BAB 2
TINJAUAN PUSTAKA

2.1 Profil Tempat Kerja Praktek

Balai Besar Bahan dan Barang Teknik (B4T) beralamat di Jalan Sangkuriang 14 Bandung 40135.Telp. 022 – 2504088, 2504828, 2510682. Fax 022 – 2502027. Website :www.b4t.go.id dan e-mail : info@b4t.go.id.

2.1.1 Sejarah Instansi

Didirikan pada tahun 1909 di Batavia (Jakarta sekarang) oleh pemerintah Hindia Belanda dengan nama Laboratorium Voor Metaal Onderzoek di bawah Burgelizke Openbake Warken (Departemen PU sekarang). Tahun 1912 diperluas menjadi Laboratorium Voor Material Onderzoek Materials. Tahun 1921 dipindahkan ke Bandung di kompleks Technische Hogeschool (ITB sekarang). Tahun 1934 kedudukan balai berada di bawah Van Ekonomische (Departemen Perekonomian / Perdagangan).

Tahun 1942 di bawah kekuasaan pemerintah Jepang berubah nama menjadi Laboratorium Zeiro Sikendya dan kemudian menjadi Laboratorium Kogio Sikendya. Tahun 1945 berubah nama menjadi Balai Penyelidikan Bahan-bahan yang berkedudukan di bawah Kementerian Kemakmuran. Tahun 1952-1960 kedudukan balai beralih ke Kementerian Perekonomian dan kemudian berada dibawah Kementerian Perindustrian. Tahun 1961 menempati Jalan Sangkuriang Bandung dengan nama Balai Penelitian Bahan – Bahan.

Tahun 1963 kedudukan balai di bawah Perindustrian Rakyat. Tahun 1971 kedudukan balai di bawah Lembaga Penelitian dan Pendidikan Industri. Tahun 1974 kedudukan balai di bawah Puslitbang Industri Logam dan Mesin Departemen Perindustrian. Tahun 1980 kedudukan balai di bawah Badan Litbang Industri Departemen Perindustrian dan berubah nama menjadi Balai Besar Penelitian dan Pengembangan Industri Bahan dan Barang Teknik. Akhir tahun 2002

menjadi Balai Besar Bahan dan Barang Teknik yang disingkat B4T. Tahun 2006 menjadi Balai Besar Bahan dan Barang Teknik (SK Menteri Perindustrian No. 43/M-IND/PER/6/2006)

2.1.2 Logo B4T

B4T (Balai Besar Bahan dan Barang Teknik) mempunyai logo yang dapat dilihat pada Gambar 2.1 sebagai berikut :



Gambar 2.1 Logo Instansi B4T

2.1.3 Visi dan Misi

Adapun **visi** dari B4T ini adalah menjadi Lembaga terkemuka dalam bidang Penjaminan dan Peningkatan Mutu Bahan dan Barang Teknik yang didukung oleh Penelitian. Dan **misi** dari B4T adalah memberikan pelayanan teknis yang profesional melalui jasa Pengujian, Kalibrasi, Inspeksi Teknik dan Litbang Terapan untuk meningkatkan mutu produk dan tenaga industri yang diakui secara nasional dan internasional.

2.1.4 Badan Hukum B4T

Balai Besar Bahan dan Barang Teknik merupakan balai resmi pemerintah yang berada di bawah naungan Badan Penelitian dan Pengembangan Industri, Departemen Perindustrian.Badan hukum oleh Menteri Perindustrian Republik Indonesia.Nomor : 13 / M – IMD / PER / 6 / 2006.

