotepad pada Linux, atau Notepad pada Windows) atau dengan menggunakan word processor, tetapi harus disimpan dengan format *"text only with line breaks"*.

Sesuai dengan namanya, bahasa ini menggunakan tanda (MarkUp) untuk menandai perintah-perintahnya. Saat ini banyak sekali aplikasi-aplikasi yang dapat digunakan untuk membuat web page, seperti Microsof Frontpage, Adobe GoLive, Macromedia Dreamweaver, Ultradev dan lain-lainnya. Namun untuk seorang web develover kemampuan dasar perintah HTML tetap dibutuhkan.

Homepage dibangun oleh sebuah file yang didalamnya terdapat kode-kode dan informasi dan dibaca oleh sebuah aplikasi yang bernama browser. Oleh browser, kode dan informasi akan disajikan sebagai sebuah halaman web yang kita kenal sebagai homepage. Kode tersebut haruslah ditulis menurut suatu aturan tertentu yang dapat dimengerti browser. Aturan tersebut dikenal dengan "bahasa pengkodean".

Halaman sebuah website dibuat menggunakan bahasa pengkodean HTML.. Sekalipun banyak orang menyebutnya sebagai salah satau bahasa pemrograman, HTML sebenarnya sama sekali bukan bahasa pemrograman karena HTML merupakan bahasa Mark Up (pengkodean) yang digunakan untuk menentukan format atau style dan teks yang ditandai. Simbol Mark Up yang digunakan oleh HTML ditandai dengan tanda lebih kecil (<) dan lebih besar (>), keduanya dinamai tag. [5]

II.2.13.2 *Hypertext Preprocessor* (PHP)

PHP (Hypertext Preprocessor) adalah sebuah bahasa *scripting* yang dibundel dengan HTML, yang berjalan disisi *server*. Sebagian besar perintahnya berasal dari bahasa *C, Java* dan *Perl* dengan beberapa tambahan fungsi khusus PHP.

Bahasa ini memungkinkan para pembuat aplikasi web yang menyajikan HTML yang dinamis dan interaktif dengan cepat dan mudah, yang dihasilkan server. PHP biasa berinteraksi dengan hampir semua teknologi *web* yang sudah ada. *Developer* biasa menulis sebuah program PHP yang mengeksekusi suatu program CGI di *server* lain. Fleksibilitas ini amat bermanfaat bagi pemilik situs-situs web yang besar dan sibuk, karena pemilik masih menggunakan aplikasi-aplikasi yang sudah terlanjur dibuat dimasa lalu dengan menggunakan CGI, ISAP atau dengan *script* seperti *Perl, awk* atau *Phyton* selama proses migrasi ke aplikasi baru yang dibuat dengan menggunakan PHP.

II.2.13.3 *Cascading Style Sheets* (CSS)

Cascading Style Sheet (CSS) merupakan salah satu bahasa pemrograman web untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. Sama halnya *styles* dalam aplikasi pengolahan kata seperti Microsoft Word yang dapat mengatur beberapa *style*, misalnya *heading*, subbab, *bodytext*, *footer*, *images*, dan *style* lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (file). Pada umumnya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML. CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran *border*, warna *border*, warna *hyperlink*, warna *mouse over*, spasi antar, spasi antar teks, *margin* kiri, kanan, atas, bawah, dan parameter lainnya [14].

II.2.13.4 *Javascript*

JavaScript adalah bahasa *script* yang digunakan pada halaman HTML. *JavaScript* adalah varian dari *java* yang sangat terkenal dalam lingkungan pemrograman *web*. *JavaScript* mempunyai keunggulan pada sisi *client*, maupun

diakses lebih cepat dari *script server*. Tetapi kelemahannya pada *script* ini belum tentu mampu diproses oleh *browser client* tergantung kompetibilitas *browser* tersebut. Cara untuk menjalankan *javascript* hanya dengan *javascript-enabled* yaitu *browser* yang mampu menjalankan *javascript* seperti *Netscape Navigator* (versi 2.0 keatas) atau *internet ekplorer* (MSIE-versi 3.0 ke atas).

