BAB 2
TINJAUAN PUSTAKA

2.1 Profil LKP TARQI Bandung

Profil lembaga kursus dan pelatihan membahas dimulai dari lembaga ini berdiri hingga berkembang sampai saat ini. Profil lembaga ini ditinjau dari sejarah, visi dan misi, struktur organisasi, deskripsi tugas, tempat dan kedudukan serta logo.

2.1.1 Sejarah LKP TARQI Bandung

LKP TARQI adalah sebuah lembaga kursus dan pelatihan yang terdaftar di Dinas Pendidikan Kota Bandung sejak tahun 2011. Lembaga ini bergerak dibidang pembinaan Al Quran dan Studi Islam dengan program dan instruktur yang telah teruji dan berpengalaman. Sejak berdirinya, LKP TARQI Bandung dibawah naungan Yayasan TARQI (Tarbiyat Al Quran Al Islami). Yayasan TARQI merupakan yayasan yang menjadikan Al Quran sebagai asas pada setiap bidang kegiatannya. Yayasan ini memiliki legalitas pembentukan yang tercantum sebagai berikut.

1. Akta Notaris: In-In Inayat Amintapura, SH., no.: 07 tanggal 23 April 2010
2. SK. Menkumham, no.: AHU.3672.AH.01.04/2010
3. Surat Keterangan Terdaftar Dinas Pendidikan Kota Bandung, no.: 421.10/282-PNFI/2011
4. SK. Kadinsos Prov. Jawa Barat, no.: 062/3252/PPSKS/2011

2.1.2 Visi dan Misi TARQI Bandung

2.1.2.1 Visi

Menjadi wadah pembinaan Al-Qur’an dan studi Islam di Jawa Barat.

2.1.2.2 Misi

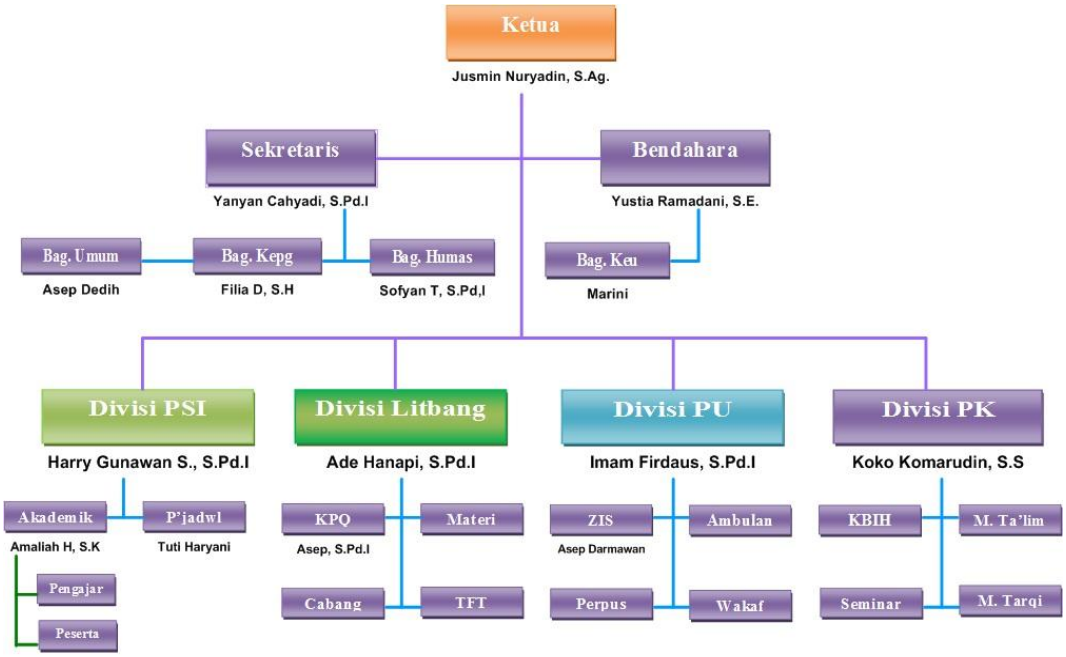
Berikut adalah misi TARQI Bandung.

1. Menyebarkan pemahaman cara membaca Al Quran dengan benar.
2. Menjembatani masyarakat untuk lebih dekat dengan Al Quran.
3. Meningkatkan kualitas Sumber Daya Manusia (SDM) melalui pembinaan dan pendidikan Al Quran yang berkesinambungan.

4. Memberikan pelayanan sosial kepada masyarakat.
5. Menjadi mitra bagi pemerintah dalam mencerdaskan kehidupan beragama, bermasyarakat dan bernegara yang berlandaskan nilai-nilai Al Quran.

2.1.3 Struktur Organisasi TARQI

Berikut adalah gambar struktur organisasi TARQI.



Gambar 2.1 Struktur Organisasi Yayasan TARQI 2013

2.1.4 Deskripsi Tugas Pengurus TARQI

Deskripsi tugas adalah rincian yang menunjukkan posisi, tanggung jawab, wewenang, fungsi dan tugas-tugas yang harus dikerjakan oleh seorang personil didalam suatu organisasi. Adapun uraian tugas bagian-bagian di LKP TARQI Bandung adalah sebagai berikut.

Ketua

1. Bertanggung jawab terhadap seluruh kegiatan yang dilaksanakan di lembaga.
2. Membuat rancangan kerja anggaran dan pendapatan (RKAP) untuk disahkan oleh Dewan Pembina.
3. Menetapkan kebijakan yang bersifat strategis untuk perkembangan lembaga.
4. Mengarahkan fungsi setiap divisi dalam menjalankan strategi lembaga.

5. Memonitor pelaksanaan kebijakan dan strategi lembaga serta memastikan kelancaran pelaksanaannya agar dapat berjalan secara maksimal dan tepat.
6. Mengevaluasi implementasi strategi agar memperoleh masukan strategis sebagai usulan untuk kebijakan tahun berikutnya.
7. Memeriksa dan meneliti laporan-laporan yang masuk dari para ketua divisi.
8. Membina hubungan baik dengan pimpinan perusahaan atau lembaga lain, terutama yang berhubungan erat dengan kegiatan operasional lembaga.
9. Membuat laporan tahunan kepada Dewan Pembina sesuai ketentuan yang berlaku.

Bagian Humas

1. Pusat informasi ke-TARQI-an.
2. Fasilitator dan negoisiator dari permintaan diluar program yang ada.
3. Konseptor marketing pusat dan cabang guna mencapai jumlah target perserta bulanan.
4. Konseptor rencana dan strategi marketing.
5. Rekap dan *follow up* permintaan presentasi.
6. Pengadaan MOU privat lembaga dan khusus.
7. Negoisasi dan pelaksana proses penandatangan MOU privat lembaga dan khusus.
8. Rekap kelas-kelas privat yang dibuka oleh karyawan (*fee marketing tahunan*).

Tugas Harian *Front Office*

1. Menerima tamu.
2. Memasukkan nama & nomor telepon pendaftar baru LKP.
3. Memasukkan biodata pendaftar ZISWAK baru.
4. Menerima pembayaran.
5. Memasukkan rekap pembayaran peserta.
6. Memasukkan pembayaran peserta per nama.
7. Menerima telepon, meliputi informasi, program LKP TARQI, ZISWAK, KBIH, jadwal belajar, *event* atau agenda TARQI, hasil ujian, komplain dan saran yang direkap.

Tugas Pekanan *Front Office*

1. Setor penerimaan harian.
2. Cek stok : buku, panduan, kuitansi dengan nomor seri, formulir, kuitansi (LKP & ZISWAK), tanda terima setoran harian, brosur.
3. Cek update pembayaran privat (khusus & luar).

Bulanan

1. Buka kelas baru (pendaftar baru & naik level).
2. Meminta form hasil ujian ke akademik.
3. Data peserta yang di SMS untuk kelas baru & *takhosus*.
4. Update penggunaan ruangan.
5. Update jadwal ujian susulan.
6. Rekap buku meliputi; fasilitas (regular & privat), fasilitas pengajar dan pembelian buku.
7. Cek kesesuaian permintaan buka kelas baru dengan jadwal dari akademik.
8. Laporan kerja *front office* meliputi jumlah pendaftar baru, jumlah pendaftar ulang/naik level, jumlah kelas regular & privat (khusus & luar), jumlah pendaftar ZISWAK, jumlah penerimaan LKP & ZISWAK, rekap buku, stok formulir, kuitansi (LKP & ZISWAK), brosur, tanda terima setoran harian.