2.1.5 Penghargaan

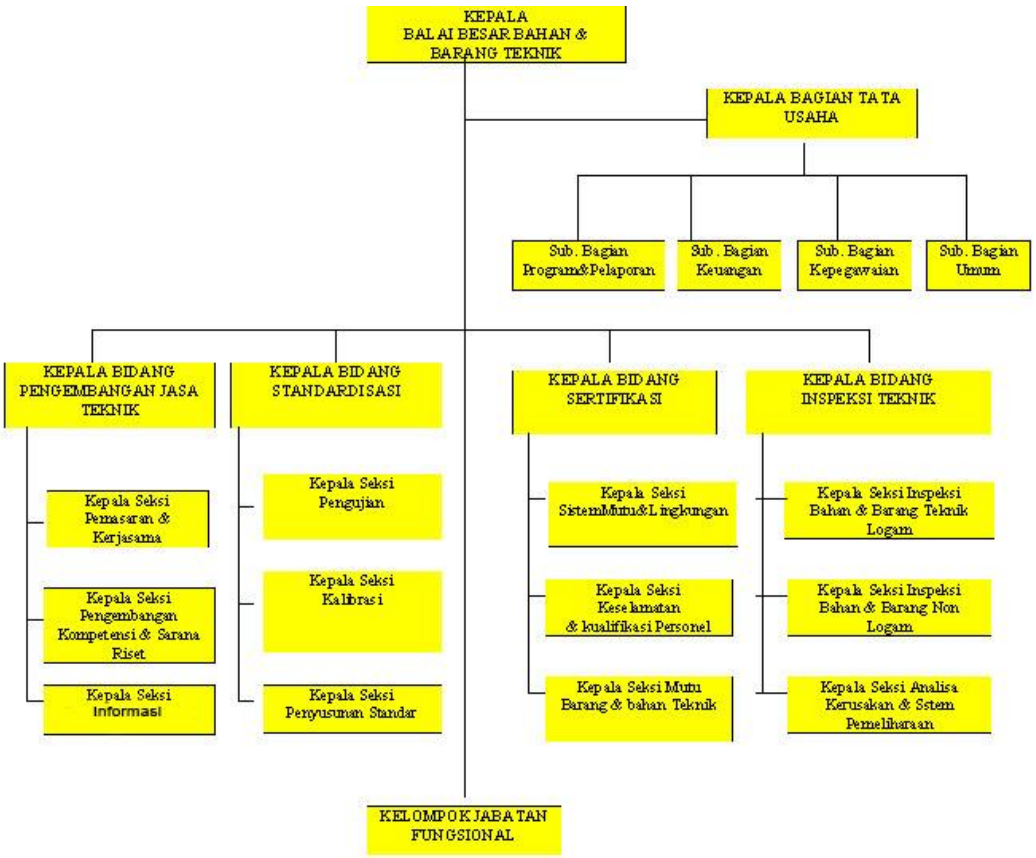
2.1 Desember 2002 mendapat Piagam penghargaan "***CITRA PELAYANAN PRIMA***" dari Menpan.

2.2 Agustus 2006

Piagam penghargaan "***UNIT PELAYANAN TERBAIK"*** dilingkungan kementrian perindustrian.

2.1.6 Stuktur Organisasi dan Job Description B4T

Berikut gambar Struktur Organisasi B4T:



Gambar 2.2 Stuktur Organisasi B4T

Balai Besar Bahan dan Barang Teknik terdiri dari :

1. Bagian TataUsaha yang terdiri dari :
    1) Sub bagian penyusun program
    2) Sub bagian kepegawaian
    3) Sub bagian keuangan
    4) Sub bagian umum
2. Bagian Pengembangan yang terdiri dari :
    1) Seksi Pemasaran dan Kerjasama
    2) Seksi Pengembangan dan Sarana Riset

3) Seksi Informasi

3. Bagian Standarisasi yang terdiri dari :
   1) Seksi Pengujian
   2) Seksi Kalibrasi
   3) Seksi Penyusunan Standar
4. Bagian Sertifikasi yang terdiri dari :
   1) Seksi Sistem Mutu Bahan dan Barang Teknik
   2) Seksi Mutu dan Lingkungan
   3) Seksi Keselamatan dan Kualifikasi Personal
5. Bagian Inspeksi Teknik yang terdiri dari :
   1) Seksi Inspeksi Bahan dan Barang Teknik Logam
   2) Seksi Nahan dan Barang Teknik Non-Logam
   3) Seksi Analisis Kerusakan dan Sistem Pemeliharaan
6. Kelompok-kelompok Jabatan Fungsional.

2.1.7 Kebijakan Mutu

Laboratorium Pengujian dan Kalibrasi Balai Besar Bahan dan Barang Teknik memiliki Kebijakan Mutu, sebagai berikut :

1. Laboratorium Balai Besar Bahan dan Barang Teknik memberikan jasa Pengujian dan Kalibrasi secara profesional dalam melayani pelanggan.
2. Laboratorium Balai Besar Bahan dan Barang Teknik menggunakan Sistem Manajemen Mutu sesuai dengan persyaratan ISO/IEC 17025:2005.
3. Laboratorium Balai Besar Bahan dan Barang Teknik memberikan pelayanan Pengujian dan Kalibrasi yang mengutamakan mutu hasil uji/kalibrasi, harga dan ketepatan waktu untuk memberikan kepuasan pelanggan.
4. Semua personil yang terlibat dalam pengujian/kalibrasi memahami dan menerapkan dokumentasi Sistem Manajemen Mutu di dalam pekerjaan sehari-hari.

5. Kegiatan pengujian/kalibrasi selalu dilaksanakan berdasarkan sistem mutu yang sesuai dengan ISO/IEC 17025:2005 dan secara berkelanjutan meningkatkan efektivitas Sistem Manajemen.

2.2 Landasan Teori

2.2.1 Konsep Dasar Framework

Dalam bahasa Inggris, *framework* berati kerangka kerja. Sementara secara harfiah, *framework* dapat diartikan sebagai sebuah tools atau alat yang berfungsi untuk membantu sebuah pekerjaan. Karena PHP adalah script untuk membuat website, maka *framework* PHP dalam kasus ini bisa diartikan sebagai alat yang berfungsi untuk mempermudah proses pembuatan sebuah website. Berbeda dengan ***Content Management System*** **(CMS)**, seperti WordPress, Joomla, dan sejenisnya.

