BAB 2
TINJAUAN PUSTAKA

2.1 Profil UNIKOM

Profil UNIKOM berisi penjelasan yang memaparkan tentang sejarah, logo, visi, misi, tujuan, dan struktur organisasi UNIKOM.

2.1.1 Sejarah UNIKOM

Universitas Komputer Indonesia (UNIKOM) secara resmi berdiri pada hari Selasa, tanggal 8 Agustus 2000 berdasarkan Surat Keputusan Menteri Pendidikan Nasional nomor 126/D/0/2000.

Awalnya dimulai pada bulan Juli tahun 1994 ketika didirikan Lembaga Pendidikan Komputer Indonesia Jerman, disingkat LPKIG, bertempat di jalan Dipati Ukur 102 Bandung. Dengan 1 ruang kelas berkapasitas 50 orang dan 1 laboratorium komputer dengan 25 unit komputer, Lembaga ini membuka program pendidikan 1 tahun dengan 5 program studi yaitu Ahli Komputer Aplikasi Bisnis, Ahli Komputer Keuangan & Perbankan, Ahli Komputer Akuntansi & Perpajakan, Ahli Komputer Manajemen & Pemasaran dan Sekretaris Eksekutif. Jumlah peserta pendidikan pada tahun pertama ini sebanyak 233 siswa.

Pada tahun kedua, 1995, dibuka jenjang pendidikan 3 tahun untuk memenuhi animo siswa tahun pertama yang ingin memperdalam ilmunya, disamping pemikiran jangka panjang pengembangan institusi. Pada tahun ini juga dibuka program studi baru, meliputi : Ahli Komputer Teknik In*form*atika, Ahli Komputer Manajemen In*form*atika dan Sekretaris Eksekutif. Ruang kelas

ditambah menjadi 2 buah dan laboratorium komputer menjadi 2 buah dengan jumlah siswa sebanyak 457 orang.

Pada tahun ketiga, 1996, dilakukan penambahan gedung kuliah baru bertempat di jalan Dipati Ukur 116 (gedung FISIP sekarang), sekaligus pemindahan pusat administrasi dan perkantoran. Digedung baru ini dilakukan

penambahan 1(satu) Lab. Komputer, 5(lima) Ruang Kuliah, Ruang Dosen dan Ruang Kemahasiswaan. Jumlah siswa dari tahun 1996 hingga tahun 1998 bertambah dari 632 orang menjadi 1184 orang.

Pada tahun kelima, 1998, dimulai pembangunan Kampus baru (Gedung Rektorat/Kampus-1 sekarang) berlantai 6 (enam) di jalan Dipati Ukur 114. Pembangunan Kampus baru ini dapat diselesaikan pada bulan Agustus 1999, sehingga pada awal perkuliahan bulan September 1999 telah dapat digunakan.

Mencermati dinamika peserta didik dan pengembangan Institusi kedepan, pada tanggal 24 Desember 1998 dibentuklah Yayasan Science dan Teknologi dan dilanjutkan dengan pengajuan pendirian STIMIK IGI dan STIE IGI ke DIKTI.

Pada bulan Juli 1999 STIE IGI diresmikan dengan keluarnya SK Mendiknas no. 119/D/O/1999 dengan 5 program studi : Akuntansi S1, Manajemen S1, Manajemen Pemasaran D3, Keuangan Perbankan D3 serta Akuntansi D3.

Pada bulan Agustus 1999 STIMIK IGI diresmikan dengan keluarnya SK Mendiknas no. 143/D/O/1999 dengan 5 program studi : Teknik In*form*atika S1, Manajemen In*form*atika D3, Teknik Komputer D3, Komputerisasi Akuntansi D3 serta Teknik In*form*atika D3

Agar Sistem Pendidikan lebih Efisien, Efektif, Produktif dengan Struktur Organisasi yang lebih baik, enam bulan kemudian dilakukan usulan ke DIKTI untuk melakukan Merger kedua Sekolah Tinggi diatas menjadi Universitas.

Pada hari Selasa, tgl. 8 Agustus 2000 keluarlah SK MENDIKNAS no. 126/D/O/2000 atas Universitas Komputer Indonesia yang disingkat dengan nama UNIKOM.Pada SK tersebut sekaligus diijinkan dibukanya 11 program studi baru: Teknik Komputer S1, Manajemen In*form*atika S1, Teknik Industri S1, Teknik Arsitektur S1, Perencanaan Wilayah dan Kota S1, Ilmu Hukum S1, Ilmu

Komunikasi S1, Ilmu Pemerintahan S1, Desain Interior D3, Desain Komunikasi Visual S1 dan Desain Komunikasi Visual D3.

Sejak berdirinya pada tahun 2000, setiap tahunnya UNIKOM menerima ± 2.000 mahasiswa baru. Terakhir pada tahun 2009 yang lalu diterima sebanyak 3.108 mahasiswa baru. Hingga tahun akademik 2009/2010 terdapat 6 Fakultas dan 23 Program Studi di UNIKOM dengan jumlah mahasiswa sebanyak 15.000 orang yang berasal dari berbagai pelosok tanah air dan dari luar negeri yang sedang menempuh pendidikan di UNIKOM.

2.1.2 Visi Misi dan Tujuan UNIKOM

Berikut penjabaran dari visi, misi, dan tujuan Universitas Komputer Indonesia adalah:

2.1.2.1 Visi

Menjadi Universitas terdepan dibidang Teknologi In*form*asi & Komputer, berwawasan Global dan menjadi pusat Unggulan dibidang ilmu pengetahuan Teknologi dan seni yang mendukung pembangunan nasional serta berorientasi pada kepentingan masyarakat, bangsa dan Negara

2.1.2.2 Misi

Menyelenggarakan Pendidikan tinggi kearah masyarakat Industri maju dengan sistem pendidikan yang kondusif, tenaga pengajar berkualitas dan program-program studi berbasis pada teknologi informasi & komputer dengan mengoptimalkan sumber daya yang ada, kualitas dan manajemen mutu berdasarkan prinsip *Quality Is Our Tradition*.

