BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Intansi

Pada tinjauan perusahaan ini akan dibahas mengenai sejarah berdirinya intansi, struktur organisasi serta uraian tugas dari masing-masing bagian yang terlibat.

2.1.1 Sejarah Pondok Pesantren Mahasiswa Miftahul Khoir

Perkembangan pondok pesantren yang ilmiah dan modern di Indonesia, khususnya di Kota Bandung, adalah suatu hal yang menggembirakan, sekaligus tantangan untuk berperan serta aktif di dalamnya.

Mimkho atau Pondok Pesantren Mahasiswa Miftahul Khoir Bandung adalah pondok pesantren mahasiswa yang memiliki konsep pembekalan ruhiyah kepada para mahasiswa muslim sekaligus tempat pemondokan bagi mahasiswa muslim.

Pesantren Miftahul Khoir didirikan pada bulan Muharram 1402 H bertepatan dengan tahun 1985 M, didirikan oleh KH. Ahmad Umar, dibantu oleh KH. AF Ghozali (alm), KH. Ahmad Rifa'I, KH. Hasan Amirudin, sedangkan nama Miftahul Khoir diberikan oleh KH. Khoirul Affandi pimpinan Pondok Pesantren Miftahul Huda Manonjaya Tasikmalaya.

Pada awalnya pesantren ini dirancang sebagai sebuah pesantren salafy (tradisional), akan tetapi lambat laun kebanyakan santri yang hadir adalah santri

yang berstatus sebagai mahasiswa, maka lambat laun pula Miftahul Khoir berubah menjadi Pesantren Mahasiswa Miftahul Khoir.

Pesantren ini dipercayakan kepada kyai Endin (Alm.) oleh KH. Ahmad Umar sebagai pengasuh dan telah memiliki santri lebih dari 200 orang, baik yang mukim ataupun yang tidak mukim. Akan tetapi, Kyai Endin meninggal pada tahun 1994 M. Sehingga Pesantren Miftahul Khoir tidak memiliki pengasuh. Hal ini menyebabkan KH. Ahmad Umar mencari pengganti yaitu KH. Djalaluddin Asy-Syatibi yang pada waktu itu masih menjadi staf pengajar di Pondok Pesantren Sumur Bandung Cililin, untuk bersedia menjadi pengasuh Pondok Pesantren Miftahul Khoir.

Sejak tahun 1994 Pondok Pesantren Miftahul Khoir resmi diasuh oleh KH. Drs. Djalaluddin Asy-Syatibi, dan pada tahun itu pula Pondok Pesantren Miftahul Khoir resmi menjadi Pondok Pesantren Mahasiswa Miftahul Khoir karena sebagian besar santrinya adalah mahasiswa. Hal ini menyebabkan pola belajar di Pesantren Mahasiswa Miftahul Khoir disesuaikan dengan kondisi santrinya yang kebanyakan adalah mahasiswa. Yaitu setelah magrib sampai jam sembilan malam, dan ba'da subuh sampai jam setengah tujuh.

Pada tahun 2004 KH. Djalaludin Asy-Syatibi dipercaya untuk menduduki kursi dewan perwakilan rakyat (DPR) pusat, sehingga Pondok Pesantren Mahasiswa Miftahuk Khoir kembali tidak memiliki pengasuh. Karena pada saat itu pula kondisi KH. Ahmad Umar sedang sakit, kemudian sebagai perwakilan dari yayasan H. Asep Ahmad Djaelani putra dari KH. Ahmad Umar mencari pengganti KH. Djalaludin Asy-Syatibi.

Pada akhirnya H. Asep Ahmad Djaelani memilih Ust. Ajil Yumna Al-Qurthuby dari Pondok Pesantren Barkatul Huda sebagai pengasuh sehari-hari santri Pondok Pesantren Mahasiswa Miftahul Khoir dari tahun 2004 sampai sekarang, dalam melaksanakan tugasnya beliau dibantu oleh para Dewan Asatidz yang lainnya, diantaranya H. Harun Djuned, Ph.D, KH. Hafidzin, KH. Drs. Anwar Nurulyamin, KH. Souban, Ust. Bambang Triyono, S.Pd dan yang lainnya, sehingga tahun 2004 hingga sekarang proses belajar mengajar di Pondok Pesantren Mahasiswa Miftahul Khoir kembali berjalan.

2.1.2 Visi dan Misi

Salah satu cara peningkatan sumber daya muslim menjadi cendikiawan muslim yang menyeluruh adalah melalui *ta'lim* (pembelajaran), *tarbiyah* (pendidikan), dan *da'wah* (seruan/ ajakan). *Ta'lim* ditujukan untuk meningkatkan wawasan dan pemahaman, *tarbiyah* ditujukan untuk pembentukan *Insan Kamil* (sempurna) dan kondisi yang kondusif, sedangkan *da'wah* ditujukan untuk menguatkan *tauhid* (peng-*esa*-an Allah swt.), sehingga terwujud masyarakat yang di ridhoi Allah swt.

2.1.3 Motto

Membentuk intelektual yang ulama dan ulama yang intelek.

2.1.4 Sistem Pendidikan

Sistem pendidikan di Pondok Pesantren Mahasiswa Miftahul khoir, menyeimbangkan antara kebutuhan ilmu duniawi dan ukhrawi. Sehingga diharapkan akan terbentuk manusia yang sempurna *(insan kamil),* memiliki

pemahaman dan pengamalan yang berorientasi pada *fiddunya hasanah wa filakhirati hasanah* (bahagia di dunia dan di akhirat).

2.1.5 Siswa dan Daftar Pengajaran

Semua siswa/ santri yang mukim di pesantren ini adalah semuanya mahasiswa dari berbagai universitas negeri maupun swasta di kota bandung, seperti ITB, UNPAD, UPI, UNISBA, UNIKOM dan lain – lain.

Adapun pengajaran yang dikaji :

1. Bahasa Inggris
2. Bahasa Arab
3. Aqidah islamiyah
4. Fiqih dan Ushul Fiqih
5. Akhlak
6. Tahsinul Quran
7. Hadist dan Ilmu Hadist
8. Tafsir Al Quran
9. Tsaqofah Islamiyah (Sejarah Kebudayaan Islam)
10. Tasawuf (Kajian kitab Al Hikam)
11. Riyadhoh
12. Muhadhoroh (latihan menjadi da’i)
13. dll.

2.1.6 Jadwal Kegiatan Pengajaran

Kegiatan talim di pesantren ini disesuaikan dengan kondisi santri yang hampir semuanya adalah mahasiswa. Secara umum jadwal *ta’lim* terbagi menjadi

dua waktu yaitu setelah shalat maghrib sampai pukul 20.30 WIB dan setelah shalat Subuh sampai pukul 06.30 WIB. Namun khusus untuk sabtu malam dan ahad pagi kegiatan talim diliburkan. Biasanya digunakan sebagian santri untuk pulang ke rumah masing – masing.

2.1.7 Tempat dan Kedudukan Instansi

Pondok Pesantren Mahasiswa Miftahul Khoir ini berada dibawah Yayasan Miftahul Khoir yang bertempat di Jl Tubagus Ismail VIII No. 60 Kelurahan Dago, Kecamatan Coblong Kabupaten Bandung 40135 Telp 022-2502757.

2.1.8 Bentuk dan Badan Hukum Instansi

Pondok Pesantren Mahasiswa Miftahul Khoir merupakan sebuah instansi yang bergerak dalam bidang pendidikan agama Islam intensif yang dikhususkan untuk para mahasiswa yang ada di perguruan tinggi baik negeri maupun swasta yang ada di sekitar daerah Bandung. Di bawah koordinasi Departemen Agama Kota Bandung.

Berdasarkan Undang – Undang Nomor 2 tahun 1989, tentang sistem Pendidikan Nasional, Peraturan Pemerintah Nomor 73 tahun 1991, tentang Pendidikan Luar Sekolah, Peraturan Pemerintah Republik Indonesia Nomor 28 tahun 1990 tentang Pendidikan Dasar, Peraturan Pemerintah Republik Indonesia Nomor 39 tahun 1992 tentang Peran Serta Masyarakat Dalam Pendidikan Nasional Surat Keputusan bersama Mendikbud dan Menteri Agama RI Nomor : I/U/KB/2000 dan Nomor MA/86/2000, tentang Pondok Pesantren Salafiyah sebagai pola wajib belajar Pendidikan Dasar Sembilan Tahun, menetapkan Yayasan/Lembaga/Organisasi Miftahul Khoir diberikan hak untuk

menyelenggarakan Pendidikan dan Pengajaran sesuai dengan aturan yang telah ditetapkan Menteri Agama dengan Nomor : Mi-14/PP.008/1897/2001, serta Nomor Statistik : 51.2.32.73.23.067.

