BAB 2
TINJAUAN PUSTAKA

2.1 Tinjauan Tempat Penelitian

Secara singkat profil perusahaan dimana pelaksanaan kerja praktek dilakukan adalah sebagai berikut :

Nama Perusahaan	:	Cahaya Bagi Negeri Indonesia
Alamat	:	Jl. Sriwijaya Kav. 5-7 Lippo Cikarang
Jenis Bidang Usaha	:	*Broadcasting* dan *Production House*
Tanggal Berdiri	:	11 April 1994
Telepon / *Fax*	:	+6221 8990 5960/ +6221 8990 5961
Email	:	mitra@cbn.or.id
Alamat *Website*	:	www.CBN.Jawaban.com

2.1.1 Sejarah Perusahaan

Cahaya Bagi Negeri Indonesia (CBN Indonesia) hadir sejak tahun 1998 untuk memberitakan kasih dan kabar baik melalui media dengan cara kreatif dan kontekstual. Yayasan Cahaya Bagi Negeri adalah salah satu bentuk pelayanan CBN *WorldReach* di Indonesia. Yayasan CBN mempunyai banyak program pelayanan. Penayangan kesaksian-kesaksian melalui media televisi dan media lainnya menjadi salah satu program andalannya. Program TV CBN yang menjadi tayangan utama dan paling dikenal adalah Solusi yang telah 13 tahun tayang di Televisi Nasional-SCTV.

Berikut ini adalah program TV CBN Indonesia :

a. Solusi

Ditayangkan sejak Bulan Mei 1999, Solusi merupakan program TV rohani yang bertahan paling lama dan tetap tayang di televisi nasional. Kesaksian kisah nyata hidup telah menjadi inspirasi dan jawaban atas permasalahan

hidup pemirsa yang menyaksikan program televisi ini. Tayang di SCTV setiap Senin, 24.30 WIB

b. Solusi Life

Program yang dikemas untuk memenuhi kebutuhan spiritual masyarakat perkotaan. Tayang setiap hari di TV Lokal Jabodetabek. Tayang di O'Channel TV (TV lokal di Jabodetabek) Senin - Jumat, 22.00 WIB.

c. Dina Hiji Mangsa

Program TV Solusi yang dialihbahasakan ke dalam bahasa sunda, menceritakan kasih dan kebaikan Tuhan. Ditayangkan di bandung dan sekitarnya.

d. Obat Malam Khusus Dewasa

Program untuk menjangkau masyarakat kalangan menengah-bawah. Kesaksian-kesaksian hidup yang dikupas secara terbuka. Dengan musik dangdut dan *talk show* membawakan topik-topik yang diminati oleh masyarakat kalangan menengah ke bawah. Telah disaksikan oleh 10.770.411 penonton.

e. www.jawaban.com

Website kristen No 1 di Indonesia. Selama tahun 2010 Jawaban.com telah dikunjungi lebih dari 3.249.182 pengunjung dengan rata-rata 9,1 *page turns* untuk setiap pengunjung.

f. Sekolah Media CBN

Sekolah Media CBN Hadir dengan visi melahirkan profesional di bidang media, yang memiliki hati untuk menjangkau bangsa dengan kasih Tuhan dan kerinduan menjadi garam dalam dunia *broadcasting*. Mendatangkan pengajar dan praktisi media yang memiliki standar internasional, serta senantiasa dikombinasikan dengan pengajaran karakter bagi para siswanya. Sepanjang tahun 2010, terdapat 85 praktisi media yang dihasilkan Sekolah Media CBN.

2.1.2 Visi dan Misi Cahaya Bagi Negeri Indonesia

a. Visi Perusahaan

Memberitakan kasih Tuhan sehingga bangsa Indonesia mendengar kabar baik.

b. Misi Perusahaan

Cahaya Bagi Negeri Indonesia memiliki misi : " Memberitakan dan menyatakan kasih Tuhan kepada bangsa Indonesia melalui pelayanan media dan pelayanan kemanusiaan dengan cara yang kreatif dan kontekstual ".

2.1.3 Logo Instansi

CBN adalah suatu perusahaan yang bergerak di bidang *Broadcasting* dan *Production House*. Perusahaan ini sudah banyak melakukan pembuatan film.

Dengan fakta tersebut, dapat disimpulkan bahwa CBN merupakan perusahaan *Broadcasting* dan *Production House* yang terpercaya baik dari segi kualitas kerja dan Pembuatan film. Berikut ini adalah logo instansi/perusahaan dari CBN :

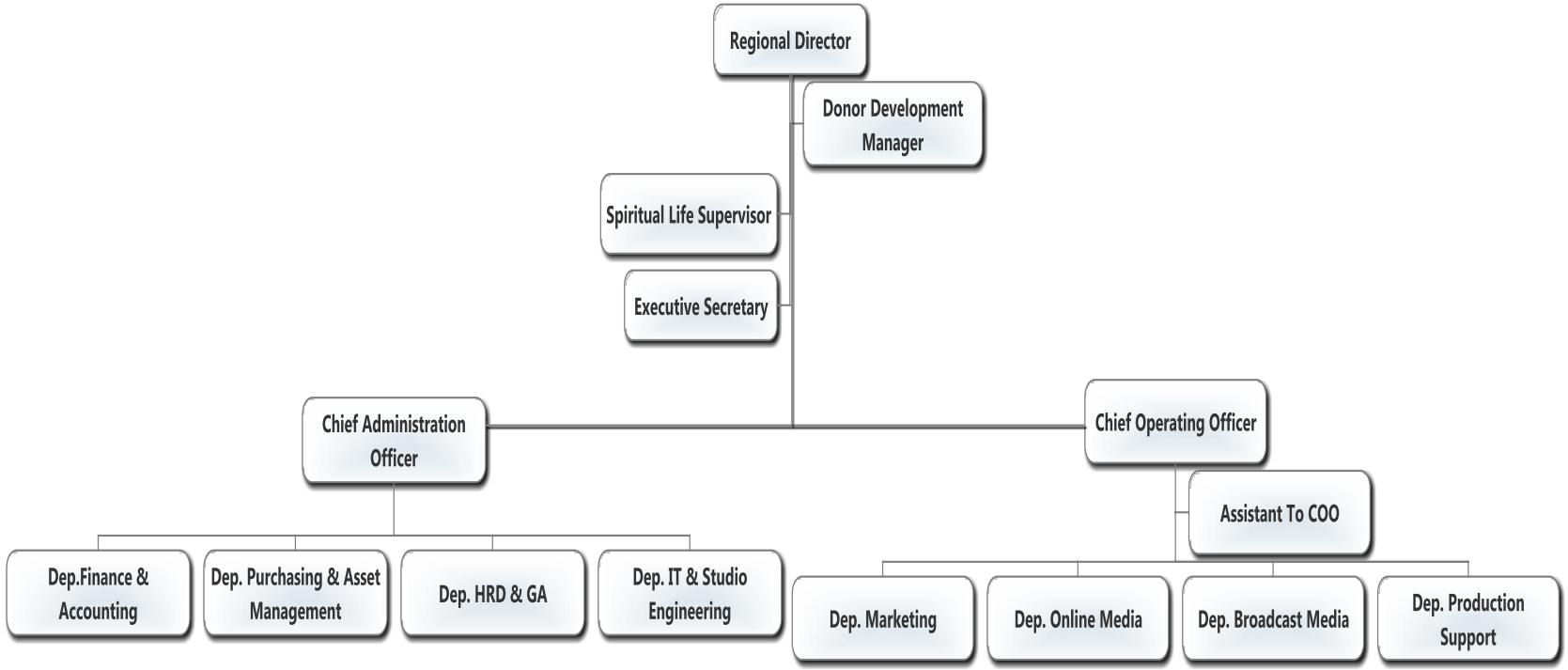


Gambar 2.1 Logo CBN Indonesia

Keterangan logo :

- Logo CBN Merupakan singkatan Cahaya Bagi Negeri
- Arti dan maksud pemakaian warna pada logo instansi tersebut adalah :
  - Merah : Mencerminkan Keuletan dan ketegasan serta keberanian dalam menghadapi berbagai macam keadaan.
  - Kuning : Mencerrminkan Haparan dan Optimis yang menyatakan kasih

2.1.4 Struktur Organisasi



Gambar 2.2 Struktur Organisasi CBN Indonesia

2.1.5 Deskripsi Pekerjaan *Marketing* CBN Indonesia

Secara garis besar, pekerjaan atau jobdesk yang telah dilakukan sebagai *marketing* di bagian *Marketing* CBN Indonesia akan di uraikan dibawah ini :

1. Menjaga dan meningkatkan hubungan dengan donatur.
2. Menyiapkan prospek klien baru.
3. Menjalin komunikasi yang baik dengan subcriber/mitra.
4. Memiliki keterampilan secara kuantitatif yang baik
5. Mempertahankan pelanggan yang telah ada.
6. Memastikan pencapaian target dalam menjaring mitra
7. Mengirim *email* pada setiap *subcriber*.
8. Membuat laporan data tracking *email* perusahaan
9. Melaporkan aktivitas penjaringan mitra perusahaan kepada atasan.

2.2 Landasan Teori

Pada landasan teori ini akan dibahas teori-teori yang dapat membantu memecahkan permasalahan yang ada. Teori-teori tersebut meliputi konsep dasar dan definisi-definisi yang berkaitan dengan perangkat lunak yang digunakan serta faktor-faktor pendukung dalam pelaksanaan perancangannya.

2.2.1 Konsep Dasar Sistem

Kata sistem mempunyai beberapa pengertian, tergantung dari sudut pandang mana kata tersebut didefinisikan. Secara garis besar ada dua kelompok pendekatan sistem, yaitu: pendekatan sistem yang lebih menekankan pada elemen-elemen atau kelompoknya didefinisikan sebagai “Suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu aturan tertentu.

Pendekatan sistem sebagai jaringan kerja dari prosedur, yang lebih menekankan urutan operasi didalam sistem didefinisikan sebagai “urutan operasi kerja (tulis-menulis), yang biasanya melibatkan beberapa orang di dalam satu atau lebih departemen, yang diterapkan untuk menjamin penanganan yag seragam dari transaksi bisnis yang terjadi. Suatu sistem mempunyai karakteristik atau sifat-sifat

yang tertentu, yaitu mempunyai komponen sistem (*component*), batasan sistem (*boundary*), lingkungan luar sistem (*environment*), penghubung sistem (*interface*), masukan sistem (*input*), keluaran sistem (*output*), pengolahan sistem (proses) dan sasaran (*objective*) atau tujuan (*goal*).