CMS menawarkan sebuah management website yang lengkap, *end-user friendly*. Jika membangun website dengan sebuah Content Management System, maka kita tinggal menjalankan *engine* dari CMS itu saja. Kita sebagai pengembang tidak perlu lagi memikirkan proses coding, penyusunan logika, dan proses menulis kode program sendiri.

Lain soal saat kita memakai *framework* PHP. Saat kita memutuskan menggunakan *framework* sebagai pondasi proses pembuatan website, kita masih harus menulis kode, menyusun logika program. Bedanya, kode-kode yang kita tulis harus menyesuaikan dengan lingkungan *framework* yang kita gunakan. Kita harus menyesuaikan pattern-pattern (pola) yang sudah menjadi acuan dalam suatu *framework*. Ya, konsekuensi dari menggunakan *framework* adalah kita harus mempelajari lagi lingkungan pengembangan, aturan-aturan, dan pola pengkodean berdasarkan *framework* yang kita gunakan. Akan tetapi, hal tersebut dirasa wajar dan akan terbayar lunas setelah kita benar-benar menguasai dan bisa menggunakan *framework* tersebut.

Disamping menyediakan lingkungan pengembangan sendiri-sendiri, sebuah *framework* PHP juga menyediakan *library* fungsi yang disiapkan

khusus, yang bisa pengembang gunakan untuk lebih memudahkan dalam pembuatan website dan aplikasi berbasis web. Jadi sat kita menggunakan *framework*, kita akan menemukan banyak kode dan fungsi-fungsi yang tidak biasa kita lihat pada saat kita membuat program sendiri. Fungsi-fungsi tersebut bukanlah fungsi-fungsi bawaan dari PHP, melainkan fungsi-fungsi yang khusus dibuat oleh pengembang *framework* masing-masing. Jumlahnya mungkin berbeda, dan gaya programingnya mungkin berbeda, tetapi kegunaannya hampir sama, yaitu mempermudah pengembang yang menggunakan *framework* tersebut.

Fungsi-fungsi yang terdapat pada masing-masing *framework* terkadang merupakan pengembangan atau penyesuaian dari fungsi asli PHP. Tujuannya agar lebih mudah digunakan atau agar lebih sesuai dengan kebutuhan pengguna *framework*. Beberapa jenis *framework* PHP yang terkenal antara lain:

1. Yii
2. CodeIgniter
3. ZendFramework
4. CakePHP
5. Panada Framework

   Referensi : Wismakarma, Komang.2010.*9 langkah menjadi master framework Codeigniter*:Lokomedia

2.2.2 Framework CodeIgniter

CodeIgniter adalah aplikasi *open source* yang berupa *framework* dengan model MVC (*Model, View, Controller*) untuk membangun website dinamis dengan menggunakan PHP. CodeIgniter memudahkan developer untuk membuat aplikasi web dengan cepat dan mudah dibandingkan dengan membuatnya dari awal. CodeIgniter dirilis pertama kali pada 28 Februari 2006. Versi stabil terakhir 2.1.2 yang dirilis pada 29 Juni 2012.

Sebelum membahas lebih jauh tentang salah satu *framework* PHP ini, mungkin kalian ada yang masih bingung apa itu *Framework,* apa itu VMC (*Model, View, Controller*)?

Framework secara sederhana dapat diartikan kumpulan dari fungsi-fungsi/prosedur-prosedur dan class-class untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang pemrograman, tanpa harus membuat fungsi atau *class* dari awal.

Ada beberapa alasan mengapa menggunakan *Framework*:

1) Mempercepat dan mempermudah pembangunan sebuah aplikasi web.
2) Relatif memudahkan dalam proses *maintenance* karena sudah ada pola tertentu dalam sebuah *framework* (dengan syarat programmer mengikuti pola standar yang ada)
3) Umumnya *framework* menyediakan fasilitas-fasilitas yang umum dipakai sehingga kita tidak perlu membangun dari awal (misalnya validasi, ORM, *pagination, multiple database, scaffolding*, pengaturan session, *error handling*, dll
4) Lebih bebas dalam pengembangan jika dibandingkan CMS Sekarang masuk ke pengertian VMC (*Model View Controller*).

Model View Controller merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web, berawal pada bahasa pemrograman Small Talk, MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, *user interface*, dan bagian yang menjadi kontrol aplikasi. Terdapat 3 jenis komponen yang membangun suatu MVC *pattern* dalam suatu aplikasi yaitu :

1) *View*, merupakan bagian yang menangani *presentation logic*. Pada suatu aplikasi web bagian ini biasanya berupa file template HTML, yang diatur oleh *controller*. View berfungsi untuk menerima dan merepresentasikan data kepada user. Bagian ini tidak memiliki akses langsung terhadap bagian model.
2) *Model*, biasanya berhubungan langsung dengan database untuk memanipulasi data *(insert, update, delete, search*), menangani validasi

dari bagian *controller*, namun tidak dapat berhubungan langsung dengan bagian *view*.

