BAB II
TINJAUAN PUSTAKA

2. 1 Profil Bank Indonesia

Pembangunan aplikasi pengolahan data pengobatan ini dilakukan di Bank Indonesia cabang Bandung. Aplikasi yang dikembangkan digunakan oleh pegawai staf Bank Indonesia pada bagian/divisi Sumber Daya Manusia (SDM).

2. 1. 1 Sejarah Indonesia

Bank Indonesia pertama kali berdiri pada tahun 1828 dengan nama De Javasche Bank yang didirikan oleh Pemerintah Hindia Belanda sebagai bank sirkulasiyang bertugas mencetak dan mengedarkan uang.Undang-Undang Pokok Bank Indonesia tahun 1953 menetapkan pendirian Bank Indonesiauntuk menggantikan fungsi De Javasche Bank sebagai bank sentral, dengan tigatugas utama di bidang moneter, perbankan, dan sistem pembayaran. Di sampingitu, Bank Indonesia diberi tugas penting lain dalam hubungannya denganPemerintah dan melanjutkan fungsi bank komersial yang dilakukan oleh DJBsebelumnya.

Undang-Undang Bank Sentral tahun 1968 mengatur kedudukan dan tugas Bank Indonesiasebagai bank sentral, terpisah dari bank-bank lain yang melakukan fungsikomersial. Selain tiga tugas pokok bank sentral, Bank Indonesia juga bertugasmembantu Pemerintah sebagai agen pembangunan mendorong kelancaranproduksi dan pembangunan serta memperluas kesempatan kerja gunameningkatkan taraf hidup rakyat.

Tahun 1999 merupakan babak baru dalam sejarah Bank Indonesia, sesuai dengan UU No.23/1999 yangmenetapkan tujuan tunggal Bank Indonesia yaitu mencapai dan memeliharakestabilan nilai rupiah.Pemerintah mengeluarkan Peraturan Pemerintah Pengganti Undang-Undang (PerPPU) No.2 tahun 2008 tentang Perubahan Kedua atas Undang-Undang No.23 tahun 1999 tentang Bank Indonesia sebagai bagian dari upaya menjaga stabilitas sistem keuangan. Amandemen dimaksudkan untuk meningkatkan ketahanan perbankan nasional

dalam menghadapi krisis global melalui peningkatan aksesperbankan terhadap Fasilitas Pembiayaan Jangka Pendek dari Bank Indonesia[1].

2. 1. 2 Logo Bank Indonesia

Berdasarkan surat edaran intern No.11/40/INTERN tanggal 29 Juni 2009, logo ataupun lambang dari Bank Indonesia adalah seperti yang terdapat pada Gambar 2.1 :



Gambar II.1Logo Bank Indonesia

Sejarah perkembangan logo BI yang telah mengalami perubahan 7 kali sejak tahun 1953 hingga 2005.Logo BI yang sekarang ini ternyata mengadaptasi logo De *Javasche* Bank dengan mengubah huruf "J" menjadi huruf "I". Makna dari warna biru pada logo BI adalah keutuhan langit dan laut kepulauan nusantara yang menyatukan Negara Kesatuan Republik Indonesia dan menyiratkan kesatuan dan persatuan [3].

2. 1. 3 Badan Hukum Bank Indonesia

Babak baru dalam sejarah Bank Indonesia sebagai Bank Sentral yang independen dimulai ketika sebuah undang-undang baru, yaitu UU No. 23/1999 tentang Bank Indonesia, dinyatakan berlaku pada tanggal 17 Mei 1999. Undang-undang ini memberikan status dan kedudukan sebagai suatu lembaga negara independen dan bebas dari campur tangan pemerintah ataupun pihak lainnya.

Sebagai suatu lembaga negara yang independen, Bank Indonesia mempunyai otonomi penuh dalam merumuskan dan melaksanakan setiap tugas dan wewenangnya sebagaimana ditentukan dalam undang-undang tersebut.Pihak luar tidak dibenarkan mencampuri pelaksanaan tugas Bank Indonesia, dan Bank Indonesia juga berkewajiban untuk menolak atau mengabaikan intervensi dalam bentuk apapun dari pihak manapun juga.