Divisi Pembinaan Al-Quran dan Studi Islam (PSI)

1. Menjalankan kegiatan belajar mengajar (KBM) secara sistematis.

Akademik

1. Menetapkan waktu belajar dari permintaan kelas yang diajukan oleh *front office* ke akademik, baik privat maupun regular.
2. Menetapkan pengajar untuk kelas yang akan dimulai.
3. Memastikan laporan kepada *front office* berupa data kelas yang akan dimulai.
4. Memastikan kelas baru berjalan sesuai waktu dan pengajar yang telah ditetapkan
5. Memastikan ketersediaan kelas belajar

6. Melakukan koordinasi dengan bagian umum untuk memastikan ketersediaan semua perangkat kegiatan belajar mengajar, papan tulis, penghapus, spidol, absensi.
7. Melakukan kontrol terhadap kelas yang sedang berjalan disetiap hari kerja.
8. Menetapkan pengajar pengganti bagi pengajar asal yang berhalangan.
9. Menetapkan waktu berakhirnya kegiatan belajar mengajar.
10. Melakukan kontrol terhadap kelas berjalan regular dan privat disetiap pekan.
11. Menetapkan SOP (*Standart Operational Procedure*) kegiatan belajar mengajar.

2.1.5 Tempat dan Kedudukan TARQI

TARQI memiliki kantor pusat yang bertempat di Graha TARQI Center yang beralamat di jalan Pahlawan no. 64, Bandung-Jawa Barat, telepon 022-7210044, email tarqicenter@gmail.com dan memiliki tiga cabang diantaranya:

1. Cabang Jatinangor, beralamat di jalan Kolonel Ahmad Syam no. 44, Jatinangor, Sumedang, telepon 08586268343
2. Cabang Panghegar, beralamat TB. Sumber Anugrah, jalan Panghegar no. 19, Bandung, telepon 022-87882227
3. Cabang Panyileukan, beralamat Komplek Citra Panyileukan, blok AB 1, no. 9, Bandung, telepon 022-77895564

2.1.6 Logo TARQI

Berikut ini adalah logo TARQI.



Gambar 2.2 Logo TARQI

2.2 Landasan Teori

Landasan teori merupakan kumpulan teori-teori yang menjadi dasar pembangunan aplikasi ini yang diambil dari berbagai sumber seperti buku dan internet.

2.2.1 Sistem Informasi

Berikut adalah penjelasaan yang berkaitan dengan sistem informasi.

2.2.1.1 Konsep Dasar Data dan Informasi

Data adalah deskripsi dari sesuatu dan kejadian yang dihadapi. Definisi data yang lain adalah data merupakan kenyataan yang menggambarkan suatu kejadian dan kesatuan nyata. Data dapat berupa teks, citra, audio dan video.

Gordon B. Davis (1985) mendefinisikan informasi sebagai data yang telah diolah menjadi bentuk yang lebih berarti dan berguna bagi penerimanya untuk mengambil keputusan masa kini maupun yang akan datang.

2.2.1.2 Konsep Dasar Sistem

Sistem adalah seperangkat komponen yang saling terkait dengan batas yang jelas bekerja sama untuk mencapai tujuan yang sama dengan menerima masukan dan menghasilkan keluaran melalui proses transformasi yang teratur.

Sistem memiliki tiga elemen dasar antara lain:

1. Masukan

   Masukan berkaitan dengan menangkap dan merakit elemen yang masuk kedalam sistem dan akan diproses.

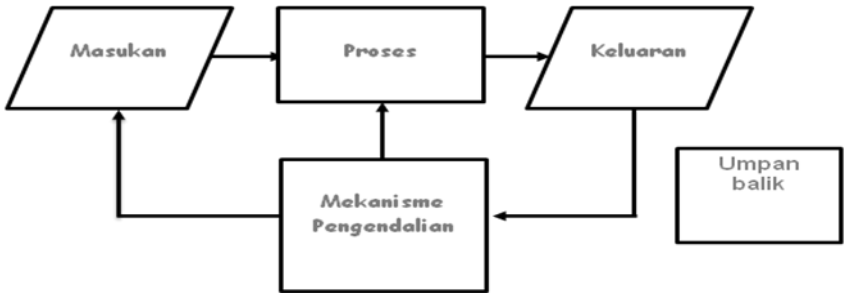
2. Proses

   Melibatkan proses transformasi yang mengubah masukan menjadi keluaran.

3. Keluaran

   Keluaran merupakan hasil dari pemrosesan untuk mencapai tujuan akhir.

Konsep sistem menjadi lebih berguna dengan menambahkan dua elemen tambahan. Elemen tersebut adalah umpan balik dan mekanisme pengendalian. Elemen umpan balik merupakan data yang berisi tentang performasi sistem yang dibangun. Sedangkan mekanisme pengendalian bertugas untuk mengawasi dan mengevaluasi data dari hasil umpan balik dan menentukan apakah sistem bergerak menuju pecapaian tujuannya.



Gambar 2.3 Alur Elemen Sistem

2.2.1.3 Konsep Dasar Sistem Informasi

Menurut Al-Bahra Bin Ladjamudin (2013), sistem informasi dapat didefinikan sebagai berikut.[10]

1. Suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi.
2. Sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan dan atau untuk mengendalikan informasi.
3. Suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

2.2.1.4 Tujuan Sistem Informasi

Menurut Hall (2000), tiga tujuan sistem informasi adalah.

1. Untuk mendukung fungsi kepengurusan manajemen.
2. Untuk mendukung pengambilan keputusan manajemen
3. Untuk mendukung kegiatan operasi perusahaan

2.2.1.5 Kemampuan Sistem Informasi

Selain itu, sistem informasi juga memiliki kemampuan. Turban, McLean dan Wetherbe (1999) mendefinisikan kemampuan sistem informasi antara lain.

1. Melaksanakan komputasi numerik, bervolume besar dengan kecepatan tinggi.
2. Menyediakan komunikasi dalam organisasi atau antarorganisasi yang murah, akurat dan cepat.
3. Menyimpan informasi dalam jumlah yang sangat besar dalam ruang yang kecil tetapi mudah diakses.

4. Memungkinkan pengaksesan informasi yang sangat banyak diseluruh dunia dengan cepat dan murah.
5. Meningkatkan efektifitas dan efisiensi orang-orang yang bekerja dalam kelompok dalam suatu tempat atau pola beberapa lokasi.
6. Menyajikan informasi dengan jelas yang menggugah pikiran manusia.
7. Mengotomasikan proses-proses bisnis yang semiotomatis dan tugas-tugas yang dikerjakan secara manual.
8. Mempercepat pengetikan dan penyuntingan.
9. Melaksanakan hal-hal diatas jauh lebih murah daripada kalau dikerjakan secara manual.

2.2.1.6 Komponen Sistem Informasi

Berikut adalah komponen sistem informasi [15]

1. Sumber daya manusia

   Terdiri dari pengguna dan *IS* specialist. Pengguna adalah orang yang menggunakan sistem informasi sedangkan *IS* specialis adalah orang yang mengembangkan dan mengoperasikan sistem informasi.
2. Sumber daya perangkat keras

   Terdiri dari semua perangkat keras yang digunakan untuk memproses informasi. Contohnya seperti mesin.
3. Sumber daya perangkat lunak

   Terdiri dari semua instruksi yang memproses informasi yang berkaitan dengan program dan prosedur.
4. Sumber daya data.
5. Sumber daya jaringan.

2.2.2 E-Learning

Berikut adalah penjelasan detail mengenai *e-learning.*

2.2.2.1 Pengertian *E-Learning*

E-learning menurut Darin E.Hartley (2001) menyatakan, *e-learning* merupakan suatu jenis belajar mengajar yang memungkinkan tersampaikannya bahan ajar ke siswa dengan menggunakan media Internet, Intranet atau media

jaringan komputer lain. Secara lebih rinci Rosenberg (2001) mengkategorikan tiga kriteria dasar yang ada dalam *e-learning*, yaitu.

1. *E-Learning* bersifat jaringan yang membuatnya mampu memperbaiki secara cepat, menyimpan, mendistribusikan dan *sharing* pembelajaran dan informasi. Persyaratan ini sangatlah penting dalam *e-learning* sehingga Rosenberg menyebutnya sebagai persyaratan absolut.
2. *E-Learning* dikirimkan kepada pengguna melalui komputer dengan menggunakan standar teknologi internet. CD ROM, Web TV, Web Cell Phones, Pagers dan alat bantu digital personal lainnya walaupun bisa menyiapkan pesan pembelajaran tetapi tidak bisa digolongkan sebagai *e-learning*.
3. *E-learning* terfokus pada pandangan pembelajaran yang paling luas, solusi pembelajaran yang mengungguli paradigma tradisional dalam pelatihan.

