

BAB II

TINJAUAN PUSTAKA

2.1 Pembangunan

Pembangunan adalah setiap usaha mewujudkan hidup yang lebih baik sebagaimana yang didefinisikan oleh suatu negara “*an increasing attainment of one’s own cultural values*” (Tjokrowinoto, 1996: 1).

Siagian (1994) memberikan pengertian tentang pembangunan sebagai “Suatu usaha atau rangkaian usaha pertumbuhan dan perubahan yang berencana dan dilakukan secara sadar oleh suatu bangsa, negara dan pemerintah, menuju modernitas dalam rangka pembinaan bangsa (nation building)”.

Sedangkan Ginanjar Kartasasmita (1994) memberikan 17 pengertian yang lebih sederhana, yaitu sebagai “suatu proses perubahan ke arah yang lebih baik melalui upaya yang dilakukan secara terencana”.

2.2 Sistem

Makna kata “sistem” didefinisikan dengan berbagai pendekatan dan beragam istilah. Menurut Lucas (1992), Sistem adalah suatu pengorganisasian yang saling berinteraksi, saling tergantung dan terintegrasi dalam kesatuan variabel atau komponen.

Jogiyanto (1999) mendefinisikan sistem ke dalam dua kelompok pendekatan, yaitu menekankan pada prosedur dan komponen atau elemennya. Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem sebagai suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkelompok dan

bekerjasama untuk melakukan kegiatan pencapaian sasaran tertentu. Makna dari prosedur sendiri, yaitu urutan yang tepat dari tahapan-tahapan instruksi yang menerangkan apa (*what*) yang harus dikerjakan, siapa (*who*) yang mengerjakan, kapan (*when*) dikerjakan dan bagaimana (*how*) mengerjakannya. Pendekatan yang menekankan pada komponen mendefinisikan “sistem” sebagai kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Beberapa penulis yang mendefinisikan “sistem” yang menekankan pada elemen atau komponennya adalah Barry E. Cushing (1974:12), Gordon B. Davis (1974:81).

2.3 Informasi

Gordon B. Davis (1991 : 28) mengatakan bahwa informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat bagi pengambilan keputusan saat ini atau mendatang.

2.4 Manajemen

Menurut Drs. Oey Liang Lee manajemen adalah seni dan ilmu perencanaan pengorganisasian, penyusunan, pengarahan dan pengawasan daripada sumberdaya manusia untuk mencapai tujuan yang telah ditetapkan.

2.5 Sistem Informasi Manajemen

Sistem Informasi Manajemen adalah suatu perencanaan untuk berbagai macam pekerjaan yang ada kaitannya dengan analisis manajemen yang dapat diselesaikan dengan cepat melalui pengendalian internal seperti pemanfaatan

sumber daya, dokumen, teknologi, dan akuntansi manajemen sebagai salah satu strategi dalam bisnis.

2.6 Aplikasi

Aplikasi dapat juga dikatakan sebagai penerjemah perintah-perintah yang dijalankan pengguna komputer untuk diteruskan ke atau diproses oleh perangkat keras.

Menurut Marimin dkk. (2011:43) Aplikasi merupakan program yang secara langsung dapat melakukan proses-proses yang digunakan dalam komputer oleh pengguna. Aplikasi merupakan kumpulan dari file-file tertentu yang berisi kode program yang menghubungkan antara pengguna dan perangkat keras Komputer.

2.7 Marketing

Pemasaran atau *marketing* adalah suatu rangkaian kegiatan untuk memenuhi kebutuhan dan kepuasan konsumen. Caranya dengan membuat produk, menentukan harganya, tempat penjualannya dan mempromosikan produk tersebut kepada para konsumen. Berikut ini merupakan pengertian *marketing* menurut para ahli.

- Jay Abraham

Menurut Jay Abraham menyatakan bahwa *marketing* bisa dianggap sukses jika kamu sudah tampak baik bagi pelanggan kamu.

- Philip Kotler

Menurut Philip Kotler menyatakan bahwa *marketing* adalah suatu kegiatan sosial dan sebuah pengaturan yang dikerjakan oleh individu atau

sekelompok supaya mendapatkan apa yang diinginkan dengan membuat suatu produk lalu menukarnya dengan nominal tertentu kepada pihak lain.

- Tung Desem Waringin

Menurut Tung Desem Waringin menyatakan bahwa *marketing* adalah mengkomunikasikan suatu nilai tambah atau *perceive value* yang lebih tinggi.

