

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

2.1.1 Konsep Dasar Sistem Informasi

Sistem informasi dalam suatu pemahaman yang sederhana dapat didefinisikan sebagai suatu sistem berbasis komputer yang menyediakan informasi bagi beberapa pemakai dengan kebutuhan yang serupa.

Menurut Pratama (2014:10) mengemukakan bahwa “Sistem informasi merupakan gabungan dari empat bagian utama. Keempat bagian utama tersebut mencakup perangkat lunak (*software*), perangkat keras (*hardware*), infrastruktur, dan Sumber Daya Manusia (SDM) yang terlatih”.

Dewasa ini sistem informasi yang digunakan lebih berfokus pada sistem informasi berbasis komputer (*computer based information system*). Harapan yang ingin diperoleh disini adalah bahwa dengan penggunaan teknologi informasi atau sistem informasi berbasis komputer, informasi yang dihasilkan dapat lebih akurat, berkualitas dan tepat waktu, sehingga pengambilan keputusan dapat lebih efektif dan efisien.

2.1.2 Perpustakaan

Menurut Hendrianto (2014:58) mengemukakan bahwa “Banyak batasan atau pengertian perpustakaan yang disampaikan oleh para pakar dibidang perpustakaan”. Ada beberapa pengertian perpustakaan di bawah ini :

1. Perpustakaan diartikan sebuah ruangan atau gedung yang digunakan untuk menyimpan buku dan terbitan lainnya yang biasanya disimpan menurut tata susunan tertentu yang digunakan pembaca bukan untuk dijual.

2. Dalam pengertian yang terbaru seperti yang tercantum dalam keputusan Presiden RI Nomor II disebutkan bahwa perpustakaan merupakan salah satu sarana pelestarian bahan pustaka sebagai teknologi dan kebudayaan dalam rangka mencerdaskan kehidupan bangsa dan menunjang pelaksanaan pembangunan nasional.
3. Menurut RUU Perpustakaan pada Bab I pasal 1 menyatakan Perpustakaan adalah institusi yang mengumpulkan pengetahuan tercetak dan terekam, mengelolanya dengan cara khusus guna memenuhi kebutuhan intelektualitas para penggunanya melalui beragam cara interaksi pengetahuan.

2.1.3 Konsep Dasar Pemrograman

1. *Adobe DreamWeaver*

Menurut Madcoms (2013:14) mengemukakan bahwa “*Adobe Dreamweaver* merupakan program *web editor* yang dapat digunakan untuk membangun halaman *web*”.

2. *XAMPP*

Menurut Pratama (2014:440) mengemukakan bahwa “*XAMPP* adalah aplikasi *web server* bersifat instan (siap saji) yang dapat digunakan baik di sistem operasi Linux maupun di sistem operasi Windows”.

3. *Web Browser*

Menurut Mauluddin (2011:41) mengemukakan bahwa “*Web browser* disebut juga sebagai perambah, adalah perangkat lunak yang berfungsi menampilkan dan melakukan interaksi dengan dokumen-dokumen yang disediakan oleh *server web*”.

4. **HTML**

Menurut Hidayatullah dan Kawistara (2015:13) mengemukakan bahwa “*Hypertext Markup Language (HTML)* adalah bahasa *standard* yang digunakan untuk menampilkan halaman web”.

5. **Java Script**

Menurut Yatini (2014:2) mengemukakan bahwa “*JavaScript* adalah bahasa *scripting* kecil, ringan, berorientasi objek yang ditempelkan pada kode *HTML* dan diproses di sisi *client*”.

6. **PHP**

Menurut Zuliarso dan Februariyanti (2013:50) mengemukakan bahwa: *PHP (Hypertext Preprocessor)*, merupakan bahasa pemrograman pada sisi *server* yang memperbolehkan programmer menyisipkan perintah-perintah perangkat lunak *web server* (Apache, IIS) atau apapun akan dieksekusi sebelum perintah itu dikirim oleh halaman ke browser yang *me-request*-nya, contohnya adalah bagaimana memungkinkannya memasukkan tanggal sekarang pada sebuah halaman *web* setiap kali tampilan tanggal dibutuhkan.