3) *Controller*, merupakan bagian yang mengatur hubungan antara bagian model dan bagian *view*, *controller* berfungsi untuk menerima *request* dan data dari *user* kemudian menentukan apa yang akan diproses oleh aplikasi.

Dengan menggunakan prinsip MVC suatu aplikasi dapat dikembangkan sesuai dengan kemampuan developernya, yaitu programmer yang menangani bagian model dan *controller*, sedangkan designer yang menangani bagian *view*, sehingga penggunaan arsitektur MVC dapat meningkatkan *maintanability* dan organisasi kode. Walaupun demikian dibutuhkan komunikasi yang baik antara programmer dan designer dalam menangani variabel-variabel yang akan ditampilkan.

Ada beberapa kelebihan CodeIgniter (CI) dibandingkan dengan Framework PHP lain,

1) Performa sangat cepat : salah satu alasan tidak menggunakan *framework* adalah karena eksekusinya yang lebih lambat daripada PHP *from the scracth*, tapi Codeigniter sangat cepat bahkan mungkin bisa dibilang codeigniter merupakan *framework* yang paling cepat dibanding framework yang lain.
2) Konfigurasi yang sangat minim (nearly zero configuration) : tentu saja untuk menyesuaikan dengan database dan keleluasaan *routing* tetap diizinkan melakukan konfigurasi dengan mengubah beberapa file konfigurasi seperti database.php atau autoload.php, namun untuk menggunakan codeigniter dengan *setting standard*, anda hanya perlu merubah sedikit saja file pada folder config.
3) Banyak komunitas: dengan banyaknya komunitas CI ini, memudahkan kita untuk berinteraksi dengan yang lain, baik itu bertanya atau teknologi terbaru.

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

Referensi : **Wismakarma, Komang.2010.*9 langkah menjadi master framework Codeigniter*:Lokomedia**

2.2.3 Bahasa Pemrograman PHP

PHP adalah sebuah bahasa pemograman yang berjalan dalam sebuah web-server (*server side*). PHP diciptakan oleh programmer unix dan Perl yang bernama Rasmus Lerdoft pada bulan Agustus-September 1994. Pada awalnya, Rasmus mencoba menciptakan sebuah script dalam wesite pribadinya dengan tujuan untuk memonitor siapa saja yang pernah mengunjungi website-nya.

Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situs personal). Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI pada sekitar tahun 1995, dan diperkenalkan kepada beberapa programmer pemula dengan alasan bahasa yang digunakan oleh PHP cukup sederhana dan mudah dipahami. Selanjutnya Rasmus menulis ulang PHP dengan bahsa C untuk meningkatkan kecepatan aksesnya.

Mulai bulan September sampai Oktober 1995, kode PHP ditulis ulang dan digabungkan menjadi PHP/F1. Baru di akhir tahun 1995 dirilis bagi umum secara gratis. Mengapa Rasmus membagikan ke publik secara gratis ? Rasmus berangapan apabila kode PHP ini berguna bagi dirinya, tentu juga akan bermanfaat untuk oranglain. Toh pada akhirnya akan kembali bermanfaat bagi dirinya sendiri.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini, interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP dirubah menjadi akronim berulang PHP: *Hypertext Preprocessing*.

Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.

Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek. Hingga saat ini (Januari 2011) PHP sudah merilis versi 5.3.5

Yang bisa dilakukan PHP adalah Script PHP adalah bahasa program yang berjalan pada sebuah webserver, atau sering disebut server-side. Oleh karena itu,PHP dapat melakukan apa saja yang bisa dilakukan program CGI lain, yaitu mengolah data dengan tipe apapun, menciptakan halaman web yang dinamis, serta menerima dan menciptakan *cookies*, dan bahkan PHP bisa melakukan lebih dari itu. Arti script server-side adalah, agar dapat menjalankan script ini dibutuhkan tiga program utama, yaitu web-server (dapat berupa IIS dari windows atau apache), modul PHP dan juga web browser. PHP dapat berjalan pada semua jenis system operasi, antara lain pada Linux dan varian Unix (HP-UX, Solaris dan OpenBSD), pada Ms Windows, Mac dan masih banyak lag, selain itu PHP juga dapat berjalan pada beberapa jenis web-server antara lain Apache, Microsoft IIS, personal webserver, Netscape dan Iplanet Server, Caudium, Xitami, Omnihttpd dan masih banyak lagi. Apabila Anda menggunakan PHP, maka Anda banyak memiliki alernatif Sistem Operasi atau webserver untuk menjalankannya, selain itu Anda juga dapat menajalankan atau menggunakan program