2.8 Bootstrap

Bootstrap merupakan salah satu jenis framework untuk CSS (Cascading Style Sheet) yang digunakan untuk perancangan situs website. Penggunaan bootstrap sangatlah membantu programmer dalam membangun tampilan sebuah website. Hal tersebut tertulis di buku Bootstrap: Responsive Web Development karangan Jake Spurlock. Menurut Jake Spurlock keunggulan dari penggunaan Bootstrap adalah dapat menyesuaikan dengan kebutuhan website, dimana dapat memilih fitur CSS dan JavaScript sesuai dengan kebutuhan. Contohnya CSS pada Bootstrap juga menyediakan fitur form, tombol, navigasi dan komponen-komponen lainnya dan JavaScript yang membantu pembuatan antarmuka lebih mudah dan stabil.

2.9 Model Rational Unified Process (RUP)

Unified Process adalah sebuah proses pengembangan perangkat lunak yang dilakukan secara iteratif (berulang) dan inkremental (bertahap dengan progres menaik). Iteratif bisa dilakukan di setiap tahap, atau iteratif tahap pada proses pengembangan perangkat lunak untuk menghasilkan perbaikan fungsi yang inkremental (bertambah menaik) dimana setiap iterasi akan memperbaiki iterasi

berikutnya. Salah satu Unified Process yang terkenal adalah *Rational Unified Process* (Rosa & Shalahudin 2016: 124).

Rational Unified Process (RUP) adalah model pengembangan perangkat lunak yang menggunakan teknik disiplin dalam penetapan tugas dan tanggung jawab untuk menghasilkan perangkat lunak berkualitas tinggi, memenuhi kebutuhan pengguna, dalam estimasi waktu dan biaya yang dapat diprediksi (Rational Software, 1998). *Rational Unified Process* didasarkan pada prinsip *use case driven, architecture centered, iterative and incremental, risk oriented* (Wazlawick, 2014: 3). Rosa & Shalahudin (2016: 125) menambahkan bahwa RUP adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*) khusus pemrograman berorientasi objek yang dikembangkan oleh *Rational Software* sebuah divisi dari IBM. RUP merupakan proses rekayasa perangkat lunak dengan pendefinisian yang baik (*well-defined*) dan penstrukturan yang baik (*well structured*).

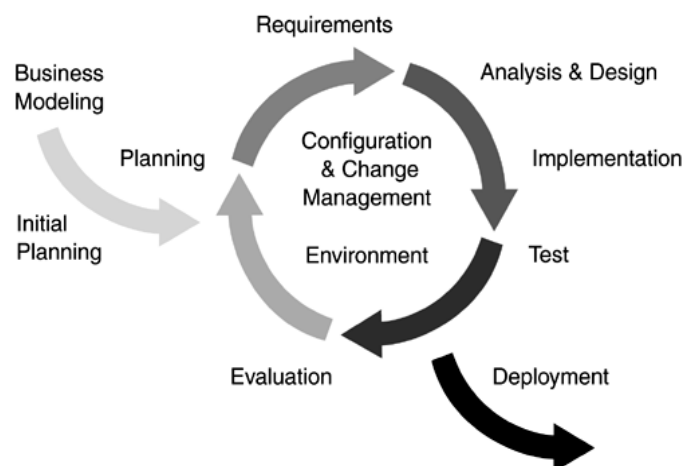
Pendekatan iteratif dari *Rational Unified Process* dapat mengakomodir beberapa kekurangan pengembangan tanpa konsep perulangan layaknya pada model *waterfall*, diantaranya sebagai berikut:

- 1) Mengakomodasi perubahan kebutuhan perangkat lunak.
- 2) Pengintegrasian bukanlah proses besar dan cepat di akhir proyek.
- 3) Risiko biasa ditemukan atau dialamatkan selama pengintegrasian awal.
- 4) Manajemen artinya membuat perubahan taktik pada produk.
- 5) *Reusable* artinya mendukung fasilitas penggunaan kembali.

- 6) Kesalahan bisa ditemukan dan diperbaiki pada beberapa iterasi sehingga menghasilkan arsitektur yang baik dan aplikasi berkualitas tinggi.
- 7) Lebih baik menggunakan anggota proyek dibandingkan susunan secara seri pada tim proyek.
- 8) Anggota tim belajar selama proyek berlangsung.
- 9) Perbaikan perangkat lunak dapat dilakukan seiring proses pengembangannya.