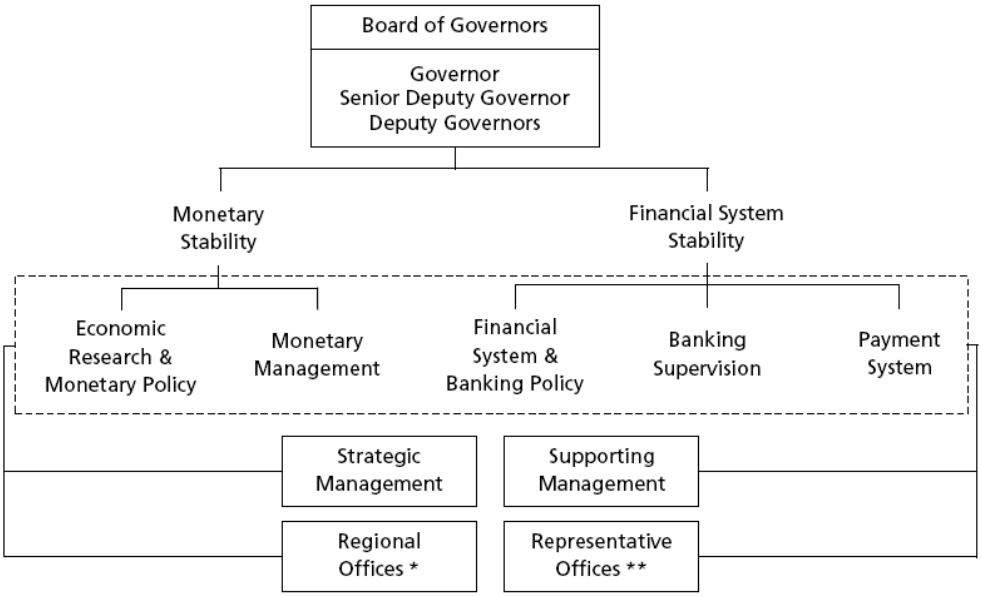
Untuk lebih menjamin independensi tersebut, undang-undang ini telah memberikan kedudukan khusus kepada Bank Indonesia dalam struktur ketatanegaraan Republik Indonesia. Sebagai Lembaga negara yang independen kedudukan Bank Indonesia tidak sejajar dengan Lembaga Tinggi Negara.

Disamping itu, kedudukan Bank Indonesia juga tidak sama dengan Departemen, karena kedudukan Bank Indonesia berada diluar Pemerintah. Status dan kedudukan yang khusus tersebut diperlukan agar Bank Indonesia dapat melaksanakan peran dan fungsinya sebagai otoritas moneter secara lebih efektif dan efisien.

Status Bank Indonesia baik sebagai badanhukum publik maupun badan hukum perdata ditetapkan dengan undang-undang. Sebagai badan hukum publik Bank Indonesia berwenang menetapkan peraturan-peraturan hukum yang merupakan pelaksanaan dari undang-undang yang mengikat seluruh masyarakat luas sesuai dengan tugas dan wewenangnya. Sebagai badan hukum perdata, Bank Indonesia dapat bertindak untuk dan atas nama sendiri di dalam maupun di luar pengadilan [6].

2. 1. 4 Struktur Organisasi Bank Indonesia

Pada Bank Indonesia memiliki bagian atau divisi yang memiliki tugasnya masing-masing sesuai dengan gambar 2.2 dan detail tugas dari masing-masing divisi seperti yang terdapat pada gambar 2.3.



