BAB II
TINJAUAN PUSTAKA

2.1 Profil Tempat Kerja Praktek

2.1.1 Sejarah Instansi

B4T didirikan pada tahun 1909 di Batavia (Jakarta sekarang) oleh pemerintah Hindia-Belanda dengan nama *Laboratorium Voor Metal Onderzoek* dan berada di bawah naungan *Burgelizke Openbake Warken* (Departemen Pekerjaan Umum sekarang).

Pada tahun 1912 diperluas menjadi *Laboratorium Voor Material Onderzoek*. Yang semula balai ini hanya mengurus benda-benda metal diperluas dengan benda material lainnya.

Pada tahun 1921 lokasi balai dipindahkan ke Bandung. Tepatnya ke lokasi *Technische Hogeschool* (Institut Teknologi Bandung sekarang). Tahun 1934 kedudukan balai berpindah ke naungan *Van Ekonomische* (Departemen Perekonomian dan Perdagangan).

Pada tahun 1942 di bawah kekuasaan pemerintah Jepang balai berubah nama menjadi *Laboratorium Zeiro Sikendya* dan kemudian berubah lagi menjadi *Laboratorium Kogio Sikendya*.

Tahun 1945 Indonesia merdeka dan balai ini kembali berubah nama menjadi Balai Penyelidikan Bahan-bahan yang berkududukan di bawah Kementrian Kemakmuran. Pada periode tahun 1952 – 1960 balai ini dua kali beralih naungan, yaitu Kementrian Perekonomian ke Kementrian Perindustrian. Pada tahun 1961 balai ini menempati Jalan Sangkuringan Bandung dengan nama Balai Penelitian Bahan-bahan. Pada tahun 1963 kedudukan balai berada di bawah naungan Perindustrian Rakyat.

Tahun 1971 kedudukan balai berada di bawah naungan Lembaga Penelitian dan Pendidikan Industri, pada tahun 1974 kedudukan balai berada 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 dan Barang teknik. Pada akhir tahun 2002 balai ini berubah nama menjadi Balai Besar Bahan dan Barang Teknik yang disingkat dengan B4T yang dipakai hingga sekarang.

Balai besar Bahan dan Barang Teknik (B4T) beralamat di Jalan Sangkuriang No. 14 Bandung 40135. Telp. 022 – 2504088, 2504828, 2510682. Fax 022 – 2502027. Website : www.b4t.go.id dan e-mail : info@b4t.go.id.

2.1.2 Logo Instansi



Gambar 2.1 Logo Instansi B4T

2.1.3 Visi dan Misi Instansi

Visi

Menjadi Lembaga Terkemuka dalam bidang Penjamin dan Peningkatan Mutu Bahan dan Barang Teknik yang didukung oleh penelitian.

Misi

Memberikan pelayanan teknis yang profesional melalui jasa Serifikasi untuk meningkatkan mutu produk dan tenaga industri yang diakui nasional maupun internasional.

2.1.4 Badan Hukum Instansi

Balai besar Bahan dan Barang Teknik (B4T) merupakan balai resmi pemerintah yang berada di bawah naungan Badan Pengkajian Kebijakan Iklim dan Mutu Industri, Kementrian Perindustrian. Badan hukum oleh Menteri Perindustrian Republik Indonesia Nomor : 43/M-IND/PER/6/2006 Tentang organisasi dan tata kerja Balai Besar Bahan dan Barang Teknik.

Berdasarkan Surat Keputusan Menteri Perindustrian No.43/M-IND/PER/6/2006 tanggal 29 Juni 2006 Tentang Organisasi dan Tata Kerja Balai Besar Bahan dan Barang Teknik, pada bab II pasal 4 dijelaskan bahwa B4T terdiri dari :

1. Bagian Tata Usaha.
2. Bidang Pengembangan Jasa Teknik.
3. Bidang Standardisasi.
4. Bidang Sertifikasi.
5. Bidang Infeksi Teknik.
6. Kelompok Jabatan Fungsional.

Adapun seksi teknologi informasi ada dibawah bidang pengembangan jasa teknik sebagaimana dijelaskan pada pasal 9 dengan rincian tugasnya terdapat dipasal 10c yaitu melakukan penyiapan bahan pengelolaan, pengembangan dan pemanfaatan teknologi informasi dan perpustakaan.