1. Komponen Sistem (*System Components*)

   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 menunjukkan 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 menpengaruhi 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

   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. Masukan 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* didalam 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. Pengolahan 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 tujuan. Sasaran sistem sangan menentukan masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.[2]

2.3 Data

2.3.1 Pengertian Data

Data adalah kumpulan dari fakta-fakta, kejadian-kejadian yang dapat berupa simbol, angka, huruf dan lain lain yang berguna bagi suatu pengolahan data (*process*) atau sebagai masukan (*input*) bagi suatu proses.[2]

2.3.2 Model Data

Data yang disimpan menggambarkan beberapa aspek dari suatu organisasi. Model data adalah himpunan deskripsi data level tinggi yang dikonstruksi untuk menyembunyikan beberapa detail dari penyimpanan level rendah. Beberapa manajemen basis data didasarkan pada model data relasional, model data hirarkis, atau model data jaringan.

1. Model Data Hirarkis

   Model hirarkis biasa disebut model pohon, karena menyerupai pohon yang dibalik. Model ini menggunakan pola hubungan orang tua-anak. Setiap simpul (biasa dinyatakan dengan lingkaran atau kotak) menyatakan sekumpulan medan. Simpul yang terhubung ke simpul pada level dibawahnya disebut orang tua. Setiap orang tua bisa memiliki satu (hubungan 1:1) atau beberapa anak (hubungan 1:M), tetapi setiap anak hanya memiliki satu orang tua. Simpul-simpul yang dibawahi oleh simpul orang tua disebuah anak. Simpul orang tua yang tidak memiliki orang tua disebut akar. Simpul yang tidak mempunyai anak disebut daun. Adapun hubungan antara anak dan orang tua disebut cabang.

2. Model Data Jaringan

   Model jaringan distandarisasi pada tahun 1971 oleh *Data Base Task Grup* (DBTG). Itulah sebabnya disebut model DBTG. Model ini juga disebut model CODASYL (*Conference on Data System Languages*), karena DBTG adalah bagian dari CODASYL.

   Model ini menyerupai model hirarkis, dengan perbedaan suatu simpul anak bisa memilki lebih dari satu orang tua. Oleh karena sifatnya demikian, model ini menyatakan hubungan 1:1 (satu arang tua punya satu anak), 1:N (satu orang tua punya banyak anak), maupun N:N (beberapa anak bisa mempunyai beberapa orangtua). Pada model jaringan, orang tua diseut pemilik dan anak disebut anggota.

3. Model Data Relasional

   Model relasional adalah model data yang paling banyak digunakan saat ini. Pembahasan pokok pada model ini adalah relasi, yang dimisalkan

sebagai himpunan dari *record*. Deskripsi data dalam istilah model data disebut skema. Pada model relasional, skema untuk relasi ditentukan oleh nama, nama dari tiap *field* (atau atribut atau kolom), dan tipe dari tiap *field*.

2.4 Konsep Dasar Informasi

Informasi (*information*) dapat didefinisikan sebagai berikut. Informasi adalah data yang sudah diolah menjadi sebuah bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya atau pengguna. Sumber dari informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal datum atau item-item. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian (*event*) adalah sesuatu yang terjadi pada saat yang tertentu. [2]

2.4.1 Kualitas Informasi

Kualitas dari suatu informasi tergantung pada tiga hal pokok yaitu:

1. Akurat (*accurate*)

   Informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan, dalam hal ini informasi harus jelas mencerminkan maksudnya.

2. Tepat waktu (*time lines*)

   Informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi karena informasi merupakan suatu landasan dalam mengambil sebuah keputusan dimana bila pengambilan keputusan terlambat maka akan berakibat fatal untuk organisasi.

3. Relevan (*relevance*)

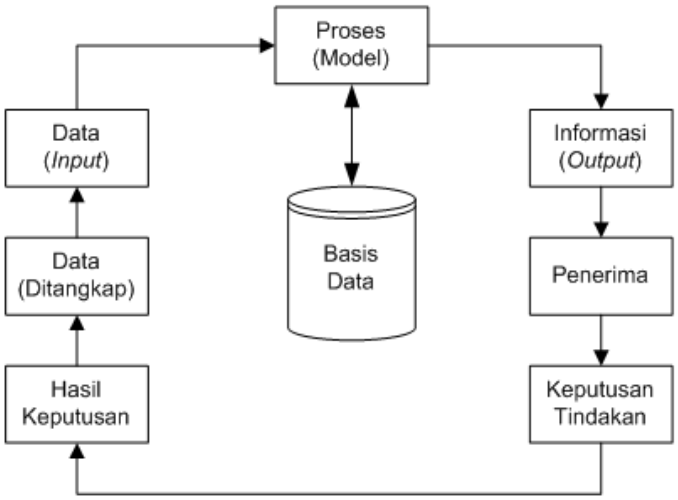
   Informasi harus mempunyai manfaat untuk pemakainya, dimana relevansi informasi untuk tiap-tiap individu berbeda tergantung pada yang menerima dan yang membutuhkan.

2.4.2 Nilai Informasi

Nilai informasi (*value of information*) ditentukan oleh dua hal yaitu manfaat dan biaya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya.

2.4.3 Siklus Informasi

Data agar menjadi lebih berarti dan berguna dalam bentuk Informasi, maka perlu diolah melalui suatu model tertentu. Data yang telah diolah tersebut kemudian diterima oleh penerima, lalu penerima membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai input, dan diproses kembali lewat suatu model dan seterusnya sehingga membentuk suatu siklus. Siklus ini disebut dengan siklus Informasi (*information cycle*) atau disebut pula siklus pengolahan data (*processing cycles*) yang dapat dilihat pada gambar 2.3.



Gambar 2.3 Siklus Informasi

2.5 Konsep Dasar 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.

Dan Sistem Informasi dapat diartikan juga sebagai kumpulan dari sub-sub sistem komponen baik phisik maupun non phisik yang saling berhubungan satu sama lain dan bekerjasama secara harmonis untuk mencapai satu tujuan yaitu mengolah data menjadi informasi yang berguna

Dari definisi diatas dapat diambil kesimpulan bahwa sistem informasi merupakan perpaduan antara manusia, alat teknologi, media, prosedur dan pengendalian yang bertujuan untuk menata jaringan komunikasi sehingga dapat membantu dalam pengambilan keputusan yang tepat. Kegiatan yang terdapat pada sistem informasi antara lain :[2]

a. *Input*, menggambarkan suatu kegiatan untuk menyediakan data yang akan diproses.
b. Proses, menggambarkan bagaimana suatu data diproses untuk menghasilkan suatu informasi yang bernilai tambah.
c. *Output*, suatu kegiatan untuk menghasilkan laporan dari proses diatas.
d. Penyimpanan, suatu kegiatan untuk memelihara dan menyimpan data.
e. Kontrol, suatu aktifitas untuk menjamin bahwa sistem informasi tersebut berjalan sesuai dengan yang diharapkan.

Sistem informasi dalam sebuah sistem meliputi pemasukan data (*input*) kemudian diolah melalui suatu model dalam pemrosesan data, dan hasil informasi akan ditangkap kembali sebagai suatu input dan seterusnya sehingga membentuk siklus informasi yang dapat diperoleh dari sistem informasi sebagai sistem khusus dalam organisasi untuk mengolah informasi tersebut.

2.5.1 Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building block*) yaitu:

a. *Hardware* yaitu suatu perangkat keras dalam komputer yang kita bisa sentuh dan rasakan.
b. *Software* yaitu suatu perangkat lunak di dalam komputer yang berfungsi untuk mengoperasikan suatu aplikasi di dalam sistem komputer.

c. Data yaitu sekumpulan karakter yang diterima sebagai masukan (*input*) untuk sistem informasi dan disimpan serta diolah.
d. Prosedur yaitu suatu urutan pekerjaan tata usaha yang biasanya melibatkan beberapa orang dalam satu bagian atau lebih, dan disusun untuk menjamin adanya perlakuan yang seragam terhadap transaksi yang terjadi.
e. *User* yaitu orang yang terlibat dalam sistem informasi seperti *operator*, pemimpin sistem informasi, dan sebagainya.

2.5.2 Tujuan Sistem Informasi

Sistem Informasi memiliki beberapa tujuan, yaitu:

a. Integrasi sistem
    1. Menghubungkan sistem individu/kelompok
    2. Pengkolektifan data dan penyambungan secara otomatis
    3. Peningkatan koordinasi dan pencapaian sinergi
b. Efisiensi pengelolaan
    1. Penggunaan basis data dalam upaya kesamaan pengadministrasian data
    2. Pengelolaan data berkaitan dengan karakteristik Informasi
    3. Penggunaan dan pengambilan Informasi
c. Dukungan keputusan untuk manajemen
    1. Melengkapi Informasi guna kebutuhan proses pengambilan kebutuhan
    2. Akuisisi Informasi eksternal melalui jaringan komunikasi
    3. Ekstraksi dari Informasi internal yang terpadu.

2.5.3 Manfaat Sistem Informasi

Sistem informasi ini memiliki beberapa manfaat, yaitu :

1. Menghemat tenaga kerja.
2. Peningkatan efisiensi.
3. Mempercepat proses.
4. Perbaikan dokumentasi.

5. Pencapaian standar.
6. Perbaikan keputusan.

2.6 *Electronic Mail*

2.6.1 Sejarah Perkembangan *Email*

Email sudah mulai dipakai pada tahun 1960-an. Pada saat itu Internet belum terbentuk, yang ada hanyalah kumpulan *'mainframe'* yang terbentuk sebagai jaringan. Mulai tahun 1980-an, surat elektronik sudah bisa dinikmati oleh khalayak umum. Sekarang ini banyak perusahaan pos di berbagai negara menurun penghasilannya disebabkan masyarakat sudah tidak memakai jasa pos lagi.[8]

Anatomi Retal, sebagai contoh :

*user*name@example.com

keterangan :

- *Email*saya : nama kotak surat (mailbox) atau nama pengguna (*user*name) yang ingin dituju dalam mailserver
- Surabaya.vibriel.net.id : nama mailserver tempat pengguna yang dituju.
  a. Surabaya : *subdomain* (milik pemegang nama domain), biasanya merujuk pada suatu komputer dalam lingkungan pemilik domain.
  b. Vibriel : nama domain, biasanya menunjukkan nama perusahaan / organisan / perorangan.
  c. *Net* : *second level domain*, menunjukan bahwa domain ini termasuk kategori *networking*.
  d. Id : *top level domain*, menunjukkan bahwa domain ini terdaftar di otoritas domain indonesia.