2.1.2.3 Tujuan

Menghasilkan Ilmuwan unggul dan berpikiran tinggi maju dibidangnya masing-masing, mahir menggunakan teknologi informasi & komputer dalam bekerja serta beriman dan bertakwa kepada Tuhan Yang Maha Esa.

2.1.3 Logo UNIKOM

Memiliki makna dari setiap bagiannya, antara lain bingkai segi lima, lingkaran dalam segi lima tempat tulisan berwarna kuning, bulatan dalam berwarna biru, komputer, stasiun relay, satelit, cakrawala, dan buku.



Gambar 2.1 Logo UNIKOM

Adapun penjabaran makna dari setiap bagiannya adalah sebagai berikut:

1. **Bingkai Segi Lima**

   Melambangkan UNIKOM berlandaskan falsafah negara yakni Pancasila dan Undang-Undang Dasar 1945.

2. **Lingkaran dalam Segi Lima Tempat Tulisan Berwarna Kuning**

   Melambangkan motto UNIKOM menuju kejayaan yakni *Quality Is Our Tradition.*

3. **Bulatan dalam Berwarna Biru**

   Melambangkan UNIKOM bertujuan menghasilkan ilmuwan unggul dan berpikiran maju yang Bertaqwa kepada Tuhan Yang Maha Esa.

4. **Komputer**

   Melambangkan ciri utama UNIKOM yang memberikan pendidikan Teknologi In*form*asi dan Komputasi pada seluruh Jurusan yang ada dilingkungan Universitas Komputer Indonesia, menjadi Universitas Terdepan

dibidang Teknologi In*form*asi dan Komputer serta sebagai Universitas komputer pertama di Indonesia.

5. Stasiun Relay

Melambangkan UNIKOM menyelenggarakan Pendidikan Tinggi kearah masyarakat industri maju dengan sistem pendidikan yang kondusif dan tenaga pengajar berkualitas untuk menghasilkan lulusan-lulusan terbaik.

6. Satelit

Melambangkan UNIKOM berwawasan Global dan menjadi pusat unggulan dibidang IPTEK & seni yang mendukung Pembangunan Nasional serta berorientasi pada kepentingan masyarakat, bangsa dan negara.

7. Cakrawala

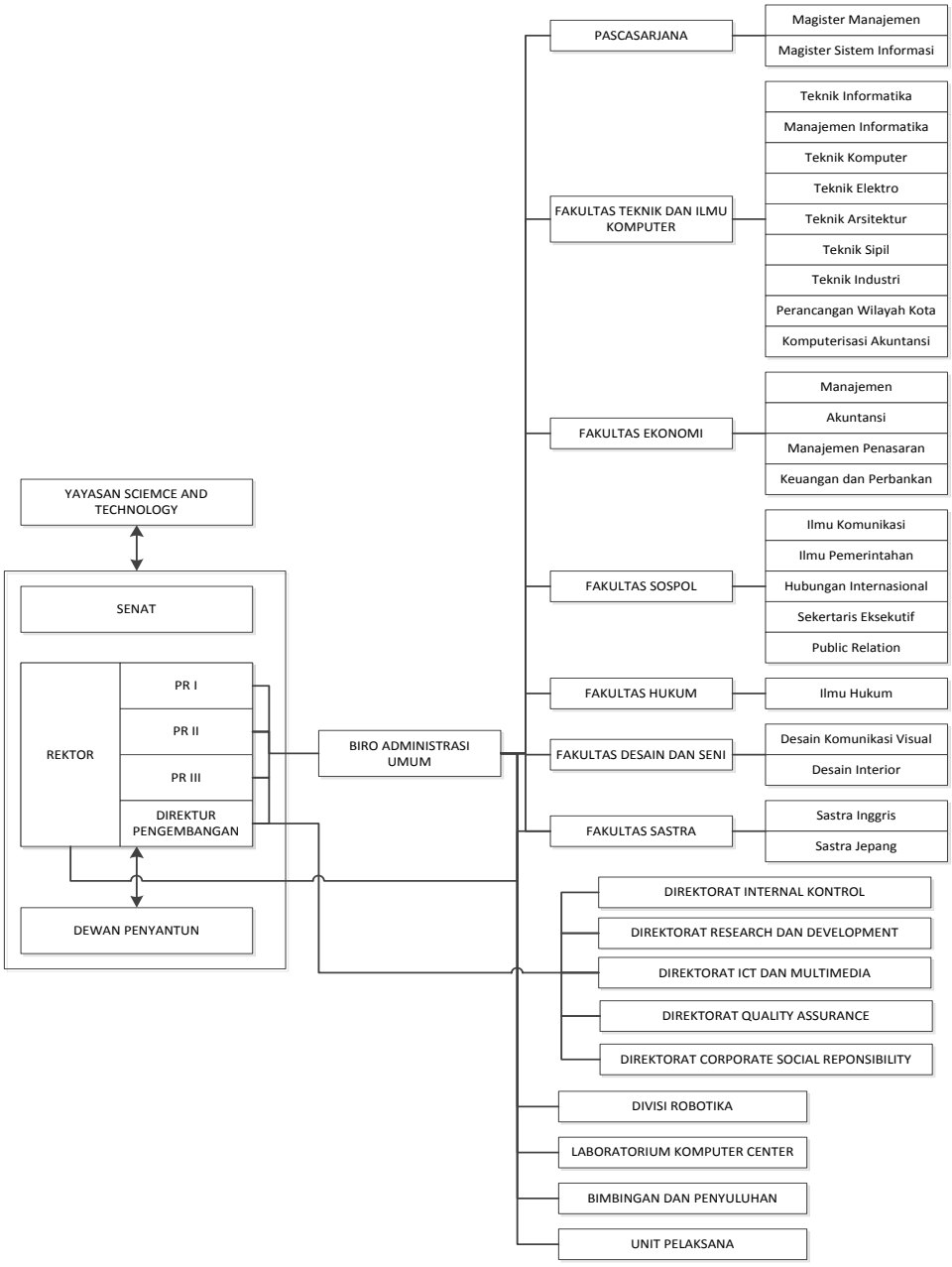
Melambangkan indahnya menggapai Cita-cita dan mengejar ilmu setinggi Langit.