7. **MySQL**

Menurut Zuliarso dan Februariyanti (2013:50) mengemukakan bahwa “MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*)”.

8. **Framework CodeIgniter**

Hidayatullah dan Kawistara (2015:295) mengemukakan bahwa “*Code Igniter* adalah salah satu *framework PHP* bahkan *framework PHP* yang paling *powerfull* saat ini karena di dalamnya terdapat fitur lengkap aplikasi *web* di mana fitur-fitur tersebut sudah dikemas menjadi satu”.

2.1.4 UML (*Unified Modelling Language*)

Menurut Pratama (2014:48) mengemukakan bahwa “*UML (Unified Modelling Language)* adalah standarisasi internasional untuk notasi dalam bentuk grafik, yang menjelaskan tentang analisis dan desain perangkat lunak yang dikembangkan dengan pemrograman berorientasi objek”.

Seperti bahasa-bahasa lainnya, *UML* mendefinisikan notasi dan *syntak* atau *semantik*. Notasi *UML* merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan *UML syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi *UML* terutama diturunkan dari tiga notasi yang telah ada sebelumnya : *Grady Booch OOD (Object Oriented Design)*, *Jim Rumbaugh OMT (Object Modelling Technique)* dan *Ivan Jacobson OOSE (Object Oriented Software Engineering)*.

Pada *UML* versi dua terdiri atas tiga kategori dan memiliki tiga belas jenis diagram yaitu :

1. *Struktur Diagram*

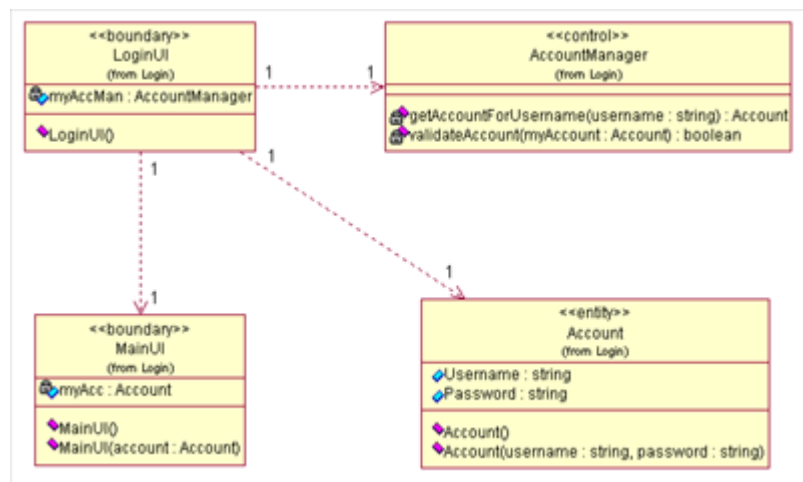
Menggambarkan elemen dari spesifikasi dimulai dengan kelas, objek, dan hubungan mereka, dan beralih ke dokumen arsitektur logis dari suatu sistem. Struktur diagram dalam *UML* terdiri atas :

- a. *Class Diagram*

Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas. *Class diagram* membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram

yang paling banyak dipakai. Selama tahap desain, class diagram berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat. *Class* memiliki tiga area pokok :

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

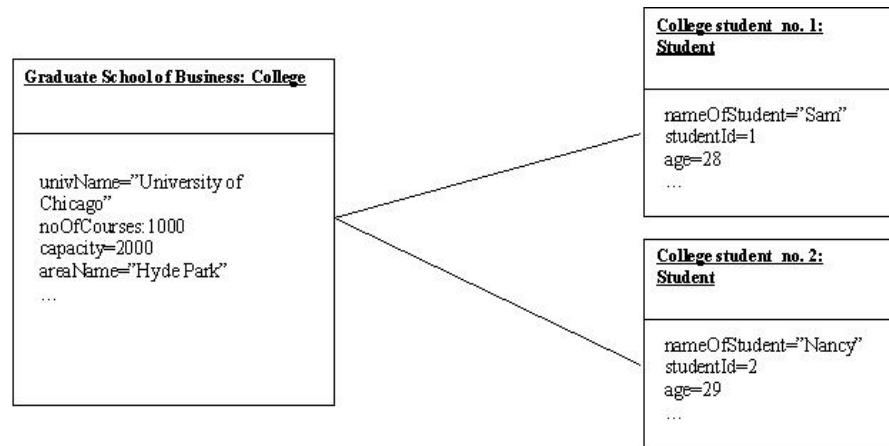


Sumber : Haviluddin (2011:3)