2.1.9 Bidang Pekerjaan Instansi

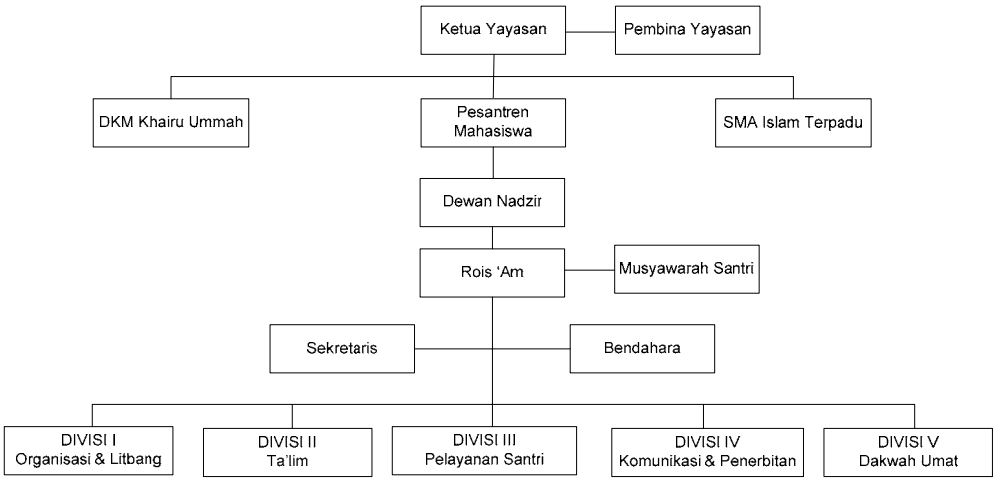
Pondok Pesantren Mahasiswa Miftahul Khoir merupakan sebuah instansi yang bergerak dalam bidang pendidikan agama Islam intensif yang dikhususkan untuk para mahasiswa yang ada di perguruan tinggi baik negeri maupun swasta yang ada di sekitar daerah Bandung. Di bawah koordinasi Departemen Agama Kota Bandung.

2.1.10 Struktur Organisasi Instansi

STRUKTUR ORGANISASI

PONDOK PESANTREN MAHASISWA MIFTAHUL KHOIR

BANDUNG



Gambar 2.1 Struktur Organisasi Pondok Pesantren Mahasiswa Miftahul Khoir

Pembina Yayasan	: Drs. H. Asep Djaelani, M.M
Ketua Yayasan	: H. Harun Djuned, Ph.D
Ketua Dewan Nadzir	: KH. Hafidzin
Pengasuh Pesantren	: Ust. Ajil Yumna Al Qurtuby
Rois ‘Am	: Dede M. Jaelani
Sekretaris	: Lukman Nur Hakim
Bendahara	: Rocky Dear Thana, A.Md.
Ketua Divisi I	: Guntur Gunara
Ketua Divisi II	: Muhammad Asyari, SE.
Ketua Divisi III	: Rifki Ferdiana
Ketua Divisi IV	: Muhajir Afandi
Ketua Divisi V	: Rachmad Muqtadir, ST.

WEWENANG DAN TANGGUNG JAWAB

Pembina Yayasan

1. Dewan musyawarah yayasan dalam menentukan kebijakan yayasan.
2. Bertindak sebagai penasihat ketua yayasan.
3. Mengawasi kinerja ketua yayasan secara keseluruhan
4. Memberikan arahan dan masukan kepada ketua yayasan.

Ketua Yayasan

1. Mengawasi seluruh aktifitas yayasan dan pesantren
2. Memberikan pengarahan kepada Dewan Nadzir

3. Melakukan kebijakan strategis mengenai pengembangan pesantren
4. Menerima laporan dari para Dewan Nadzir dan Rois ‘am mengenai keberlangsungan proses belajar mengajar.
5. Menerima laporan keuangan pesantren.

Dewan Nadzir

1. Mengawasi aktivitas ta’lim dan kegiatan keseharian santri
2. Bertanggung jawab menentukan kurikulum pesantren
3. Mengeluarkan santri (*drop out*) jika ada yang melanggar peraturan santri

Musyawarah Santri

1. Melakukan pergantian pengurusan santri dari yang lama ke yang baru.
2. Mengawasi kepengurusan santri
3. Mengingatkan dan memberi masukan kepada pengusus apabila terjadi sesuatu hal terjadi.

Rois ‘Am

1. Melaksanakan ketetapan yang telah digariskan oleh Musyawarah Santri.
2. Memimpin, mengarahkan, dan mengkoordinasikan semua fungsi yang ada dibawahnya untuk mencapai tujuan yang telah ditetapkan.
3. Membuat kebijakan-kebijakan dengan efisien dan efektivitas kerja.
4. Dalam kondisi yang tidak memungkinkan Rois dapat mengambil kebijakan tanpa mengadakan rapat dengan pengurus inti.
5. Mengingatkan seluruh pengurus terhadap ketentuan yang berlaku.
6. Memberikan pertanggungjawaban atas segala kebijakan dan kegiatan Pesantren di Musyawarah santri.

Sekretaris Umum

1. Mengingatkan Rois terhadap ketentuan yang berlaku maupun janji memenuhi undangan.
2. Menggantikan tugas Rois apabila Rois berhalangan.
3. Mengatur rapat dan membuat notulen rapat dengan persetujuan Rois.
4. Menyelenggarakan pencatatan dan pengumpulan data secara terperinci untuk bahan penyusunan laporan.
5. Membuat agenda surat keluar dan surat masuk.
6. Mengontrol dan mengarahkan seluruh administrasi kegiatan.
7. Memberikan pertanggungjawaban atas segala kebijakan administrasi pada Rois Dalam menjalankan tugasnya.
8. Membuat data invertaris barang PPM Miftahul Khoir dibantu divisi pelayanan santri.

Bendahara Umum

1. Menjalankan dan merealisasikan kebijakan-kebijakan yang berhubungan dengan keuangan.
2. Bertanggung jawab atas segala permasalahan keuangan organisasi.
3. Mencatat penerimaan dan pengeluaran uang dalam kegiatan pesantren.
4. Memberikan keterangan mengenai kondisi keuangan organisasi.
5. Penagihan SPP Santri.
6. Mengeluarkan dana kegiatan dengan persetujuan Rois.
7. Membuat kebijakan keuangan atas persetujuan Rois.
8. Mengingatkan Rois terhadap ketentuan yang berlaku.

9. Melaporkan keuangan bulanan kepada Rois.
10. Penagihan listrik SMAIT Miftahul Khoir
11. Pembayaran biaya bulanan
12. Menggantikan tugas Rois dan sekretaris umum apabila mereka berhalangan.

Ketua Divisi

1. Bertanggung jawab terhadap pelaksanaan kegiatan Divisi kepada Rois.
2. Menjabarkan, merencanakan, mengorganisasikan, melaksanakan, mengarahkan, memimpin dan mengontrol semua kegiatan divisi yang telah ditetapkan berdasarkan persetujuan Rois.
3. Mengingatkan Rois dan anggota divisinya terhadap ketentuan yang berlaku.
4. Menggantikan tugas Rois dan sekretaris umum serta bendahara umum, apabila ketiganya berhalangan.

Divisi I Organisasi dan Litbang

1. Bertanggung jawab terhadap pelaksanaan kegiatan kaderisasi santri
2. Mengadakan pelatihan-pelatihan pengembangan kualitas santri.
3. Melakukan studi banding ke berbagai lembaga yang dapat memberikan informasi untuk pengembangan pesantren.

Divisi II Ta’lim

1. Menyediakan fasilitas sarana dan prasarana bagi pelaksanaan ta’lim.
2. Bertanggung jawab penuh terhadap pelaksanaan ta’lim.

3. Merencanakan pengembangan metode ta’lim untuk peningkatan kualitas santri.
4. Bertanggung jawab mengkondisikan santri untuk mengikuti ta’lim dibantu oleh semua pengurus.