8. Buku

Melambangkan sumber ilmu yang tiada habis-habisnya.

2.1.4 Struktur Organisasi UNIKOM

Unikom terdiri dari bagian atau divisi yang memiliki peranannya masing-masing. Adapun susunan organisasi Unikom dapat dilihat pada Gambar II-2



Gambar 2.2 Struktur Organisasi UNIKOM

Keterangan:

PR I : Pembantu Rektor Bidang Akademik

PR II : Pembantu Rektor Bidang Keuangan, Administrasi, dan Kepegawaian

PR III : Pembantu Rektor Bidang Kemahasiswaan dan Kerjasama

2.2 Landasan Teori

Landasan teori merupakan penjelasan berbagai konsep dasar dan teori-teori yang berkaitan dalam pembangunan aplikasi sistem invormasi inventory. Beberapa teori yang terkait dengan pembangunan aplikasi adalah bagian perlengkapan UNIKOM, sistem informasi, bahasa pemograman PHP, *framework* Codeigniter, Mysql, client server dan UML.

2.2.1 Sistem Informasi

Pada dasarnya sistem informasi bukanlah sistem yang selalu terkait dengan komputer. Adapun beberapa penjelasan terkait dengan sistem informasi adalah pengertian sistem, pengertian data, pengertian informasi, dan pengertian sistem informasi itu sendiri.

Menurut Teguh Wahyono, *"Sistem informasi diartikan sebagai suatu jaringan daripada beberapa elemen-elemen yang saling berhubungan serta membentuk satu kesatuan untuk menyediakan informasi yang dibutuhkan oleh organisasi untuk beroperasi dengan cara yang sukses, serta member sinyal kepada manajemen dan yang lain terhadap kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan"* [1]

2.2.2 Pengertian Sistem

Pengertian sistem bisa dilihat dari dua pendekatan, yaitu berdasarkan prosedur dan komponennya. Sistem dilihat dari pendekatan prosedur berarti kumpulan prosedur yang saling bekerja sama untuk mencapai suatu tujuan tertentu sedangkan sistem dilihat dari pendekatan komponennya berarti sistem terdiri dari komponen-komponen yang saling berinteraksi untuk mencapai suatu tujuan tertentu. [2]

2.2.3 Pengertian Data

Data merupakan fakta mengenai suatu objek seperti manusia, benda, peristiwa, konsep, keadaan dan sebagainya yang dapat dicatat dan mempunyai arti

secara impilisit. Data dapat dinyatakan dalam bentuk angka, karakter atau simbol, sehingga bila data dikumpulkan dan saling berhubungan maka dikenal dengan istilah basis data (database) [3]

2.2.4 Pengertian Informasi

In*form*asi merupakan data yang memiliki nilai lebih bagi penerimanya karena telah melewati proses pengolahan tertentu sehingga bisa dimanfaatkan untuk mengambil keputusan baik itu untuk saat ini ataupun untuk masa yang akan datang atau informasi didefinisikan sebagai data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

2.2.5 Pengertian Sistem Informasi

Berdasarkan pemaparan sebelumnya mengenai sistem, data, dan informasi maka dapat diambil kesimpulan bahwa pengertian dari sistem informasi adalah sekumpulan komponen-komponen dan prosedur-prosedur yang saling bekerja sama untuk mencapai suatu tujuan tertentu.

2.2.6 Pengertian Inventory

Menurut A.A.K Oka Sudana (2007) inventori (persediaan) adalah bahan baku, produk setengah jadi, produk jadi yang berada di dalam sistem produksi pada suatu waktu, yang bersifat sebagai buffer (penyangga) yang belum digunakan (idle) yang mempunyai nilai ekonomis di masa mendatang pada saat aktif. Fungsi dari manajemen inventory:

1. perencanaan inventory: menentukan kebutuhan material untuk memenuhi kebutuhan sesuai rencana operasi dan produksi yang telah disusun, yaitu berapa banyak yang harus dipesan.

2. pengendalian inventory: menentukan tingkat inventory yang sesuai dimana pemesanan harus dilakukan kembali, persediaan pengaman, dan kondisi inventory tersebut yang terkait

Sistem perencanaan dan pengendalian inventory yang efektif akan memberikan pemenuhan kebutuhan secara tepat baik waktu, jumlah maupun spesifikasi, dengan total biaya persediaan yang optimal.

Alasan-alasan perlunya inventory:

1. kebutuhan tidak bisa konstan / sesuai dengan due date yang telah ditentukan.
2. fluktuasi (naik/turunnya) harga barang yang dipesan.
3. ketidakpastian kebutuhan bahan terhadap order yang diterima.

Biaya-biaya yang terkait dalam penentuan total biaya inventory:

1. Harga: harga beli perunit jika item diperoleh dari vendor (supplier luar) atau biaya produksi per unit bila item tersebut diproduksi sendiri.
2. Capital cost: jumlah modal yang terkait pada item yang disimpan, suku bunga i% dikalikan dengan harga beli / produksi.
3. Carrying cost (holding cost), dapat meliputi:
   a. biaya depresiasi
   b. biaya asuransi dan pajak
   c. biaya pemakaian area / ruang serta fasilitas-fasilitas dalam ruang penyimpanan maupun fasilitas penanganan baik secara fisik maupun yang berkaitan dengan data / informasi inventory.
4. Ordering cost: meliputi biaya-biaya persiapan dan peletakkan order (pesanan) inventory, biaya handling dan pengiriman order, biaya pemeriksaan order yang datang. jika item diproduksi sendiri maka disebut 20 setup cost, yang meliputi biaya persiapan / setup mesin untuk produksi. biaya ini ditentukan untuk setiap kali pesanan / setup/
5. Shortage cost: biaya ini timbul jika inventory tidak tersedia pada saat dibutuhkan. biaya ini meliputi biaya karena kehilangan kesempatan untuk memperoleh keuntungan dari penjualan biaya karena kehilangan/ turunya reputasi perusahaan, ketidakpuasan pelanggan, biaya-biaya administrasi

yang terkait dalam usaha untuk mempertahankan permintaan yang tidak dipenuhi langsung.