berorientasi object atau sering disebut *Object Oriented Programming* (OOP). Output yang dihasilkan PHP bukan hanya HTML, namun juga dalam bentuk gambar, file PDF, serta gambar animasi menggunakan Libswf dan Ming. Output yang lain dengan jenis teks dapat berupa file XHTML dan XML. PHP juga banyak mendukung database populer yang sering digunakan dalam beberapa web server, antara lain Adabas D, Empress, FrontBase, PostgreSQL FIlePro(read only), mSQL, Solid, Hyperwave, Direct MS-SQL, Sybase, IBM DB2, MySQL, Velocis, Informix, ODBC dan Unix dbm. PHP juga mendukung koneksi denganlayanan jasa yang menggunakan protokol seperti LDAP, IMAP, SNMP, NNTP, POP3, HTTP dan COM (pada windows). Selain itu, juga dapat melakukan koneksi dengan jaringan yang menggunakan protokol lain, PHP mempunyai dukungan WDDX, sehingga dapat bertukar data antar bahasa pemograman web.

System kerja dari PHP diawali dengan permintaan yang beasal dari halaman website oleh browser. Berdasarkan URL atau alamat website dalam jaringan internet, browser akan menemukan sebuah alamat dari webserver, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh webserver. Selanjutnya webserver akan mencarikan berkas yang diminta dan menampilkan isinya di browser. Browser yang mendapatkan isinya segera menerjemahkan kode HTML dan menampilkannya.

Referensi : **http://agiptek.com/index.php/php/101-php.html**

2.2.4 **DataBase**

Database adalah suatu kumpulan data-data yang disusun sedemikian rupa sehingga membentuk informasi yang sangat berguna. Database terbentuk dari sekelompok data-data yang memiliki jenis/sifat sama. Ambil contoh, data-data berupa nama-nama, kelas-kelas, alamat-alamat. Semua data tersebut dikumpulkan menjadi satu menjadi kelompok data baru, sebut saja sebagai data-data mahasiswa. Demikian juga, kumpulan dari data-data

mahasiswa, data-data dosen, data-data keuangan dan lainnya dapat dikumpulkan lagi menjadi kelompok besar, misalkan data-data politeknik elektronika. Bahkan dalam perkembangannya, data-data tersebut dapat berbentuk berbagai macam data, misalkan dapat berupa program, lembaran-lembaran untuk *entry* (memasukkan) data, laporan-laporan. Kesemuanya itu dapat dikumpulkan menjadi satu yang disebut dengan database.

Data secara umum dapat dikatakan sebagai segala sesuatu yang dapat dikumpulkan. Tentu saja hal ini akan membuat segala sesuatu di dunia ini menjadi data, dan masing masing dapat dikumpulkan menurut jenisnya. Segala bentuk catatan mengenai data-data tersebut sebenarnya dapat dianggap sebagai database (tempat kumpulan data-data). Biasanya catatan dari data-data tersebut dilakukan dengan relatif sederhana dan dilakukan dengan cara manual (dicatat di atas lembaran-lembaran kertas, atau paling tidak diketik menggunakan program aplikasi tertentu). Setelah data-data tersebut dikumpulkan, biasanya diperlukan untuk pembuatan laporan, pengambilan keputusan atau segala sesuatu bentuk pengolahan yang berhubungan dengan data tersebut.

Jika data-data tersebut tercatat secara manual, maka segala bentuk pengolahan juga dilakukan secara manual (disusun, dihitung atau dibuat laporannya secara manual). Cara ini tentu saja membutuhkan ekstra tenaga dan waktu. Dan lebih sering lagi, diperlukan pengumpulan data-data yang sejenis secara berkali-kali dan dilakukan juga pengolahan dan pembuatan laporan secara berkali-kali pula. Bisa dibayangkan ini merupakan pekerjaan yang sangat membosankan.

Dari kenyataan tersebut, akan lebih mudah jika dibuat suatu sistem yang digunakan untuk menyimpan data-data tersebut secara lebih terorganisasi, dan dengan bantuan program-program aplikasi tertentu, data-data tersebut dapat diolah dan dibuat laporannya secara lebih cepat dan lebih mudah. Hal inilah yang menjadikan perlunya dibuat sistem database.

Meskipun sebenarnya tujuan dari database tersebut sama, yaitu lebih mempermudah dalam pengolahan data, namun caranya ada berbagai macam.

Macam dari database tersebut dapat dilihat dari bentuk konfigurasi sistemnya atau dari bentuk/isi dari database tersebut.

Ada beberapa jenis dari database, mulai dari yang menggunakan text biasa, menggunakan exel, lotus, foxpro, dbase, paradoc, access, oracle, SQL dan banyak lagi. Masing-masing dapat berbeda dari sisi format datanya, fasilitas yang disediakan dan teknik pengolah databasenya (database engine).

Referensi : **lecturer.eepis-its.edu/~eru/artikel/Database1/DataBase.DOC**

2.2.5 OOP

Pemrograman berorientasi objek (*object oriented programming* disingkat OOP) merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya,

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat. Konsep dasar dari Pemrograman Berorientasi Objek Pemrograman orientasi-objek menekankan konsep berikut:

1) Kelas Adalah kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh 'class of dog' adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi objek. Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun terkait dengan domain permasalahan yang ada, dan kode yang

terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak menggunakan OOP). Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.