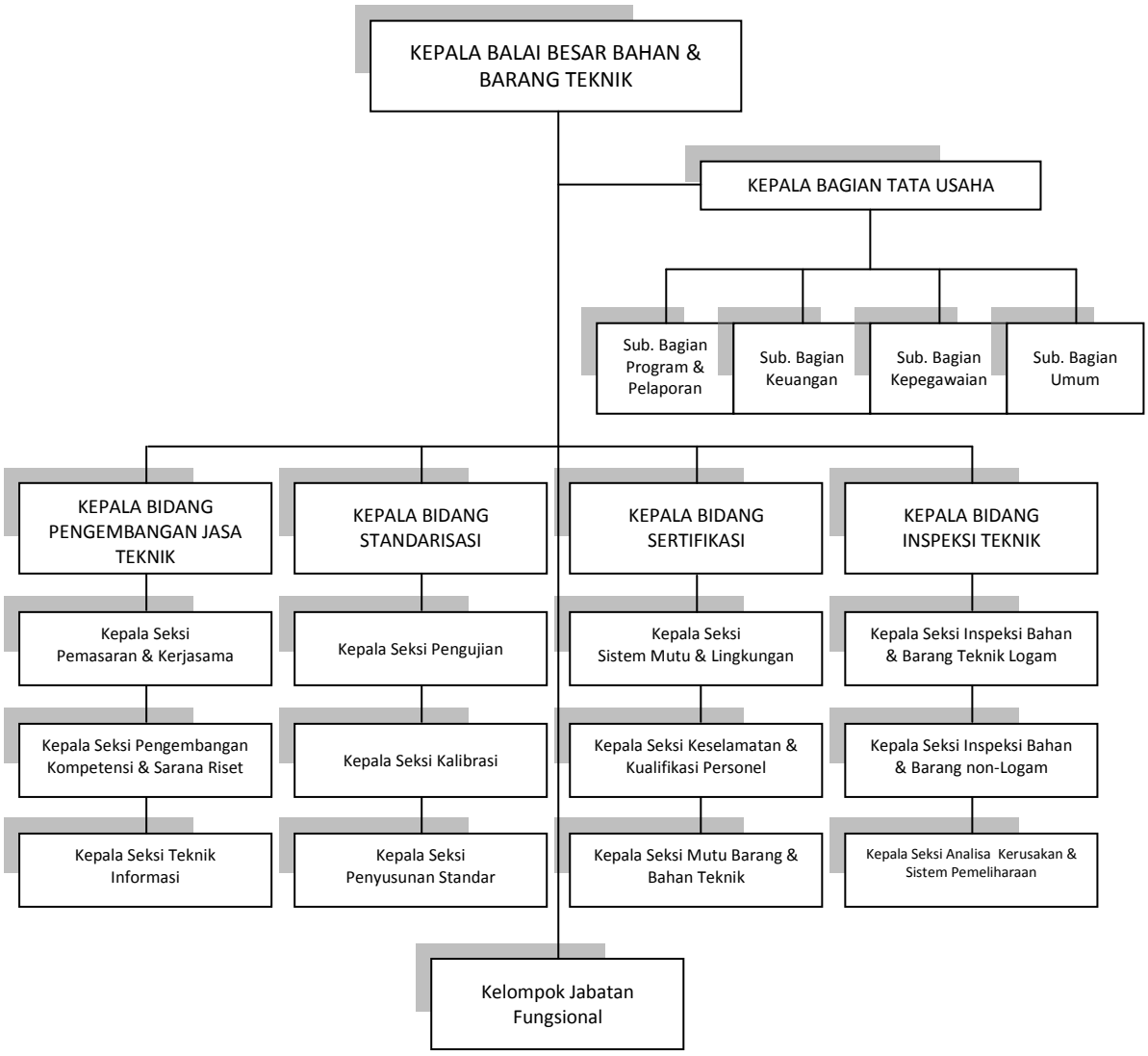
2.1.5 Bidang Pekerjaan Perusahaan

Balai Besar Bahan dan Barang Teknik (B4T) menyiapkan layanan teknik jasa bagi industri yang didukung peralatan lodern dan handal, sumber daya manusia yang terlatih dan berkualifikasi, laburatorium kalibrasi. Lembaga inspeksi dan lembaga sertifikasi serta lembaga pelatihan teknik yang terakreditasi nasional maupun internasional.