2.2.7 Personal Home Page (PHP)

PHP atau *Personal Home Page* merupakan sebuah bahasa *scripting* yang dibundel dengan HTML, yang dijalankan di sisi *server*. Sebagain besar intinya berasal dari C, Java dan Perl dengan beberapa tambahan fungsi khusus PHP. Bahasa ini memungkinkan para pembuat aplikasi web menyajikan halaman HTML dinamis dan interaktif dengan cepat dan mudah, yang dihasilkan *server*. PHP juga dimaksudkan untuk mengganti teknologi lama seperti *Common Gateway Interface* (CGI).

PHP membuat proses pengembangan aplikasi menjadi mudah karena kelebihan-kelebihannya, yaitu :

1. Script (kode program) terintegrasi dengan file HTML, sehingga developer bisa berkonsentrasi langsung pada penampilan dokumen webnya.
2. Tidak ada proses *compiling* dan *linking*.
3. Berorientasi obyek.
4. Sintaksis pemogramannya mudah dipelajari, dan menyerupai C dan Perl.

Integrasi yang sangat luas ke berbagai server *database*. Menulis web yang terhubung ke *database* menjadi sangat sederhana. Pada umumnya PHP menggunakan MySQL sebagai *database*, namun PHP juga mendukung *database* yang lain seperti Oracle, Sybase, MySQL, Solid, ODBC, PostgreSQL, Adabas D, FilePro, Velocis, In*form*ix, dBase, UNIX dbm.

2.2.8 *Framework* CodeIgniter

Framework adalah kumpulan perintah atau fungsi dasar yang membentuk aturan-aturan tertentu dan saling berinteraksi satu sama lain sehingga dalam pembuatan aplikasi website kita harus mengikuti aturan dari *framework* tersebut. Dengan *framework* (dalam hal ini *framework* php), kita tidak perlu memikirkan

kode perintah/fungsi dasar dari aplikasi website kita. Seperti bagaimana mengambil data dari *database* untuk ditampilkan.

Hal-hal penunjang lainnya seperti koneksi *database* validasi *form*, GUI, dan keamanan; telah disediakan oleh *framework* sehingga jumlah baris kode yang kita buat jauh lebih sedikit dibandingkan jika semua kode dari kita. Jadi, keuntungan yang dapat diperoleh dari pengguna *framework* adalah:

1. Waktu pembuatan aplikasi website kita jauh lebih singkat.
2. Kode aplikasi website menjadi lebih mudah dibaca, karena sedikit dan sifatnya pokok.
3. Website kita menjadi lebih mudah diperbaiki,karena kita tidak perlu fokus ke semua komponen kode website, terutama kode *system framework.*
4. Kita tidak perlu lagi membuat kode penunjang aplikasi website seperti koneksi *database*,validasi *form*,GUI,dan keamanan.
5. Pikiran kita menjadi lebih terfokus ke kode alur permasalahan website, apa yang ditampilkan dan layanan apa saja yang diberikan dari aplikasi website tersebut.
6. Jika dikerjakan *team work*, maka akan lebih terarah karena *system framework*, mengharuskan adanya keteraturan peletakan kode. seperti bagian pengambilan *database* terpisah dengan bagian pengaturan tampilan untuk pengunjung.

Jadi intinya banyak keuntungan yang dapat diperoleh, terutama bagi developer web yang selalu dihantui waktu deadline proyek websitenya. Tidak selesai waktunya sesuai target, klien akan kecewa dan mungkin akan lari ke developer lain. Jika anda bersikeras membuat website dari nol kode maka akan butuh banyak waktu dan pasti akan ketinggalan dengan teman anda yang menggunakan *framework.*

Framework CI paling mudah dikuasai untuk seorang pemula, karena help-nya cukup tersedia diwebsitenya. bahkan saat kita mendownload *framework* ini

dari internet, sudah disertai help-nya yang mudah dimengerti, mulai dari langkah instalasi sampai fungsi-fungsinya tersedia. Adanya help/dokumentasi sangat mudah bagi pemula dalam mempelajari lingkungan pengembangan website dengan Codeigniter.

Framework Codeigniter dapat dioperasikan dalam PHP 4.3.2 + maupun 5 sehingga kita membuat aplikasi website pada sebuah *server* yang masih belum support PHP5, ini tidak akan menjadi masalah,karena masih tetap dapat berkerja. MVC (*model view controller*) adalah sebuah pola program yang bertujuan memisahkan logika bisnis,logika data, dan logika tampilan (*interface*), atau secara sederhana memisahkan antara proses, data dan tampilan mvc mengatur arsitektur sebuah aplikasi. umumnya apliksai yang dibangun dengan konsep MVC adalah apliksai yang dibangun dengan konsep MVC adalah aplikasi yang cukup besar, karena salah satu keuntungan dari MVC itu adalah kemudahan maintenance, dan pengembangan apliksai tersebut.CI menggunakan konsep ini anda harus memisahkan kode *database* ke folder model, kode proses ke folder controller dan kode tampilan ke folder view. [4]

2.2.8.1 Jenis MVC pada website :

1. Server Side MVC

Server Side MVC biasa terjadi pada aplikasi web tradisional, yang tidak melibatkan *client side* seperti Javascript, Java applet, Flash dan lain-lain. *Server Side* MVC menyerahkan keseluruhan proses bisnis pada server, aplikasi pada sisi pengguna hanya dapat menerima. MVC jenis ini kadang-kadang disebut juga dengan nama *Thin Client*.