2.6.2 Komponen *Email*

Sebuah *email* terdiri dari beberapa komponen penyusun. Komponen-komponen tersebut adalah :

1. *Mail User Agent*

Merupakan program yang digunakan untuk membuat dan membaca *email*. MUA bisa disebut juga sebagai *email reader* yang dapat menerima bermacam - macam perintah untuk pembuatan, penerimaan dan penjawaban pesan. Selain itu MUA juga dapat digunakan untuk memanipulasi *mailbox email* pengguna.

Beberapa MUA mengizinkan pemakaian format *Multipurpose Internet Mail Extension* (MIME) yang dapat digunakan untuk melampirkan *file* ke dalam suatu pesan. Dengan MIME ini, pesan yang dikirimkan dapat melampirkan file-file lain. *MailX*, pine dan *netscape* merupakan beberapa contoh MUA.

2. *Mail Transport Agent*

Pada saat proses pengiriman *email*, *email* tersebut diberikan ke MTA untuk dilakukan proses yang lebih lanjut.

Fungsi dari MTA adalah sebagai berikut :

- MTA menggunakan alamat tujuan untuk menentukan bagaimana pesan tersebut harus dikirimkan.
- MTA dapat menggunakan aliases/daftar distribusi untuk mengirimkan salinan dari sebuah pesan ke berbagai tujuan.
- MTA menerima dan memproses *email* yang masuk dari mesin lain dalam jaringan.

3. *Mail Channel* dan *Delivery Agent*

Mail channel memiliki dua komponen utama yaitu :

- Tabel yang menentukan saluran yang akan digunakan untuk mengirimkan pesan *email*.
- *Delivery agent* yang akan melakukan pengiriman pesan bagi saluran yang telah ditentukan.

MTA akan mengenali saluran pengiriman pertama yang sesuai dengan alamat pesan pada tabel dan menggunakan agen pengiriman yang sesuai

untuk mengirimkan pesan. Alamat yang tidak sesuai dengan saluran yang ada akan dianggap sebagai alamat yang tidak dapat dikirimkan dan akan dikembalikan ke pengirim. Jenis-jenis saluran yang digunakan :

- Local channel yang menangani pengiriman *email* ke pengguna lokal.
- SMTP channel akan mengirimkan *email* melalui jaringan TCP/IP dengan menggunakan DNS dan SMTP.
- Badhost channel yang secara default akan mengantrikan pesan-pesan yang sementara waktu tidak dapat dikirimkan karena nama *server* tidak siap.

Pesan *email* akan di-*route* dari pengirim ke penerima oleh MTA pada host *email* yang berbeda. Komunikasi diantara *host* yang berbeda ditangani oleh program yang disebut *mailer*.

2.6.3 Bentuk Komunikasi Dengan *Email*

Pada dasarnya *email* memang meniru snail *email*, meski begitu akan tetapi *email* ini memiliki kelebihan yang terlihat jelas dan kemungkinan benruk komunikasinya yang bisa digunakan dengan *email* Sekurang kurangnya ada 5 macam dintaranya :

1. ***Poin To Poin***

   Maksud dari bentuk poin to poin yaitu pengiriman *email* yang secara langsung ke sebuah alamat *email* tertentu di indonesia yang dimana sebelumnya telah mengadakan keepakatan antara penerima *email* dengan pengirim yang sering kirim mengirim *email*, dan cara ini juga bias disebut japri, dimana japri merupakan kepanjangan dari jalur pribadi.

2. ***Carbon Copy* /Cc**

   Seperti telah dijelaskan sebelumnya bentuk *Carbon copy* ini adalah sebuah *email* yang selain ditunjukan ke sebuah alamat *email* utama yang kita tuliskan pada bagian yang diawali dengan “TO” , sedangkan alamat *email* tembusannya kita tuliskan pada bagian “CC”.

3. ***Blind Carbon Copy* / Bcc**

Blind carbon copy ini merupakan variasi dari CC dimana orang yang menerima *email* alamatnya dituliskan pada bagian "TO" oleh si pengirim, dan pengirim juga bisa melihat kapada siapa atau ke alamat *email* mana saja *email*-nya dikirimkan sebagai tembusan. Untuk melihatnya bias dilihat pada bagian "CC". Akan tetapi kadang apabila kita mengirim *email* kepada seseorang dan kita tidak ingin orang tersebut mengetahuinya bahwa kita juga mengirimkan *email* tembusan kepada alamat *email* orang lain. Oleh karena itu agar tidak diketahui digunakanlah BCC. Tapi ada beberapa *email client* yang menyediakan bagian "BCC", selain "TO" dan"CC" pada *outlook Express* ver, 5.01 pada bagian BCC akan tampak apabila kita masuk ke program sebuah *email* baru dan mengklik pilihan *view* dan *all header*.

4. ***Distribution List***

   Distribution list ini merupakan *distribution list* yang agak berbeda dengan tiga varian bentuk diatas,karena *distribution list* ini lebih berkenaan dengan pengaturan *distribusi email* yang dilakukan secara eksternal, atau Pengaturan *distribusi email* satu arah yang bisa digunakan untuk menyebarluaskan informasi tertentu dan tidak mengharapkan adanya respon atau balasan dari penerima.

5. ***Discussion List***

   Discussion list ini bisa disebut juga *mailing list* atau milist. Yang mana penjelasannya hampir sama dengan *distribution list*, hanya saja cara penggunaannya memungkinkan siapapun bisa tergabung menjadi *subscriber* untuk mengirimkan respon. Akan tetapi untuk bisa tergabung menjadi *subscriber* terkadang suka ada milis yang mempersyaratkan calon anggotanya untuk memberikan keterangan mengenai jati dirinya sebelum permohonan bergabung disetujui, dan apabila administratur milis menyetujuinya untuk bergabung maka biasanya tidak lama kita akan dikirimkan informasi mengenai tatacara pengiriman *email* dan keterangan keterangan lainnya tentang milis tersebut seperti tujuan milis, yang boleh

dilakukan dan yang tidak boleh dilakukan, cara keluar dari milis, alamat *email* administratur dan lain sebagainya.

2.6.4 Format Isi *Email*

Format isi *Email* adalah bagaimana penampakan visual sebuah pesan *email*. Ada dua macam yaitu :

1. ***PlainText***

   Dengan *plain text*, dimaksudkan bahwa tulisan yang kita ketikkan hanya berupa data dalam format ASCII (*American Standard Code for Information Interchange*). Tulisan yang disajikan dalam format *plain text* sama sekali tidak memungkinkan kita untuk membuat modifikasi pada tampilan visualnya, seperti membuat tebal, atau membuat miring sebuah teks, mengganti warna, ukuran huruf, latar belakang, atau membubuhkan hiasan-hiasan tertentu.

2. **HTML**

   HTML singkatan dari *HyperText Mark-up Language*, yaitu sebuah bahasa standard yang digunakan untuk menampilkan / mengatur penampilan materi-materi informasi di internet. *Email* yang ditulis dengan menggunakan format HTML akan memungkinkan isinya disusun secara variatif, mulai dari pengaturan huruf, warna, lay out, sampai peletakan gambar.Seringkali bila kita mengirim *email* ke sebuah forum diskusi / milis ada himbauan untuk hanya mengirimkan *email* dalam format *plain text* saja. Hal ini disebabkan biasanya *email* dalam format HTML memiliki ukuran yang lebih besar dari yang berformat *plain text*. Alasan lain adalah karena tidak semua orang dalam sebuah milis menggunakan *email* client yang bisa menampilkan HTML, sehingga ketika sebuah *email* HTML dibuka, isinya hanya serangkaian kode-kode yang tidak bisa dipahami sama sekali. Sebenarnya hampir semua *email* client dewasa ini sudah mendukung format HTML, sehingga kini jarang ditemui keluhan seseorang yang tidak bisa membaca sebuah *email*, hanya karena *email* yang dibacanya itu dibuat dalam format HTML.[8]

2.6.5 *Email Tracking*

Email tracking adalah metode untuk memantau pengiriman *email* ke penerima yang dimaksud. Teknologi pelacakan yang paling memanfaatkan beberapa bentuk digital dicap waktu untuk mengungkapkan catatan waktu yang tepat dan tanggal yang e-*mail* diterima atau dibuka, serta alamat IP penerima. *email tracking* berguna ketika pengirim ingin tahu apakah penerima dituju benar-benar menerima *email*, atau jika mereka mengklik *link*. Namun, karena sifat teknologi, *email* pelacakan tidak dapat dianggap sebagai indikator benar-benar akurat bahwa pesan dibuka atau dibaca oleh penerima.

Beberapa hal yang di *tracking* dari *email* yaitu sebagai berikut :

1. ***Open rate***

   Open rate merupakan *tracking* untuk mengetahui sudah berapa banyak *subscriber* melakukan pembukaan *email* dan untuk mengetahui berapa jumlah orang yang sudah membuka *email*.

2. ***Click rate***

   Click rate merupakan *tracking* untuk mengetahui jumlah orang yang mengklik *link* dalam *email* serta jumlah kali link diklik di *email*.

3. ***Unsubcribe***

   Uncubcribe untuk mengetahui jumlah orang yang dihapus *email* mereka dari daftar anda, menggunakan *link* manajemen berlangganan (*platform email* akan selalu menyertakan ini pada *email* yang dikirim melalui sistem mereka) di bagian bawah *email*.

4. ***Bounce***

   Bounce untuk mengetahui jumlah orang yang tidak mendapatkan *email* misalnya akun *email* mereka tidak bisa dihubungi.

5. ***Complaint***

   Complaint untuk mengetahui jumlah dari kontak melaporkan pesan anda sebagai *spam* di klien *email* mereka.

2.6.6 Metode Pengiriman

Untuk mengirim surat elektronik kita memerlukan suatu program *mail*-client. *E-mail* yang kita kirim akan melalui beberapa poin sebelum sampai di tujuan. Mulai dari *email* dikirim → *Internet* → POP3 *server* penyedia *email* penerima → e*mail client* (di komputer si penerima) → *email* dibaca si penerima

Terlihat *email* yang terkirim hanya melalui 5 poin (selain komputer pengirim dan penerima). Sebenarnya lebih dari itu sebab setelah *email* meninggalkan POP3 *Server* maka itu akan melalui banyak *server-server* lainnya. Tidak tertutup kemungkinan *email* yang kita kirim disadap orang lain. Maka dari itu bila *email* yang kita kirim mengandung isi yang sensitif sebaiknya kita melakukan tindakan pencegahan, dengan mengacak data dalam *email* tersebut (contohnya menggunakan PGP, sertifikat digital, dan lain-lain).[7]

2.6.7 Keamanan

Keamanan data di surat elektronik tidaklah terjamin dan selalu ada risiko terbuka untuk umum, dalam artian semua isinya dapat dibaca oleh orang lain. Hal ini disebabkan oleh karena surat elektronik itu akan melewati banyak *server* sebelum sampai di tujuan. Tidak tertutup kemungkinan ada orang yang menyadap surat elektronik yang dikirimkan tersebut.