2) Objek Adalah membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer; objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.
3) Abstraksi Adalah Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.
4) Enkapsulasi Adalah Memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek tersebut yang diberi izin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.
5) Polimorfisme melalui pengiriman pesan. Tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan; metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesa

tersebut dikirim. Contohnya, bila sebuah burung menerima pesan "gerak cepat", dia akan menggerakan sayapnya dan terbang. Bila seekor singa menerima pesan yang sama, dia akan menggerakkan kakinya dan berlari. Keduanya menjawab sebuah pesan yang sama, namun yang sesuai dengan kemampuan hewan tersebut. Ini disebut polimorfisme karena sebuah variabel tungal dalam program dapat memegang berbagai jenis objek yang berbeda selagi program berjalan, dan teks program yang sama dapat memanggil beberapa metode yang berbeda di saat yang berbeda dalam pemanggilan yang sama. Hal ini berlawanan dengan bahasa fungsional yang mencapai polimorfisme melalui penggunaan fungsi kelas-pertama.

6) Dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Sebagai contoh anggap kita memiliki sebuah departemen yang memiliki manager, sekretaris, petugas administrasi data dan lainnya. Misal manager tersebut ingin memperoleh data dari bag administrasi maka manager tersebut tidak harus mengambilnya langsung tetapi dapat menyuruh petugas bag administrasi untuk mengambilnya. Pada kasus tersebut seorang manager tidak harus mengetahui bagaimana cara mengambil data tersebut tetapi manager bisa mendapatkan data tersebut melalui objek petugas adminiistrasi. Jadi untuk menyelesaikan suatu masalah dengan kolaborasi antar objek-objek yang ada karena setiap objek memiliki deskripsi tugasnya sendiri.

Referensi : http://id.wikipedia.org/wiki/Pemrograman_berorientasi_objek

2.2.6 MySQL

MySQL adalah sebuah perangkat lunak Pembuat database yang bersifat terbuka atau open source dan berjalan disemua platform baik Linux

maupun Si Windows, MySQL merupakan program pengakses database yang bersifat network sehingga dapat digunakan untuk aplikasi Multi User (Pengguna Banyak).

MySQL adalah Relational Database Management System (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL dapat dikatakan lebih unggul dibandingkan database server lainnya dalam query data. Hal ini terbukti untuk query yang dilakukan oleh single user, kecepatan query MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase.

Referensi : **http://www.maniacms.web.id/2012/01/pengertian-mysql.html**

2.2.7 Unified Modelling Language

2.2.7.1 Pengertian UML

Unified Modeling Language (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat tool untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan oleh *Object Management Group*, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an.[1] Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP. UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari IBM.

UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat.Ini merupakan standar terbuka yang menjadikannya sebagai bahasa

pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem.

Sampai era tahun 1990 puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah: Metodologi Booch, Metodologi Coad, Metodologi OOSE, Metodologi OMT, Metodologi Shlaer-Mellor, Metodologi Wirfs-Brock, dsb. Masa itu terkenal dengan masa perang metodologi (*method war*) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan kelompok/perusahaan lain yang menggunakan metodologi yang berlainan.

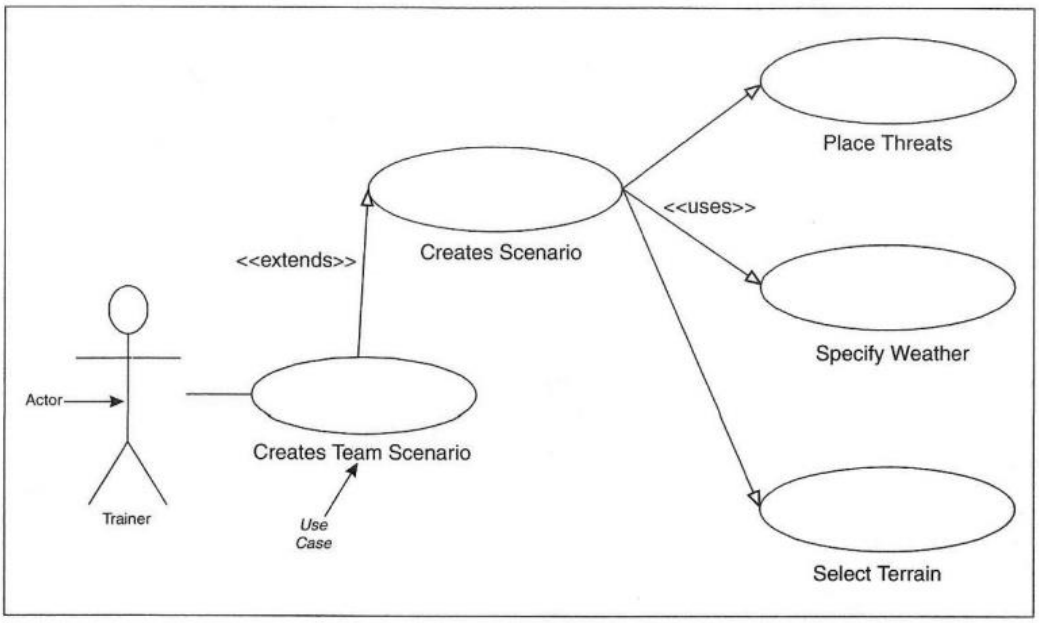
Dimulai pada bulan Oktober 1994 Booch, Rumbaugh dan Jacobson, yang merupakan tiga tokoh yang boleh dikata metodologinya banyak digunakan mempelopori usaha untuk penyatuan metodologi pendesainan berorientasi objek.[3] Pada tahun 1995 direlease draft pertama dari UML (versi 0.8)[3]. Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh *Object Management Group* (OMG)