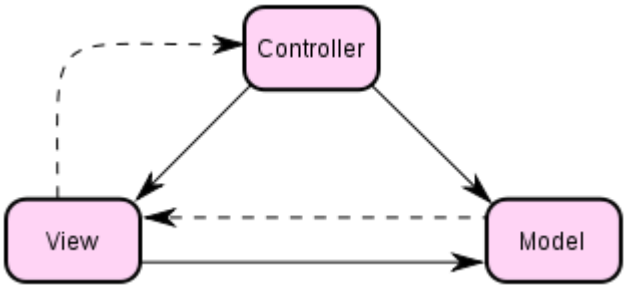
2. Mixed Client Side and Server Side MVC

Pada Mixed Client Side and Server Side MVC 1 client tidak menggunakan model sebagai jembatan untuk melakukan komunikasi pada server, dibandingkan dengan Server Side MVC, arsitektur ini memiliki tingkat kompleksitas yang lebih tinggi karena lebih banyak komponen yang terlibat.

Untuk selanjutnya arsitektur ini disebut, dengan Mixed MVC 1. Pada Mixed Client Side and Server Side MVC 2, client menggunakan model sebagai jembatan untuk melakukan komunikasi pada server, dibandingkan dengan arsitektur MVC yang lain, arsitektur ini memiliki tingkat kompleksitas yang paling tinggi karena lebih banyak komponen yang terlibat, sehingga membutuhkan sumber daya yang lebih besar pula. Untuk selanjutnya arsitektur ini disebut dengan Mixed MVC

3. Rich Internet Application MVC

Application MVC Rich Internet Application (RIA) disebut juga dengan nama *Fat Client*, merupakan aplikasi web yang memiliki kemampuan dan fungsi hampir seperti aplikasi desktop. RIA pada sisi client, memiliki mesin untuk mengambil data yang berada pada server, sehingga pada client terdapat bagian MVC sendiri dan hanya membutuhkan bagian model pada sisi server.



Gambar 2.3 Konsep MVC pada Framework Codeigniter

2.2.9 *Hyper Text Markup Language* (HTML)

HTML (*Hyper Text Markup Language*) adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman web dan menampilkan berbagai informasi di dalam sebuah *browser* Internet. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML *(Standard Generalized Markup Language)*, HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium*(W3C).

HTML berupa kode-kode tag yang menginstruksikan browser untuk menghasilkan tampilan sesuai dengan yang diinginkan. Sebuah file yang merupakan file HTML dapat dibuka dengan menggunakan *browser* web seperti *Mozilla Firefox* atau *Microsoft Internet Explorer*. HTML juga dapat dikenali oleh aplikasi pembuka email ataupun dari PDA dan program lain yang memiliki kemampuan browser.

HTML dokumen tersebut mirip dengan dokumen teks biasa, hanya dalam dokumen ini sebuah teks bisa memuat instruksi yang ditandai dengan kode atau lebih dikenal dengan TAG tertentu. Sebagai contoh jika ingin membuat teks ditampilkan menjadi tebal seperti: Tambil Tebal, maka penulisannya dilakukan dengan cara: Tampil Tebal. Tanda digunakan untuk mengaktifkan instruksi cetak tebal, diikuti oleh teks yang ingin ditebalkan, dan diakhiri dengan tanda untuk menonaktifkan cetak tebal tersebut.

Secara garis besar, terdapat beberapa jenis elemen dari HTML:

Structural. tanda yang menentukan level atau tingkatan dari sebuah teks (contoh,<h1>Golf</h1> akan memerintahkan *browser* untuk menampilkan "Golf" sebagai teks tebal besar yang menunjukkan sebagai Heading 1

Presentational. tanda yang menentukan tampilan dari sebuah teks tidak peduli dengan level dari teks tersebut (contoh, boldface akan menampilkan bold. Tanda presentational saat ini sudah mulai digantikan oleh CSS dan tidak direkomendasikan untuk mengatur tampilan teks,

Hypertext. tanda yang menunjukkan pranala ke bagian dari dokumen tersebut atau pranala kedokumen lain (contoh, Web Saya akan menampilkan Web Saya sebagai sebuah hyperlink ke URL tertentu), Elemen widget yang membuat objek-objek lain seperti tombol (<button>), list (<li>), dan garis horizontal (<hr>). Selain markup presentational, markup yang lin tidak menentukan bagaimana tampilan dari sebuah teks. Namun untuk saat ini, penggunaan tag HTML untuk menentukan tampilan telah

dianjurkan untuk mulai ditinggalkan dan sebagai gantinya digunakan *Cascading Style Sheets.*

2.2.10 *Structured Query Language* (SQL)

SQL (*Structured Query Language*) adalah sebuah bahasa yang dipergunakan untuk mengakses data dalam basis data relasional. Bahasa ini secara *de facto* merupakan bahasa standar yang digunakan dalam manajemen basis data relasional. Saat ini hampir semua server basis data yang ada mendukung bahasa ini untuk melakukan manajemen datanya.

2.2.10.1 Sejarah *Structured Query Language* (SQL)

Sejarah SQL dimulai dari artikel seorang peneliti dari IBM bernama EF Codd yang membahas tentang ide pembuatan basis data relasional pada bulan Juni 1970. Artikel ini juga membahas kemungkinan pembuatan bahasa standar untuk mengakses data dalam basis data tersebut. Bahasa tersebut kemudian diberi nama SEQUEL (*Structured English Query Language*).

Setelah terbitnya artikel tersebut, IBM mengadakan proyek pembuatan basis data relasional berbasis bahasa SEQUEL. Akan tetapi, karena permasalahan hukum mengenai penamaan SEQUEL, IBM pun mengubahnya menjadi SQL. Implementasi basis data relasional dikenal dengan *System/R*.