Ruang lingkup kegiatan B4T sebagai mitra industri meliputi :

1. Pengujian Bahan
2. Pengujian Barang Teknik
3. Kalibrasi
4. Inspeksi Teknik
5. Pelatihan Teknik
6. Serifikasi Sistem Manajemen Mutu
7. Serifikasi Produk
8. Setifikasi Personil
9. Serifikasi Manajemen Lingkungan
10. Penyusunan Standar
11. Konsultasi Teknik
12. Penelitian dan Pengembangan

2.1.6 Struktur Organisasi



Gambar 2.2 Struktur Organisasi

2.2 Landasan Teori

Selama pelaksanaan kerja praktek di Balai Besar Bahan dan Barang Teknik (B4T), kelompok kerja praktek menggunakan pengetahuan yang diperoleh selama masa perkuliahan sebagai landasan teori dalam pengembangan sistem intranet di B4T dan beberapa pengetahuan baru yang didapat pada saat pelaksanaan kerja praktek berlangsung.

2.2.1 Intranet

Intranet adalah sebuah jaringan komputer berbasis protokol TCP/IP seperti internet hanya saja digunakan dalam internal perusahaan, kantor, bahkan warung internet (WARNET) pun dapat di kategorikan Intranet. Antar Intranet dapat saling berkomunikasi satu dengan lainnya melalui sambungan Internet yang memberikan tulang punggung komunikasi jarak jauh. Akan tetapi sebetulnya sebuah Intranet tidak perlu sambungan luar ke Internet untuk berfungsi secara benar. Intranet menggunakan semua protocol TCP/IP dan aplikasinya sehingga kita memiliki "*private*" Internet.

2.2.2 *Framework*

Tidak ada definisi pasti dari framework, tapi dapat ditarik kesimpulan bahwa framework adalah sekumpulan class dan library yang terpadu sehingga memudahkan menyelesaikan permasalahan secara menyeluruh. Efeknya waktu untuk membuat program menjadi lebih singkat. Beberapa contoh framework di PHP adalah CakePHP, Zend Framework, CodeIgniter dan Symfony.

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.

2.2.3 *Trigger MySql*

Trigger dalam database diartikan sebagai *"procedural code that is automatically executed in response to certain events on a particular table or view in a database"*. Singkatnya, *trigger* merupakan sekumpulan perintah atau sintaks yang akan secara otomatis dijalankan jika terjadi operasi tertentu dalam tabel atau *view*. *Trigger* digunakan untuk memanggil satu atau beberapa perintah SQL secara otomatis sebelum atau sesudah terjadi proses *INSERT*, *UPDATE* atau *DELETE* dari suatu tabel. Sebagai contoh misalnya kita ingin menyimpan id pelanggan secara otomatis ke tabel 'log' sebelum menghapus data di tabel pelanggan.

Di MySQL, *Triggers* mulai dikenal di versi MySQL 5.0, dan di versi saat ini (5.1.4) fungsionalitasnya sudah bertambah. Pada versi selanjutnya pihak pengembang MySQL berjanji akan lebih menguatkan (menambah) fitur trigger ini. Trigger sering digunakan, antara lain untuk :

1. Melakukan update data otomatis jika terjadi perubahan. Contohnya adalah dalam sistem penjualan, jika dientri barang baru maka stock akan bertambah secara otomatis.
2. Trigger dapat digunakan untuk mengimplementasikan suatu sistem log. Setiap terjadi perubahan, secara otomatis akan menyimpan ke tabel log.
3. Trigger dapat digunakan untuk melakukan validasi dan verifikasi data sebelum data tersebut disimpan.

2.2.4 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, images, dan style lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas.

Pada umumnya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML. CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran border, warna border, warna hyperlink, warna mouse over, spasi antar 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 format yang berbeda.

2.2.5 Code Igniter

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. 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 programer, tanpa harus membuat fungsi atau *class* dari awal.

CodeIgniter (ellisLab) memiliki karakteristik ringan dan fleksible. *Framework* yang didukung komunitas dan perusahaan ini terkenal dengan kemudahan penggunaannya karena semua tata struktur penulisan scripting

diserahkan sepenuhnya pada developer. *CodeIgniter* terasa ideal pada website dengan kompleksitas ringan.

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.
5) dll.

2.2.6 UML (*Unified Modelling Language*) *Object Oriented Analysis & Design*

2.2.6.1 Konsep Objek

Objek dalam *'software analysis & design'* adalah sesuatu berupa konsep (*concept*), benda (*thing*), dan sesuatu yang membedakannya dengan lingkungannya. Secara sederhana objek adalah mobil, manusia, *alarm* dan lain-lainnya. Tapi objek dapat pula merupakan sesuatu yang abstrak yang hidup didalam sistem seperti tabel, *database, event, system messages.*

Objek dikenali dari keadaannya dan juga operasinya. Sebagai contoh sebuah mobil dikenali dari warnanya, bentuknya, sedangkan manusia dari suaranya. Ciriciri ini yang akan membedakan objek tersebut dari objek lainnya. Alasan mengapa saat ini pendekatan dalam pengembangan software dengan *object-oriented*, pertama adalah *scalability* dimana objek lebih mudah dipakai untuk menggambarkan sistem yang besar dan komplek. Kedua *dynamic modeling*, adalah dapat dipakai untuk permodelan sistem dinamis dan *real time*.