2.2.2.2 Tipe *E-Learning*

Terdapat berbagai macam penggunaan *e-learning* sehingga ada pembagian atau pembedaan *e-learning*. Dalam penggunaannya *e-learning* menurut Effendi dan Hartono (2005) mempunyai dua tipe yaitu[12]

1. *Synchronous training*

   Synchronous training adalah tipe pelatihan ketika proses pembelajaran terjadi pada saat yang sama ketika pengajar sedang mengajar dan murid sedang belajar. Pembelajaran dilakukan dengan memanfaatkan perangkat komputer, jaringan komputer dan aplikasi yang dapat mendukung pembelajaran tersebut. Keuntungan dari tipe ini diantaranya.

   a. Efisien untuk peserta yang berbeda secara geografis.
   b. Mudah memberikan revisi materi.
   c. Efisiensi dalam pengiriman materi pembelejaran
   d. Dapat berinteraksi langsung dengan pengajar.
   e. Efektif ketika diperlukan diskusi antar peserta.

2. *Asynchronous training*

   Asynchronous training adalah pembelajaran yang dilakukan dengan waktu yang berbeda antara pengajar dan peserta. Pembelajaran dilakukan dengan

mandiri. Tidak pengajar yang dapat menjawab pertanyaan serta melakukan diskusi. Keuntungan dari tipe ini adalah.

a. Peserta dapat menjadi mandiri dan terbiasa mencari informasi dari bermacam sumber.
b. Menghemat waktu karena tidak harus menyediakan waktu khusus untuk pembelajaran.

2.2.2.3 Pemodelan Pembelajaran Berbasis E-Learning

Menurut Haughey (1998) ada tiga kemungkinan dalam pengembangan sistem pembelajaran berbasis internet, yaitu *web course, web centric course,* dan *web enhanced course. Web course* adalah penggunaan internet untuk keperluan pendidikan, peserta didik dan pengajar sepenuhnya terpisah dan tidak diperlukan adanya tatap muka. Seluruh bahan ajar, diskusi, konsultasi, penugasan, latihan, ujian dan kegiatan pembelajaran lainnya sepenuhnya disampaikan melalui internet. Dengan kata lain model ini menggunakan sistem jarak jauh.[14]

Web centric course adalah penggunaan internet yang memadukan antara belajar tanpa tatap muka (jarak jauh) dan tatap muka (konvensional). Sebagian materi disampaikan melalui internet dan sebagian lagi melalui tatap muka. Fungsinya saling melengkapi. Dalam model ini pengajar bisa memberikan petunjuk pada siswa untuk mempelajari materi pelajaran melalui web yang telah dibuatnya. Siswa juga diberikan arahan untuk mencari sumber lain dari situs-situs yang relevan. Dalam tatap muka, peserta didik dan pengajar lebih banyak diskusi tentang temuan materi yang telah dipelajari melalui internet tersebut.

Model *web enhanced course* adalah pemanfaatan internet untuk menunjang peningkatan kualitas pembelajaran yang dilakukan di kelas. Fungsi internet adalah untuk memberikan pengayaan dan komunikasi antara peserta didik dengan pengajar, sesama peserta didik, anggota kelompok, atau peserta didik dengan nara sumber lain. Oleh karena itu peran pengajar dalam hal ini dituntut untuk menguasai teknik mencari informasi di internet, membimbing mahasiswa mencari dan menemukan situs-situs yang relevan dengan bahan pembelajaran, menyajikan materi melalui web yang menarik dan diminati, melayani bimbingan dan komunikasi melalui internet dan kecakapan lain yang diperlukan.

2.2.2.4 Manfaat *E-Learning*

Manfaat penggunaan internet, khususnya dalam pendidikan terbuka dan jarak jauh (Elangoan, 1999, Soekartawi, 2002; Mulvihil, 1997; Utarini, 1997), antara lain dapat disebutkan sbb:

1. Tersedianya fasilitas e-moderating di mana guru dan siswa dapat berkomunikasi secara mudah melalui fasilitas internet secara regular atau kapan saja kegiatan berkomunikasi itu dilakukan dengan tanpa dibatasi oleh jarak, tempat dan waktu.
2. Guru dan siswa dapat menggunakan bahan ajar atau petunjuk belajar yang terstruktur dan terjadwal melalui internet sehingga keduanya bisa saling menilai sampai berapa jauh bahan ajar dipelajari.
3. Siswa dapat belajar atau me-review bahan ajar setiap saat dan di mana saja kalau diperlukan mengingat bahan ajar tersimpan di komputer.
4. Bila siswa memerlukan tambahan informasi yang berkaitan dengan bahan yang dipelajarinya, siswa dapat melakukan akses di internet secara lebih mudah.
5. Baik guru maupun siswa dapat melakukan diskusi melalui internet yang dapat diikuti dengan jumlah peserta yang banyak sehingga menambah ilmu pengetahuan dan wawasan yang lebih luas.
6. Berubahnya peran siswa dari yang biasanya pasif menjadi aktif.
7. Relatif lebih efisien. Misalnya bagi mereka yang tinggal jauh dari perguruan tinggi atau sekolah konvensional, bagi mereka yang sibuk bekerja, bagi mereka yang bertugas di kapal, di luar negeri dan sebagainya.

2.2.3 Dashboard

Berikut adalah penjelasan detail tentang dashboard.

2.2.3.1 Konsep Dashboard

Stephen Few (2006) mendefinisikan dashboard adalah sebuah tampilan visual dari informasi yang paling penting yang dibutuhkan untuk mencapai satu atau lebih tujuan, digabungkan dan diatur pada sebuah layar menjadi informasi yang dibutuhkan dan dapat dilihat secara sekilas. Dashboard cenderung sangat visual (grafik) dalam menyajikan informasi. Dashboard disajikan dengan benar,

gambar dari data yang diproses dapat diamati dan dipahami jauh lebih cepat daripada data yang sama disajikan sebagai teks.[5]

Shadan Malik (2005) menggunakan istilah *enterprise dashboard* yang didefinisikan sebagai sebuah antar muka komputer yang banyak menampilkan bagan, laporan, indikator visual dan mekanisme *alert*, yang dikonsolidasikan ke dalam *platform* informasi yang dinamis dan relevan. *Enterprise dashboard* berperan sebagai *live console* untuk mengelola inisiatif bisnis. Sedangkan Wayne Eckerson (2006) mendefinisikan dashboard sebagai mekanisme penyajian informasi secara visual didalam sistem manajemen kinerja, yang menyajikan informasi kritis mengenai kinerja proses operasional secara sekilas. Wayne menitikberatkan penggunaan dashboard untuk monitoring kinerja dari proses operasional.

2.2.3.2 Tujuan Penggunaan Dashboard

Wayne Eckerson (2006) menyatakan bahwa dashboard didesain untuk membantu organisasi dalam mencapai tujuan strategisnya. Dashboard digunakan untuk mengukur proses yang telah berjalan, memonitor kinerja yang sedang berjalan dan memprediksi kinerja di masa mendatang. Dengan melakukan hal tersebut, organisasi dapat membuat, menilai, menyesuaikan dan menyusun kembali strategi yang telah dibuat untuk mengoptimalkan kinerjanya. Selain itu tiga manfaat utama dashboard adalah[5]

1. Mengkomunikasikan strategi. Dashboard digunakan untuk mengkomunikasikan stategi dan tujuan yang dibuat oleh eksekutif kepada semua pihak yang berkepentingan sesuai dengan peran dan levelnya dalam organisasi.
2. Memonitor dan menyesuaikan pelaksanaan strategi. Dashboard digunakan untuk memonitor pelaksanaan dari rencana dan strategi yang telah dibuat. Dashboard memungkinkan pihak eksekutif untuk mengidentifikasi permasalahan kritis dan membuat strategi untuk mengatasinya.
3. Menyampaikan wawasan dan informasi ke semua pihak. Dashboard menyajikan informasi secara sekilas menggunakan grafik, simbol, bagan dan

warna-warna yang memudahkan pengguna dalam memahami dan mempersepsi informasi secara benar.