Divisi III Pelayanan Santri

1. Menyediakan, mengatur dan merawat fasilitas atau sarana-prasarana PPM Miftahul khoir.
2. Membantu Sekretaris membuat data invertaris barang PPM Miftahul Khoir
3. Menyelenggarakan kegiatan kerja bakti.
4. Membuat dan mengatur jadwal piket kebersihan dan perawatan kebun PPM Miftahul Khoir
5. Menyediakan dan merawat peralatan kebersihan
6. Mengadakan kegiatan untuk meningkatkan kesehatan mental dan kesehatan jasmani santri.
7. Membentuk dan mengelola seni islami.
8. Membentuk dan mengelola time sepak bola Mimkho.
9. Menyediakan perlengkapan dan peralatan olah raga.
10. Bertanggung jawab terhadap pengelolaan gudang pesantren beserta isinya.
11. Bertanggung jawab terhadap pemeliharaaan tempat sampah.
12. Bertanggung jawab terhadap pemeliharaaan tempat jemuran.
13. Bertanggung jawab terhadap pengelolaan koran.

Divisi IV Komunikasi dan Penerbitan

1. Membuat bulletin mingguan PPM Mimkho.

2. Penanggung jawab dan pengelola papan pengumuman pesantren.
3. Penanggung jawab dan pengelola Website PPM Mimkho.
4. Menjalin komunikasi intern dan ekstern.

Divisi V Da'wah Umat

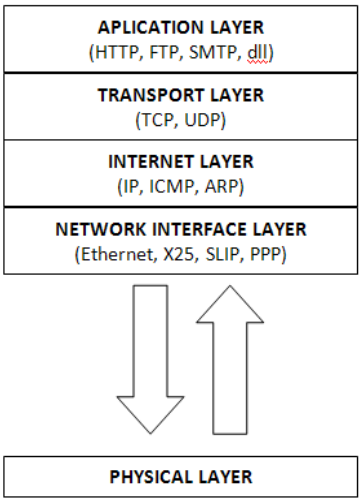
1. Menyelenggaraan kegiatan dakwah di pesantren dan sekitarnya, terlebih kepada masyarakat luas.
2. Menyelenggarakan kegiatan PHBI (Peringatan Hari Besar Islam).
3. Menyelenggaraan kegiatan Reuni Akbar Pesantren
4. Penanggung jawab kegiatan sosial.
5. Mennyelenggarakan kegiatan pengajian dan pendidikan nonformal anak-anak PPM Mimkho.
6. Penanggung jawab pengajian ibu-ibu dan pengajian umum.

2.2 Landasan Teori

2.2.1 Internet

Menurut COL[7], internet adalah himpunan informasi dan sumber daya komputer yang paling besar di dunia. Internet merupakan koneksi berbagai macam jaringan komputer dengan berbagai jenis *hardware* dan teknologi yang berbeda. Komunikasi yang dibangun pada jaringan internet menggunakan mekanisme tertentu yang disebut dengan *protocol*. *Protocol* yang digunakan untuk menjalankan komunikasi antar jaringan komputer pada internet ini adalah *Transmission Control Protocol/ Internet Protocol (TCP/IP).* Dengan menggunakan protokol yang sama yaitu TCP/IP, perbedaan jenis komputer dan

sistem operasi tidak menjadi masalah. TCP/IP pada awal perkembangannya digunakan oleh *Defense Advanced Research Project Agency* (DARPA) untuk membangun jaringan untuk keperluan militer. Nama TCP/IP sendiri diambil dari nama *protokol* pada layer *network* yaitu *Internet Protocol* (IP) dan *protokol* pada layer transfort yaitu *Transmission Control Protocol (TCP)*. Dengan mengacu pada nama ini, TCP/IP internet kemudian sering disebut sebagai Internet. Arsitektur protokol TCP/IP ini dapat dimodelkan dengan empat layer sebagaimana yang dapat dilihat pada gambar berikut.



Gambar 2.2. Arsitektur Protokol TCP/IP

TCP/IP terdiri dari empat lapis kumpulan protokol yang bertingkat. Empat tingkat itu adalah :

Application Layer, yaitu layer yang bertanggung jawab terhadap aplikasi-aplikasi yang menggunakan TCP/IP.

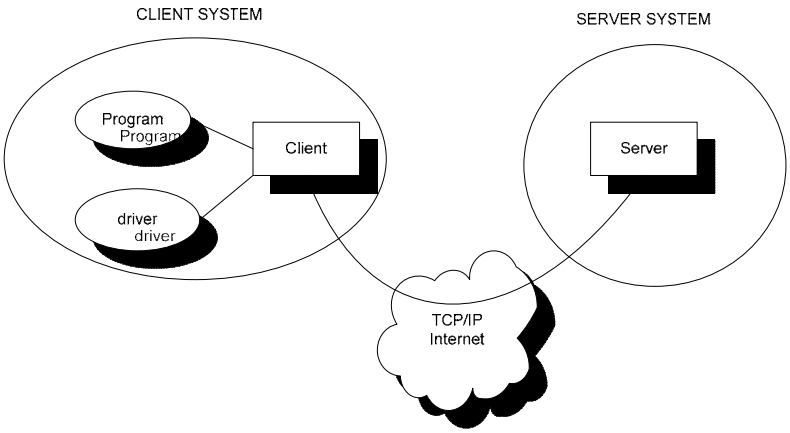
Transport Layer, yaitu layer yang bertanggung jawab memberikan layanan pengiriman data dari satu node ke node yang lain.

Internet Layer, yaitu layer yang menentukan jenis datagram dan menangani perutean data yang dikirim.

Network Interface Layer, yaitu layer yang bertanggung jawab untuk mengirim dan menerima data pada media fisik transmisi yang digunakan.

2.2.1.1 Model Client Server Internet

Proses yang terjadi pada internet mempunyai mekanisme yang telah diatur berdasarkan protokol standar. Proses yang terjadi di internet adalah ada satu program yang meminta kepada program yang lain untuk memberikan suatu layanan. Dua program tersebut dapat terletak dalam subnet yang sama ataupun berbeda. Komputer dengan program yang meminta layanan disebut dengan client sedangkan yang memberikan layanan disebut *server*. COL[7]



Gambar 2.3. Model Client Server Internet

2.2.2 Teknologi Website (WWW)

World Wide Web (WWW) merupakan suatu sistem penyedia informasi dengan skala yang besar yang mengelola informasi tersebut secara terdistribusi dalam internet dengan menggunakan teknologi hypermedia.

WWW mempunyai karakteristik :

a. Dukungan *interface* yang terintegrasi

   WWW menyediakan suatu *interface* yang terintegrasi terhadap protokol, format data, sistem pengalamatan, dan lain-lain sehingga berbagai macam layanan dan basis data yang ada di internet dapat diakses secara langsung.

b. Kemudahan pada sisi pengguna

   WWW mendukung secara transparan sebagian besar aplikasi-aplikasi yang terdapat di internet seperti layanan *telnet, gopher, anonymous ftp, finger*, dan lain – lain. Kombinasi layanan ini memberikan *user interface* yang terintegrasidengan aplikasi – aplikasi internet yang lain.

c. Kemudahan dalam perkembangan

   Kapabilitas server WWW dapat dikembangkan secara mudah dengan menggunakan standar PHP antara server WWW dengan alplikasi yang lain. Program PHP juga memungkinkan perubahan informasi secara dinamis yang dapat diperoleh secara *real time*.

d. Tidak tergantung pada *flatform* tertentu

   WWW memungkinkan seseorang dapat membangun server WWW di berbagai sistem yang berlainan dan memberikan informasi dalam bentuk hypermedia. WWW tidak didesain untuk mendukung sistem tertentu.

WWW menggunakan model client server. WWW juga menggunakan ekspresi informasi, transfer informasi, metode penamaan informasi yang standar sehingga dapat memproses dan mentransfer informasi secara terdistribusi yang dilakukan secara sistematik.

Informasi yang terdistribusi disimpan di dalam *server* WWW dan *user* mengakses informasi tersebut menggunakan suatu *software* yang disebut dengan browser. *Server* menyimpan informasi di dalam file-file terstruktur menggunakan *Hypertext Markup Language* (HTML). Hypertext merupakan kumpulan teks-teks yang saling berhubungan satu sama lainnya. *Hypermedia* serupa dengan *hypertext*, hanya saja media yang digunakan bukan hanya teks akan tetapi meliputi juga *audio*, *image*, dan *video*.