Di akhir tahun 1970-an, muncul perusahaan bernama *Oracle* yang membuat *server* basis data populer yang bernama sama dengan nama perusahaannya. Dengan naiknya kepopuleran *Oracle*, maka SQL juga ikut populer sehingga saat ini menjadi standar *de facto* bahasa dalam manajemen basis data.

2.2.10.2 *Standarisasi Structured Query Language* (SQL)

Secara umum, SQL terdiri dari dua bahasa, yaitu *Data Definition Language* (DDL) dan *Data Manipulation Language* (DML). Implementasi DDL dan DML berbeda untuk tiap sistem manajemen basis data (SMBD), namun secara umum implementasi tiap bahasa ini memiliki bentuk standar yang ditetapkan ANSI.

1. *Data Definition Language* (DDL)

DDL digunakan untuk mendefinisikan, mengubah, serta menghapus basis data dan objek-objek yang diperlukan dalam basis data, misalnya tabel, *view*, *user*, dan sebagainya. Secara umum, DDL yang digunakan adalah `CREATE` untuk membuat objek baru, `USE` untuk menggunakan objek, `ALTER` untuk mengubah objek yang sudah ada, dan `DROP` untuk menghapus objek. DDL biasanya digunakan oleh administrator basis data dalam pembuatan sebuah aplikasi basis data.

2. *Data Manipulation Language* (DML)

DML digunakan untuk memanipulasi data yang ada dalam suatu tabel. Perintah yang umum dilakukan adalah:

SELECT untuk menampilkan data

INSERT untuk menambahkan data baru

UPDATE untuk mengubah data yang sudah ada

DELETE untuk menghapus data

2.2.10.3 MySql

MySQLadalah sebuah perangkat lunak sistem manajemen basis dataSQL (bahasa Inggris : *database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek-proyek seperti *Apache*, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh

sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah : David Axmark, Allan Larsson dan Michael "Monty" Widenius.

MySQL memiliki beberapa keistimewaan, antara lain :

1. *Portabilitas*. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti *Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga,* dan masih banyak lagi.
2. Perangkat lunak sumber terbuka. MySQL dipenyerahankan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
3. *Multi-user*. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. *Performance tuning*, MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed / unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.
6. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).
7. Keamanan. MySQL memiliki beberapa lapisan keamanan seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protocolTCP/IP, Unis Socket (UNIX), atauNamed Pipes (NT).
10. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. Antar Muka. MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API(*Application Programming Interface*).
12. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
13. Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan basis data lainnya semacam *Posgre*SQL ataupun *Oracle*.

2.2.11 *Cascading Style Sheet* (CSS)

Cascading Style Sheet (CSS) merupakan salah satu bahasa pemrograman web untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam.

Sama halnya *styles* dalam aplikasi pengolahan kata seperti *Microsoft Word* yang dapat mengatur beberapa *style*, misalnya *heading*, *subbab*, *bodytext*, *footer*, *image,* dan *style* lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (*file*). Pada umumnya CSS dipakai untuk mem*form*at tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML.

CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran border, warnaborder, warna hyperlink, warna*mouse over*, spasiantar paragraf, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameter lainnya. CSS adalah bahasa *style sheet* yang digunakan untuk mengatur tampilan dokumen. Dengan adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan *form*at yang berbeda.

2.2.12 Java Script

JavaScript pertama kali diperkenalkan oleh *Netscape* pada tahun 1995. Pada awalnya bahasa yang sekarang disebut *JavaScript* ini dulunya dinamai *"LiveScript""* yang berfungsi sebagai bahasa sederhana untuk *browserNetscape Navigator* 2 yang sangat populer pada saat itu. Kemudian sejalan dengan sedang giatnya kerjasama antara *Netscape* dan Sun (pengembang bahasa pemrograman "*Java*") pada masa itu, maka *Netscape* memberikan nama "*JavaScript*" kepada bahasa tersebut pada tanggal 4 desember 1995.

Pada saat yang bersamaan Microsoft sendiri mencoba untuk mengadaptasikanteknologi ini yang mereka sebut sebagai "*Jscript*" di *browser* milik mereka yaitu Internet Explorer 3. *JavaScript* sendiri merupakan modifikasi dari bahasa pemrograman C++ dengan pola penulisan yang lebih sederhana dari bahasa pemrograman C++.

JavaScript adalah bahasa pemrograman berbasis *prototipe* yang berjalan disisi klien. Jika kita berbicara dalam konteks web, sederhananya, kita dapat memahami *JavaScript* sebagai bahasa pemrograman yang berjalan khusus untuk di *browser* atau halaman web agar halaman web menjadi lebih hidup. Kalau dilihat dari suku katanya terdiri dari dua suku kata, yaitu *Java* dan *Script*. *Java* adalah Bahasa pemrograman berorientasi objek, sedangkan *Script* adalah serangkaian instruksi program.

Secara fungsional, *JavaScript* digunakan untuk menyediakan akses *script* pada objek yang dibenamkan (embedded). Contoh sederhana dari penggunaan *JavaScript* adalah membuka halaman pop up, fungsi validasi pada *form* sebelum data dikirimkan ke *server*, merubah *image cursor* ketika melewati objek tertentu, dan lain lain.

2.2.13 JQuery

JQuery adalah javascript *library*, JQuery dirancang untuk memperingkas kode-kode javascript. JQuery adalah javascript *library* yang cepat dan ringan untuk menangani dokumen HTML, event, membuat animasi dan interaksi ajax. [6]

Kemampuan library JQuery :

1. Kemudahan mengakses elemen-elemen HTML
2. Memanipulasi elemen HTML
3. Memanipulasi CSS
4. Penanganan event HTML
5. Efek – efek javascript dan animasi
6. Modifikasi HTML DOM
7. AJAX
8. Menyederhanakan kode javascript lainya

JQuery merupakan salah satu librari yang membuat program web di sisi klien, tidak terlihat sebagai program JavaScript biasa, yang harus secara eksplisit disisipkan pada dokumen web. Pada teknik pemrograman sisi klien dengan menggunakan JavaScript biasa, setiap elemen yang akan memiliki event, akan secara eksplisit terlihat ada event yang dilekatkan pada elemen tersebut.