2.2.3.3 Karakteristik Dashboard

Shadan Malik (2005) menjabarkan karakteristik dashboard yang dapat dilihat pada tabel berikut.[11]

Tabel 2.1 Karakteristik Dashboard

Karakteristik	Penjelasan
Synergetic	Ergonomis dan memiliki tampilan visual yang mudah dipahami oleh pengguna. Dashboard mensinergikan informasi dari berbagai aspek yang berbeda dalam satu layar.
Monitor	Menampilkan *Key Performance Indicator* yang diperlukan dalam pembuatan keputusan dalam domain tertentu, sesuai dengan tujuan pembangunan dashboard tersebut.
Accurate	Informasi yang disajikan harus akurat dengan tujuan untuk mendapatkan kepercayaan dari penggunanya.
Responsive	Merespon *threshold* yang didefinisikan dengan memberikan *alert* (seperti bunyi alarm, blinker, email) untuk mendapatkan perhatian pengguna terhadp hal-hal yang kritis.
Timely	Menampilkan informasi terkini yang diperlukan untuk pengambilan keputusan.
Interactive	Pengguna dapat melakukan *drill down* dan mendapatkan informasi lebih detail, analisis sebab akibat dan sebagainya.
More data history	Pengguna dapat melihat tren sejarah dari KPI, misalkan melihat perbandingan *market share* periode saat ini dengan beberapa tahun yang lalu untuk mengetahui apakah kondisi sekarang lebih baik atau tidak.
Personalized	Penyajian informasi harus spesifik untuk setiap jenis pengguna sesuai dengan domain tanggung jawab, hak akses dan batasan akses data.
Analytical	Memberikan fasilitas bagi pengguna untuk melakukan analisis, seperti analisis sebab akibat.
Collaborative	Memberikan fasilitas pertukaran catatan (laporan) antar pengguna mengenai hasil pengamatan dashboardnya masing-masing sebagai sarana untuk komunikasi dalam rangka melakukan fungsi manajemen dan kontrol.
Trackability	Memungkinkan setiap pengguna untuk mengkustomisasi metrik yang akan dilacaknya.

2.2.3.4 Jenis-jenis Dashboard

Wayne Eckerson (2006) menyatakan dashboard dapat dikelompokkan menjadi tiga berdasarkan level manajemen yang didukungnya, yaitu *operational dashboard, tactical dashboard* dan *strategic dashboard.* Berikut adalah penjelasan dari ketiga jenis dashboard tersebut.[5]

Tabel 2.2 Ciri Khas Tiap Jenis Dashboard

Strategic Dashboard	***Tactical Dashboard***	***Operational Dashboard***
Mendukung manajemen level strategis.	Mendukung manajemen level taktikal.	Mendukung manajemen level operasional.
Memberikan informasi untuk	Memberikan informasi	Memberikan informasi

membuat keputusan bisnis, memprediksi peluang dan memberikan arahan pencapaian tujuan strategis.	yang diperlukan oleh analisis untuk mengetahui penyebab suatu kejadian.	mengenai aktifitas yang sedang terjadi, beserta perubahannnya secara *real-time* untuk memberikan kewaspadaan terhadap hal-hal yang perlu direspon secara cepat.
- fokus pada pengukuran kinerja *high level* dan pencapaian tujuan strategis organisasi. - mengadopsi konsep *Balance Score Card.*	Fokus pada proses analisis untuk menemukan penyebab dari suatu kondisi atau kejadian tertentu.	Fokus pada monitoring aktifitas dan kejadian yang berubah secara konstan.
- Informasi yang disajikan tidak terlalu detail. - Konten informasi tidak terlalu banyak dan disajikan secara ringkas.	Memiliki konten informasi yang lebih banyak (analisis perbandingan, pola/tren, evaluasi kinerja)	Informasi yang disajikan sangat spesifik dengan tingkat kedetailan yang cukup dalam.
Informasi disajikan dengan mekanisme yang sederhana melalui tampilan yang *"unidirectional".*	Menggunakan media penyajian yang "cerdas" yang memungkinkan pengguna melakukan analisis terhadap data yang kompleks berserta keterhubungannnya.	- Menggunakan media penyajian yang sederhana. - *Alert* disajikan dengan cara yang mudah dipahami dan mampu menarik perhatian pengguna.
Tidak didesain untuk berinteraksi dalam melakukan analisis yang lebih detail.	- Didesain untuk berinteraksi dengan data. - Dengan fungsi *drill-down* dan navigasi yang baik.	Didesain untuk berinteraksi dengan data untuk mendapatkan informasi yang lebih detail maupun informasi pada level yang lebih atas (*higher-level data*).
Tidak memerlukan data *real-time*	Tidak memerlukan data *real-time.*	Bersifat dinamis sehingga memerlukan data *real-teime.*

2.2.4 Internet dan World Wide Web (WWW)

Internet adalah sebuah solusi jaringan yang dapat menghubungkan beberapa jaringan lokal yang ada pada suatu daerah, kota atau bahkan pada sebuah Negara. Internet dapat menghubungkan beberapa jaringan lokal yang ada pada setiap tempat. Tidak hanya itu juga dengan adanya fasilitas internet setiap orang dapat melakukan kegiatan apa saja seperti mengirim email, mencari data, bermain game, akses bank, pembelajaan atau bahkan pembelajaran.[13]

Untuk dapat menghubungkan beberapa komputer sehingga menjadi sebuah kelompok jaringan, dibutuhkan suatu media penghubung yang bernama TCP/IP yaitu sebuah protokol yang mengidentifikasi sebuah komputer yang terhubung di

dalam jaringan. TCP/IP memiliki teknik mengidentifikasi dengan menggunakan penomoran yang dinamakan nomor IP /IP address (Internet Protokol Address). Dengan menggunakan nomor ini sebuah komputer dapat terhubung dengan komputer lain dalam sebuah jaringan atau dalam jaringan global yang disebut internet.[13]

WWW atau yang sering disebut World Wide Web (Jaringan Dunia Luas) adalah sebuah bagian dari internet yang sangat dikenal dalam dunia internet, dengan adanya WWW seorang pengguna dapat menampilkan sebuah halaman virtual yang disebut dengan Web Site.

Jika dilihat dari proses kerjanya WWW (World Wide Web) dapat dibagi menjadi beberapa komponen sebagai berikut:

a. *Protocol*

   Sebuah media yang distandarkan untuk dapat mengakses komputer di dalam sebuah jaringan. Halaman yang dapat diakses adalah halaman Web Site. WWW memiliki standar *protocol* yang bernama HTTP atau (*Hypertext Transfer Protocol*). Dengan menggunakan *protocol* ini sebuah halaman yang ada di dalam komputer jaringan dapat dibuka dan diakses.

b. *Address*

   Merupakan alamat yang berkaitan dengan penamaan sebuah komputer di dalam jaringan. Alamat ini sebenarnya merupakan sebuah nomor yang dimiliki sebuah komputer yang sering disebut dengan nomor IP, akan tetapi dengan adanya perkembangan jaman, maka dibentuklah metode baru yang bernama Domain Name sehingga no IP tersebut digantikan dengan sebuah alamat yang dinamakan URL (*Uniform Resource Locator*) yang berkaitan dengan nama sutu instansi pemiliki komputer tersebut misalnya, *http://www.akakom.ac.id.*

c. *HTML*

   Selain dari kedua media tersebut masih membutuhkan sebuah media lagi yaitu HTML (*Hypertext Markup Language*) yaitu sebuah bahasa Scripting yang dapat menghasilkan halaman Web Site sehingga halaman tersebut dapat diakses pada setiap komputer pengakses (*client*).

Dari beberapa komponen yang berkaitan dengan WWW di atas dapat dijelaskan dengan penggambaran seperti berikut.



Gambar 2.4 Hubungan antara Protokol, Address dan HTML

2.2.5 Basis Data

Berikut adalah penjelasan detail mengenai basis data.

2.2.5.1 Pengertian Basis Data

Menurut Fathansyah (2004) basis data terdiri atas dua kata, yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (siswa, pembeli, pelanggan) barang, hewan, peristiwa, konsep, keadaan dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya.[4]

Basis Data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti.