WWW menggunakan standar untuk mencapai konsistensi dalam menghasilkan dan melakukan transfer informasi. WWW menggunakan HTML sebagai standar produksi informasi dengan menggunakan *Hypertext Transfer Protocol* (HTTP) sebagai standar transfer informasi. WWW menggunakan standar penamaan yang disebut dengan *Universal Resource Locator* (URL) untuk mendukung penyediaan informasi secara *universal* dalam *server* yang terdistribusi.

2.2.2.1 Universal Resource Locator (URL)

URL digunakan untuk menentukan lokasi dari informasi terdistribusi pada suatu server WWW. User dapat mengakses informasi yang tersimpan di suatu

server dengan menentukan lokasi server sesuai dengan ekspresi URL. Ekspresi URL mempunyai bentuk sebagai berikut :

Protocol ://Alamat Internet [: nomor port]/[directory]/[nama file]

Pada bagian protokol pada URL tersebut adalah protokol transfer data yang dapat berupa *http, ftp, gopher, mailto, news, telnet,* dan lain-lain.

2.2.2.2 Hypertext Transfer Protocol (HTTP)

HTTP merupakan sebuah protokol yang didesain untuk men-*trasfer* informasi dalam bentuk hypermedia antara server dengan sebuah client. HTTP juga men-*transfer* data suatu informasi melalui *header*-nya. *Header* HTTP ini merupakan bentuk pengembangan dari *Multipurpose Internet Mail Extentions* (MIMEs). Pengembangan ini memungkinkan HTTP untuk ment-*ransfer* informasi dalam bentuk biner dan informasi dalam bentuk yang tidak standar yang berhasil dinegosiasi antara *server* dan *client*. Secara umum, delay akan terjadi pada saat melakukan suatu negosiasi sebelum proses transfer data. Karena lama dari *delay* yang disebabkan oleh *overhead* akan menjadi relative lebih besar pada transfer data yang sebetulnya singkat.

HTTP merupakan protokol yang sederhana yang didesain untuk mereduksi *delay*. HTTP merupakan *protokol* yang bersifat *stateless*, sehingga *server* akan memproses setiap *request* dari *user* secara terpisah dari *request* yang lain, *independen* terhadap request yang sebelumnya. HTTP menggunakan 8 bit untuk mentransfer semua tipe data yang mungkin.

Mekanisme yang terjadi pada HTTP bila suatu client menginginkan layanan dari server dibagi dalam 4 langkah, yaitu :

1. *Connection Setup*

   Client mengakses sebuah *server* dengan menggunakan *internet address* dan port number. *Default* dari *port number* adalah 80.

2. *Request*

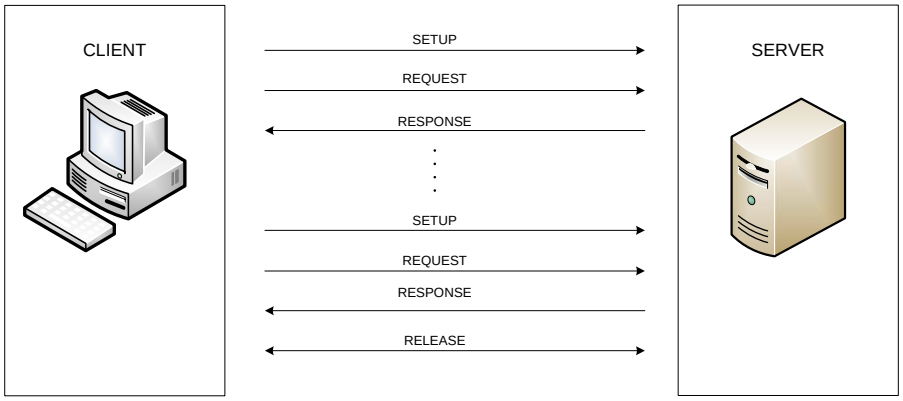
   Client mengirimkan *message* berupa informasi dari metode transaksi dan kapabilitas *client*.

3. *Response*

   Server mengirimkan response kepada *client* sesudah *client* itu menyelesaikan *reques*t-nya. *Response message* meliputi informasi dari transaksi dan data yang diminta.

4. *Connection Release*

   Client mengakhiri koneksi ke *server*.



Gambar2.4. Proses Koneksi Protokol HTTP antara client dan server

2.2.3 HTML

Untuk menyediakan informasi terdistribusi dalam suatu bentuk yang terintegrasi, diperlukan suatu standar untuk mengekspresikan informasi. HTML merupakan bahasa yang digunakan oleh WWW untuk konsistensi ekspresi suatu informasi. Sebuah dokumen hypermedia meliputi teks, image, suara, video, dan tipe informasi yang lain. HTML juga dapat menghubungkan informasi dalam dokumen yang berbeda. Bahasa HTML bersifat independen terhadap *flatform* sehingga dapat digunakan tanpa adanya batasan dari jenis *hardware* maupun *software*.

2.2.4 Cascadding Style Sheet (CSS)

Menurut WIK [11], *Cascadding Style Sheet (CSS) adalah suatu bahasa style sheet yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa markup*

. Penggunaan yang paling umum dari CSS adalah untuk memformat halaman web yang ditulis dengan HTML dan XHTML. Walaupun demikian, bahasanya sendiri dapat dipergunakan untuk semua jenis dokumen XML termasuk SVG dan XUL. Spesifikasi CSS diatur oleh *World Wide Web Consortium* (W3C).

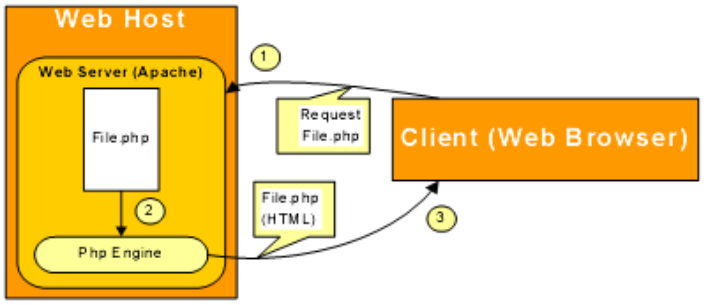
CSS digunakan pada halaman web untuk menentukan warna, jenis huruf, tata letak dan berbagai aspek tampilan dokumen. CSS digunakan terutama untuk memisahkan antara isi dokumen (yang ditulis dengan HTML atau bahasa markup lainnya) dengan presentasi dokumen (yang ditulis dengan CSS). Pemisahan ini dapat meningkatkan aksesibilitas isi, memberikan lebih banyak keleluasaan dan

control terhadap tampilan, dan mengurangi kompleksitas serta pengulangan pada struktur isi.

CSS memungkinkan halaman yang sama untuk menampilkan dengan cara yang berbeda untuk metode presentasi yang berbeda, seperti melalui layar, cetak, suara (sewaktu dibacakan oleh browser basis suara atau pembaca layar), dan juga alat pembaca Braille. Halaman HTML atau XML yang sama juga dapat ditampilkan scara berbeda, baik dari segi gaya tampilan atau skema warna dengan menggunakan CSS.

2.2.5 PHP

Menurut KAR[3], PHP masuk kedalam kategori *server-side scripting* dimana *browser* pada *client* tidak lagi bertanggung jawab dalam menjalankan kode-kode PHP, melainkan *web server* p*roses* ini diilustrasikan kedalam gambar berikut.