Referensi : **http://id.wikipedia.org/wiki/Unified_Modeling_Language**

2.2.7.2 Jenis-jenis Notasi UML

1) Diagram Use Case

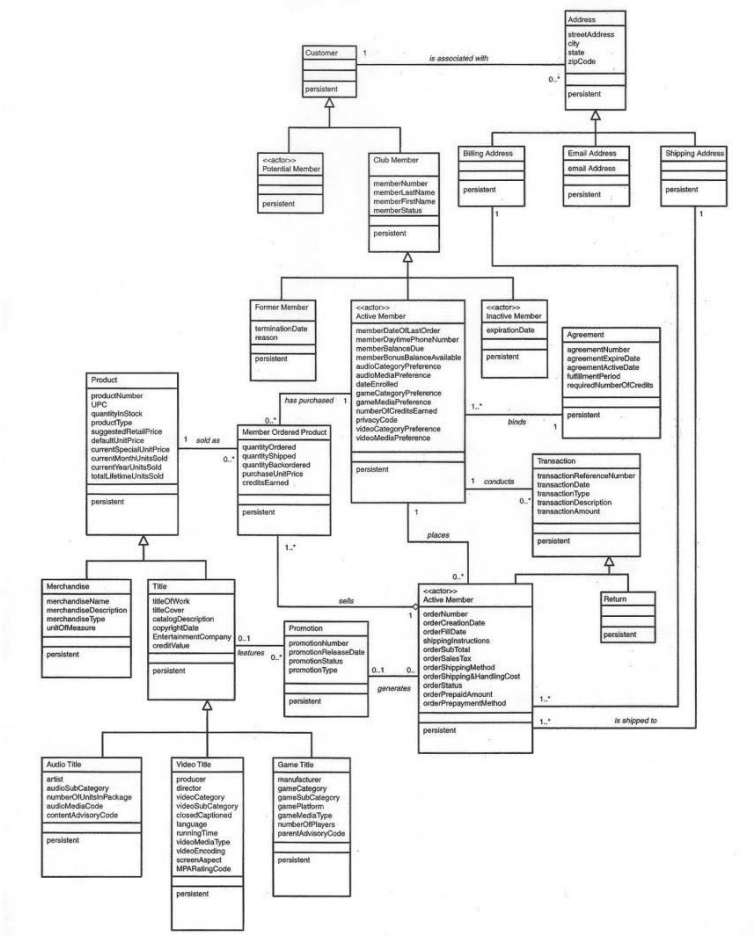
Use case adalah deskripsi fungsi dari sebuah dari sudut pandang pengguna. Use case bekerja dengan cara mendeskripsikan tipikal interkasi antar user (pengguna) sebuah system dengan system itu sendiri dan menjelaskan bagaimana system itu bekerja.



Gambar 2.3 Use Case Diagram

2) Diagram Class

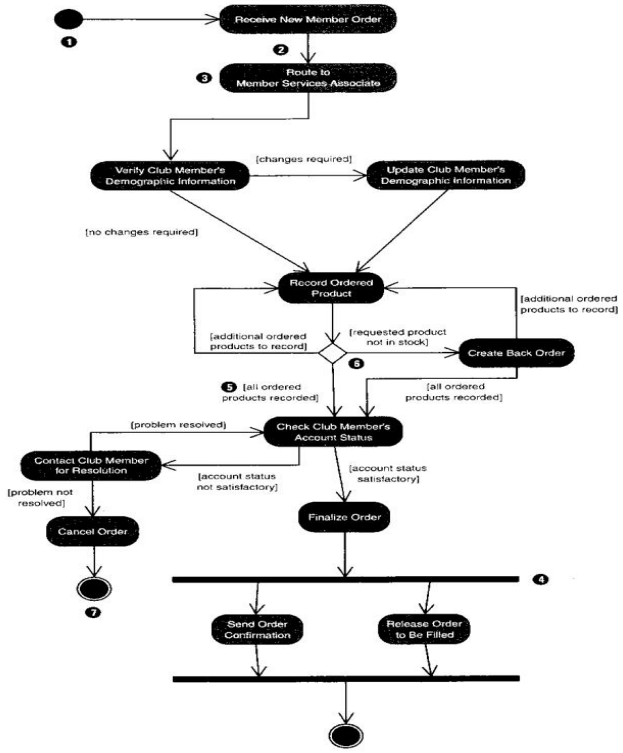
Class diagram adalah sebuah spesifikasi yang jika diinstansiasi maka akan menghasilkan objek yang merupakan inti dari pengembangan dan desain berorientasi objek. Kelas menggambarkan atribut atau properti dari sebuah system sekaligus menawarkan layanan apa saja yang bisa dilakukan dengan objek tersebut (method/fungsi). Jadi, kelas memiliki 3 pokok penting yaitu: nama, atribut dan method.