2.2.6.2 Definisi *Unified Modeling Language* (UML)

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

1. Menurut (Hend, 2006) "*Unified Modeling Language* (UML) adalah bahasa yang telah menjadi standard untuk visualisasi, menetapkan, membangun dan mendokumentasikan artifak suatu sistem perangkat lunak".
2. Menurut (Adi Nugroho : 2005). "*Unified Modeling Language* (UML) adalah alat bantu analisis serta perancangan perangkat lunak berbasis objek".
3. Menurut (Joomla dari http://soetrasoft.com : 2007). *"Unified Modeling Language* (UML) merupakan standard modeling language yang terdiri dari kumpulan-kumpulan diagram, dikembangkan untuk membantu para pengembang sistem dan *software* agar bisa menyelesaikan tugas-tugas seperti: Spesifikasi, Visualisasi, Desain Arsitektur, Konstruksi, Simulasi dan testing serta Dokumentasi".

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

2.2.6.3 Bangunan Dasar Metodologi *Unified Modeling Language* (UML)

Menurut (Adi Nugroho : 2005). "Bangunan dasar metodologi *Unified Modeling Language* (UML) menggunakan tiga bangunan dasar untuk mendeskripsikan sistem/perangkat lunak yang akan dikembangkan yaitu :

1. Sesuatu *(things)*

   Ada 4 (empat) *things* dalam *Unified Modeling Language* (UML), yaitu:

   a. *Structural things*

      Merupakan bagian yang relatif statis dalam model *Unified Modeling Language* (UML). Bagian yang relatif statis dapat berupa elemen-*elemen* yang bersifat fisik maupun konseptual.

   b. *Behavioral things*

      Merupakan bagian yang dinamis pada model *Unified Modeling Language* (UML), biasanya merupakan kata kerja dari model *Unified Modeling Language* (UML), yang mencerminkan perilaku sepanjang ruang dan waktu.

   c. *Grouping things*

      Merupakan bagian pengorganisasi dalam *Unified Modeling Language* (UML). Dalam penggambaran model yang rumit kadang *diperlukan* penggambaran paket yang menyederhanakan model. Paket-paket ini kemudian dapat didekomposisi lebih lanjut. Paket berguna bagi pengelompokkan sesuatu, misalnya model-model dan subsistem-subsistem.

   d. *Annotational things*

      Merupakan bagian yang memperjelas model *Unified Modeling Language* (UML) dan dapat berupa komentar-komentar yang menjelaskan fungsi serta ciri-ciri setiap elemen dalam model *Unified Modeling Language* (UML).

2. Relasi *(Relationship)*

   Ada 4 (empat) macam *relationship* dalam *Unified Modeling Language* (UML), yaitu :

a. Kebergantungan

Merupakan hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(*independent*)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri *(independent).*

b. Asosiasi

Merupakan apa yang menghubungkan antara objek satu dengan objek lainnya, bagaimana hubungan suatu objek dengan objek lainnya. Suatu bentuk asosiasi adalah agregasi yang menampilkan hubungan suatu objek dengan bagian-bagiannya.

c. Generalisasi

Merupakan hubungan dimana objek anak (*descendent)* berbagi perilaku dan struktur data dari objek yang ada diatasnya objek induk (*ancestor)*. Arah dari atas kebawah dari objek induk ke objek anak dinamakan spesialisasi, sedangkan arah berlawanan sebaliknya dari arah bawah keatas dinamakan generalisasi.

d. Realisasi

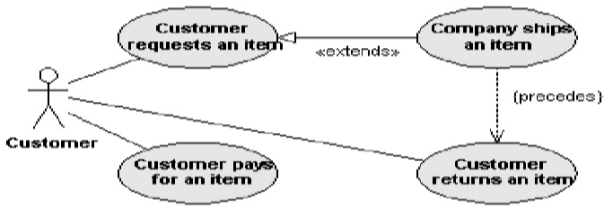
Merupakan operasi yang benar-benar dilakukan oleh suatu objek.