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

2.2.5.2 Manfaat Basis Data

Berikut adalah penjelasan manfaat basis data menurut Fathansyah (2004).[4]

a. Kecepatan dan kemudahan (*Speed*)

   Pemanfaatan basis data memungkinkan untuk dapat menyimpan data atau melakukan perubahan/manipulasi terhadap data atau menampilkan kembali

data tersebut dengan lebih cepat dan mudah daripada jika menyimpan data secara manual (non elektronis) atau secara elektronis (tetapi tidak dalam bentuk penerapan basis data, misalnya dalam bentuk *spread sheet* atau dokumen teks biasa).

b. Efisiensi ruang penyimpanan (S*pace*)

Karena keterkaitan yang erat antar kelompok data dalam sebuah basis data maka redundansi (pengulangan) data pasti akan selalu ada. Banyaknya redundansi ini tentu akan memperbesar ruang penyimpanan (baik di memori utama maupun memori sekunder) yang harus disediakan. Dengan basis data, efisiensi/optimalisasi penggunaan ruang penyimpanan dapat dilakukan, karena dapat melakukan penekanan jumlah redundansi data, baik dengan menerapkan sejumlah pengkodean atau dengan membuat relasi-relasi (dalam bentuk file) antar kelompok data yang saling berhubungan.

c. Keakuratan (*Accuracy*)

Pemanfaatan pengkodean atau pembentukan relasi antar data bersama dengan penerapan aturan/batasan (*constraint*) tipe data, domain data, keunikan data dan sebagainya yang secara ketat dapat diterapkan dalam sebuah basis data, sangat berguna untuk menekan ketidakakuratan pemasukan/penyimpanan data.

d. Ketersediaan (*Availability*)

Pertumbuhan data (baik dari sisi jumlah maupun jenisnya) sejalan dengan waktu akan semakin membutuhkan ruang penyimpanan yang besar. Padahal tidak semua data itu selalu digunakan atau dibutuhkan. Karena itu kita dapat memilah adanya data utama/master/referensi, data transaksi, data histori hingga data kadaluarsa. Data yang sudah jarang atau bahkan tidak pernah lagi digunakan. Dapat diatur untuk dilepaskan dari sistem basis data yang sedang aktif (menjadi *off-line*) baik dengan cara penghapusan atau dengan memindahkannya ke media penyimpanan *off-line (*seperti *removable disk* atau tape*)*

e. Kelengkapan (*Completeness*)

Lengkap tidaknya data yang dikelola dalam sebuah basis data bersifat relatif (baik terhadap kebutuhan pemakai maupun terhadap waktu).

f. Keamanan (*Security*)

Dapat menentukan siapa-siapa (pemakai) yang boleh menggunakan basis data berserta objek-objek di dalamnya dan menentukan jenis-jenis operasi apa saja yang boleh dilakukannya.

g. Kebersamaan Pemakaian (*Sharability*)

Pemakai basis data seringkali tidak terbatas pada satu pemakai saja atau di satu lokasi saja atau oleh satu sistem/aplikasi saja. Basis data yang dikelola oleh sistem (aplikasi) yang mendukung lingkungan multiuser akan dapat memenuhi kebutuhan ini tetapi tetap dengan menjaga/menghindari terhadap munculnya persoalan baru seperti inkonsistensi data (karena data yang sama diubah oleh banyak pemakai pada saat yang bersamaan) atau kondisi *deadlock* (karena ada banyak pemakai yang saling menunggu untuk menggunakan data).

2.2.6 Berioentasi Objek

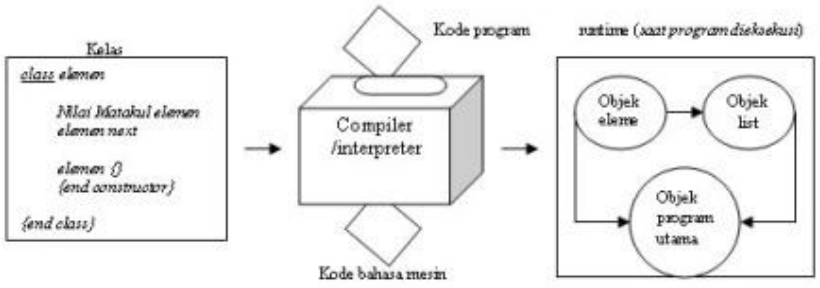
Berikut adalah penjelasan detail mengenai orientasi objek.

2.2.6.1 Pendekatan Berorientasi Objek

Pendekatan berorientasi objek adalah cara memandang persoalan menggunakan model-model yang diorganisasikan seputar konsep objek yang mengkombinasikan struktur data dan perilaku suatu entitas. Pada pendekatan berorientasi objek, blok pembangun utama sistem perangkat lunak adalah objek dan/atau kelas. Berdasarkan Abbot, kemudian diikuti oleh Rumbaugh menyatakan bahwa kandidat-kandidat kelas umumnya dapat merupakan sesuatu yang dapat diperoleh dari kosa kata (*vocabolary*) di ruang persoalan dan/atau ruang solusi. Kelas adalah deskripsi himpunan objek yang serupa. Kelas merupakan cetak biru dari objek. Setiap objek memiliki identitas (berbeda dengan objek lain), *state* (data yang diasosiakan dengannya) dan perilaku (kita dapat melakukan sesuatu pada objek dan objek dapat melakukan sesuatu pada objek lain).

Pendekatan berorientasi objek menawarkan pemakaian bersama di beberapa tingkat berbeda. Pewarisan struktur data dan perilaku memungkinkan struktur dan perilaku dipakai bersama di antara beberapa subklas serupa tanpa harus terjadi redundansi. Pemakaian bersama kode di hirarki pewarisan adalah salah satu keunggulan utama pendekatan berorientasi objek yang diterapkan pada

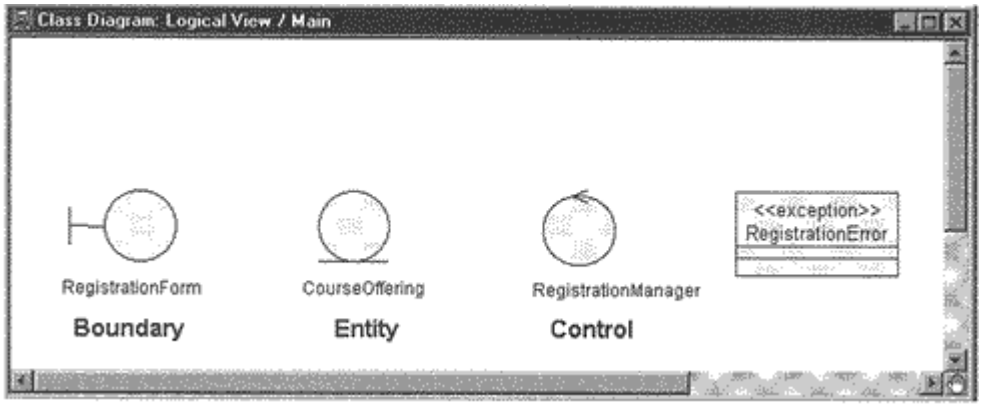
bahasa berorientasi objek. Pengembangan berorientasi objek tidak hanya memungkinkan informasi dipakai bersama di dalam satu aplikasi, tapi juga penggunaan ulang (*reuse*) rancangan dan kode pada proyek-proyek berikutnya.



Gambar 2.5 Ilustrasi Kelas dan Objek[3]

2.2.6.1.1Mendefinisikan Class

Sebuah mekanisme yang digunakan untuk mengkategorikan kelas-kelas dikenal dengan sebutan stereotype. Stereotype memberikan kemampuan untuk membuat elemen pemodelan yang baru. Beberapa stereotype untuk kelas adalah boundary, entity, dan control. Sebuah mekanisme pengembangan perangkat lunak, yaitu Rational Unified Process menggunakan *boundary*, *control* dan *entity* class. Jenis stereotype ini digambarkan pada gambar 2.6 sebagai contohnya dan disertai stereotype eksepsi.[16]



Gambar 2.6 Stereotypes pada kelas

Ketiga stereotypes ini menggambarkan sebuah sudut pandang model-view-controller sehingga membuat seorang analisis dapat membagi sistem dengan memisahkan sudut pandang dari domain dari control yang dibutuhkan oleh sistem. [16]. Berikut adalah penjelasan stereotype pada kelas.

1. *Entity Class*

   Entity class memodelkan informasi dan operasi yang biasanya lama. Tipe kelas ini menggambarkan entitas dunia nyata atau entitas yang dibutuhkan untuk melakukan tugas internal sistem. Tipe ini tidak terikat oleh komunikasi antara sistem dengan lingkungannya. Tipe ini tidak terikat dengan aplikasi dan dapat digunakan lebih dari satu aplikasi.

2. *Boundary Class*

   Boundary Class menangani komunikasi antara lingkungan sistem dan kedalam bagian sistem. Tipe kelas ini menyediakan interface ke penggguna atau sistem lain (misalnya interface ke actor). Tipe kelas ini digunakan untuk memodelkan sistem interfase.