Rational Unified Process menyajikan pengembangan perangkat lunak sebagai aktivitas (*workflow*) berulang disusun pada empat tahapan fase yaitu *inception*, *elaboration*, *construction*, dan *transition* (Sommerville, 2011: 121).

RUP biasanya digambarkan dari tiga perspektif: 1) Perspektif dinamis, menunjukkan fase model dari waktu ke waktu. 2) Perspektif statis, menunjukkan aktivitas proses yang diberlakukan. 3) Perspektif praktik, menyarankan praktik yang baik digunakan selama proses.



Gambar 2.1 Model Pengembangan Perangkat Lunak RUP

Fase dalam RUP lebih terkait dengan bisnis daripada masalah teknis. Berikut penjelasan empat fase pengembangan RUP:

1. *Inception*

Tahap *inception* memiliki fokus utama terhadap memodelkan proses bisnis yang dibutuhkan (*business modeling*) dan mendefinisikan kebutuhan sistem yang akan dibuat (*requirements*). Tujuan dari fase *inception* adalah menetapkan business case sistem. Mengidentifikasi semua entitas eksternal (orang dan sistem) yang akan berinteraksi dengan sistem dan menentukan interaksinya. Selanjutnya menggunakan informasi tersebut untuk menilai kontribusi yang diberikan sistem kepada bisnis. Apabila kontribusi tidak signifikan, proyek bisa dibatalkan.

2. *Elaboration*

Tahap ini lebih pada proses analisis dan desain perancangan arsitektur sistem. Mempunyai tujuan untuk mengembangkan pemahaman terkait domain masalah, 42 membangun kerangka kerja arsitektur sistem, mengembangkan rencana proyek, mengidentifikasi risiko proyek utama. Setelah menyelesaikan fase ini, diharuskan memiliki requirements model untuk sistem, yang mungkin berupa serangkaian UML *use-cases*, deskripsioarsitektur, dan rencana pengembangan untuk perangkat lunak.

3. *Construction*

Tahap ini lebih pada pengembangan komponen dan fitur-fitur sistem, khususnya implementasi dan pengujian sistem yang fokus pada kode program. Tahap *construction* melibatkan desain sistem, pemrograman, dan pengujian. Bagian dari sistem dikembangkan secara paralel dan terintegrasi

selama fase ini. Setelah menyelesaikan fase ini, diharuskan memiliki sistem perangkat lunak yang berfungsi dan dokumentasi terkait yang siap dikirim ke pengguna.

4. *Transition*

Tahapan *transition* lebih fokus terhadap *deployment* sistem supaya diketahui oleh user atau pengguna. Berkaitan dengan pemindahan sistem dari komunitas pengembangan ke komunitas pengguna dan membuatnya bekerja di lingkungan nyata. Setelah menyelesaikan fase ini, diharuskan memiliki sistem perangkat lunak terdokumentasi yang berfungsi dengan benar di lingkungan operasionalnya. Iterasi dalam RUP didukung dengan dua cara. Setiap fase dapat diberlakukan secara *iteratif* dengan hasil yang dikembangkan secara bertahap. Selain itu, seluruh rangkaian fase juga dapat diberlakukan secara bertahap seperti yang ditunjukkan oleh panah perulangan dari pada Gambar 8. 43 Gambar 8. Iterasi dalam RUP Perspektif statis dari Rational Unified Process berfokus pada kegiatan yang terjadi selama proses pengembangan. Ini disebut alur kerja (*workflow*) dalam deskripsi RUP. Ada enam alur kerja proses inti (*core process workflows*) dan tiga alur kerja pendukung inti (*core supporting workflows*) yang diidentifikasi dalam proses. RUP telah dirancang bersama dengan UML, sehingga uraian alur kerja berorientasi di sekitar model UML terkait seperti sequence models, object model, dan component model.

2.10 Perangkat Desain


2.10.1 Unified Modeling Language (UML)

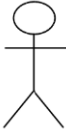

UML merupakan singkatan dari “*Unified Modelling Language*” yaitu suatu metode permodelan secara visual untuk sarana perancangan sistem berorientasi objek, atau definisi UML yaitu sebagai suatu bahasa yang sudah menjadi standar pada visualisasi, perancangan dan juga pendokumentasian sistem software. Saat ini UML sudah menjadi bahasa standar dalam penulisan *blue print software*.



Menurut Booch (2005:7) UML adalah Bahasa standar untuk membuat rancangan software. UML biasanya digunakan untuk menggambarkan dan membangun, dokumen artifak dari *software – intensive system*.