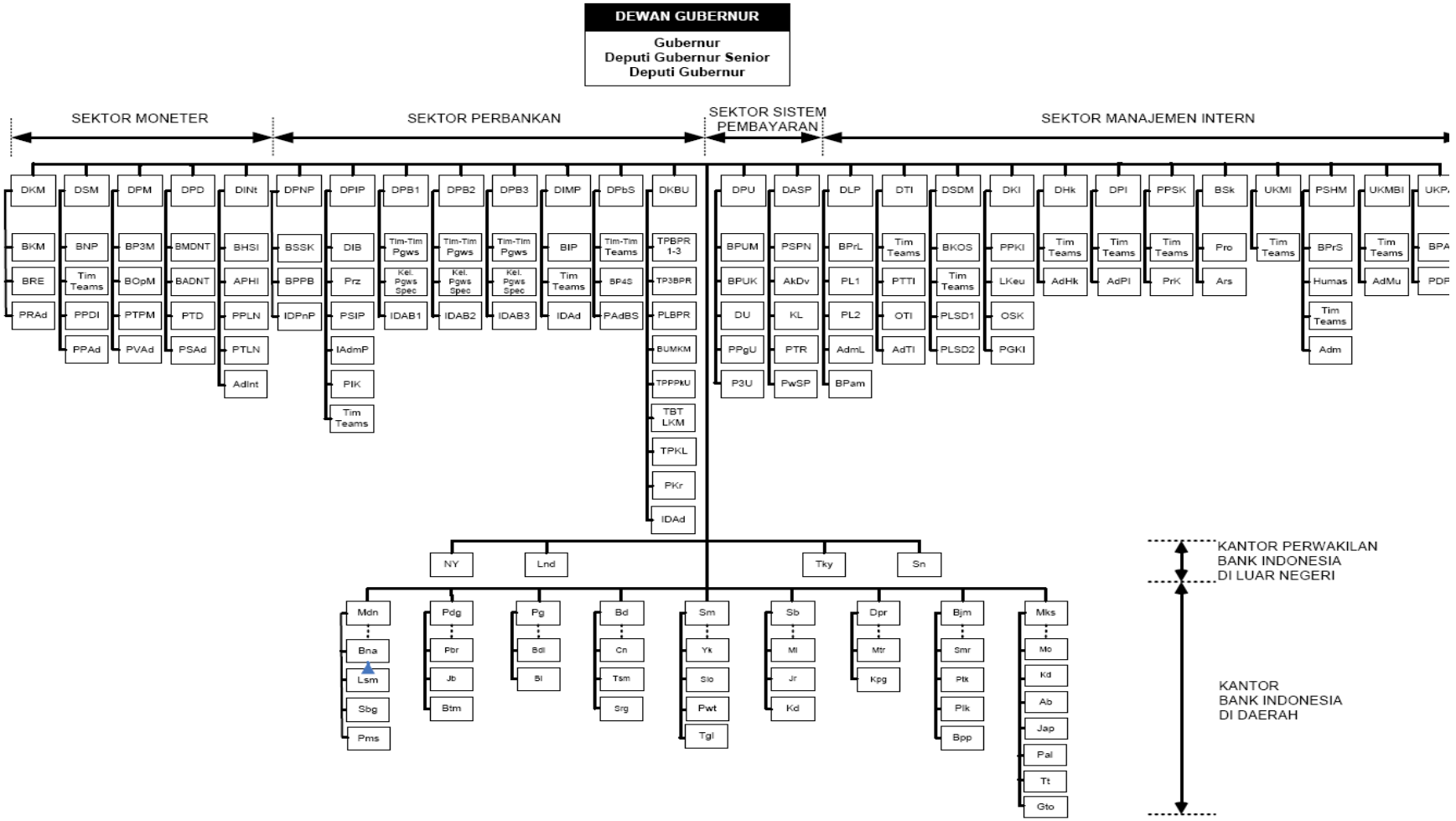
Gambar II.2 Struktur Organisasi Bank Indonesia

Keterangan *

Medan, Banda Aceh, Pematang Siantar, Lhokseumawe, Sibolga, Padang, Pekanbaru, Jambi, Batam, Palembang, Bengkulu, Bandarlampung, Bandung, Serang, Cirebon, Tasikmalaya, Tegal, Semarang, Yogyakarta, Solo, Purwokerto, Surabaya, Malang, Kediri, Jember, Denpasar, Mataram, Kupang, Banjarmasin, Pontianak, Palangkaraya, Gorontalo, Samarinda, Balikpapan, Makassar, Manado, Palu, Kendari, Ternate, Ambon, & Jayapura.

Keterangan **

Singapore, Tokyo, London, & New York [8].



Gambar II.3Detail Struktur Organisasi Bank Indonesia

2. 1. 5 *Job Descrption* Manajemen SDM Bank Indonesia

Berdasarkan surat edaran No.9/12/INTERN tanggal 30 Maret 2007, *job description* di Bank Indonesia dibagi kedalam 4 bagian, yaitu sebagai berikut :

1. Pemimpin Bank Indonesia (PBI).

Mengarahkan dan menetapkan pelaksanaan kebijakan perbankan syariah dalam rangka memastikan keberhasilan kebijakan.