Surat elektronik dapat diamankan dengan melakukan teknik pengacakan (enkripsi). Salah satu program enkripsi yang populer adalah PGP (*Pretty Good Privacy*). Dengan memakai PGP maka isi akan dienkrip, dan hanya orang yang tertuju dapat mendekripsi dan membaca surat elektronik tersebut. Kerugiannya adalah membuat repot pihak pengirim dan penerima (karena keduanya harus memiliki program PGP, dan pengirim juga harus memiliki kunci umum penerima, dan melakukan enkripsi pesan dengan kunci tersebut).

2.7 *Email Marketing*

2.7.1 Pengertian *Email Markekting*

Email marketing merupakan salah satu strategi promosi online yang ampuh dan dahsyat. Sebagian besar internet marketer atau pebisnis *online* menggunakan *email marketing* untuk melakukan promosi di dunia online. Namanya *email marketing* jadi mekanismenya juga seperti kita berkirim pesan via *email*. Namun mekanisme kirim pesan dilakukan secara otomatis oleh jasa penyedia layanan *email marketing*. Jadi anda tidak perlu repot-repot mengirimkan *email* untuk iklan anda satu per satu ke calon pelanggan anda.

Email marketing adalah pemasaran pesan langsung secara komersial untuk sekelompok orang yang menggunakan *email*. Dalam arti luas, setiap *email* yang dikirim ke pelanggan potensial atau saat ini bisa dianggap pemasaran *email*. Ini biasanya melibatkan menggunakan *email* untuk mengirim iklan, meminta bisnis, atau meminta penjualan atau sumbangan, dan dimaksudkan untuk membangun loyalitas, kepercayaan, atau *brand awareness*. *Email* pemasaran dapat dilakukan baik daftar dijual atau *database* pelanggan saat ini. Secara umum, istilah ini biasanya digunakan untuk merujuk kepada mengirim pesan *email* dengan tujuan meningkatkan hubungan pedagang dengan pelanggan saat ini atau sebelumnya, untuk mendorong loyalitas pelanggan dan bisnis yang berulang, mendapatkan pelanggan baru atau pelanggan saat ini meyakinkan untuk membeli sesuatu segera, dan menambahkan iklan ke pesan *email* yang dikirim oleh perusahaan lain untuk pelanggan mereka.[6]

2.7.2 Jenis Pemasaran *Email* (*Types of Email Marketing*)

Email pemasaran dapat dilakukan melalui berbagai jenis *email*:

1. *Email* Transaksional (*Transactional emails*)

   Email transaksional biasanya dipicu berdasarkan tindakan pelanggan dengan perusahaan. Untuk memenuhi syarat sebagai pesan transaksi atau hubungan, tujuan utama komunikasi ini harus “untuk memfasilitasi, lengkap, atau mengkonfirmasi transaksi komersial yang

penerima sebelumnya telah setuju untuk masuk ke dalam dengan pengirim", bersama dengan beberapa definisi sempit lainnya transaksional *messaging*. pesan transaksional Dipicu termasuk menjatuhkan pesan keranjang, *email reset password*, pembelian atau *email* konfirmasi pesanan, *email* status *order*, menyusun ulang *email* dan penerimaan *email*.

Tujuan utama dari *email* transaksional adalah untuk menyampaikan informasi mengenai tindakan yang memicu itu. Namun, karena tarif terbuka yang tinggi (51,3% dibandingkan dengan 36,6% untuk *newsletter email*), *email* transaksional merupakan kesempatan untuk terlibat pelanggan: untuk memperkenalkan atau memperpanjang hubungan *email* dengan pelanggan atau pelanggan, untuk mengantisipasi dan menjawab pertanyaan atau silang menjual atau up-menjual produk atau jasa.

Banyak vendor perangkat lunak *newsletter email* menawarkan dukungan *email* transaksional, yang memberikan perusahaan kemampuan untuk menyertakan pesan promosi dalam tubuh *email* transaksional. Ada juga vendor *software* yang menawarkan jasa pemasaran *email* transaksional khusus, yang meliputi penyediaan ditargetkan dan personal pesan *email* transaksional dan menjalankan kampanye pemasaran tertentu (seperti program *customer referral*).

2. *Email* Langsung (*Direct Email*)

Email langsung atau gangguan pemasaran berbasis melibatkan mengirimkan *email* semata-mata untuk mengkomunikasikan pesan promosi (misalnya, pengumuman penawaran khusus atau katalog produk). Perusahaan biasanya mengumpulkan daftar pelanggan atau *email* prospek alamat untuk mengirim pesan promosi langsung, atau mereka juga bisa menyewa daftar alamat *email* dari perusahaan jasa, tapi *mail* yang aman *marketing* juga digunakan.

2.8 Analisis dan Perancangan Berorientasi Objek

2.8.1 Konsep Dasar Pendekatan Objek

Konsep dasar pendekatan objek adalah suatu teknik atau cara pendekatan baru dalam melihat permasalahan dari sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata. Ada banyak cara untuk mengabstraksikan dan memodelkan objek-objek tersebut, mulai dari abstraksi objek, kelas, hubungan antar kelas sampai abstraksi sistem.

2.8.2 Objek dan Kelas

Objek adalah abstraksi dari sesuatu yang mewakili dunia nyata seperti benda, manusia, satuan, organisasi, tempat, kejadian, struktur, status, dan hal-hal lainnya yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berperngaruh pada status objeknya. Dalam konteks OOP (*Object Oriented Programming*), objek adalah instansi yang dibentuk secara seketika dari kelas pada saat dieksekusi atau dapat dikatakan objek adalah instansi dari kelas. Objek memiliki siklus hidup mulai dari diciptakan, dimanipulasi, dan dihancurkan.

Kelas adalah kumpulan dari objek-objek yang memiliki karakteristik yang sama. Sebuah kelas akan mempunyai sifat (attribut), kelakuan (operasi), hubungan (*relationship*), dan arti. Suatu kelas dapat diturunkan dari kelas lain, dimana atribut dari kelas semula dapat diwariskan ke kelas yang baru.

2.8.3 Sistem Berorientasi Objek

Sistem berorientasi objek adalah sebuah sistem yang komponennya dibungkus (dienkapsulasi) menjadi kelompok data dan fungsi. Ada beberapa karakteristik atau sifat-sifat yang dimiliki oleh sebuah sistem berorientasi objek yaitu:

1. Abstraksi

   Prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana, dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

2. Enkapsulasi

   Pembungkusan atribut data dan layanan (operasi-operasi) yang dimiliki oleh objek, untuk menyembunyikan implementasi dari objek sehingga objek lain tidak mengetahui cara kerjanya. Konsep enkapsulasi ini mendukung *information hiding*. Untuk mengakses nilai atribut kelas harus melalui sebuah operasi. Detail implementasi dari data dan prosedur disembunyikan dari dunia luar, sehingga mengurangi efek samping ketika terjadi perubahan dalam kelas.

3. Pewarisan (*Inheritance*)

   Inheritance merupakan pewarisan sifat dari sebuah *class* ke *class* yang baru. *Subclass* Y merupakan pewaris dari *superclass* X, maka *subclass* Y mewarisi semua atribut dan operasi yang dimiliki oleh *superclass* X. Hal ini mendukung konsep *reuse*. Pada setiap level hirarki *class*, atribut dan operasi baru dapat ditambahkan ke *class* yang telah diwarisi dari level yang lebih tinggi dalam hirarki. Pada *inheritance* juga memungkinkan terjadinya *overriding*. *Overriding* terjadi ketika atribut dan operasi yang diwarisi, dimodifikasi untuk kebutuhan spesifik dari *class* yang baru

4. *Reusability*

   Reusability adalah pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut.

5. Generalisasi dan Spesialisasi

   Generalisasi dan spesialisasi menunjukkan hubungan antar kelas dan objek yang umum dengan kelas dan objek yang khusus.

6. Komunikasi antar Objek

   Komunikisi antar objek dilakukan lewat pesan (*message*) yang dikirim dari satu objek ke objek lainnya.

7. *Polymorphism*

   Polymorphism adalah kemampuan suatu objek untuk digunakan di banyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program. Hal ini membuat objek saling terpisah dari objek lainnya dan membuat setiap objek lebih independen.

2.8.4 Keuntungan Metodologi Berorientasi Objek

Pengembangan sistem dengan metode berorientasi objek dapat memberikan keuntungan-keuntungan sebagai berikut:

1. Meningkatkan produktivitas.

   Karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lainnya yang melibatkan objek tersebut (*reusable*).

2. Kecepatan pengembangan.

   Karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengkodean.

3. Kemudahan pemeliharaan.

   Karena dengan objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dari pola-pola yang mungkin sering berubah.

4. Adanya konsistensi.

   Karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan, maupun pengkodean.

5. Meningkatkan kualitas perangkat lunak.

   Karena pendekatan pengembangan lebih dekat dengan dunia nyata dan adanya konsistensi pada saat pengembangannya, perangkat lunak yang

dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.

2.9 *Unified Modelling Language* (UML)

2.9.1 Sejarah *Unified Modelling Language* (UML)

Tahun 1994, Grady Boch dan James Rumbaugh bergabung untuk menggunakan metode berorientasi objek. Ivan Jacobson bergabung pada tahun 1995, dan mereka bertiga fokus membuat suatu bahasa pemodelan objek standar sebagai ganti dari pendekatan atau metode objek standar. Berdasarkan kerja mereka dan hasil kerja lainnya pada industri, *Unified Modeling Language* **(**UML**)** versi 1.0 dirilis pada tahun 1997.[1]

Unified Modeling Language **(**UML**)** tidak menentukan metode untuk sistem-sistem pengembangan, tetapi sudah diterima luas sebagai standar untuk pemodelan objek. *Object Management Group* (OMG), badan standar industri, mengadopsi UML pada bulan November 1997 dan terus bekerja sama untuk meningkatkannya berdasarkan kebutuhan industri. Pada saat ini, salah satu industri telah merilis sebuah *sofware* yang mendukung UML yaitu *Visual Paradigm* 6.4 *Interprise edition*. Berbagai industri juga bermunculan dan mendukung penggunaan UML dengan berbagai produk, diantaranya *Rational Rose*, *SmartDraw*, *StarUML*, dan lain-lain.