II.2.13.5 *Asynchronous Javascript* and XML (AJAX)

AJAX (*Asynchronous JavaScript and XML*) sesungguhnya bukanlah suatu bahasa pemrogarman, melainkan hanya suatu teknik yang memungkinkan untuk membuat aplikasi *web* yang interaktif. Dengan menggunakan AJAX, aplikasi *web* dapat berinteraksi dengan server di latar belakang sehingga tidak mempengaruhi halaman *web* secara keseluruhan. Sebagai gambaran, Anda mungkin sering mendaftarkan diri pada suatu situs *web* sebagai anggota baru. Setelah Anda selesai mengetikan *user ID* saat registrasi, sistem bisa memberitahukan bahwa nama yang Anda pakai sudah digunakan oleh orang lain atau belum.

hal penting yang perlu diperhatikan dalam lapisan AJAX adalah :

1. Lapisan AJAX meminta layanan ke *server* atau tidak, tergantung pada jenis layanan antarmuka. Sebagai contoh, untuk validasi terhadap objek formulir tidak perlu melakukan permintaan ke *server*. Cukup ditangani di sisi *client*. Namun untuk pemeriksaan keabsahan *password*, mau tidak mau harus dilayangkan ke *server* karena harus berhubungan dengan database.
2. Sekalipun penambahan lapisan AJAX berkesan memperlambat proses (karena komunikasi ke server tidak lagi dilakukan secara langsung), pada kenyataanya yang terjadi adalah sebaliknya. Hal ini karena permintaan yang dilakukan oleh lapisan AJAX menghasilkan informasi yang jauh lebih sedikit dibanding kalau yang diminta adalah keseluruhan halaman.
3. Lapisan AJAX bekerja dengan pendekatan asinkron. Asinkron berarti bahwa *client* bisa meminta layanan dari *server* dan tidak perlu menunggu sampai *server* melayaninya. Dengan demikian, saat informasi dari *server* belum didapat, layanan kepada pemakai tetap bisa dilakukan.

4. Komunikasi antara *client* dan *server* tidak perlu mengubah keseluruhan antarmuka dalam halaman *web*, tetapi cukup hanya bagian tertentu yang terpengaruh. Hal seperti ini tidak bisa dilakukan pada aplikasi *web* yang tidak menggunakan AJAX.[7].

II.2.13.6 Code Igniter

CodeIgniter adalah aplikasi open source yang berupa framework dengan model MVC (*Model*, *View*, *Controller*) untuk membangun website dinamis dengan menggunakan PHP. CodeIgniter memudahkan developer untuk membuat aplikasi web dengan cepat dan mudah dibandingkan dengan membuatnya dari awal. CodeIgniter dirilis pertama kali pada 28 Februari 2006.

CodeIgniter memiliki beberapa kelebihan dibandingkan dengan beberapa *framework* PHP yang lain yaitu.

1. Performa sangat cepat

   salah satu alasan tidak menggunakan *framework* adalah karena eksekusinya yang lebih lambat daripada PHP *from the scracth*, tapi Codeigniter sangat cepat bahkan mungkin bisa dibilang codeigniter merupakan *framework* yang paling cepat dibanding framework yang lain.
2. Konfigurasi yang sangat minim (*nearly zero configuration*)

   tentu saja untuk menyesuaikan dengan *database* dan keleluasaan *routing* tetap diizinkan melakukan konfigurasi dengan mengubah beberapa file konfigurasi seperti database.php atau autoload.php, namun untuk menggunakan codeigniter dengan *setting standard*, anda hanya perlu mengubah sedikit saja file pada *folder config*.
3. Banyak komunitas

   dengan banyaknya komunitas CI ini, memudahkan kita untuk berinteraksi dengan yang lain, baik itu bertanya atau teknologi terbaru.
4. Dokumentasi yang sangat lengkap

Setiap paket instalasi codeigniter sudah disertai *user guide* yang sangat bagus dan lengkap untuk dijadikan permulaan, bahasanya pun mudah dipahami.

II.2.13.7 MySQL

MySQL adalah *multi user database* yang menggunakan bahasa *Strucktured Query Language* (SQL). MySQL dalam operasi *client-server* melibatkan *server* daemon MySQL disisi *server* dan berbagai macam program serta *library* yang berjalan disisi *client*. MySQL mampu menangani data yang cukup besar. Perusahaan yang mengembangkan MySQL yaitu TcX, mengaku menyimpan data lebih dari 40 *database*, 10.000 tabel dan sekitar 7 juta baris, totalnya kurang lebih 100 Gigabyte data.

SQL adalah bahasa standar yang digunakan untuk mengakses *database server*. Bahasa ini pada awalnya dikembangkan oleh IBM, namun telah diadopsi dan digunakan sebagai standar industri. Dengan menggunakan SQL, proses akses database menjadi lebih *user-friendly* dibandingkan dengan menggunakan dBASE atau Clipper yang masih menggunakan perintah-perintah pemrograman.

MySQL merupakan *software database* yang paling populer di lingkungan Linux, kepopuleran ini karena ditunjang performa query dari *database*nya yang saat ini bisa dikatakan paling cepat dan jarang bermasalah. MySQL ini juga sudah dapat berjalan pada lingkungan Windows[5].