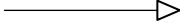
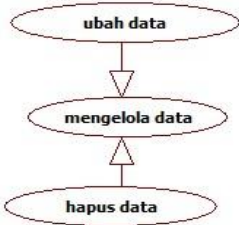

2.10.2 Use Case Diagram

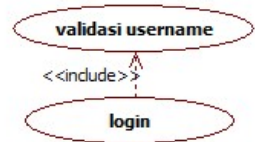
Menurut Rosa dan M. Shalahuddin (2015:155), *Use Case* atau *Diagram Use Case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut adalah symbol-simbol yang ada pada *use case diagram* :

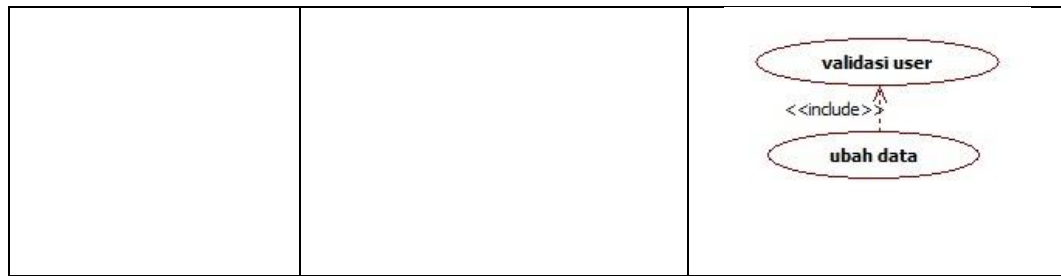
Simbol	Nama	Deskripsi
	<i>Use Case</i>	Fungsionalitas yang disediakan sistem sebagai unit yang saling bertukar pesan antar unit atau aktor,

		<p>biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i>.</p>
	<i>Actor</i>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
	<i>Association</i>	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>

<p><<extend>></p> 	<p>Ekstensi/<i>Extend</i></p>	<p>Relasi <i>use case</i> tambahan kesebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal</p>  <p>Arah panah mengarah pada <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use</i> <i>case</i> yang menjadi induknya.</p>
---	-------------------------------	--

	<p>Generalisasi/<i>Generalization</i></p>	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:</p> <div data-bbox="1054 792 1294 1016">  </div> <p>Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).</p>
<p><<include>></p> 	<p>Menggunakan/<i>Include</i></p>	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini. Ada dua sudut pandang</p>

		<p>yang cukup besar mengenai <i>include</i> di <i>use case</i> :</p> <ul style="list-style-type: none"> • <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, misal pada kasus berikut:  <pre> graph TD A([validasi username]) -.-> <<include>> B([login]) </pre> <p>The diagram illustrates an include relationship between two use cases. The top use case is labeled 'validasi username' and the bottom use case is labeled 'login'. A dashed line with an open arrowhead points from the 'login' use case to the 'validasi username' use case. The label '<<include>>' is placed on the line between the two use cases.</p> <ul style="list-style-type: none"> • <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan telah dijalankan sebelum <i>use case</i> tambahan di jalankan, misal pada kasus berikut:
--	--	---

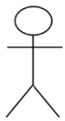

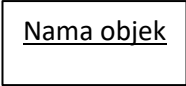

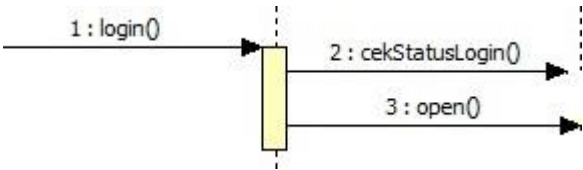


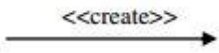
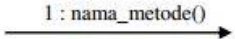
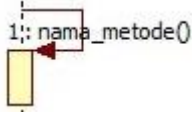
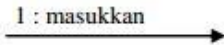
Sumber : (Rosa A.S dan M. Shalahuddin, 2015:156)

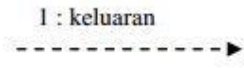
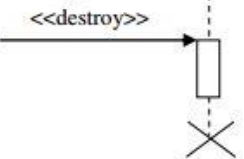
Tabel 2.1 Simbol *Use Case Diagram*

2.10.3 Sequence Diagram

Menurut Rosa dan M. Shalahuddin (2015:165), diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Berikut adalah simbol-simbol yang ada pada *sequence diagram* :