2. Deputi PBI Bidang Ekonomi Moneter (Ekmon).

a. Deputi PBI bidang Ekmon

Mengarahkan dan menetapkan pelaksanaan kebijakan perbankan syariah dalam rangka memastikan keberhasilan kebijakan.

b. Analis Madya Senior

Merencanakan dan Menentukan Peraturan dan Pedoman Operasionalisasi Kebijakan yang efektif dan sesuai dengan prinsip syariah

c. Analis Madya

Merencanakan, menentukan dan mengontrol pelaksanaan Perizinan dan Pengawasan Bank Syariah yang efektif dalam rangka memastikan keberhasilan kebijakan

d. Peneliti Ekonomi Madya

Merealisasikan Penelitian dan Pembangunan Operasionalisasi Kebijakan Perbankan Syariah, melalui research-based policy, dalam rangka memastikan keberhasilan kebijakan

e. Analis Madya

Merealisasikan kegiatan pengaturan melalui kegiatan review dan harmonisasi ketentuan yang dapat memberikan nilai tambah dalam rangka memastikan keberhasilan kebijakan

3. Deputi PBI Bidang Sistem Pembayaran Manajemen Intern (SPMI).

a. Deputi PBI bidang SPMI

Mengarahkan dan menetapkan pelaksanaan kebijakan perbankan syariah dalam rangka memastikan keberhasilan kebijakan.

b. Kepala Bidang Manajemen Intern

Merencanakan dan Menentukan Peraturan dan Pedoman Operasionalisasi Kebijakan yang efektif dan sesuai dengan prinsip syariah.

c. Kepala Bidang Sistim Pembayaran

Merencanakan, menentukan dan mengontrol pelaksanaan Perizinan dan Pengawasan Bank Syariah yang efektif dalam rangka memastikan keberhasilan kebijakan.

d. Kasir Madya

Merealisasikan Penelitian dan Pembangunan Operasionalisasi Kebijakan Perbankan Syariah, melalui research-based policy, dalam rangka memastikan keberhasilan kebijakan

4. Deputi PBI Bidang Pebankan.

a. Deputi PBI Bidang Perbankan

Mengarahkan dan menetapkan pelaksanaan kebijakan perbankan syariah dalam rangka memastikan keberhasilan kebijakan

b. Pengawas Bank Madya Senior

Merencanakan dan Menentukan Peraturan dan Pedoman Operasionalisasi Kebijakan yang efektif dan sesuai dengan prinsip syariah

c. Pengawas Bank Madya

Merencanakan, menentukan dan mengontrol pelaksanaan Perizinan dan Pengawasan Bank Syariah yang efektif dalam rangka memastikan keberhasilan kebijakan

d. Kepala Bagian Idab

Merealisasikan Penelitian dan Pembangunan Operasionalisasi Kebijakan Perbankan Syariah, melalui research-based policy, dalam rangka memastikan keberhasilan kebijakan [7].

2. 2 Landasan Teori

Untuk mengembangkan aplikasi pengolahan data pengobatan ini , diperlukan teori-teori yang mendukung serta berkaitan dengan aplikasi yang akan dikembangkan. Teori-teori tersebut dapat menjadi landasan dari aplikasi yang

akan dikembangkan. Landasan teori dari Pembangunan aplikasi pengolahan data pengobatan ini adalah sebagai berikut :

2. 2. 1 Java

Java adalah bahasa pemrograman berorientasi objek yang dikembangkan oleh Sun Micro*system*s sejak tahun 1991. Bahasa ini dikembangkan dengan model yang mirip dengan bahasa C++ dan *Smalltalk*, namun dirancang agar lebih mudah dipakai dan *platform independent*, yaitu dapat dijalankan di berbagai jenis sistem operasi dan arsitektur komputer. Bahasa ini juga dirancang untuk pemrograman di Internet sehingga dirancang agar aman dan portabel.

1. Platform Independent

Platform independent berarti program yang ditulis dalam bahasa Java dapat dengan mudah dipindahkan antar berbagai jenis sistem operasi dan berbagai jenis arsitektur komputer. Aspek ini sangat penting untuk dapat mencapai tujuan Java sebagai bahasa pemrograman Internet di mana sebuah program akan dijalankan oleh berbagai jenis komputer dengan berbagai jenis sistem operasi. Sifat ini berlaku untuk level source code dan binary code dari program Java. Berbeda dengan bahasa C dan C++, semua tipe data dalam bahasa Java mempunyai ukuran yang konsisten di semua jenis platform. Source code program Java sendiri tidak perlu dirubah sama sekali jika Anda ingin mengkompile ulang di platform lain. Hasil dari mengkompile source code Java bukanlah kode mesin atau instruksi prosesor yang spesifik terhadap mesin tertentu, melainkan berupa bytecode yang berupa file berekstensi `.class`. Bytecode tersebut dapat langsung Anda eksekusi di tiap platform yang dengan menggunakan Java Virtual Machine (JVM) sebagai interpreter terhadap bytecode tersebut.