Gambar 2.5 PHP Server side scripting

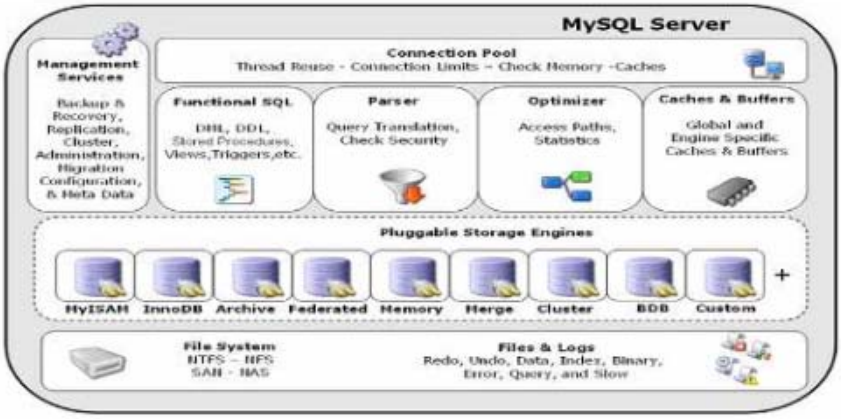
Pertama-tama web *browser* pada *client* me-*request* sebuah file (1). Dalam kasus ini bagaimanapun juga file yang di-*request* ber-*ekstensi*/berakhiran .php (contoh: File.php), tanda bahwa didalam file tersebut terkandung kode-kode PHP yang perlu diproses oleh *server*. Web *server* mengenali file ini dan tidak mengirim

file tersebut langsung ke *browser*, tetapi dikirim ke PHP *scripting engine* (2) (mesin pengolah kode-kode PHP). PHP engine merupakan komponen perangkat lunak dari server yang mampu mengartikan kode-kode PHP dan memberikan output dalam kode HTML. Setiap kode PHP dapat memberikan *output* kode HTML yang berbeda, tergantung pada jenis *request* dari *client* (*browser*). Proses tersebut membangkitkan halaman HTML secara dinamis lalu dikirimkan kembali ke *client* (*browser*) untuk merespon terhadap request yang sebelumnya telah dikirimkan.

Pemrograman disisi *server* biasanya digunakan untuk membuat sebuah website yang interaktif yang dihubungkan kedalam basis data atau data store lain.

2.2.6 MySQL

MySQL merupakan sebuah *database management system* dengan kemampuan multi-threaded, multi-user dan meningkatan fungsi *Structured Query Language* (SQL) terutama dengan PHP5 yang sudah memiliki fungsi built-in untuk melakukan koneksi dengan MySQL.



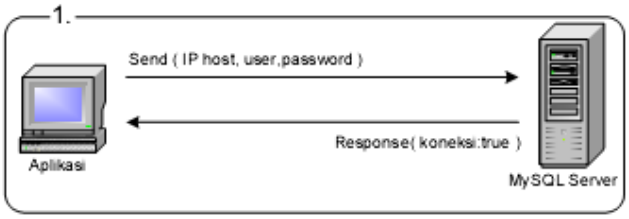
Gambar 2.6. MySQL Multiple Storage Engine Architecture

MySQL merupakan sebuah *relational database management system*, yang menyimpan data pada tabel yang terpisah daripada menyatukan semua data dalam satu tabel yang besar, hal ini dapat meningkatkan kecepatan dan fleksibilitas.

Tabel-tabel dihubungkan melalui relasi-relasi tertentu sehingga memungkinkan untuk mengkombinasikan data dari berbagai tabel sesuai dengan permintaan, melalui SQL yang telah menjadi bahasa standar yang umum untuk mengakses database.

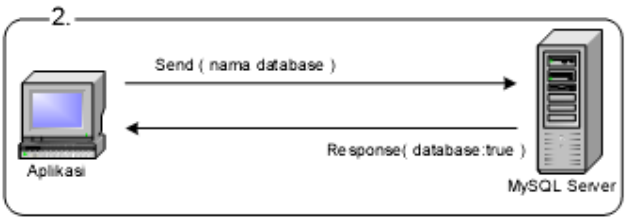
Untuk dapat mengendalikan MySQL server dari dalam program ada tiga hal yang perlu diperhatikan:

1. Buat koneksi dengan MySQL server dengan memasukkan alamat host dari MySQL, biasanya berupa alamat *Internet Protocol* (IP) lalu, user terdaftar beserta kata sandinya. Jika *host* ditemukan dan *user* tersebut terdaftar maka koneksi akan dimulai.



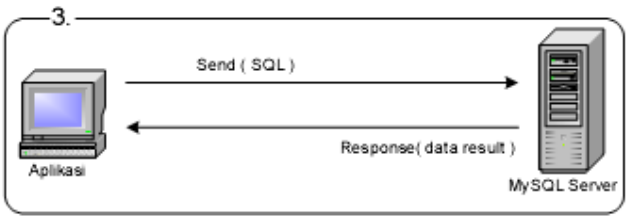
Gambar 2.7 Buat Koneksi

2. Pemilihan database yang akan digunakan. Pemilihan database dilakukan dengan menggunakan fungsi khusus dari PHP.



Gambar 2.8. Pilih Database

3. Pengiriman perintah kepada MySQL dengan menggunakan SQL.



Gambar 2.9. Pengiriman perintah MySQL

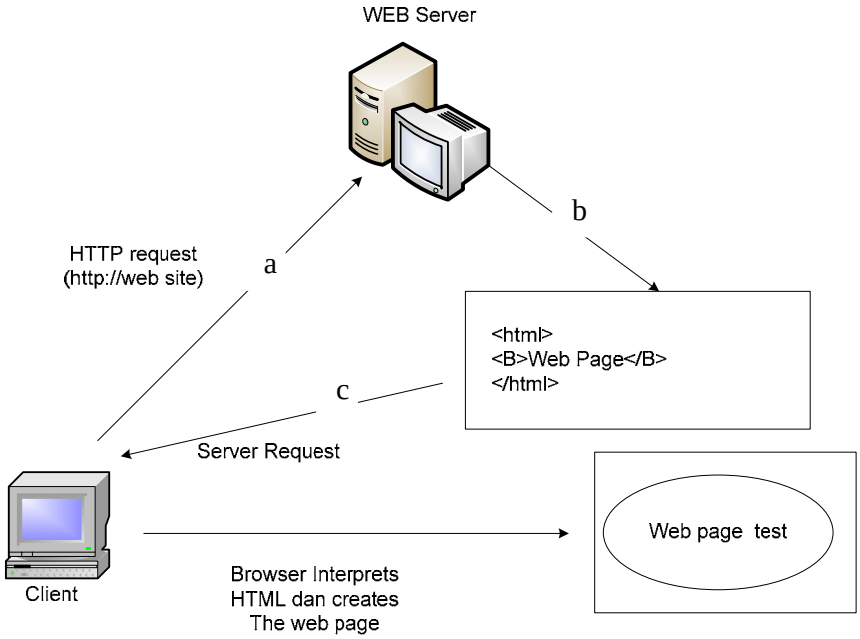
2.2.7 Aplikasi Program Berbasis Web

Banyak situs internet yang memiliki halaman dengan sifat statis seperti profil perusahaan, artikel, dan keterangan-keterangan lain. Situs ini mempunyai dokumen dengan teks yang sederhana, image dan hyperlinks ke dokumen yang dimilikinya. Untuk mengembangkan situs yang bersifat statis, kita menggunakan teknologi *client side*. HTML dan *Cascading Style Sheet* (CSS) dapat digunakan untuk mengatur struktur dan menampilkan halaman isi. Seandainya ingin diperindah dapat ditambahkan script yang sifatnya client side, seperti JavaScript, Jscript ataupun VBScript.

Dengan berkembangnya internet, situs yang ada di internet tidak hanya berfungsi untuk mempresentasikan content tetapi cenderung berupa aplikasi yang kebanyakan terhubung ke suatu basis data. Pada tahapan ini situs akan bersifat

dinamis, karena content yang dipresentasikan akan bervariasi dan berubah-ubah sesuai dengan data yang diminta dan action dari user. Untuk mengembangkan situs yang dinamis diperlukan teknologi server side seperti PHP, ASP, Perl dan CGI yang lain. Dengan teknologi *server side* kita dapat mengembangkan suatu aplikasi berbasis internet yang dapat mengahsilkan dan menampilkan content secara dinamis. SAR[9]

Cara kerja situs dengan content yang statis dapat dijelaskan dengan gambar berikut ini :



Gambar 2.10. Web yang Statis

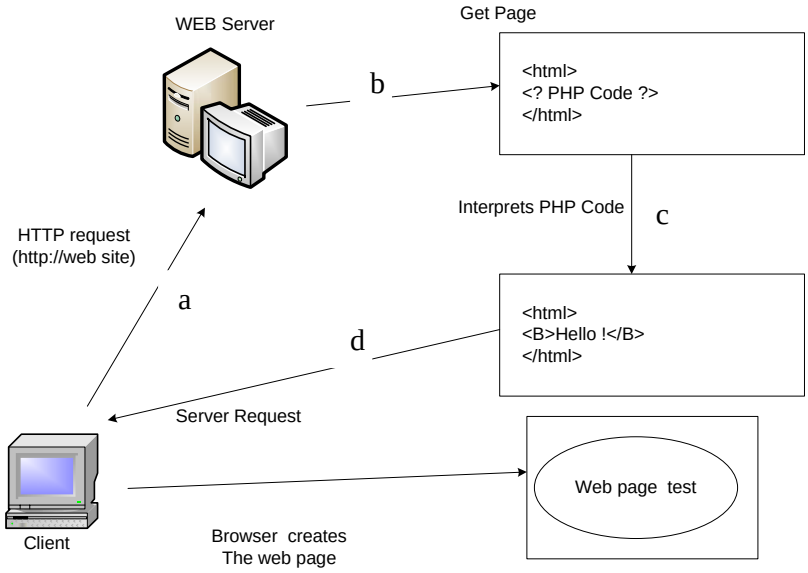
Pada saat ada suatu *request* dari *browser*, *server* web akan melakukan langkah-langkah :

a. Membaca *request* yang dikirim oleh *browser*

b. Mencari dan menemukan halaman di *server*

c. Mengirim halaman yang diminta melalui internet ke *browser*.