Simbol	Nama	Deskripsi
	<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda diawal frase nama aktor.
	Garis hidup/ <i>Lifeline</i>	Menyatakan kehidupan suatu objek
	Objek	Menyatakan objek baru pada diagram
	Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang berhubungan dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya: 

		<p>Maka cekStatusLogin() dan open() dilakukan dalam metode login().</p> <p>Aktor tidak memiliki waktu aktif.</p>
	<p>Pesan tipe <i>create</i></p>	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>
	<p>Pesan tipe <i>call</i></p>	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.</p>  <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.</p>
	<p>Pesan tipe <i>send</i></p>	<p>Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>



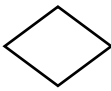


	Pesan tipe <i>return</i>	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
	Pesan tipe <i>destroy</i>	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>destroy</i> .

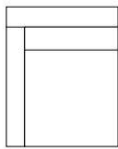
Sumber : (Rosa A.S dan M. Shalahuddin, 2015:165)

Tabel 2.2 Simbol *Sequence Diagram*

2.10.4 Activity Diagram

Menurut Rosa dan M. Shalahuddin (2015:161), diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut adalah simbol-simbol yang ada pada diagram aktivitas :

Simbol	Nama	Deskripsi
	Status awal	Status awal aktivitas sistem, sebuah aktivitas memiliki sebuah status awal.
	<i>Activity</i>	Aktivitas yang dilakukan sistem, biasanya diawali dengan kata kerja.
	<i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
	Status akhir	Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
	<i>Fork and Join</i>	<i>Fork and Join</i> digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.

	<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas terjadi.
---	-----------------	---

Sumber : (Rosa A.S dan M. Shalahuddin, 2015:162)



Tabel 2.3 Simbol *Activity Diagram*




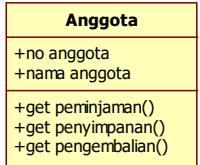
2.10.5 Class Diagram

Menurut Rosa dan M. Shalahuddin (2015:141), diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan *method* atau operasi. Berikut penjelasan *attribute* dan *method* :

- Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas.
- Operasi atau *method* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Berikut adalah simbol-simbol yang ada pada diagram kelas :

Simbol	Nama	Keterangan
	<i>Association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity.
	<i>Aggregation</i>	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>).

	<i>Composition</i>	Menunjukkan hubungan kelas induk dan anak, kelas anak tidak bisa berdiri sendiri
	<i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.
	<i>Realization</i>	Menunjukkan realisasi dari suatu kelas, digunakan untuk menunjukkan kelas implement dari suatu interface.
 <pre> classDiagram class Anggota { +no anggota +nama anggota +get peminjaman() +get penyimpanan() +get pengembalian() } </pre>	<i>Class</i>	<i>Class</i> adalah blok-blok pembangunan pada program berorientasi objek. Sebuah class digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah nama <i>class</i> , bagian tengah mendefinisikan atribut dan bagian akhir mendefinisikan <i>method-method</i> dari sebuah <i>class</i> .

Sumber : (Rosa A.S dan M. Shalahuddin, 2015:146)

Tabel 2.4 Simbol *Class Diagram*

2.11 Perangkat Koding

2.11.1 Java

Java merupakan bahasa pemrograman berorientasi objek dan bebas *platform*, dikembangkan oleh *SUN Micro System* dengan sejumlah keunggulan yang memungkinkan java dijadikan sebagai bahasa pengembangan enterprise. (RijalulFikri, Ipam Fuadina Adam, ImamPrakoso : 2005 : 15)

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (*bytecode*) dan dapat dijalankan pada berbagai *Mesin Virtual Java* (JVM). Java merupakan bahasa pemrograman yang bersifat umum atau non-spesifik (*general purpose*), dan secara khusus didesain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun". Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak ataupun aplikasi.

2.11.2 Java Development Kit (JDK)

Menurut Rachmad Hakim dan Sutarto (2009: 7) mengungkapkan: “Java Development Kit merupakan seperangkat aplikasi yang digunakan untuk mengembangkan program dengan bahasa java. Di dalamnya terdapat tiga komponen utama, yaitu java API, java virtual mechine, dan java compiler.”