2. Kompilasi

Kompilasi kode program Java dilakukan menggunakan toolcommand-line yang bernama javac, atau biasa disebut kompiler Java. Tahap kompilasi ini bertujuan untuk mengonversi kode sumber ke program biner yang berisi bytecode, yaitu instruksi-instruksi mesin.

3. Interpretasi

Kode program Java tidak dieksekusi di komputer secara langsung, tetapi berjalan di atas komputer hipotesis yang distandardisasikan, yang disebut Java Virtual Machine.

Untuk menginterpretasi bytecode, kita menggunakan tool bernama java, atau biasa disebut interpreter Java.

4. Library

Selain kompiler dan interpreter, bahasa Java sendiri memiliki library yang cukup besar yang dapat mempermudah Anda dalam membuat sebuah aplikasi dengan cepat. Library ini sudah mencakup untuk grafik, desain user interface, kriptografi, jaringan, suara, database, dan lain-lain.

5. Berorientasi Objek

Java adalah bahasa pemrograman berorientasi objek. Pemrograman berorientasi objek secara gamblang adalah teknik untuk mengorganisir program dan dapat dilakukan dengan hampir semua bahasa pemrograman. Namun Java sendiri telah mengimplementasikan berbagai fasilitas agar seorang programer dapat mengoptimalkan teknik pemrograman berorientasi objek.

2. 2. 2 NetBeans

NetBeans merupakan sebuah proyek kode terbuka yang sukses dengan pengguna yang sangat luas, komunitas yang terus tumbuh, dan memiliki hampir 100 mitra (dan terus bertambah). Sun Micro*system*s mendirikan proyek kode terbuka NetBeans pada bulan Juni 2000 dan terus menjadi sponsor utama.

Saat ini terdapat dua produk : NetBeans IDE dan NetBeans Platform. The NetBeans IDE adalah sebuah lingkungan Pembangunan - sebuah kakas untuk pemrogram menulis, mengompilasi, mencari kesalahan dan menyebarkan

program. Netbeans IDE ditulis dalam Java – namun dapat mendukung bahasa pemrograman lain. Terdapat banyak modul untuk memperluas Netbeans IDE.Netbeans IDE adalah sebuah produk bebas dengan tanpa batasan bagaimana digunakan [4].

2. 2. 3 iReport

IReport adalah report designer yang digunakan oleh Jasper Report. Karena kemiripannya dengan Crystal Report, IReport menjadi salah satu pilihan utama untuk melakukan proses desain report di Java. Tool untuk reporting yang paling populer di Java adalah iReport. iReport penggunaanya mirip dengan Crystal Report yang sudah banyak dikenal orang. Dengan iReport kita dapat membuat report dengan mudah. Cukup dengan drag n drop dan juga wizard yang sangat mudah digunakan. iReport sendiri tersedia dalam bentuk aplikasi tunggal dan yang paling baru adalah iReport sudah dapat diintegrasikan dengan Netbeans IDE. Artinya kita tidak perlu lagi menggunakan 2 tools yang berbeda untuk membuat aplikasi yang lengkap dengan reportnya. Dengan Netbeans IDE + iReport plugin maka tools Pembangunan aplikasi kita udah lengkap [11].

2. 2. 4 Unified Modelling Language (UML)

UML sebagai sebuah bahasa yang memberikan vocabulary dan tatanan penulisankata-kata dalam ‘MS Word’ untuk kegunaan komunikasi. Sebuah bahasa modeladalah sebuah bahasa yang mempunyai vocabulary dan konsep tatanan / aturanpenulisan serta secara fisik mempresentasikan dari sebuah sistem. Seperti halnyaUML adalah sebuah bahasa standard untuk pengembangan sebuah software yangdapat menyampaikan bagaimana membuat dan membentuk model-model, tetapitidak menyampaikan apa dan kapan model yang seharusnya dibuat yangmerupakan salah satu proses implementasi pengembangan software.