Gambar 2.4 Class Diagram

3) Diagram Activity

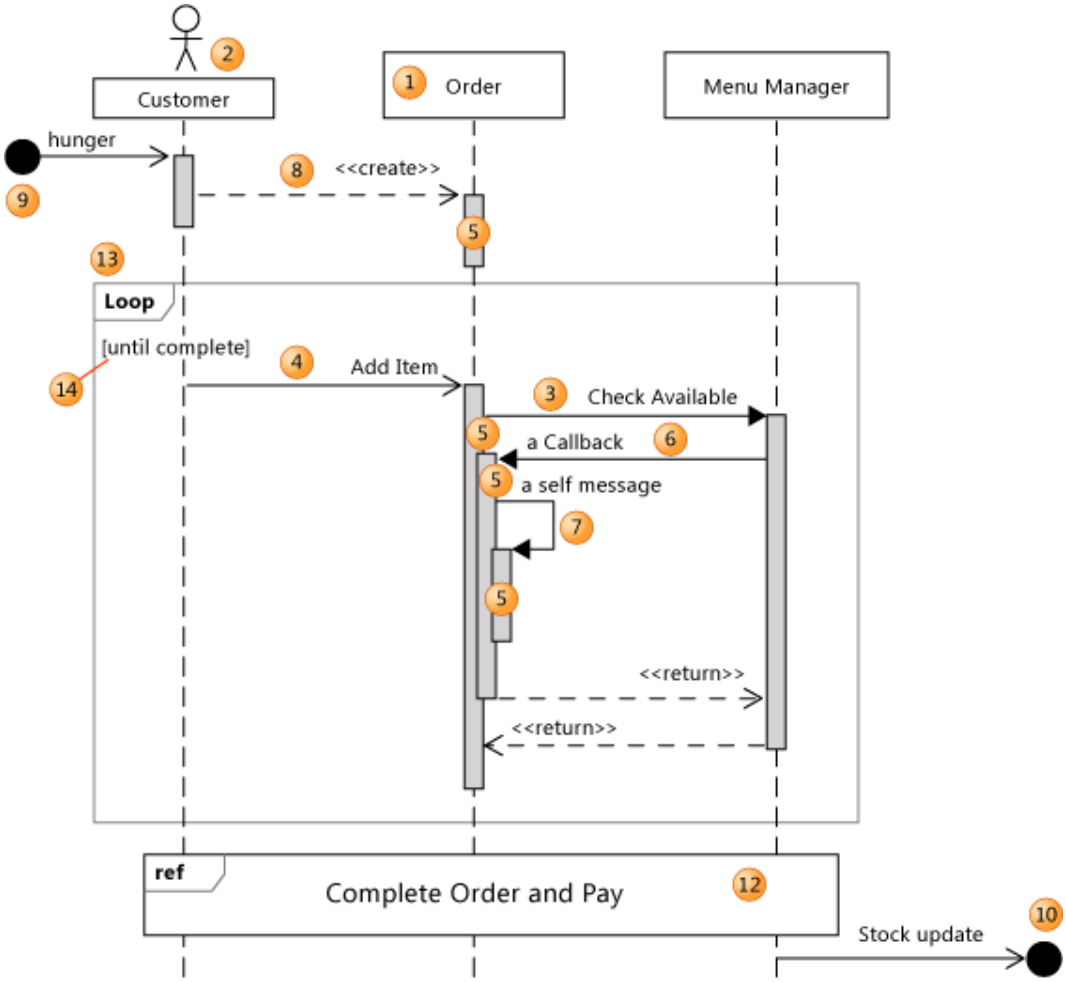
Activity diagram menggambarkan berbagai alir aktifitas dalam system yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses parallel yang mungkin erjadi pada beberapa eksekusi.



Gambar 2.5 Activity Diagram

4) Diagram Sequence

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah sekenario. Diagram ini menunjukkan sejumlah contoh objek dan message (pesan) yang diletakkan di antara objek-objek ini di dalam use case.



Gambar 2.6 Diagram Sequence

Referensi : **http://cookies.web.id/2012/10/apa-itu-pengertian-uml.html**