2.11.3 Java Runtime Environment (JRE)

Java Runtime Environment (JRE) adalah satu set alat perangkat lunak untuk pengembangan aplikasi Java. Ini menggabungkan *Java Virtual Machine* (JVM), kelas platform inti dan mendukung *library*. JRE merupakan bagian dari *Java Development Kit* (JDK), tapi dapat didownload secara terpisah. JRE pada awalnya dikembangkan oleh *Sun Microsystems Inc*, anak perusahaan yang sepenuhnya dimiliki oleh *Oracle Corporation*. Juga dikenal sebagai runtime Java.

2.11.4 Framework Spring

Framework Spring adalah framework open source berbasis Java yang menyediakan infrastruktur yang komprehensif dalam mengembangkan aplikasi Java dengan mudah dan cepat. Spring pertama kali ditulis dan dirilis oleh Rod Johnson dengan lisensi Apache 2.0 pada bulan Juni 2003. Spring akan membantu pengguna dalam pengembangan aplikasi dengan build yang sederhana, portable, cepat dan sistem berbasis JVM yang fleksibel.

Beberapa Fitur yang disediakan *Spring Framework* adalah sebagai berikut.

1. *Depedency Injection*
2. *Aspect Oriented Programming*
3. *Spring MVC dan Restful Web Service*
4. *Support koneksi database, dsb.*

Dalam pembangunan sistem ini, penulis menggunakan salah satu jenis Framework Spring yaitu Spring Boot.

Spring Boot merupakan salah satu jenis framework dari Spring. Namun di Spring Boot, kita lebih dipermudah dalam pembuatan program karena :

1. Sudah disediakan Tomcat dan beberapa server lain, sehingga kita hanya perlu *run*.
2. Menggunakan Maven sebagai *build manager* yang dapat kita atur di *Project Object Model (POM)*
3. Anotasi-anotasi ajaib yang mempermudah kita dalam menentukan komponen kelas-kelas.

2.11.5 Cascading Style Sheet (CSS)

CSS (*Cascading Style Sheet*) merupakan bahasa pemrograman web yang digunakan untuk mengendalikan dan membangun komponen dalam web sehingga tampilan web akan lebih rapi, terstruktur, interaktif, dan seragam. Program ini wajib dikuasai oleh setiap pembuat web program (Web Programmer), terutama oleh Web Designer (Saputra dan Feni Agustin, 2011).

2.11.6 HyperText Markup Language (HTML)

HyperText Markup Language (HTML) merupakan suatu bahasa markup yang digunakan untuk melakukan markup terhadap sebuah dokumen teks. Dalam dokumen atau skrip tersebut terdapat kode-kode atau perintah-perintah yang nantinya akan ditransfer oleh http kedalam web browser. Web browser seperti

Internet Explorer, Mozilla Firefox dan lain-lain berfungsi untuk menguji kode-kode HTML. Didalam sebuah file HTML terdapat tag-tag HTML yang secara umum terbagi ke dalam dua bagian yaitu head (kepala) dan body (tubuh). File HTML biasanya diawali dengan tag <html> dan diakhiri dengan tag </html> (Bunafit Nugroho, 2008).

2.11.7 XAMPP

Menurut (Maulana, Susilo, & Riyadi, 2015) “ XAMPP merupakan software web server yang berguna dalam pengembangan website yang di dalamnya sudah tersedia database server MySQL dan dapat mendukung pemrograman PHP “. XAMPP merupakan software gratis, dapat dijalankan di sistem operasi Windows, Linux maupun Mac OS. Software XAMPP bisa didapatkan gratis pada <http://www.apachefriends.org/en/xampp.html> sesuai dengan operating system yang digunakan.

2.11.8 MySQL

MySQL digunakan untuk mengakses dan mengatur konfigurasi MySQL database. Menurut (Djaelangkara et al., 2015) “ MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional RDBMS yang didistribusikan secara gratis di bawah lisensi GPL (General Public License) “. Berdasarkan teori diatas maka dapat disimpulkan MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional RDBMS yang didistribusikan di bawah lisensi GPL. MySQL adalah sebuah aplikasi yang digunakan untuk membuat query dalam pembuatan database, tabel maupun

manipulasi data dengan RDBMS (Relational Date-base Management System) yaitu aplikasi sistem yang menjalankan fungsi pengolahan data.

Sebagaimana dikemukakan oleh para ahli di atas, pembangunan adalah semua proses perubahan yang dilakukan melalui upaya-upaya secara sadar dan terencana (Riyadi dan Deddy Supriyadi Bratakusumah, 2005).

2.11.9 Database

Database adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Database adalah representasi kumpulan fakta yang saling berhubungan disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.