Gambar II.1. Notasi *Class* Diagram

b. *Object* Diagram

Object diagram menggambarkan kejelasan kelas dan warisan dan kadang-kadang diambil ketika merencanakan kelas, atau untuk membantu pemangku kepentingan non-program yang mungkin menemukan diagram kelas terlalu abstrak.

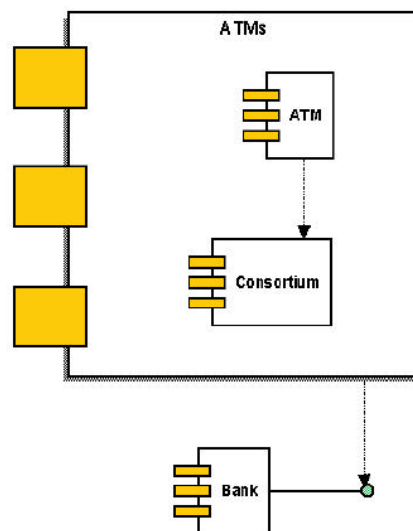


Sumber : Haviluddin (2011:3)

Gambar II.2. Notasi *Object Diagram*

c. *Component Diagram*

Component diagram menggambarkan struktur fisik dari kode, pemetaan pandangan logis dari kelas proyek untuk kode aktual di mana logika ini dilaksanakan.

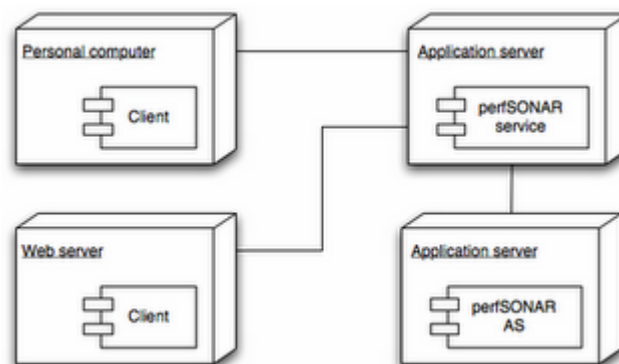


Sumber : Haviluddin (2011:3)

Gambar II.3. Notasi *Component Diagram*

d. *Deployment Diagram*

Deployment diagram memberikan gambaran dari arsitektur fisik perangkat lunak, perangkat keras, dan artefak dari sistem. *Deployment* diagram dapat dianggap sebagai ujung spektrum dari kasus penggunaan, menggambarkan bentuk fisik dari sistem yang bertentangan dengan gambar konseptual dari pengguna dan perangkat berinteraksi dengan sistem.

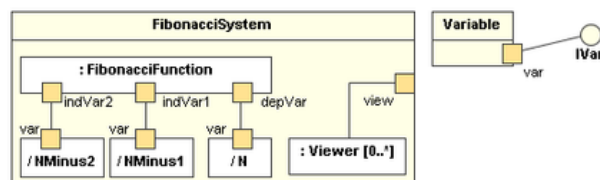


Sumber : Haviluddin (2011:4)

Gambar II.4. Notasi *Deployment* Diagram

e. *Composite Structure Diagram*

Sebuah diagram struktur komposit mirip dengan diagram kelas, tetapi menggambarkan bagian individu, bukan seluruh kelas. Kita dapat menambahkan konektor untuk menghubungkan dua atau lebih bagian dalam atau ketergantungan hubungan asosiasi.