2.2.14 Pemograman Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientasi objek didasarkan pada penerapan prinsip-prinsip pengolahan kompleksitas. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemograman berorientasi objek. Dan pengujian berorientasi objek. [5]

Pada saat ini metode berorientasi objek banyak digunakan karena metodologi lama banyak menimbulkan masalah seperti adanya kesulitaan pada saat mentrans*form*asi hasil dari satu tahap pengembangan ke tahap berikutnya, misalnya pada metode perkembangan terstruktur, jenis aplikasi yang dikembangkan saat ini sangat beragam (aplikasi bisnis, *real-time*, *utility*, dana sebagainya) dengan plat*form* yang berbeda-beda, sehingga menimbulkan tuntutan kebutuhan metodologi pengembangan yang dapat mengakomodasi ke semua jenis aplikasi tersebut.

Berikut beberapa keuntungan menggunakan metodologi berorientasi objek, antara lain :

1. Meningkatkan produktivitas

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

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

   Karena dengan model objek , pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin sering berubah-ubah
4. Adanya konsistensi

   Karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan dan pengkodean
5. Meningkatkan kualitas perangkat lunak

   Karena pendekatan pengembangan lebih dekat dengan dunia nyata dan adanya konsistensi pada saat pengembangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.

2.2.15 UML

Pada perkembangan teknologi perangkat lunak, diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai Negara dapat mengerti pemodelan perangkat lunak. Seperti yang kita ketahui bahwa menyatukan banyak kepala untuk menceritakan sebuah ide dengan tujuan untuk memahami hal yang sama tidaklah mudah, oleh karena itu diperlukan sebuah bahasa pemodelan perangkat lunak yang dapat dimengerti oleh banyak orang.

Banyak orang yang telah membuat bahasa pemodelan pembangunan perangkat lunak sesuai dengan teknologi pemograman yang berkembang pada saat itu, misalnya yang sempat berkembang dan digunakan oleh banyak pihak adalah *Data Flow Diagram* (DFD) untuk memodelkan perangkat lunak yang menggunakan pemograman prosedural atau structural kemudian juga ada *State Transition Diagra*m (STD) yang digunakan untuk memodelkan sistem *real time* (waktu nyata).

Pada perkembangan teknik pemograman berorientasi objek, munculah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun , dan dokumentasi dari sistem perangkat lunak. UML, merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada teknologi tertentu, meskipun pada kenyataannya UML banya digunakan pada pemograman berorientasi objek.

Banyak hal di dunia sistem informasi yang tidak dapat dilakukan, semua tergantung kebutuhan, lingkungan dan konteksnya. Begitu juga pada

perkembangan penggunaan UML bergantung pada level abstraksi penggunanya. Jadi, belum tentu pandangan yang berbeda dalam penggunaan UML adalah suatu yang salah, tetapi perlu ditelaah dimanakah UML digunakan dan hal apa yang ingin divisualkan. [6]

2.2.15.1 Sejarah UML

Bahasa pemograman berorientasi objek yang pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangan pada tahun 1967.

Bahasa pemograman ini kurang berkembang dan dikembangkan lebih lanjut, namun dengan kemunculannya telah memberikan sumbangan yang besar pada developer pengembang bahasa pemograman berorientasi objek.

Perkembangan aktif dari pemograman berorientasi objek mulai mennggeliat ketika berkembangnya bahasa pemograman *Smalltalk* pada awal 1980-an yang kemudian diikuti perkembangan bahasa pemograman berorientasi objek lainya, seperti C objek, C++, *Eiffel*, dan *CLOS*. Secara aktual, penggunaan bahasa pemograman berorientasi objek pada saat itu masih terbatas, namun telah banyak menarik perhatian di saat itu. Sekitar lima tahun setelah *Smalltalk* berkembang, maka berkembang pula metode pengembangan berorientasi objek. Metode yang pertama diperkenalkan oleh Sally Shalaer dan Stephen Mellor (Shlaer–Melor, 1988) dan Peter Coad dan Edward Yourdon (Coad–Yourdon, 1991), diikuti oleh Grady Booch (Booch, 1991), James R. Rumbaugh, Michael R. Blaha, Wiliam Lorensen, Frederick Eddy, Wiliam Premerlani (Rumbaugh-Blaha-Lorensen- Eddy- Premerlani,1991) dan masih banyak lagi. [7]

Karena banyaknya metodologi-metodologi yang berkembang pesat saat itu, maka munculah ide untuk membuat sebuah bahasa yang dapat dimengerti semua orang. Usaha penyatuan ini banyak mengambil dari metodologi-metodologi yang berkembang pesat saat itu. Maka dibuat bahasa yang merupakan gabungan dari beberapa konsep seperti konsep *Object Modeling Technique* (OMT) dari Rumbaugh dan Booch (1991), konsep *The Classes*, *Responsibilities*,