3. Diagram

Ada 5 (lima) macam diagram dalam *Unified Modeling Language* (UML), yaitu :

a. *Use Case Diagram*

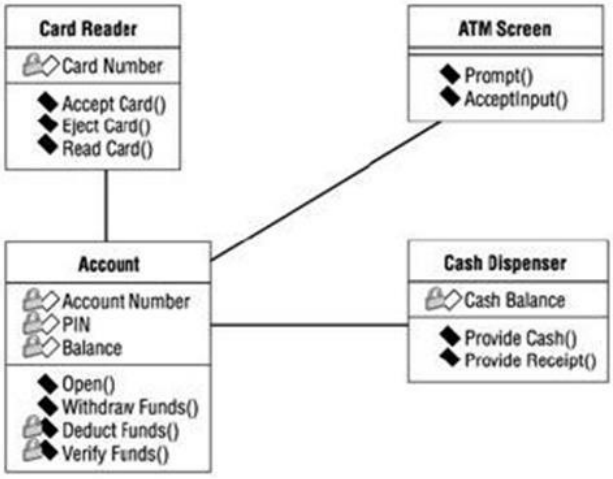
Diagram ini memperihatkan himpunan use case dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna.



Gambar 2.3 Contoh use case diagram

b. *Class Diagram*

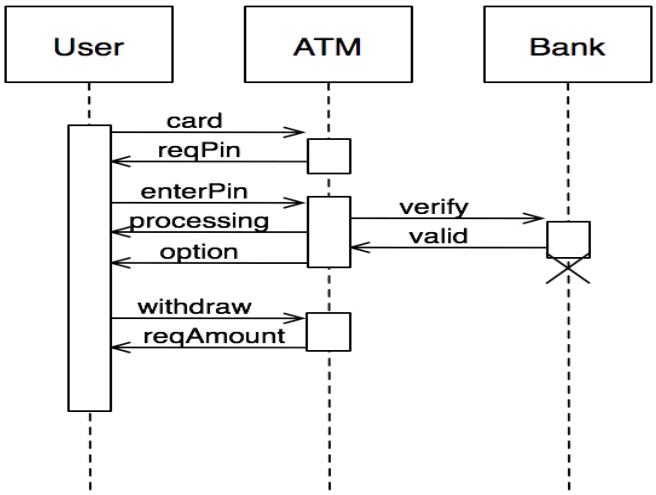
Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi dan relasi-relasi antar objek.



Gambar 2.4 Contoh class diagram

c. *Sequence Diagram*

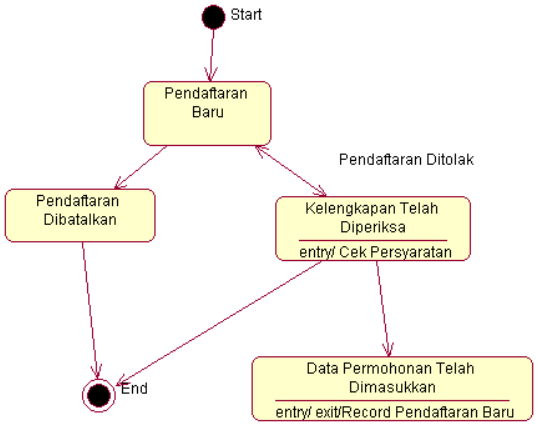
Diagram ini memperlihatkan interaksi yang menekankan pada pengiriman pesan *(message)* dalam suatu waktu tertentu.



Gambar 2.5 Contoh Sequence diagram

d. *State Chart Diagram*

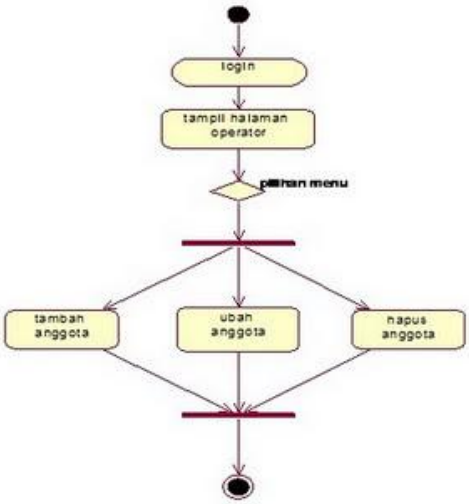
Diagram ini memperlihatkan state-state pada sistem, memuat state, transisi, event, dan aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka, kelas, kolaborasi dan terutama penting pada pemodelan sistem-sistem yang reaktif.



Gambar 2.6 Contoh Statechart diagram

e. *Activity Diagram*

Diagram ini memperlihatkan aliaran dari suatu aktifitas ke aktifitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek.



Gambar 2.7 Contoh Activity diagram