2.9.2 Definisi *Unified Modelling Language* (UML)

Berikut ini definisi *Unified Modeling Language* (UML) menurut para ahli:

1. *Unified Modeling Language* (UML) adalah bahasa yang telah menjadi standard untuk visualisasi, menetapkan, membangun dan mendokumentasikan artifak suatu sistem perangkat lunak.[1]
2. *Unified Modeling Language* (UML) adalah alat bantu analisis serta perancangan perangkat lunak berbasis objek.[2]

   Berdasarkan beberapa pendapat yang dikemukakan diatas dapat ditarik

kesimpulan bahwa "*Unified Modeling Language* (UML) adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk menvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis OO (*Object Oriented*)".

2.9.3 Konsep Pemodelan Menggunakan UML

Pemodelan menggunakan *Unified Modeling Language* merupakan metode pemodelan berorientasi objek dan berbasis visual. Karenanya pemodelan menggunakan UML merupakan pemodelan objek yang fokus pada pendefinisian struktur statis dan model sistem informasi yang dinamis daripada mendefinisikan data dan model proses yang tujuannya adalah pengembangan tradisional. UML menawarkandiagram yang dikelompokan menjadi lima perspektif berbeda untuk memodelkan suatu sistem. Seperti satu set *blue print* yang digunakan untuk membangun sebuah rumah.

UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan bahasa pemodelan visual yang memungkinkan pengembang sistem membuat *blue print* atas visinya dalam bentuk yang baku. UML berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui sejumlah elemen grafis yang bisa dikombinasikan menjadi *diagram*. UML mempunyai banyak *diagram* yang dapat mengakomodasi berbagai sudut pandang dari suatu perangkat lunak yang akan dibangun. *Diagram-diagram* tersebut digunakan untuk :

1. Mengkomunikasikan ide.
2. Melahirkan ide-ide baru dan peluang-peluang baru.
3. Menguji ide dan membuat prediksi memahami struktur dan relasi-relasinya.

2.9.4 Tujuan Penggunaan UML

Adapun tujuan dari penggunaan *Unifed Modelling Language* (UML) adalah:

1. Memberikan bahasa pemodelan yang bebas dari berbagai bahas pemrograman dan proses rekayasa.
2. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.
3. Memberikan model yang siap pakai, bahsa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
4. UML bisa juga berfungsi sebagai sebuah (*blue print*) cetak biru karena sangat lengkap dan detail. Dengan cetak biru ini maka akan bias diketahui informasi secara detail tentang coding program atau bahkan membaca program dan menginterpretasikan kembali ke dalam bentuk *diagram* (*reserve enginering*).

2.9.5 Fokus *Unified Modeling Language* (UML)

Kerangka spesifikasi *Unified Modeling Language* (UML) menyediakan model-model yang tepat, tidak mendua arti *(ambigu)* serta lengkap. Secara khusus, *Unified Modeling Language (uml* menspesifikasikan langkah-langkah penting dalam pengambilan keputusan analisis, perancangan serta implementasi dalam sistem yang sangat bernuansa perangkat lunak *(software intensive system).*[1]

Dalam hal ini, *Unified Modeling Language* (UML) bukanlah merupakan bahasa pemprograman tetapi model-model yang tercipta berhubungan langsung dengan berbagai macam bahasa pemprograman, sehingga adalah mungkin melakukan pemetaan *(mapping)* langsung dari model-model yang dibuat dengan *Unified Modeling Language* (UML) dengan bahasa-bahasa pemprograman berorientasi obyek, seperti *Java, Borland Delphi, Visual Basic, C++,* dan lain-lain. Pemetaan *(mapping) Unified Modeling Language* (UML) bersifat dua arah yaitu :

1. Generasi kode bahasa pemprograman tertentu dari *Unified Modeling Language* (UML) *forward engineering.*
2. Generasi kode belum sesuai dengan kebutuhan dan harapan pengguna, pengembang dapat melakukan langkah balik bersifat *iterative* dari implementasi ke *Unified Modeling Language* (UML) hingga didapat sistem/peranti lunak yang sesuai dengan harapan pengguna dan pengembang.

2.9.6 Bangunan Dasar Metodologi UML

Unified Modelling Language (UML) bahasa yang telah menjadi standard untuk visualisasi, menetapkan, membangun dan mendokumentasikan artifak suatu sistem perangkat lunak". UML memiliki standar penulisan sebuah sistem blueprint meliputi proses bisnis, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database* dan komponen yang diperlukan dalam pembangunan aplikasi atau sistem. Dalam bahasa standar UML untuk pengembangan atau pembangunan sebuah aplikasi atau sistem yang dapat menyampaikan bagaimana membuat dan membentuk model-model. Elemen utama dari UML adalah building block, aturan-aturan yang menyatakan bagaimana building block diletakkan secara bersamaan dan beberapa mekanisme umum (*common*). Building block dalam UML adalah sebagai berikut:[1]

1. *Things* (Benda)

Things merupakan elemen yang paling penting dalam pemodelan UML,yaitu menjelaskan dari sebuah konsep. Komponen dari things adalah *classes, interface, collaboration, usecase, active classes, component, nodes, interactions, state machines, packages* dan *notes*.

2. *Relationship* (Hubungan)

Relationship merupakan elemen yang menghubungkan *thing* dengan *thing* lainnya. Terdapat empat macam relationship dalam UML yaitu :

a. *Dependency*

Dependency merupakan hubungan semantik antara dua benda yang apabila satu benda berubah maka benda yang berhubungan mengalami perubahan. *Dependency* digambarkan dengan sebuah panah dengan garis putus-putus.

b. *Association*

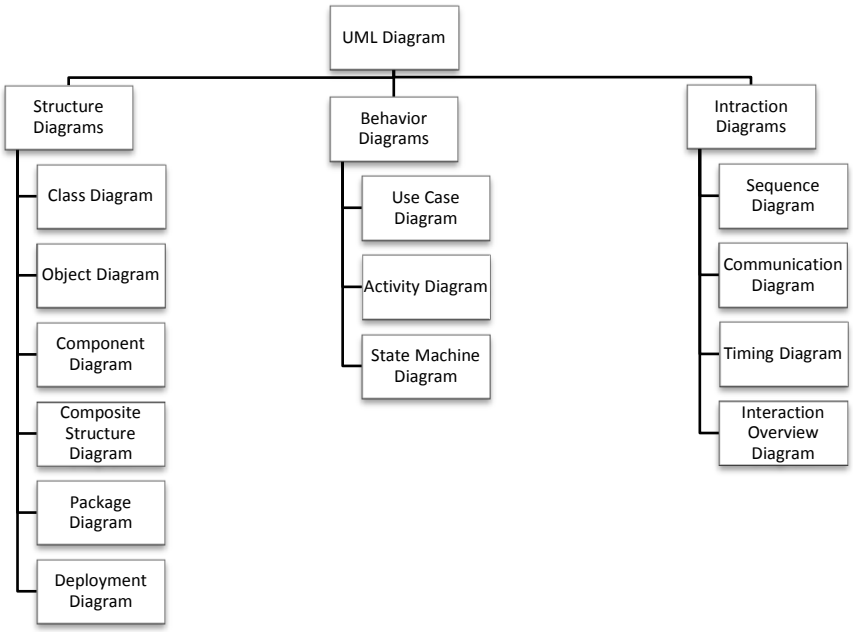
Assosiation merupakan hubungan antar benda yang berhubungan. Hubungan ini menggambarkan hubungan struktural antara seluruh atau sebagian objek. *Association* digambarkan dengan garis yang diberi label, nama dan status hubungan.

c. *Generalization*

Generalzation merupakan hubungan abtara objek *child*/siswa yang menggantikan objek *parent*/induk. Objek *child* atau siswa memiliki pengaruh dalam hal struktur dan tingkah laku objek parent atau induk. *Generalization* digambar kan dengan garis panah.

3. *Diagrams* (Bagan)

Pada UML ada 13 macam *diagram* yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam *diagram* tersebut dapat dilihat pada gambar 2.4.[4]



Gambar 2.4 *Diagram* UML

Berikut ini penjelasan singkat dari pembagian kategori tersebut.

a. *Structure diagrams* yaitu kumpulan *diagram* yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
b. *Behavior diagrams* yaitu kumpulan *diagram* yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
c. *Interaction diagrams* yaitu kumpulan *diagram* yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antarsubsistem pada suatu sistem.

1. Use Case

Use case merupakan situasi dimana sistem atau aplikasi dapat memenuhi satu atau lebih kebutuhan dari pengguna. *Use case* menggambarkan bagian dari fungsionalitas yang disediakan oleh sistem atau aplikasi. *Use case* merupakan pemodelan *diagram* yang penting karena sebuah *use case* akan mempengaruhi ke semua elemen *diagram* lainnya dalam sistem atau aplikasi.

2. Activity Diagrams

Activity diagram merupakan sebuah bagan bagaimana sistem atau aplikasi akan mencapai tujuannya. *Activity diagram* menunjukkan tindakan untuk mewakili suatu proses yang terjadi dalam sistem atau aplikasi. *Activity diagram* adalah sebuah proses bisnis yang terjadi dalam sistem atau aplikasi.

3. Class Diagrams

Class diagram dalam sebuah sistem atau aplikasi yang menggunakan *object programming* memiliki hubungan satu dengan yang lainnya. *Class diagram* menggambarkan apa yang terjadi pada class-class dalam sebuah sistem atau aplikasi. *Class diagram* tidak menjelaskan apa yang terjadi jika class pada sistem atau aplikasi berhubungan dengan class yang lainnya.

4. Sequence Diagrams

Sequence diagram adalah bagian penting dalam pemodelan sistem atau aplikasi menggunakan UML. *Sequence diagram* atau *diagram* interaksi merupakan memodelkan urutan interaksi dari bagian-bagian yang membentuk sebuah sistem atau aplikasi. *Sequence diagram* menggambarkan interaksi mana yang akan bereaksi ketika penggunaan fungsi-fungsi dilakukan dan untuk tujuan apa sebuah unteraksi bereaksi. Secara sederhana *sequence diagram* merupakan cara sederhana dan efektif bagaiman cara berkomunikasi urutan peristiwa dalam suatu interaksi.