Cara Kerja situs dengan *content* yang dinamis dapat ditunjukkan dengan gambar berikut ini :



Gambar 2.11. Web yang dinamis

Pada saat ada suatu *request* dari *browser*, *server* web akan melakukan langkah – langkah :

a. Membaca *request* yang dikirim dari *browser*.

b. Mencari dan menemukan halaman di *server*.

c. Menterjemahkan perintah yang diberikan oleh bahasa program *server-side* menjadi halaman HTML.

d. Mengirim halaman yang diminta melalui internet ke *browser*.

Perbedaan utama antara HTML dengan bahasa *server side* adalah HTML diterjemahkan oleh *client browser,* tidak dieksekusi di *server*.

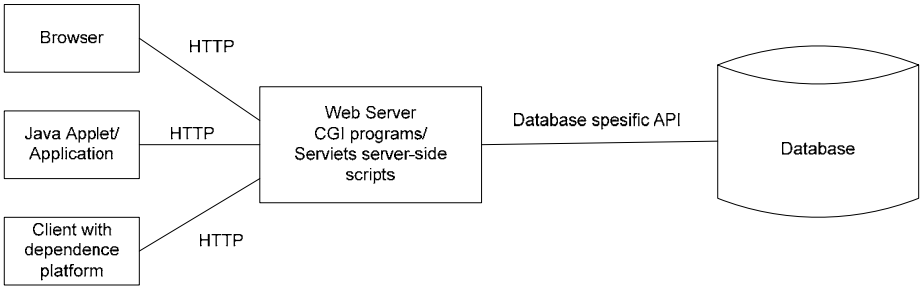
Dengan membuat kode yang dapat di eksekusi pada *server*, kita dapat menciptakan banyak sekali aplikasi yang bersifat dinamis dan dapat dikendalikan oleh *user* melalui *browser*.

Beberapa kelebihan teknologi *server side* dibandingkan dengan HTML :

a. Memberikan kemudahan untuk mengedit suatu *content* suatu halaman web, pengeditan dapat dilakukan dengan meng-*update content* dalam suatu basis data dan tidak lagi pada kode HTML nya.
b. Dapat membuat halaman yang dapat di kostumisasi penampilannya sesuai dengan keinginan *user*.
c. Dapat menampilkan dan melakukan perubahan data pada basis data yang dapat dilakukan melalui halaman web itu sendiri.
d. Memperoleh *feedback* dari *user* yang mengembalikan informasi berdasarkan isian yang disediakan untuk *user*.

2.2.8 Aplikasi Berbasis Web Menggunakan Server Basis Data

Web yang dinamis harus didukung oleh ketersediaan content. Content ini harus mudah untuk dimodifikasi dan diakses. Untuk memudahkan proses tersebut, content dari suatu web yang dinamis disimpan dalam suatu basis data.



Gambar 2.12. Arsitektur aplikasi berbasis web dengan menggunakan server basis data

Arsitektur dari aplikasi yang berbasis web dengan menggunakan basis data mempunyai layer :

a. Client

Client dapat berupa user yang mengakses aplikasi dari server web menggunakan web *browser*, java applet, aplikasi java, ataupun *platform* lain yang tergantung pada *client program*.

b. Logika Aplikasi

Logika aplikas merupakan algoritma yang dapat digunakan untuk membangun sebuah aplikasi dan diimplementasikan dalam script program CGI, script program *server side*, modul – modul khusus dalam server web, atau aplikasi yang tergantung pada server web.

c. Konektifitas dengan server basis data

Konektifitas dengan server basis data dapat menggunakan *application Programming Interface* (API) suatu basis data, atau menggunakan *protocol* umum yang mendukung konektifitas dengan basis data seperti ODBC dan JDBC.

d. Server basis data

Server basis data yang dapat mendukung aplikasi berbasis web antara lain MySQL, ORACLE, Microsoft SQL Server, PostgreSQL dan lain – lain.

2.2.9 Konsep Objek Oriented

Konsep berorientasi objek memandang segala sesuatu sebagai objek. Konsep ini bukan sekedar algoritma yang diterapkan pada bahasa pemrograman, tetapi menyentuh level *mikro* yaitu perancangan sampai pada level tertinggi dalam pemrograman. Sehingga dalam konsep berorientasi objek harus dipertimbangkan penggunaan analisis berorientasi objek, perancangan berorientasi objek, analisis domain berorientasi objek, system basis data berorientasi objek dan perangkat Bantu rekayasa perangkat lunak berorientasi objek. HAR[2]

Konsep ini secara mendasar berbeda dengan konsep konvensional dalam artian berorientasi fungsional dan struktur data yang oleh Niklaus Wirth dinyatakan dengan persamaan sebagai berikut: HAR[2]

Algorithms + DataStructures = Programs

Persamaan diatas menyatakan bahwa program perangkat lunak adalah sekumpulan mekanisme yang melakukan aksi-aksi tertentu pada data tertentu. Artinya pada pemrograman konvensional, terdapat dua fokus dalam memandang perangkat lunak, yaitu berfokus pada fungsi atau berfokus pada data. Sedangkan dalam konsep berorientasi objek, menggunakan konsep penggabungan fokus pada fungsi dan data secara bersamaan dalam bentuk objek.

Dalam pemrograman berorientasi objek, perangkat lunak secara keseluruhan adalah sebuah sistem yang dibuat dari berbagai komponen atau objek,

masing masing dengan fungsi khusus dan berinteraksi satu dengan lainnya dengan cara yang telah ditetapkan. HAR[2]

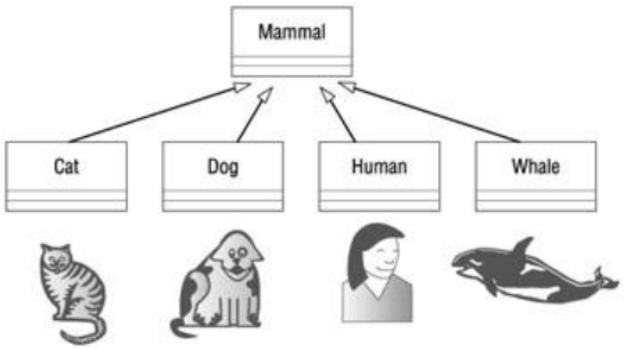
Pada metoda ini penekanannya ada pada obyek dan bukan pada proses, pendekatan berorientasi obyek ini memungkinkan seorang analisis dan perancang istem berkonsentrasi pada “apa” dan bukan pada “bagaimana”. Paradigma berorientasi obyek memandang aplikasi dengan cara yang berbeda.

Dengan pendekatan ini, kita membagi aplikasi menjadi banyak potongan-potongan kecil, atau obyek, yang cukup bebas antara satu dengan lainnya. Kita kemudian dapat membangun aplikasi dengan menggabungkan/merekatkan semua obyek ini bersama.