3. *Control Class*

   Control Class memodelkan urutan kelakuan (*behaviour*) khusus untuk satu atau lebih use case. Pada awal phase *elaboration*, sebuah *control class* ditambahkan untuk setiap pasangan actor atau use case. Control class bertanggung jawab untuk aliran kejadian-kejadian dalam use case.

2.2.6.2 Konsep Berorientasi Objek

Menurut Booch (1994) ada empat konsep utama dalam konsep berorientasi objek, tanpa salah satu dari konsep utama ini maka tidak dapat dikatakan sebagai pendekatan berorientasi objek. Empat konsep itu adalah sebagai berikut.[2]

1. Abstraksi (*Abstraction)*

   Abstraksi adalah kemampuan untuk mengenali keserupaan di antara objek-objek, situasi-situasi atau proses-proses di dunia nyata serta keputusan untuk berkonsentrasi pada keserupaan-keserupaan tersebut dan mengabaikan yang disebut sebagai perbedaan-perbedaan kecil.

2. Pengkapsulan (*Encapsulation*)

   Prinsip pengkapsulan adalah penyembunyian informasi-informasi. Prinsip ini digunakan ketika pengembangan seluruh struktur program yang masing-masing komponen dari program mengkapsulkan atau menyembunyikan satu keputusan rancangan tunggal. Antarmuka masing-masing modul didefinisikan

dengan suatu cara sehingga sekecil mungkin menyatakan cara kerja di dalamnya.

3. Modularitas (*Modularity*)

   Modularitas merupakan sesuatu yang esensi untuk memecah program menjadi modul-modul kecil dimana masing-masing modul saling berinteraksi dengan lainnya lewat (sejumlah kecil) antarmuka sempit yang terdefinisi bagus.

4. Hirarki (*Hierarchy*)

   Hirarki pada sistem yang kompleks adalah struktur kelas hirarki dan struktur objek yang merupakan bagian dari hirarki. Contoh dari hirarki adalah pewarisan yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dan dirinya.

2.2.6.3 Rekayasa Perangkat Lunak Berorientasi Objek

Rekayasa perangkat lunak berorientasi objek merupakan disiplin rekayasa yang mempunyai tujuan dari disiplin rekayasa adalah mengelola program dengan mememecahnya menjadi potongan-potongan dalam ukuran yang dapat dikelola serta menyembunyikan potongan-potongan ini di belakang antarmuka dan menjadikan kebergantungan di antara ptongan-potongan itu terhadap antarmuka. Pada rekayasa perangkat lunak berorientasi objek, potongan-potongan ini adalah objek-objek dari suatu kelas.

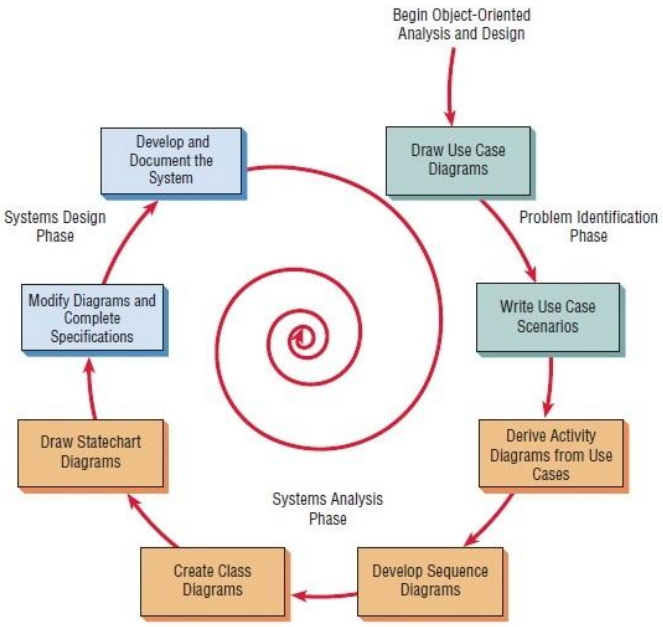
2.2.6.4 Keunggulan Menggunakan Pendekatan Berorientasi Objek

Pendekatan berorientasi objek mempunyai keunggulan sebagai berikut.[6]

a. Pendekatan objek menuntun penggunaan ulang (*reuse*) komponen-komponen program sebelumnya. Penggunaan kembali menuntun pengembangan perangkat lunak yang lebih cepat dan berkualitas lebih tinggi.
b. Perangkat lunak yang dikembangkan dengan berorientasi objek mempermudah pemeliharaan karena strukturnya secara inheren sudah menyatu.
c. Sistem berorientasi objek lebih mudah diadaptasi dan di skala sistem lebih besar karena sistem-sistem lebih besar dibuat dengan merakit subsistem-subsistem yang dapat digunakan ulang.

2.2.7 Analisis dan Perancangan Sistem Berorientasi Objek

Analisis dan perancangan berorientasi objek adalah sebuah pendekatan yang dimaksudkan untuk memfasilitasi pengembangan sistem yang berkembangan sistem yang harus berubah dengan cepat dalam menanggapi lingkungan bisnis yang dinamis. Teknik berorientasi objek dapat bekerja dengan baik sekalipun di dalam sistem informasi yang rumit yang sedang menjalani perawatan berkelanjutan, adaptasi dan perencanaan ulang. Pendekatan berorientasi objek menggunakan standar industri untuk sistem pemodelan berorientasi objek yang biasanya disebut dengan UML (*Unified Modeling Language*). Berikut gambar yang menjelaskan tentang analisis dan perancangan sistem berorientasi objek menggunakan UML menurut Kendall (2011).



Gambar 2.7 Analisis dan Perancangan Sistem Berorientasi Objek[9]

Langkah-langkah berikut memberikan penjelasan singkat mengenai gambar diatas.

1. Menentukan model use case

   Pada tahap ini, analisis memulai untuk mengidentifikasi *actor* dan apa saja yang dilakukan oleh *actor.* Seringkali, analisis akan memulai untuk menggambar diagram tongkat untuk merepresentasikan *actor* dan panah yang menunjukkan bagaimana hubungan *actor.* Ini biasanya disebut dengan *use case*

diagram (Bab 2.2.8.4) dan ini merepresentasikan alur standar kegiatan di sistem. Kemudian seorang menulis skenario use case (Bab 2.2.8.4) yang menggambarkan langkah-langkah yang dilakukan.

2. Pada tahapan sistem analisis, sudah memulai untuk menggambar UML diagram. Di tahapan kedua ini, analisis akan menggambar *Activity Diagram* (Bab 2.2.8.4) yang menjelaskan semua aktifitas *use case*. Selain itu, analisis akan membuat satu atau banyak *sequence diagram* (Bab 2.2.8.4) untuk masing-masing *use case,* yang menunjukkan urutan kegiatan berdasarkan waktu. Pada tahap ini, diberi kesempatan untuk meninjau *use case,* memikirkan kembali dan memodifikasi mereka jika memang perlu.
3. Lanjutan dari tahapan analisis, pembuatan *class diagram* (Bab 2.2.8.4)

   Benda di *use case* adalah objek yang berpotensial untuk dikelompokkan ke dalam kelas.
4. Masih dalam tahap analisis yaitu menggambar *statechart diagram* (Bab 2.2.8.4)

   Class diagram digunakan untuk menggambar *statechart diagram* yang membantu dalam proses kompleks yang tidak dapat sepenuhnya digambarkan oleh *sequence diagram. Statechart diagram* sangat berguna dalam memodifikasi *class diagram,* untuk proses yang berulang untuk melanjutkan pemodelan UML.
5. Memulai untuk perancangan sistem dengan memodifikasi UML diagram. Dan kemudian menyelesaikan spesifikasi.

   Sistem perancangan berarti memodifikasi sistem yang ada dan menjelaskan perubahan diagram pada tahapan sebelumnya. Diagram ini diperlukan *class, attributes* dan *method*. Para analisis harus menulis spesifikasi kelas untuk masing-masing kelas yang terdiri dari *attribute, method* dan penjelasan lainnya. Mereka juga harus mengembangkan spesifikasi *method* yang jelas dengan persyaratan *input* dan *output* serta dengan penjelasan rinci tentang proses internal *method.*

6. Mengembangkan dan mendokumentasikan sistem

   UML memang sebuah bahasa pemodelan. Seorang analisis dapat membuat model dengan indah, tapi jika sistem tidak dikembangkan dan tidak ada gunanya model yang dibangun. Dokumentasi sangat penting. Semakin lengkap informasi yang diberikan oleh tim pengembangan melalui dokumentasi dan *diagram* UML, maka semakin cepat pengembangan dan lebih bagus untuk hasil yang telah dibangun.