2.10 *Fuzzy Multi-Attribut Decision Making* (FMADM)

Fuzzy Multiple Attribute Decision Making (FMADM) adalah metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu. FMADM menentukan nilai bobot untuk setiap atribut, lalu melakukan proses perankingan dari alternatif yang sudah ada. FMADM mempunyai tiga pendekatan untuk mencari nilai atribut, yaitu secara subyektif, obyektif, dan integrasi antara subyektif dan obyektif. Berikut beberapa metode yang bisa digunakan untuk menyelesaikan masalah FMADM:[4]

1. *Simple Additive Weighting* (SAW)
2. *Weighted Product* (WP)
3. ELECTRE
4. *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS)
5. *Analytic Hierarchy Process* (AHP)

2.11 *Analytic Hierarchy Process* (AHP)

AHP (*Analytical Hierarchy Process*) adalah salah satu metode dalam sistem pengambilan keputusan yang menggunakan beberapa variabel dengan proses analisis bertingkat. Analisis dilakukan dengan memberi nilai prioritas dari tiap-tiap variabel, kemudian melakukan perbandingan berpasangan dari variabel-variabel dan altematif - alternatif yang ada.[10]

AHP merupakan suatu model pengambilan keputusan yang bersifat komprehensif. AHP mempunyai kemampuan untuk memecahkan masalah yang multiobyektif dan multi-kriteria yang berdasar pada perbandingan preferensi dari setiap elemen dalam hirarki. Berikut ini adalah beberapa kelebihan AHP:[5]

1. Struktur yang berhirarki, sebagai konsekuensi dari kriteria yang dipilih, sampai pada subsubkriteria yang paling dalam.
2. Memperhitungkan validitas sampai dengan batas toleransi inkonsistensi berbagai kriteria dan altematif yang dipilih oleh para pengambil keputusan.
3. Memperhitungkan daya tahan atau ketahanan output analisis sensivitas pengambil keputusan.

2.11.1 Prinsip Kerja AHP

Prinsip kerja AHP adalah penyederhanaan suatu persoalan kompleks yang tidak terstruktur, stratejik, dan dinamik menjadi bagian-bagiannya, serta menata dalam suatu hierarki. Kemudian tingkat kepentingan setiap variabel diberi nilai numerik secara subjektif tentang arti penting variabel tersebut secara relatif dibandingkan dengan variabel lain. Dari berbagai pertimbangan tersebut

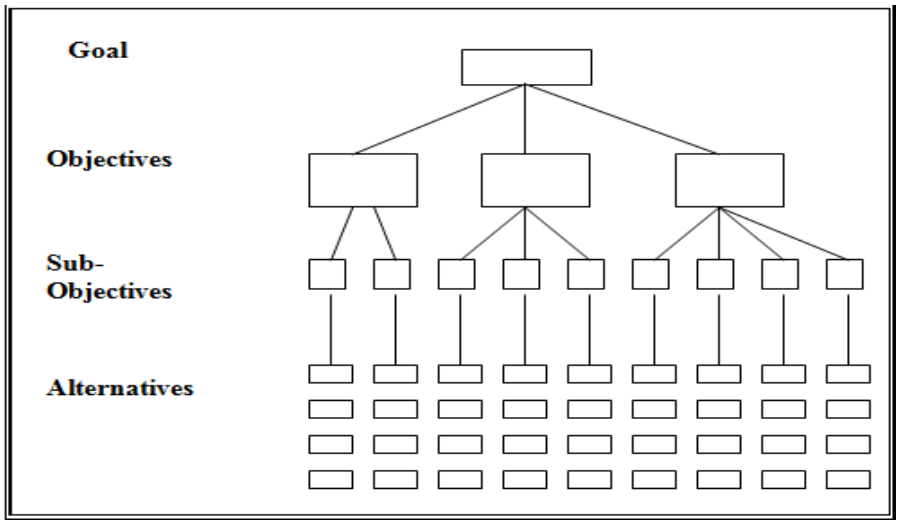
kemudian dilakukan sintesa untuk menetapkan variabel yang memiliki prioritas tinggi dan berperan untuk mempengaruhi basil pada sistem tersebut.

Penggunaan AHP dimulai dengan membuat struktur hirarki dari permasalahan (dekomposisi), melakukan pembandingan berpasangan antar variabel, melakukan analisis/evaluasi, dan menentukan altematif terbaik.[10]

Lebih lanjut mengemukakan bahwa pada dasarnya langkah - langkah dalam metode AHP diuraikan sebagai berikut: [5]

a. Menyusun hirarki dari permasalahan yang dihadapi.

   Persoalan yang akan diselesaikan, diuraikan menjadi unsur – unsurnya, yaitu kriteria dan alternatif, kemudian disusun menjadi struktur hierarki seperti gambar 2.5 di bawah ini.



Gambar 2.5 Struktur Hierarki AHP

b. Penilaian Kriteria dan Alternatif

   Kriteria dan altematif dinilai melalui perbandingan berpasangan. Untuk berbagai persoalan, skala 1 sampai 9 adalah skala terbaik dalam menekspresikan pendapat. Nilai dan definisi pendapat kualitatif dari skala perbandingan saaty dapat dilihat pada table 2.1.[5]

Tabel 2.1 Skala penilaian perbandingan berpasangan

Intensitas Kepentingan	Keterangan
1	Kedua elemen sama pentingnya
3	Elemen yang satu sedikitlebih penting daripada elemen yang lainnya
5	Elemen yang satu lebih penting daripada yang lainnya
7	Satu elemen jelas lebih mutlak penting daripada elemen lainnya
9	Satu elemen mutlak penting daripada elemen lainnya
2,4,6,8	Nilai-nilai antara dua nilaipertimbangan-pertimbangan yang berdekatan

Perbandingan dilakukan berdasarkan kebijakan pembuat keputusan dengan menilai tingkat kepentingan satu elemen terhadap elemen lainnya Proses perbandingan berpasangan, dimulai dari level hirarki paling atas yang ditujukan untuk memilih kriteria, rnisalnya A, kemudian diambil elemen yang akan dibandingkan, rnisal Al, A2, dan A3. Selanjutnya susunan elemen - elemen yang dibandingkan tersebut akan tampak seperti pada gambar 2.2 matriks di bawah ini:

Tabel 2.2 contoh matriks perbandingan berpasangan

	A1	A2	A3
A1	1		
A2		1	
A3			1

Untuk menentukan nilai kepentingan relatif antar elemen digunakan skala bilangan dari 1 sampai 9. Apabila suatu elemen dibandingkan dengan dirinya sendiri maka diberi nilai 1. Jika elemen 1 dibandingkan dengan elemen j mendapatkan nilai tertentu, maka elemen j dibandingkan dengan elemen i merupakan kebalikannya.

Dalam AHP ini, penilaian alternatif dapat dilakukan dengan metode langsung (direct), yaitu metode yang digunakan untuk memasukkan data kuantitatif. Biasanya nilai-nilai ini berasal dari sebuah analisis sebelumnya atau dan pengalaman dan pengertian yang detail dari masalah keputusan tersebut. Jika si pengambil keputusan memiliki pengalaman ataupemahaman yang besar mengenai masalah keputusan yang dihadapi, maka dia dapat langsung memasukkan pembobotan dari setiap alternatif.

c. Penentuan Prioritas

Untuk setiap kriteria dan alternatif, perlu dilakukan perbandingan berpasangan (painvise comparisons). Nilai-nilai perbandingan relatif kemudian diolah untuk menentukan peringkat altematif dari seluruh altematif.

Baik kriteria kualitatif, maupun kriteria kuantitatif, dapat dibandingkan sesuai dengan perulaian yang telah ditentukan untuk menghasilkan bobot dan proritas. Bobot atau prioritas dihitung dengan manipulasi matriks atau melalui penyelesaian persamaan matematik.

Pertimbangan-pertimbangan terhadap perbandingan berpasangan disintesis untuk memperoleh keseluruhan prioritas melalui tahapan-tahapan berikut:

a. Kuadratkan matriks hasil perbandingan berpasangan.
b. Hitung jumlah nilai dari setiap baris, kemudian lakukan normalisasi matriks.

d. Konsistensi Logis

Semua elemen dikelompokkan secara logis dan diperingatkan secara konsisten sesuai dengan suatu kriteria yang logis.

Matriks bobot yang diperoleh dari hasil perbandingan secara berpasangan tersebut harus mempunyai hubungan kardinal dan ordinal. Hubungan tersebut dapat ditunjukkan sebagai berikut:[5]

Hubungan kardinal : 3ii . aik = aik

Hubungan ordinal : Ai> Ai, Ai > Ak maka Ai > Ak

Hubungan diatas dapat dilihat dari dua hal sebagai berikut :

a. Dengan melihat preferensi multiplikaf, misalnya bila anggur lebih enak empat kalli dari mangga dan mangga lebih enak dua kali dari pisang maka anggur lebih enak delapan kali dari pisan.
b. Dengan melihat preferensi transitif, misalnya anggur lebih enak dari mangga dan mangga lebih enak dari pisang maka anggur lebih enak dari pisang.

Pada keadan sebenarnya akan terhadi beberapa penyimpangan dari hubungan tersebut, sehingga matriks tersebut tidak konsisten sempurna. Hal ini teradi karena ketidakkonsistenan dalam preferensi seseorang.

Penghitungan konsistensi logis dilakukan dengan mengikuti langkah-langkah sebagai berikut :

a. Mengalikan matriks dengan proritas bersesuaian.
b. Menjumlahkan hasil perkalian per baris.
c. Hasil penjumlahan tiap baris dibagi prioritas bersangkutan dan hasilnya dijumlahkan.
d. Hasil c dibagi jumlah elemen, akan didapat λmaks.
e. Indeks Konsistensi (CI) = (λmaks-n) I (n-1)

f. Rasio Konsistensi = CII RI, dimana RI adalah indeks random konsistensi. Jika rasto konsistensi < 0.1, hasil perhitungan data dapat dibenarkan.

Nilai RI didasarkan pada Penelitian yang dilakukan oleh [10], yang ditujukan pada tabel 2.3.

Tabel 2.3 Nilai indeks *random*

Ukuran	1	2	3	4	5	6	7	8	9	10
RI	0,00	0,00	0,52	0,89	1,11	1,25	1,35	1,40	1,45	1,49

2.12 Perangkat Lunak Yang Digunakan

2.12.1 PHP

PHP adalah singkatan dari *PHP Hypertext Preprocessor*. PHP adalah bahasa scripting yang menyatu dengan HTML dan dijalankan pada *server* side. Artinya semua sintaks yang kita berikan akan sepenuhnya dijalankan pada *server* sedangkan yang dikirimkan ke browser hanya hasilnya saja.PHP Pertama kali ditemukan pada 1995 oleh seorang *Software Developer* bernama Rasmus Lerdrof. Ide awal PHP adalah ketika itu Radmus ingin mengetahui jumlah pengunjung yang membaca *resume* onlinenya. *Script* yang dikembangkan baru dapat melakukan dua pekerjaan, yakni merekam informasi *visitor* dan menampilkan jumlah pengunjung dari suatu website dan sampai sekarang kedua tugas tersebut masih tetap populer digunakan oleh dunia web saat ini. Kemudian, dari situ banyak orang dimilis mendiskusikan script buatan Rasmus Lerdrof, hingga akhirnya rasmus mulai membuat sebuah *tool/script*, bernama Personal *Home Page*(PHP).