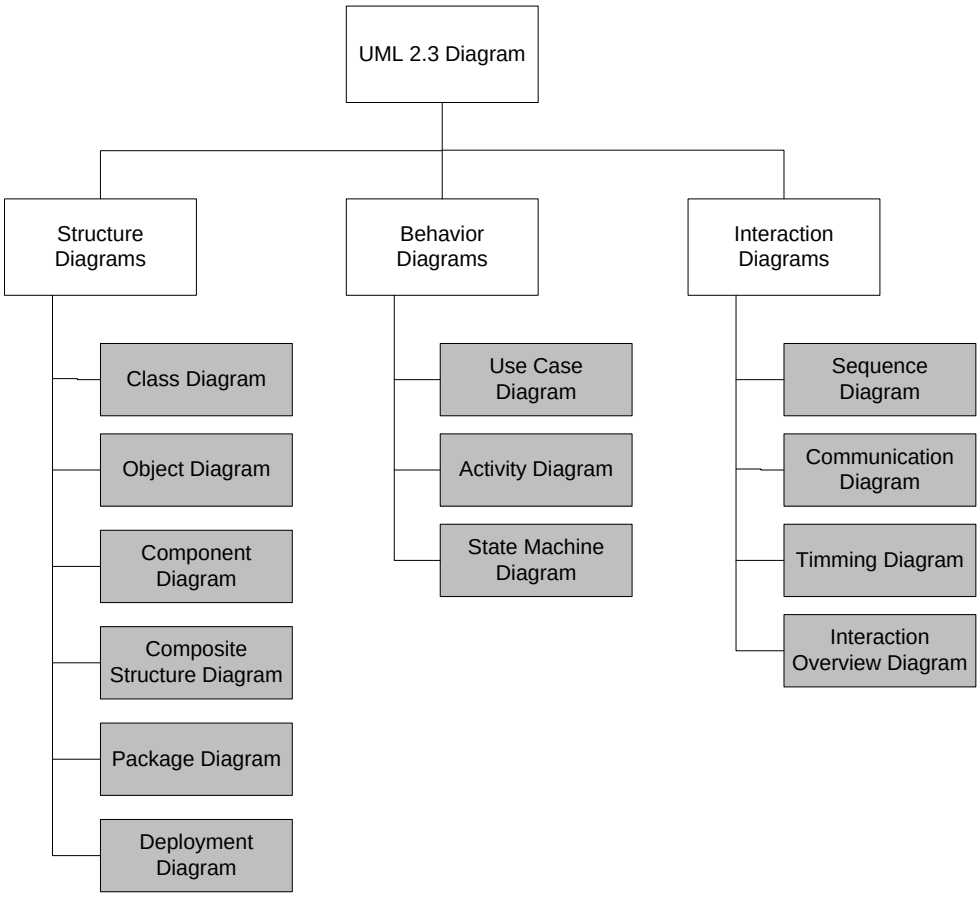
Collaborators (CRC) dari Rebeca Wirfs-Brock (1990), konsep pemikiran Ivar Jacobson, dan bebrapa konsep lainnya dimana James Rumbaigh, Grady Booch dan ivar Jacobson bergabung dalam sebuah Perusahaan yang bernama Rational Software Corporation menghasilkan bahasa yang disebut dengan *Unified Modeling Language* (UML).

Pada tahun 19916, *Object Management Group* (OMG) mengajukan proposal agar adanya standarisasi pemodelan berorientasi objek dan pada bulan September 1997 UML diakomodasi oleh OMG sehingga sampai saat ini UML telah memberikan kontribusi yang cukup besar di dalam metodologi berorientasi objek dan hal-hal yang terkait di dalamnya.

Secara fisik UML adalah sekumpulan spesifikasi yang dikeluarkan oleh OMG. UML terbaru dalah UML 2.3 yang terdiri dari 4 macam spesifikasi, yaitu Diagram Interchange Spesification, UML infrastructure, UML Superstructure, dan Object Constraint Language (OCL). Seluruh spesifikasi tersebut dapat diakses di website http://www.omg.org

2.2.15.2 Diagram UML

Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar berikut.



Gambar 2.4 Diagram UML

1. Structure Diagram

   Kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan, terdiri dari :

   a. *Class Diagram* untuk memodelkan struktur kelas.
   b. *Object Diagram* untuk memodelkan struktur *object*.
   c. *Component Diagram* untuk memodelkan komponen *object*.
   d. *Composite Structure Diagram* untuk menggambarkan struktur dari bagian-bagian yang saling terhubung
   e. *Package Diagram* menyediakan cara mengumpulkan elemen-elemen yang saling terkait dalam diagram *UML*
   f. *Deployment Diagram* untuk memodelkan penyerahan aplikasi.

2. Behavior Diagram

Kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem, terdiri dari :

a. *Usecase Diagram* untuk memodelkan proses bisnis
b. *Activity Diagram* untuk memodelkan perilaku *Usecases* dan *objects* di dalam *system.*
c. *State Diagram* untuk memodelkan perilaku *objects* di dalam sistem.

3. Interaction Diagrams

Kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada siatu sistem, terdiri dari :

a. *Sequence Diagram* untuk memodelkan pengiriman pesan (*message*) antar *objects*
b. *Communication Diagram*
c. *Timming Diagram*
d. *Interaction Overview Diagram*

Berikut akan dijelaskan empat macam diagram yang paling sering digunakan dalam pembangunan aplikasi berorientasi object, yaitu use case diagram, sequence diagram, collaboration diagram, dan class diagram.

1. Usecase Diagram

Usecase diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. *Usecase* diagram terdiri atas diagram untuk *use case* dan *aktor*. *Aktor* merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi.

Usecase merepresentasikan operasi-operasi yang dilakukan oleh aktor. *Usecase* digambarkan berbentuk elips dengan nama operasi dituliskan di

dalamnnya. *Aktor* yang melakukan operasi dihubungkan dengan garis lurus ke *usecase*.

2. Sequence Diagram

Sequence diagram menjelaskan secara detil urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari use case: interaksi yang terjadi antar class, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi.

3. Collaboration Diagram

Collaboration diagram dipakai untuk memodelkan interaksi antar objek di dalam sistem. Berbeda dengan sequence diagram yang lebih menonjolkan kronologis dari operasi-operasi yang dilakukan, collaboration diagram lebih fokus pada pemahaman atas keseluruhan operasi yang dilakukan oleh objek.

4. Class Diagram

Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. Class diagram menunjukkan hubungan antar class dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan.

Class memiliki tiga area pokok :

1. Nama (dan stereotype)
2. Atribut
3. Metoda

Atribut dan metode 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.