2.2.8 UML

Berikut adalah penjelasan detail mengenai UML.

2.2.8.1 Pengertian UML

UML adalah bahasa pemodelan untuk menspesifikasikan, memvisualisasikan, membangun dan mendokumentasikan artifak-artifak dari sistem. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.[6]

2.2.8.2 Tujuan UML

Tujuan utama perancangan UML adalah.[6]

1. Menyediakan bahasa pemodelan visual yang ekspresif dan siap pakai untuk mengembangkan dan pertukaran model-model yang berarti.
2. Menyediakan mekanisme perluasan dan spesialisasi untuk memperluas konsep-konsep inti.
3. Mendukung spesifikasi independen bahasa pemrograman dan proses pengembangan tertentu.
4. Menyediakan basis formal untuk pemahaman bahasa pemodelan.
5. Mendorong pertumbuhan pasar kakas berorientasi objek.
6. Mendukung konsep-konsep pengembangan level lebih tinggi seperti komponen, kolaborasi, framework dan pattern.

2.2.8.3 Kegunaan Diagram

Kegunaan diagram pada pemodelan adalah untuk formalisasi ekspresi model objek secara koheren, presisi dan mudah dirumuskan. Pemodelan berorientasi objek memerlukan kakas untuk mengekspresikan model. UML

(*Unified Modelling Language*) menyediakan sejumlah diagram untuk mengekspresikan pemodelan berorientasi objek yang dilakukan.

2.2.8.4 Diagram UML

Seperti yang tercatum pada gambar analisis dan perancangan sistem berorientasi objek diatas, berikut adalah penjelasan diagram UML.[19]

1. *Use case diagram*

   Pemodelan yang digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

2. *Use case scenario*

   Skenario *use case* digunakan untuk memudahkan dalam menganalisa skenario yang akan digunakan pada fase-fase selanjutnya.

3. *Activity diagram*

   Workflow (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. *Activity diagram* juga banyak digunakan untuk urutan pengelompokkan tampilan dari sistem / *user interface* yang setiap aktifitas dianggap memiliki sebuah rancangan antarmuka tampilan.

4. *Sequence diagram*

   Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu, untuk menggambar sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat scenario yang ada pada *use case.*

5. *Class diagram*

   Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

6. *Statechart diagram*

   Statechart diagram digunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem atau objek.

2.2.9 Teori Pendukung Program

Membuat sebuah aplikasi harus disertakan dengan teori pendukung itu sendiri sebagai sebagai referensi.

2.2.9.1 HTML

Untuk membangun sebuah *web page* dibutuhkan sebuah bahasa pemrograman yang lebih dikenal dengan sebuat web *scripting*. Dikatakan *script* karena perintah kode program tersebut akan di *interpreter* dan tidak ada kompilasi untuk menjadikannya *executable*. Berdasarkan letak proses *interpreter* maka *web scripting* dibagi menjadi dua kategori, yaitu yang bersifat *client side* dan *server side*. *Client side* dilakukan oleh *web browser* seperti *Internet Explorer, Netscape, Opera* dan *Firefox*. Untuk contoh bahasa *client side* adalah HTML, CSS, Javascript. Sedangkan *server side* dilakukan oleh *web server* seperti Apache. Untuk bahasa *server side* adalah PHP.

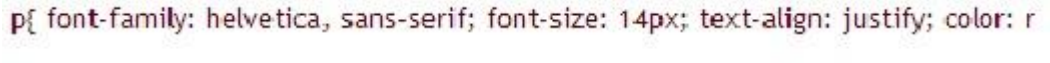
Web scripting yang bersifat *client side* akan menghasilkan *web page* yang statis artinya lebih menekankan pada desain format tampilan informasi dan informasi yang disajikan tidak dapat *update* seketika, karena tidak dapat dilakukan *request* interaktif dari pengguna dan proses tidak terjadi di *server* sehingga tidak akan menghasilkan *output* apa pun. Sehingga sangatlah tidak mungkin untuk menciptakan aplikasi *web* yang bersifat dinamis dengan *web scripting* bersifat *client side* ini, tetapi harus dikombinasikan juga dengan *web scripting* yang bersifat *server side*.

HTML adalah bahasa dasar untuk *web scripting* bersifat *client side* yang memungkinkan untuk menampilkan informasi dalam bentuk teks, grafik serta multimedia dan juga untuk menghubungkan antar tampilan *web page*.[20]

2.2.9.2 CSS

CSS merupakan singkatan dari *Cascading Style Sheet* merupakan bahasa pemrograman web yang didesain khusus untuk mengendalikan dan membangun berbagai komponen dalam web sehingga tampilan web lebih rapih, terstruktur dan seragam. Tujuan utama dari CSS adalah untuk memisahkan konten utama dengan tampilan dokumen lainnya (html dan sejenisnya). Web yang menggunakan CSS akan lebih ringan dan mudah untuk dibuka dibandingkan dengan web yang tidak

menggunakan CSS.[18] Setiap file CSS dalam bentuk .css maupun yang di-*embed* pada halaman web mempunyai satu atau lebih kode yang dinamakan statement. Statement CSS ini secara umum mempunyai dua buah komponen, yaitu Selector dan Deklarator. Selector adalah bagian dari statement yang menyeleksi elemen-elemen dalam sebuah halaman web HTML. Sedangkan deklarator adalah bagian dari statement yang mendeskripsikan bagaimana elemen yang telah dipilih selector harus ditampilkan pada halaman web. Deklarator mempunyai property dan nilai yang membedakan antara selector yang satu dengan selector yang lainnya. Berikut adalah gambar contoh untuk selector dan deklarator.



Gambar 2.8 Selektor dan Deklarator

Pada gambar diatas merupakan statement pada sebuah style sheet. Selector pada statement tersebut adalah p, statement tersebut akan memberikan pengaruh pada elemen <p> pada setiap halaman web yang terhubung dengan style sheet tersebut. Statement tersebut mempunyai deklarator yang diapit tanda kurung awalnya, didalamnya terdapat empat buah property (color, font-family, font-size, text-align) yang masing-masing properti tersebut mempunyai nilai.

2.2.9.3 PHP (*hypertext processing*)

Salah satu program yang dapat dijalankan di server dan cukup andal adalah PHP. PHP adalah salah satu bahasa pemrograman yang berjalan dalam sebuah web server dan berfungsi sebagai pengolah data pada sebuah server. Dengan menggunakan program PHP, sebuah website akan lebih interaktif dan dinamis. Data yang dikirim oleh pengunjung website/ komputer client akan diolah dan disimpan pada database *web server* dan dapat ditampilkan kembali apabila diakses. Beberapa keunggulan yang dimiliki program PHP adalah[1]

1. PHP memiliki tingkat akses yang lebih cepat
2. PHP memiliki tingkat *lifecycle* yang cepat sehingga selalu mengikuti perkembangan teknologi internet.
3. PHP memiliki tingkat keamanan yang tinggi/
4. PHP mampu berjalan di beberapa server yang ada misalnya Apache, Microsoft IIS, PWS, AOLServer, phttpd, fhttpd dan Xitami.

5. PHP juga mendukung akses ke beberapa database yang sudah ada, baik yang bersifat free/gratis ataupun komersial. Database itu antara lain MySQL, PosgreeSQL, mSQL, Informix dan MicrosoftSQL server.
6. PHP bersifat free atau gratis.

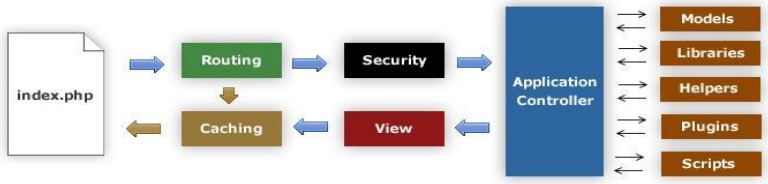
2.2.9.4 Framework CodeIgniter

Berikut adalah penjelasan detail mengenai Framework CodeIgniter.

2.2.9.4.1Pengenalan Framework CodeIgniter

Framework dapat diartikan sebagai koleksi atau kumpulan potongan-potongan program yang disusun atau diorganisasikan sedemikian rupa, sehingga dapat digunakan untuk membantu membuat aplikasi utuh tanpa harus membuat semua kodenya dari awal. CodeIgniter adalah framework yang digunakan untuk membuat sebuah aplikasi berbasis web yang disusun dengan menggunakan bahasa PHP. CodeIgniter dibuat berdasarkan kaidah Model-View-Controller (MVC). Dengan MVC maka memungkinkan pemisahan antara layar *application-logic* dan *presentation.*[1]

Untuk lebih memahami cara kerja framework CodeIgniter, berikut adalah diagram cara kerja framework CodeIgniter.