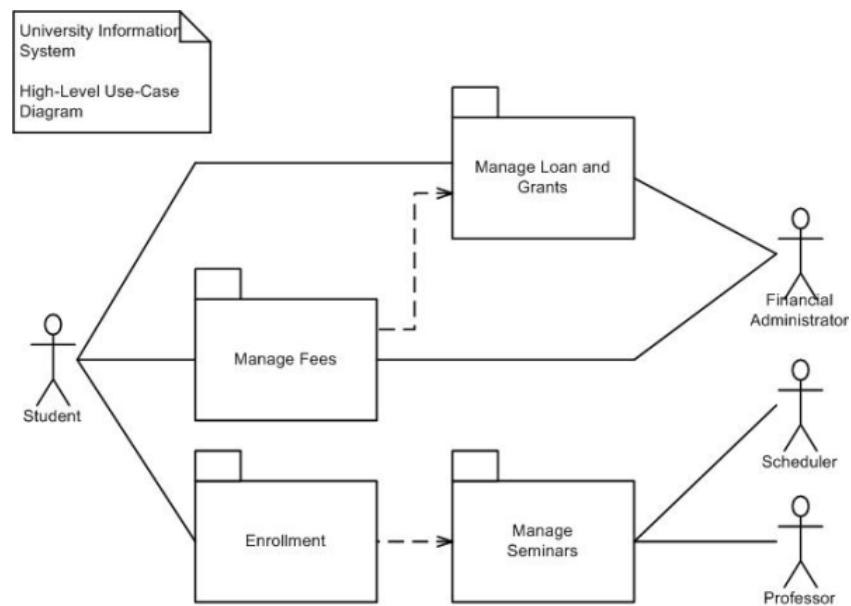


Sumber : Haviluddin (2011:4)

Gambar II.5. Notasi *Composite* Diagram

f. *Package Diagram*

Paket diagram biasanya digunakan untuk menggambarkan tingkat organisasi yang tinggi dari suatu proyek *software*. Atau dengan kata lain untuk menghasilkan diagram ketergantungan paket untuk setiap paket dalam pohon model.



Sumber : Haviluddin (2011:4)

Gambar II.6. Notasi *Package Diagram*

2. *Behaviour Diagram*

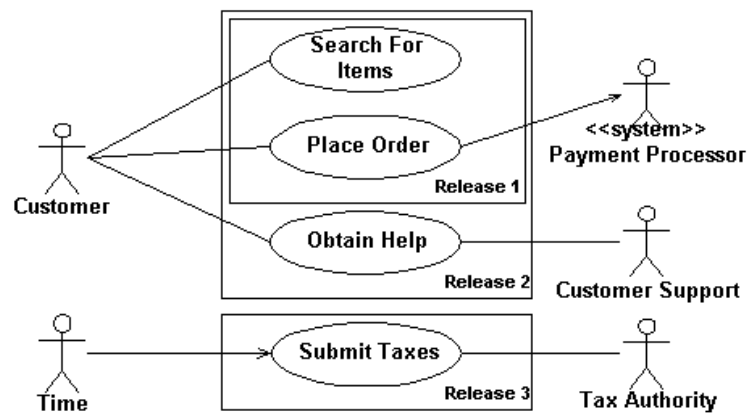
Menggambarkan ciri-ciri *behavior*/metode/fungsi dari sebuah sistem atau *business process*. *Behavior* diagram dalam UML terdiri atas :

a. *Use Case Diagram*

Diagram yang menggambarkan *actor*, *use case* dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah *use case* digambarkan sebagai elips horizontal dalam suatu diagram UML *use case*. *Use Case* memiliki dua istilah :

- 1) *System use case*; interaksi dengan sistem.