Kebutuhan PHP sebagai *tool* yang serba guna membuat Lerdorf melanjutkan untuk mengembangkan PHP hingga menjadi suatu bahasa tersendiri yang mungkin dapat mengkonversikan data yang di inputkan melalui *Form* HTML menjadi suatu variable, yang dapat dimanfaatkan oleh sistem lainnya. Untuk merealisasikannya, akhirnya Lerdrof mencoba mengembangkan PHP menggunakan bahasa C ketimbang menggunakan Perl. Tahun 1997, PHP versi 2.0 di rilis, dengan nama Personal *Home Page Form Interpreter* (PHP-FI). PHP Semakin popular, dan semakin diminati oleh *programmer web* dunia.

Rasmus Lerdrof benar-benar menjadikan PHP sangat populer, dan banyak sekali Team Developer yang ikut bergabung dengan Lerdrof untuk mengembangkan PHP hingga menjadi seperti sekarang, Hingga akhirnya dirilis versi ke 3-nya, pada Juni1998, dan tercatat lebih dari 50.000 programmer menggunakan PHP dalam membuat website dinamis.

Pengembangan demi pengembangan terus berlanjut, ratusan fungsi ditambahkan sebagai fitur dari bahasa PHP, dan di awaal tahun 1999, netcraft mencatat, ditemukan 1.000.000 situs di dunia telah menggunakan PHP. Ini membuktikan bahwa PHP merupakan bahasa yang paling populer digunakan oleh dunia web development. Hal ini mengagetkan para developernya termasuk Rasmus sendiri, dan tentunya sangat diluar dugaan sang pembuatnya. Kemudian Zeev Suraski dan Andi Gutsman selaku *core developer* (*programmer* inti) mencoba untuk menulis ulang PHP *Parser*, dan diintegrasikan dengan menggunakan *Zend scripting engine*, dan mengubah jalan alur operasi PHP. Dan semua fitur baru tersebut dirilis dalam PHP4.

13 Juli 2004, evolusi PHP, PHP telah mengalami banyak sekali perbaikan disegala sisi, dan wajar jika *netcraft* mengumumkan PHP sebagai bahasa *web* populer didunia, karena tercatat 19 juta domain telah menggunakan PHP sebagai *server side* scriptingnya. PHP saat ini telah Mendukung XML dan *Web Services*, Mendukung SQLite. Tercatat lebih dari 19 juta domain telah menggunakan PHP sebagai *server* scriptingnya. Benar-benar PHP sangat mengejutkan.

Yang menjadikan PHP berbeda dengan HTML adalah proses dari PHP itu sendiri. HTML merupakan bahasa statis yang apabila kita ingin merubah konten/isinya maka yang harus dilakukan pertama kali nya adalah, membuka file-nya terlebih dahulu, kemudian menambahkan isi kedalam file tersebut. Beda hal nya dengan PHP. Bagi anda yang pernah menggunakan CMS seperti *wordpress* atau *joomla* yang dibangun dengan PHP tentunya, ketika akan menambahkan konten kedalam *website*, anda tinggal masuk kedalam halaman admin, kemudian pilih new artikel untuk membuat halaman/content baru. Artinya hal ini, seorang *user* tidak berhubungan langsung dengan scriptnya. Sehingga seorang pemula sekalipun dapat menggunakan aplikasi seperti itu.

2.12.2 MySQL

MySQL adalah sebuah aplikasi *Relational Database Management Server* (RDBMS) bersifat *open source* yang memungkinkan data diakses dengan cepat oleh banyak pemakai secara bersamaan dan juga memungkinkan pembatasan akses pemakai berdasarkan *privilege* (hak akses) yang diberikan. *MySQL* menggunakan bahasa SQL (*structured query language)* yang merupakan bahasa standar pemograman *database*.

MySQL dipublikasikan sejak tahun 1996, akan tetapi sebenarnya sudah dikembangkan sejak tahun 1979. *MySQL* telah memenangakan penghargaan *Linux Journal Reader's Choice Award* selama tiga tahun. MySQL sekarang tersedia di bawah lisensi *open source*, tapi ada juga lisensi utuk menggunakan *MySQL* yang bersifat komersial.

Keunggulan dari *MySQL* adalah :

1. Bersifat *open source*.
2. Sistem *software*-nya tidak memberatkan kerja *server* atau komputer karena dapat bekerja di background.

2.12.3 PHPStorm 7.1.3

PhpStorm adalah PHP IDE ringan dan cerdas berfokus pada produktivitas pengembang yang sangat memahami kode Anda, menyediakan penyelesaian kode cerdas, navigasi cepat dan memeriksa *on-the-fly* kesalahan. Itu selalu siap untuk membantu Anda membentuk kode Anda, menjalankan unit-tes atau memberikan debugging visual. PhpStorm mencakup semua fungsi *WebStorm* (HTML / CSS *Editor*, Javascript *Editor*) dan menambahkan dukungan penuh untuk PHP. http://www.jetbrains.com/phpstorm/

2.12.4 Apache

Server HTTP *Apache* atau *Server Web*/WWW Apache adalah *server web* yang dapat dijalankan dibanyak sistem operasi (unix, BSD, Linux, Windows, Novotel Netware dan lainya) yang berguna untuk melayani dan memfungsikan

situs web. Protokol yang digunakan untuk melayani fasilitas web/www ini menggunakan HTTP.

Apache memiliki fitur-fitur yang canggih seperti pesan kesalahan yang dapat dikonfigur, autentikasi berbasis basis data dan lain-lain. *Apache* juga didukung oleh sejumlah antar muka pengguna berbasis grafik (GUI) yang memungkinkan penanganan *server* menjadi mudah.

Apache merupakan *software open soucre* dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan *Apache Software Foundation.*

2.12.5 Laravel

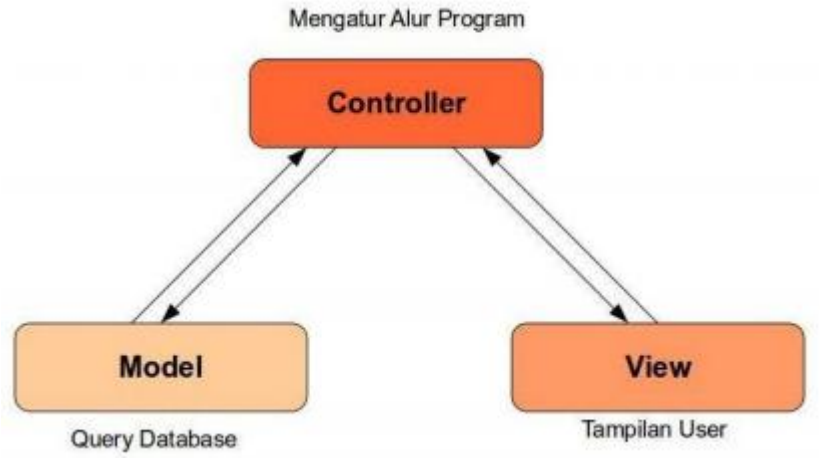
Laravel adalah sebuah Framework PHP 5.3 yang bersifat opensource yang ditulis oleh Taylor Otwell dengan lisensi dibawah MIT License. Laravel dibuat untuk membatu para developer khususnya dalam membuat sebuah web dengan sintak yang sederhana, elegan, ekspresif dan menyenangkan.

Laravel adalah aplikasi web dengan sintak yang ekspresif dan elegan. Dengan Laravel, tugas-tugas umum developer dapat dikurangi pada sebagian besar proyek-proyek web seperti routing, session dan caching. Disamping itu, laravel berusaha menggabungkan pengalaman-pengalaman development dalam bahasa lain, seperti Ruby on Rails, ASP.NET, MVC dan Sinatra.

2.12.6 MVC (*Model-View-Controller*)

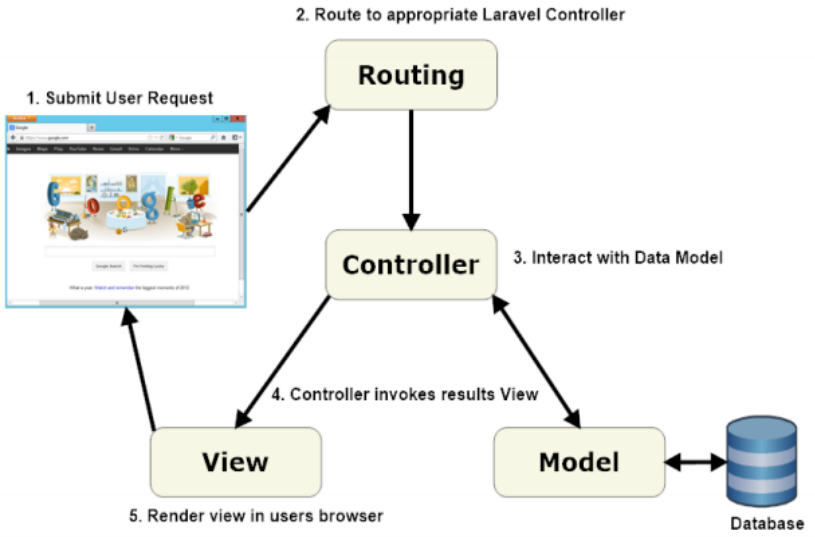
Laravel MVC atau kepanjangan dari Model-View-Controller adalah sebuah metode yang digunakan dalam pengembangan suatu aplikasi yang memisahkan data (model) dari tampilan / frontend (View) dan logic dari aplikasi itu sendiri (Controller). MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, antarmuka pengguna dan kontrol dalam sebuah aplikasi. Model digunakan untuk proses query atau manipulasi data ke/dari *database*. Sedangkan View kaitannya erat dengan antarmuka / frontend tampilan sebuah web seperti HTML, CSS dan JS dan data yang bersifat client. Controller adalah logika dari sebuah web.

Menjembatani komunikasi antara Model dan View. Kalau digambarkan alur proses MVC adalah sebagai berikut :



Gambar 2.6 Model - View - Controller

Adapun gambar simulasi proses MVC pada Laravel diperlihatkan pada gambar 2.8 dibawah ini:



Gambar 2.7 MVC Laravel

2.13 *Internet*

Internet merupakan jaringan global yang terdiri dari berbagai komputer yang saling berhubungan dan bekerjasama dengan cara berbagi 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*).