Dalam konsep konsep *Object Oriented* dikenal *Polimorfisme*, *Inheritance* (Pewarisan), dan *Encapsulation* (Pembungkusan). PRA[8]

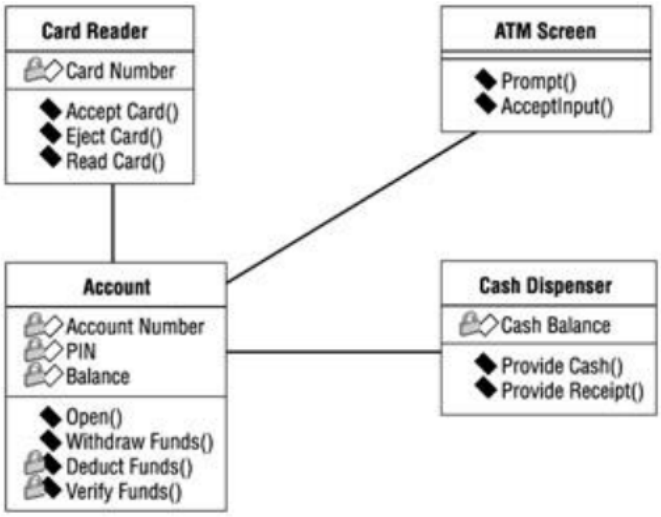
Polimorfisme, yaitu konsep yang menyatakan bahwa suatu fungsi yang sama dapat diterapkan dan dapat dimiliki oleh kelas-kelas yang berlainan.

Inheritance, pewarisan ini pada dasarnya adalah berbagi atribut dan operasi antarkelas berdasarkan hierarki kelas. Pewarisan akan sangat mengurangi perulangan penulisan kode dan mungkin merupakan salah satu keunggulan utama rekayasa perangkat lunak, karena memungkinan penulisan kode yang lebih sedikit tanpa mengorbankan esensi perangkat lunak secara keseluruhan.



Gambar 2.13. Inheritance : Natural Model

Encapsulation, Pembungkusan berarti meninggalkan aspek eksternal dari objek yang dapat diakses oleh objek lain dan memfokuskan diri pada implementasi internal suatu objek. Keuntungan pembungkusan adalah memungkinkan seseorang menggunakan (memanfaatkan fungsi-fungsi serta layanan) suatu komponen yang diciptakan dengan konsep pembungkusan tanpa mengetahui rincian implementasinya.



Gambar 2.14. Encapsulation: Banking Model

2.2.10 Unified Modelling Language (UML)

Unified Modeling Language (UML) merupakan sistem arsitektur yang bekerja dalam *Object Oriented Analysis and Design* (OOAD) dengan satu bahasa yang konsisten untuk menentukan visualisasi, mengkontruksi, dan mendokumentasikan *artifact* yang terdapat dalam sistem *software*.

2.2.10.1 Sejarah UML

Sejarah UML sendiri cukup panjang. Sampai era tahun 1990 seperti telah diketahui puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Masa itu terkenal dengan masa perang metodologi (*method war*) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi-notasi sendiri-sendiri yang mengakibatkan timbul masalah baru apabila bekerjasama dengan group atau perusahaan lain yang menggunakan metodologi yang berlainan.

Metodologi adalah cara sistematis untuk mengerjakan pekerjaan analisis dan desain. dengan metodologi pihak yang membangun sistem *software* dapat merencanakan dan mengulanmgi pekerjaannya di lain waktu.

Metodologi juga menghilangkan kesalahpahaman dan menghilangkan perbedaan notasi untuk suatu hal yang sama karena setiap orang akan berbicara dalam bahasa yang sama.

2.2.10.2 Semantik Dalam UML

Object Management Group (OMG) sebuah organisasi internasional yang mempromosikan teori dan praktek-praktek *object oriented technology* telah menetapkan semantik (makna istilah) semua notasi UML dan model struktural

dan model behavioral. Model *structural* (disebut juga sebagai model statis) menekankan struktur objek dalam sebuah sistem, menyangkut kelas-kelas, *interface*, *attribute*, dan hubungan antar komponen. Model behavioral (juga disebut sebagai model dinamis) menekankan perilaku objek dalam sebuah sistem, termasuk metode, interaksi, kolaborasi, dan *state history*. UML memiliki notasi untuk menjelaskan secara visual mengenai elemen-elemen pemodelan.

2.2.10.3 Tujuan UML

Tujuan UML diantaranya untuk :

1. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
2. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
3. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan

2.2.10.4 Cakupan UML

Cakupan UML diantaranya :

1. UML merupakan bahasa pemodelan tunggal yang umum dan digunakan secara luas oleh para *user* dan bahkan para *user* metode lainnya.
2. UML menekankan pada apa yang dapat dikerjakan dengan metode-metode tersebut.
3. UML berfokus pada suatu bahasa pemodelan standar, bukan pada proses standar.

Dengan menggunakan UML programmer dapat membuat model untuk semua jenis aplikasi perangkat lunak, dimana aplikasi tersebut dapat berjalan pada perangkat keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun.

Karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan perangkat lunak dalam bahas-bahasa berorientasi objek seperti PHP (diatas versi 4), SmallTalk, C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

2.2.10.5 Notasi dalam UML (Unified Modeling Language)

UML menyediakan beberikan notasi dan artifak standar yang bisa yang bisa digunakan sebagai alat para pelaku dalam proses analisis dan desain. Artifak di dalam UML didefinisikan sebagai informasi dalam berbagai bentuk yang digunakan atau dihasilkan dalam proses pengembangan perangkat lunak/

a. Actor

Actor adalah segala sesuatu yang berinteraksi dengan sistem aplikasi komputer. Jadi actor ini bisa berupa orang, perangkat keras, atau mungkin objek lain dalam sistem yang sama biasanya yang dilakukan actor adalah memberikan informasi pada sistem dan atau memerintahkan sistem untuk melakukan sesuatu

b. Class

Class merupakan pembentuk utama dari sistam berorientasi objek katrena class menunjukkan kumpulan objek yang memiliki atribut dan operasi yang

sama. *Class* digunakan untuk mengabstaksikan elemen – elemen dari sistem yang sedang dibangun.

c. *Interface*

Interface merupakan kumpulan operasi tanpa implementasi dari suatu *class*.

d. *Use Case*

Use case menjelaskan urutan kegiatan yang dilakukan actor dan sistem untuk mencapai suatu tujuan tertentu walaupun menjelaskan kegiatan. Namun *use case* hanya menjelaskan apa yang dilakukan oleh actor dan sistem, bukan bagaimana actor dan sistem melakukan kegiatan tersebut.

e. *Interaction*

Interaction digunakan untuk menunjukkan baik aliran pesan atau informasi antar objek maupun hubungan antar objek.

f. *Package*

Package adalah kontainer atau wadah konspetual yang digunakan untuk mengelompokkan elemen – elemen dari sistem yang sedang di bangun, sehingga bisa dibuat model yang lebih sederhana. Tujuannya adalah untuk mempermudah penglihatan (visibility) dari model yang sedang dibangun.

g. *Note*

Note digunakan untuk memberikan keterangan dan komentar tambahan dari satu elemen sehingga bisa langsung terlampir dalam model. *Note* ini bisa ditampilkan ke semua elemen notasi yang lain.

h. Dependency

Depedency merupakan relasi yang menunjukan bahwa perubahan pada salah satu elemen memberi pengaruh pada elemen lain.

i. Boundary Class

Boundary class adalah class yang mengkoordinasi aktivitas dalam sistem. *Class* ini menghubungkan boundary class dengan entity.

j. Control Class

Control class adalam class yang mengkoordinasi aktivitas dalam sistem *class* ini menghubungkan *boundary class* dengan *entity class*.

k. Entity Class

Entity class adalah *class* yang menghubungkan dengan data atau informasi yang digunakan oleh sistem. *Entity class* ini adalah *class* yang menyimpan dan mengelola data dapat dikombinasikan.

2.2.10.6 Diagram Grafis UML

Unified Modeling Language (UML) adalah sebuah bahasa untuk menentukan, visualisasi, kontruksi, dan mendokumentasikan *artifacts* dari sistem *software*, untuk memodelkan bisnis, dan sistem *nonsoftware* lainnya. *Artifact* adalah sepotong informasi yang digunakan atau dihasilkan dalam suatu proses rekayasa *software*. *Artifact* dapat berupa model, deskripsi atau *software*.