- 2) *Business use case*; interaksi bisnis dengan konsumen atau kejadian nyata

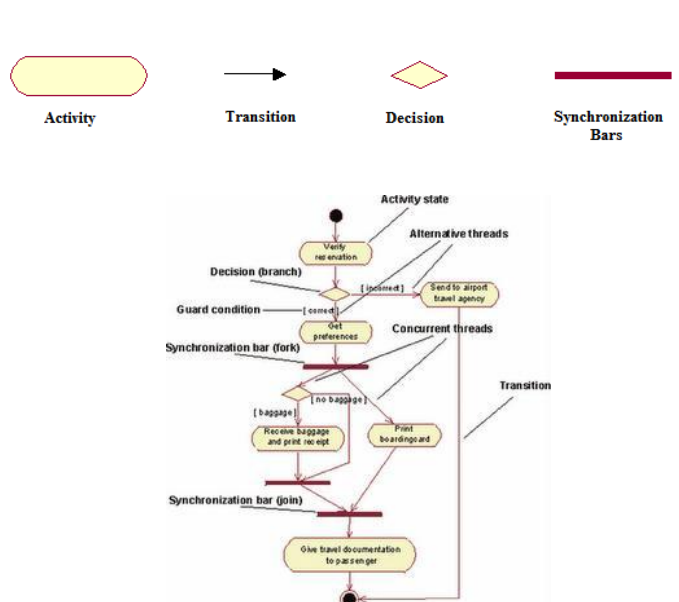


Sumber : Haviluddin (2011:4)

Gambar II.7. Notasi *Use Case Diagram*

b. *Activity Diagram*

Menggambarkan aktifitas-aktifitas, objek, *state*, transisi *state* dan *event*.

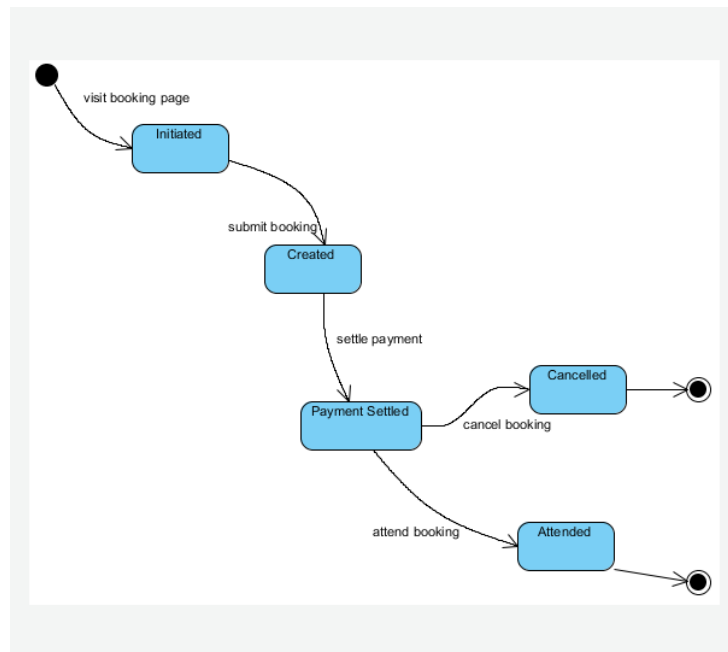


Sumber : Haviluddin (2011:4)

Gambar II.8. Notasi *Activity Diagram*

c. *State Machine Diagram*

Menggambarkan state, transisi state dan event.



Sumber : Haviluddin (2011:5)

Gambar II.9. Notasi *State Machine Diagram*

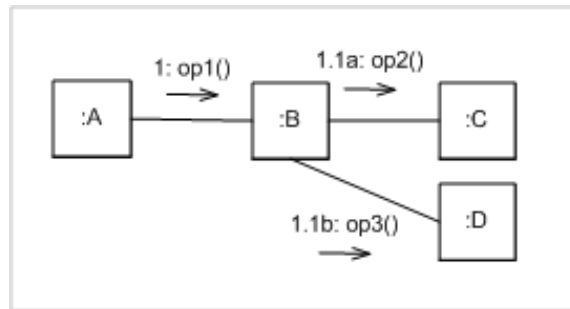
3. *Interaction Diagram*

Bagian dari *behavior diagram* yang menggambarkan interaksi objek.

Interaction diagram dalam *UML* terdiri atas :

a. *Communication Diagram*

Serupa dengan *sequence diagram*, tetapi diagram komunikasi juga digunakan untuk memodelkan perilaku dinamis dari *use case*. Bila dibandingkan dengan *sequence diagram*, diagram komunikasi lebih terfokus pada menampilkan kolaborasi benda daripada urutan waktu.