Gambar 2.9 Gambar Cara Kerja Framework CodeIgniter[17]

Berikut adalah penjelasan cara kerja Framework CodeIgniter

1. File index.php sebagai file yang bertindak sebagai kontrol halaman depan. File ini menginisialisasi semua *resource* yang diperlukan untuk menjalankan CodeIgniter atau halaman situs.
2. Routing memeriksa setiap *request* HTTP dan melakukan apa yang perlu dilakukan terhadap *request* tersebut, tergantung *rule* yang ada.
3. Apabila terdapat Cache maka sistem akan mengirimkan *cache* tersebut secara langsung ke browser tanpa harus melewati birokrasi sistem secara normal.

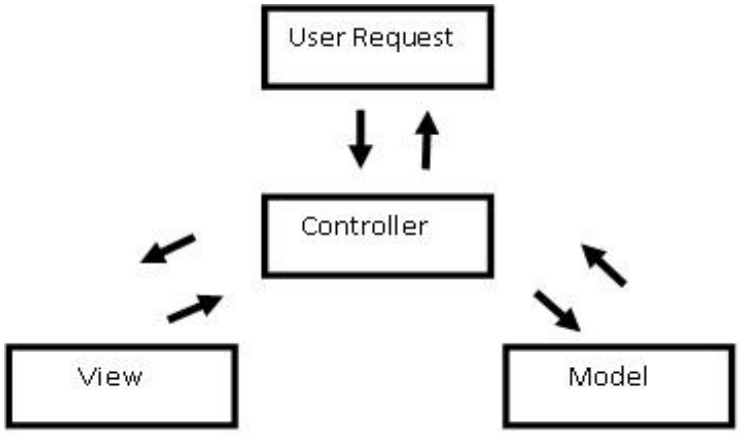
4. Sebelum membuka *Application Controller*, setiap *request* HTTP dan data yang dikirim oleh pengguna maka akan di-*filter* terlebih dahulu melalui sebuah sistem pengamanan.
5. Controller kemudian memuat seluruh *resource*, baik itu model, *library* maupun *helper* yang digunakan untuk memproses *request* atau data yang diinginkan.
6. Tahapan akhir adalah view, berfungsi me-render hasil olahan controller dan mengirimkannya ke *web browser*. Apabila sistem Cache diaktifkan maka view akan di-*cache* terlebih dahulu sebelum muncul ke *browser*.

2.2.9.4.2Konsep MVC pada CodeIgniter

Deacon (2009) mendefinisikan *Model-View-Controller* (MVC) adalah sebuah konsep yang diperkenalkan oleh penemu *Smalltalk* (*Trygve Reenskaug*) untuk meng-*enkapsulasi* data bersama dengan pemrosesan (*model*), mengisolasi dari proses manipulasi (*controller*) dan tampilan (*view*) untuk direpresentasikan pada sebuah *user interface*. MVC mengikuti pendeketan yang paling umum dari *Layering*. *Layering* hanyalah sebuah logika yang membagi kode kita ke dalam fungsi di kelas yang berbeda. Pendekatan ini mudah dikenal dan yang paling banyak diterima. Menurut Satish (2004), keuntungan utama dalam pendekatan ini adalah penggunaan ulang (*reusability*) kode.

Deacon (2009) mendefinisikan *Model-View-Controller* (MVC) adalah sebuah konsep yang diperkenalkan oleh penemu *Smalltalk* (*Trygve Reenskaug*) untuk meng-*enkapsulasi* data bersama dengan pemrosesan (*model*), mengisolasi dari proses manipulasi (*controller*) dan tampilan (*view*) untuk direpresentasikan pada sebuah *user interface*. MVC mengikuti pendeketan yang paling umum dari *Layering*. *Layering* hanyalah sebuah logika yang membagi kode kita ke dalam fungsi di kelas yang berbeda. Pendekatan ini mudah dikenal dan yang paling banyak diterima. Menurut Satish (2004), keuntungan utama dalam pendekatan ini adalah penggunaan ulang (*reusability*) kode.

Berikut adalah gambar konsep MVC pada program aplikasi berbasis *framework* CodeIgniter.



Gambar 2.10 Model - View - Controller

Keterangan:

Ketika datang sebuah *user request* maka permintaan tersebut akan ditangani oleh *controller*, kemudian *controller* akan memanggil model jika memang diperlukan operasi database. Hasil dari query oleh model kemudian akan dikembalikan ke *controller*. Selanjutnya, *controller* akan memanggil view yang tepat dan mengkombinasikannya dengan hasil query model. Hasil akhir dari operasi ini akan ditampilkan ke browser yang selanjutnya bisa dilihat di user.

Dalam konteks CodeIgniter dan aplikasi berbasis web, maka penerapan konsep MVC mengakibatkan kode program dapat dibagi menjadi tiga kategori, yaitu :

1. Model. Kode program (berupa OOP class) yang digunakan untuk memanipulasi database.
2. View. Berupa template html/xhtml atau php untuk menampilkan data pada browser.
3. Controller. Kode program (berupa OOP class) yang digunakan untuk mengontrol aliran aplikasi (sebagai pengontrol model dan view)

2.2.9.4.3Keuntungan Menggunakan Framework CodeIgniter

Adapun beberapa keuntungan menggunakan Framework CodeIgniter, diantaranya.

1. Gratis

   CodeIgniter berlisensi dibawah Apache/BSD opensource, jadi dapat digunakan secara bebas.

2. Berukuran kecil

   Ukuran CodeIgniter yang kecil merupakan keunggulan tersendiri. Dibanding framework lain yang berukuran besar serta membutuhkan resource yang besar pula untuk berjalan. Pada CodeIgniter, bisa diatur agar sistem meload library yang dibutuhkan saja sehingga sistem dapat berjalan ringan dan cepat.

3. Menggunakan konsep M-V-C

   CodeIgniter menggunakan konsep M-V-C (*Model-View-Controller*) yang memungkinkan pemisahan antara *layer application-logic* dan *presentation.*

4. URL yang sederhana

   Secara default, URL yang dihasilkan CodeIgniter sangat bersih dan *Search Engine Friendly* (SEF).

5. Memiliki paket library yang lengkap

   CodeIgniter memiliki library yang lengkap untuk mengerjakan operasi-operasi yang umum dibutuhkan oleh sebuah aplikasi berbasis web, misalnya mengakses database, mengirim email, memvalidasi form, menangani session dan sebagainya.

6. Extensible

   Sistem dapat dikembangkan dengan mudah dengan menggunakan *pluggin* dan *helper* atau dengan menggukan *hooks.*

7. Dokumentasi lengkap dan jelas

   Dari sekian banyak framework CodeIgniter adalah satu-satunya framework dengan dokumentasi yang lengkap dan jelas.

2.2.9.5 MySQL

MySQL merupakan database yang paling digemari dikalangan programmer web, dengan alasan bahwa program ini merupakan database yang sangat kuat dan cukup stabil untuk digunakan sebagai mendia penyimpanan data. Sebagai sebuah database server yang mampu untuk memanajemen database dengan baik, MySQL terhitung merupakan database yang paling digemari dan paling banyak digunakan

dibanding database lainnya. Penggunaan MySQL ini dipadukan dengan menggunakan program aplikasi PHO karena dengan menggunakan kedua program tersebut diatas telah terbukti akan kehandalannya dalam menangani permintaan data.

Pada database ini, MySQL memiliki query yang telah distandarkan oleh ANSI/ISO yaitu menggunakan bahasa SQL sebagai bahasa permintaannya, hal tersebut juga telah dimiliki oleh bentuk-bentuk database server seperti Oracle, PostgreSQL, MSQL, SQL Server maupun bentuk-bentuk database yang berjalan pada mode grafis (sifatnya visual) seperti interbase yang diproduksi oleh Borland. Kemampuan lain yang dimiliki MySQL adalah mampu mendukung Relasional Database Manajemen Sistem (RDBMS) sehingga dengan kemampuan ini MySQL akan mampu menangani data-data sebuah perusahaan yang berukuran sangat besar hingga ukuran Giga Byte. Selain itu MySQL merupakan sebuah *software* database yang bersifat gratis karena MySQL dilisensi dibawah GNU General Public License (GPL).