Untuk membuat suatu model, UML memiliki diagram grafis sebagai berikut :

1. *Use case diagram.*
2. *Class diagram.*

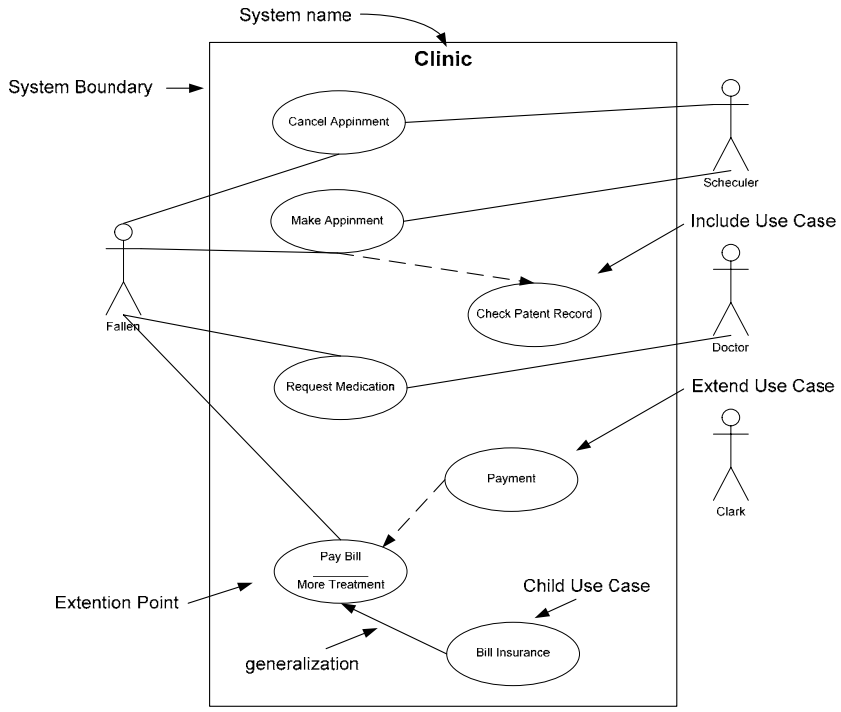
3. *Behavior diagram.*
    a. *Statechart diagram.*
    b. *Activity diagram.*
    c. *Interaction diagram* : *sequence diagram, collaboration diagram.*
4. *Implementation diagram*
    a. *Component diagram*.
    b. *Deployment diagram*.

Diagram-diagram tersebut diberi nama berdasarkan sudut pandang yang berbeda-beda terhadap sistem dalam proses analisis atau rekayasa. Orang mungkin mengetahui beberapa diagram lain kelak ketika membuat suatu model, misal *data-flow diagram*. Diagram ini tidak terdapat dalam UML. *Data flow diagram* dan tipe diagram lain yang tidak terdapat dalam UML tidak termasuk dalam paradigma *object-oriented*.

Activity diagram dan *collaboration diagram* yang terdapat dalam UML menggantikan *data-flow diagram*. *Activity diagram* juga sangat bermanfaat untuk membuat *workflow*.Berikut ini penjelasan dari konsep dasar UML, diantaranya :

1. Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* mempresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, seorang atau sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.



Gambar 2.15. contoh use case diagram

2. Class Diagram

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

Class memiliki 3 area pokok :

l. Nama dan *Stereotype*

m. Atribut

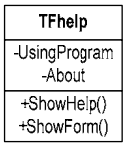
n. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

1. *Private,* tidak dapat dipanggil dari luar class yang bersangkutan.

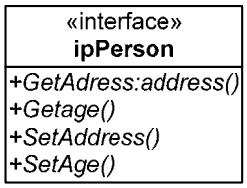
2. *Protected*, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya.

3. *Public*, dapat dipanggil oleh siapa saja.



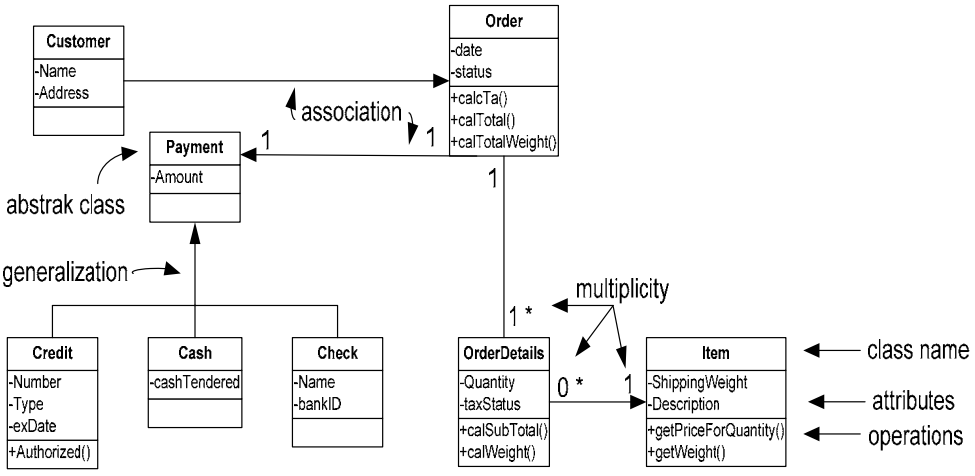
Gambar 2.16. contoh atribut

Class dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansikan, tetapi diimplementasikan dahulu menjadi sebuah *class*.



Gambar 2.17. contoh interface

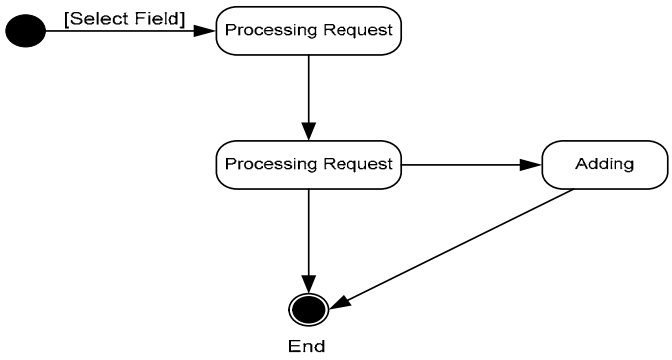
Dibawah ini terdapat hubungan antar kelas di dalam lingkungan *class diagram* :



Gambar 2.18. contoh class diagram

3. Statechart Diagram

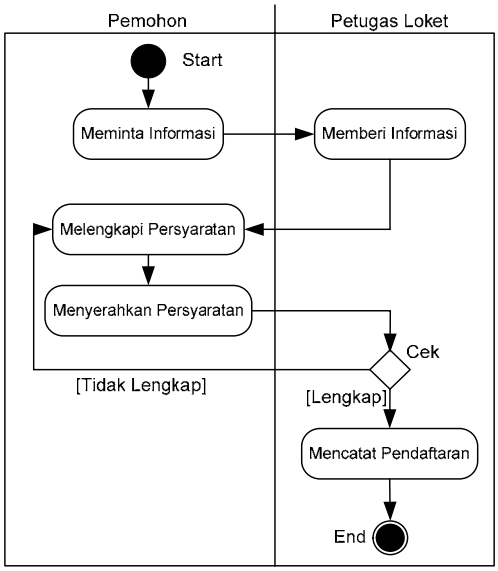
Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari stimuli yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).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



Gambar 2.19. contoh statechart diagram

4. Activity Diagram

Activity diagram menggambarkan berbagai alir aktifitas 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.



Gambar 2.20. contoh activity diagram

5. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal.

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan output tertentu.

6. Collaboration Diagram

Pada *collaboration diagram* terdapat beberapa objek, link dan message. Diagram ini digunakn sebagai alat untuk menggambarkan interaksi yang mengungkapkan keputusan mengenai perilaku sistem. *Collaboration diagram* menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran atau hubungan masing-masing objek dan bukan pada

waktu penyampaian *message*. Setiap *message* memiliki *sequence number*, di mana *message* dari level tertinggi memiliki nomor 1. *Message* dari level yang sama memiliki prefiks yang sama.

7. Component Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen perangkat lunak, termasuk ketergantungan (*depedency*) di antaranya. Komponen perangkat lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*.

Umumnya konponen terbentuk dari beberapa class atau package, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa interface, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

8. Deployment Diagram

Setiap model hanya memiliki satu diagram deployment dan memperlihatkan pemetaan *software* kepada *hardware*. Deployment physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastuktur sistem, dimana komponen akan terletak, hubungan antar node (misalnya TCP/IP) dan requirement dapat juga didefinisikan dalam diagram ini.