UML tidak hanya merupakan sebuah bahasa pemograman visual saja, namunjuga dapat secara langsung dihubungkan ke berbagai bahasa pemograman,seperti JAVA, C++, Visual Basic, atau bahkan dihubungkan secara langsung kedalam sebuah object-oriented database. Begitu juga

mengenaipendokumentasian dapat dilakukan seperti; requirements, arsitektur, design,source code, project plan, tests, dan prototypes [5].

Dari berbagai penjelasan rumit yang terdapat di dokumen dan buku-buku UML, sebenarnya konsepsi dasar UML bisa kita rangkumkan seperti dalam gambar 2.6.

Major Area	*View*	*Diagrams*	*Main Concepts*
structural	static view	class diagram	class, association, generalization, dependency, realization, interface
	use case view	use case diagram	use case, actor, association, extend, include, use case generalization
	implementation view	component diagram	component, interface, dependency, realization
	deployment view	deployment diagram	node, component, dependency, location
dynamic	state machine view	statechart diagram	state, event, transition, action
	activity view	activity diagram	state, activity, completion transition, fork, join
	interaction view	sequence diagram	interaction, object, message, activation
		collaboration diagram	collaboration, interaction, collaboration role, message
model management	model management view	class diagram	package, subsystem, model
extensibility	all	all	constraint, stereotype, tagged values

Gambar II.4 Konsepsi Dasar UML

Seperti juga tercantum pada Gambar 2.6, UML mendefinisikan diagram-diagram sebagai berikut:

1. Use Case Diagram

Use case diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. Use case diagram terdiri atas diagram untuk use case dan actor. Actor merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi.

Use case merepresentasikan operasi-operasi yang dilakukan oleh actor. Use case digambarkan berbentuk elips dengan namaoperasi dituliskan di dalamnya. Actor yang melakukan operasi dihubungkan dengan garis lurus ke use case [9].

2. Class Diagram

Class diagram dapat membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai.Class diagram banyak memperhatikan hubungan antarkelas dan penjelasan detail tiap kelas dalam pemodelan desain (dalam logical view) dari suatu sistem.Selama proses analisa, class diagram memperhatikan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama tahap desain, class diagram berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat [9].

Hubungan antar kelas dalam pemodelan dengan tools UML digambarkan sebagai berikut :

a. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.
b. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).
c. Pewarisan, yaitu hubungan hirarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
d. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.

3. Statechart Diagram

Statechart diagram digunakan untuk memodelkan perilaku dinamis satu kelas atau objek. Statechart diagram memperlihatkan urutan keadaan sesaat (state) yang dilalui sebuah objek, Kejadian yang menyebabkan sebuah transisi dari suatu state atau aktivitas kepada yang lainnya.Statechart diagram khusus digunakan untuk memodelkan tahap-tahap diskrit dari sebuah siklus hidup objek,sedangkan Activity diagram paling cocok untuk memodelkan urutan aktifitas dalam suatu proses [9].

4. Activity Diagram

Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (internal processing). Activity diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Activity diagram dapat dibagi menjadi beberapa object swimlane untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu [2].

5. Sequence Diagram

Sequence Diagram menjelaskan interaksi objekyang disusun dalam suatu urutan waktu. Diagram inisecara khusus berasosiasi dengan use-case. Sequencediagram memperlihatkan tahap demi tahap apa yang arusterjadi untuk menghasilkan suatu didalam use-case diagram. Tipe diagram yang digunakan sebaiknya digunakan diawal tahap desain atau analisis karena kesederhanaannya dan mudah untuk di mengerti [9].

6. Collaboration Diagram

Collaboration diagram menggambarkan interaksi antar objek seperti sequence diagram, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian message. Setiap message memiliki sequence number, di mana message dari level tertinggi memiliki nomor 1. Messages dari level yang sama memiliki prefiks yang sama. [9].

7. Component Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (dependency) di antaranya. Komponen piranti lunak adalah modul berisi code, baik berisi source code maupun binary code, baik library maupun executable, baik yang muncul pada compile time, link time, maupun run time. Umumnya komponen terbentuk dari beberapa class dan/atau package, tapi dapat juga dari komponen-komponen yang lebih kecil.Komponen dapat juga berupa interface, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain [9].

8. Deployment Diagram

Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal

Sebuah *node* adalah server, *workstation*, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini [9].

2. 2. 5 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang multi thread, 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[10].