2.13.1 Sejarah *Internet*

Internet adalah singkatan dari *Interconnected Network*. Secara umum, Internet adalah sebuah sistem komunikasi global yang menghubungkan berbagai mesin computer dan jaringan – jaringan computer di seluruh dunia melalui kabel maupun tanpa kabel (*wireless*). Mesin komputer tersebut dapat berupa *server*, komputer pribadi (PC), telepon seluler (*hanpdhone*), PDA, dan lain-lain.

Pada awalnya Internet merupakan jaringan komputer yang dibentuk oleh Departmen Pertahanan Amerika Serikat pada tahun 1969, melalui proyek ARPA yang disebut ARPANET (*Advanced Research Project Agency Network*), dimana mereka mendomenstrasikan bagaimana dengan *hardware* dan *software* komputer yang berbasis *UNIX*, anda bisa melakukan komunikasi dalam jarak yang tak terhingga melalui saluran telepon.

Tujuan awal dibangunnnya proyek itu adalah untuk keperl;uan militer. Pada saat itu Departemen Pertahanan Amerika Serikat (*US Department Of Defense*) membuat sistem jaringan komputer yang terbesar dengan menghubungkan komputer di daerah-daerah vital untuk mengatasi masalah bila terjadi serangan nuklir dan untuk menghindari terjadinya informasi terpusat, yang apabila terjadi perang dapat mudah dihancurkan.

Pada mulanya ARPANET hanya menghubungkan 3 situs saja yaitu *Stamford Research Institute*, University of California, Santa Barbara, University

of Utah, dimana mereka membentuk satu jaringan terpadi di tahun 1969, dan secara umum ARPANET diperkenalkan pada bulan Oktober 1972. Tidak lama kemudian, proyek ini berkembang pesat di seluruh daerah, dan semua universitas di negara tersebut ingin bergabung, sehingga membuat ARPANET dipecah menjadi dua, yaitu "MILNET" untuk keperluan militer dan "ARPANET" baru lebih kecil keperluan *non*-militer seperti, universitas - universitas. Gabungan kedua jaringan akhirnya dikenal dengan nama DARPA Internet, yang kemudian disederhanakan menjadi Internet.

2.13.2 Kegunaan Internet

Kegunaan *internet* yang utama antara lain :

a. Fungsi komunikasi

   Internet adalah alat komunikasi, kegunaan yang sangat penting dari *internet* adalah pertukaran pesan dengan menggunakan *electronic mail* (*e-mail*).

b. Fungsi *Resource Sharing*

   Dengan *internet*, kita dapat mencari *software*, *essay*, data dan program dari ribuan titik distribusi di seluruh dunia.

c. Fungsi *Resource Discovery*

   Navigasi untuk mencari file tertentu, dokumen, *host* atau orang diantara jutaan *host*.

d. Fungsi Komunitas

   Masyarakat pengguna internet dapat berhubungan dan membuat perkumpulan/komunitas tertentu.

2.13.3 Perkembangan *Internet*

Jumlah pengguna *internet* yang besar dan semakin berkembang, telah mewujudkan budaya *internet*. *Internet* juga mempunyai pengaruh yang besar atas ilmu, dan pandangan dunia. Dengan hanya menggunakan mesin pencari seperti Google, pengguna di seluruh dunia mempunyai akses yang mudah atas

bermacam-macam informasi. Dibanding dengan buku dan perpustakaan, *internet* melambangkan penyebaran (*decentralization*) informasi dan data secara ekstrim.

Perkembangan internet juga telah mempengaruhi perkembangan ekonomi. Berbagai transaksi jual beli yang sebelumnya hanya bisa dilakukan dengan cara tatap muka (dan sebagian sangat kecil melalui pos atau telepon), kini sangat mudah dan sering dilakukan melalui *internet*.

Terkait dengan teknologi pembuatan *website* sebagai konten dari teknologi *internet* itu sendiri memicu berkembangan teknik pembuatan *website* yang semakin canggih dengan fitur-fitur yang sangat beragam tidak hanya terbatas pada teks dan gambar saja, akan tetapi sudah merambah kedalam dunia multimedia dan telkomunikasi visual. Bahkan dengan munculnya teknologi konten manajemen sistem pengguna yang awam dengan kemampuan teknis yang sedikipun bisa membuat *website* sesuai dengan keinginannya tanpa harus berhubungan dengan *webmaster* dengan konten yang dinamis dan selalu *uptodate*.

2.13.4 *Web Server*

Web server adalah seuatu program untuk menawarkan pelayanan yang bisa diperoleh seluruh jaringan. *Web server* merupakan suatu tipe *server* khusus yang dapat berkomunikasi langsung dengan *client* menggunakan HTTP, *web server* menerima permintaan dari *client* dan meresponnya, biasanya dengan mengembalikan sebuah dokumen atau gambar.

2.13.5 *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* computer 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 menampilkanya dengan benar. Teks ditampilkan sebagai teks dan gambar ditampilkan sebagai gambar. Umumnya *browser web* 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. Browserlah yang memiliki kuasa penuh dalam menterjemahkan perintah-perintah tadi. Meskipun sudah dibuat *consensus* untuk menstandarkan format dan elemen-elemen HTML, setiap jenis browser bisa menterjemahkan file HTML secara berbeda.

Beberapa *server web* memiliki *feature* seperti *server side programming*, *sevurity control* dan lain sebagainya. Meskipun beragam macamnya, secara fungsional semua jenis *server* web adalah sama saja, yaitu berfungsi melayani permintaan-permintaan dari *browser web*.

Banyak *web browser* yang bisa digunakan untuk mengakses *web*, diantaranya internet explorer, mozilla firefox, opera, safari, dan masih banyak lagi.

2.13.6 *Website*

Website merupakan *webpage* beserta *homepage*, yang merupakan sistem yang luas dari *server* yang menawarkan informasi untuk semua orang melalui jaringan. Informasi tersebut tersebut berupa teks, gambar, suara, dan tipe data *link*, yang merupakan navigasi kehalaman lainnya (*hyperlink*). *Homepage* merupakan halaman yang pertama kali muncul ketika pertama kali dibuka,sedang *webpage* adalah halaman yang ada pada *website*.

2.13.7 *World Wide Web* (WWW)

Sebuah situs *web* adalah sebutan bagi sekelompok halaman *web* (*web page*), yang umumnya merupakan bagian dari suatu nama domain (*domain name*) atau subdomain di *World Wide Web* (WWW) di Internet. WWW terdiri dari seluruh situs web yang tersedia kepada publik. Halaman-halaman sebuah situs *web* diakses dari sebuah URL yang menjadi "akar" (*root*), yang disebut *homepage* (halaman induk; sering diterjemahkan menjadi "beranda", "halaman muka"), dan biasanya disimpan dalam *server* yang sama. Tidak semua situs *web* dapat diakses dengan gratis. Beberapa situs *web* memerlukan pembayaran agar dapat menjadi pelanggan, misalnya situs-situs yang menampilkan pornografi, situs-situs berita, layanan surat elektronik (*email*), dan lain-lain.

Terminologi *website* adalah kumpulan dari halaman-halaman situs, yang biasanya terangkum dalam sebuah domain atau subdomain, yang tempatnya berada didalam *World Wide Web* (WWW) di *Internet*. Sebuah *web page* adalah dokumen yang ditulis dalam format HTML (*Hyper Text Markup Language*), yang hampir selalu bisa diakses melalui HTTP, yaitu protokol yang menyampaikan informasi dari *server website* untuk ditampilkan kepada para pemakai melalui *web browser*. Semua publikasi dari *website-website* tersebut dapat membentuk sebuah jaringan informasi yang sangat besar.

2.13.8 *Universal Resource Locator* (URL)

URL adalah rangkaian karakter menurut suatu format standar tertentu, yang digunakan untuk menunjukkan alamat suatu sumber seperti dokumen dan gambar di *internet*. URL menunjukkan sumber daya internet alamat sebuah halaman *web* (*homepage*) yaitu halaman suatu dokumen atau program yang ingin ditampilkan atau digunakan. Nama file ini tidak hanya menunjukan direktori dan nama filenya saja, tetapi juga nama mesinnya dalam jaringan. URL dapat disediakan (ada atau diakses) dengan berbagai metode dan bisa jadi bukan sekedar file. Karena URL dapat menunjukan *query-query* dokumen dapat disimpan dalam *database*, hasil dari suatu perintah *finger* atau *achie* atau apapun yang berkaitan dengan data hasil proses.

2.13.9 *Domain Name Service*(DNS)

DNS digunakan untuk mempermudah penggunaan internet, dengan memetakan IP address ke nama *host*. Agar data nama host dapat didistribusikan di banyak *server*, format data yang digunakan harus mencerminkan terdistribusinya data tersebut. Untuk itu, digunakan format *tree* dengan masing-masing nodenya disebut domain. Penulisan nama host secara lengkap disebut sebagai *Full Qualifield Domain Name* (FQDN).

DNS bekerja secara hirarki dan berbentuk seperti pohon (tree). Bagian atas adalah *Top Level Domain* (TLD) seperti COM, ORG, EDU, MIL dsb. Seperti pohon DNS mempunyai cabang-cabang yang dicari dari pangkal sampai ke ujung.

Pada waktu kita mencari alamat misalnya linux.or.id pertama-tama DNS bertanya pada TLD *server* tentang DNS *Server* yang melayani domain .id misalnya dijawab ns1.id, setelah itu dia bertanya pada ns1.id tentang DNS *Server* yang bertanggung jawab atas .or.id misalnya ns.or.id kemudian dia bertanya pada ns.or.id tentang linux.or.id dan dijawab 64.29.24.175.

Sedangkan untuk mengubah IP menjadi nama *host* melibatkan domain in-addr.arpa. Seperti domain lainnya domain in-addr.arpa pun bercabang-cabang. Yang penting diingat adalah alamat IP-nya ditulis dalam urutan terbalik di bawah in-addr.arpa. Misalnya untuk alamat IP 64.29.24.275 prosesnya seperti contoh linux.or.id: cari *server* untuk arpa, cari *server* untuk in-addr.arpa, cari *server* untuk 64.in-addr.arpa, cari *server* 29.64.in-addr.arpa, cari *server* untuk 24.29.64.in-addr.arpa. Dan cari informasi untuk 275.24.29.64.in-addr.arpa. Pembalikan urutan angkanya memang bisa membingungkan.