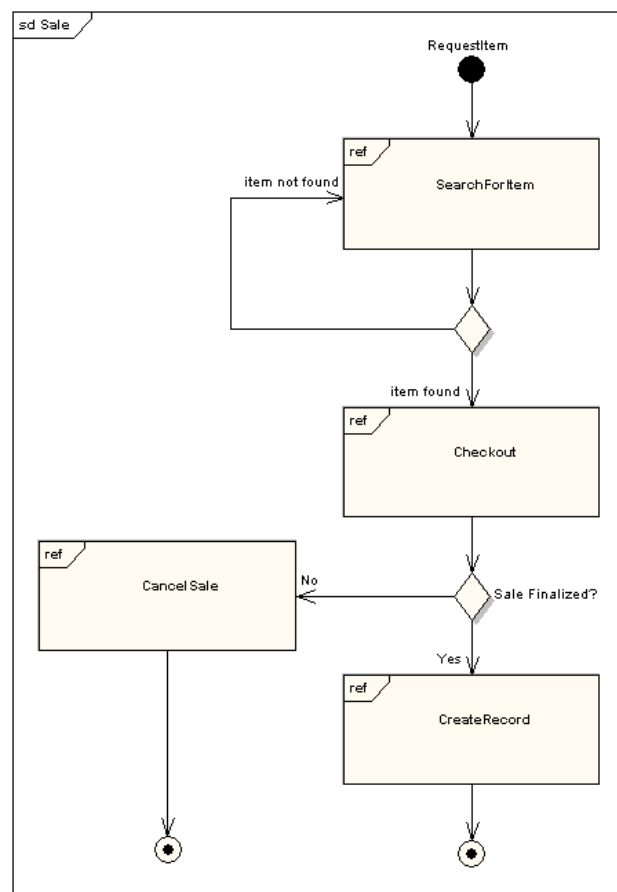


Sumber : Haviluddin (2011:5)

Gambar II.10. Notasi *Communication Diagram*

b. *Interaction Overview Diagram*

Interaksi *overview* diagram berfokus pada gambaran aliran kendali interaksi dimana node adalah interaksi atau kejadian interaksi.

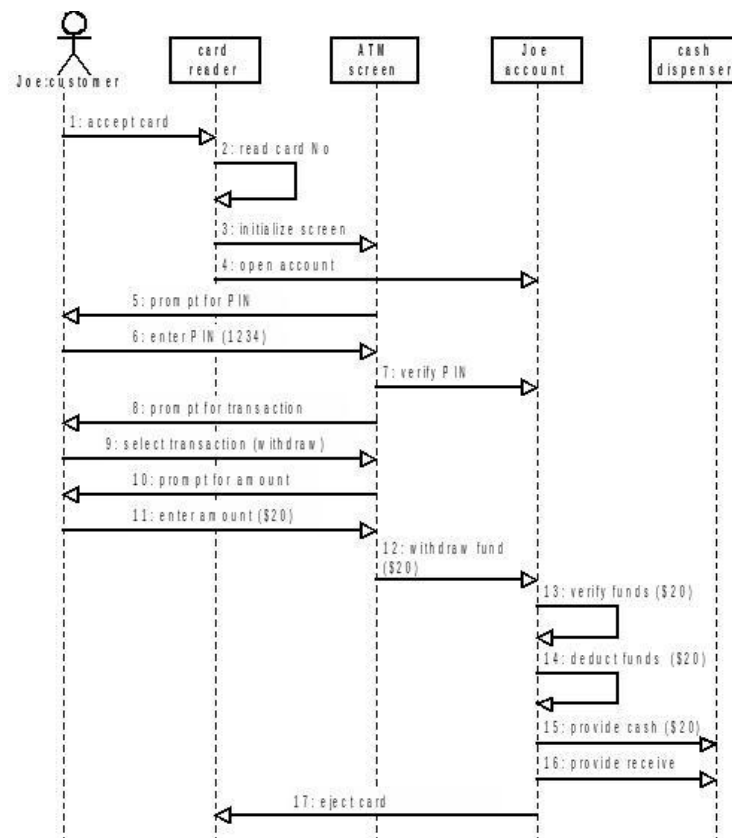


Sumber : Haviluddin (2011:5)

Gambar II.11. Notasi *Overview Diagram*

c. *Sequence Diagram*

Sequence diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence* diagram adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case* diagram.

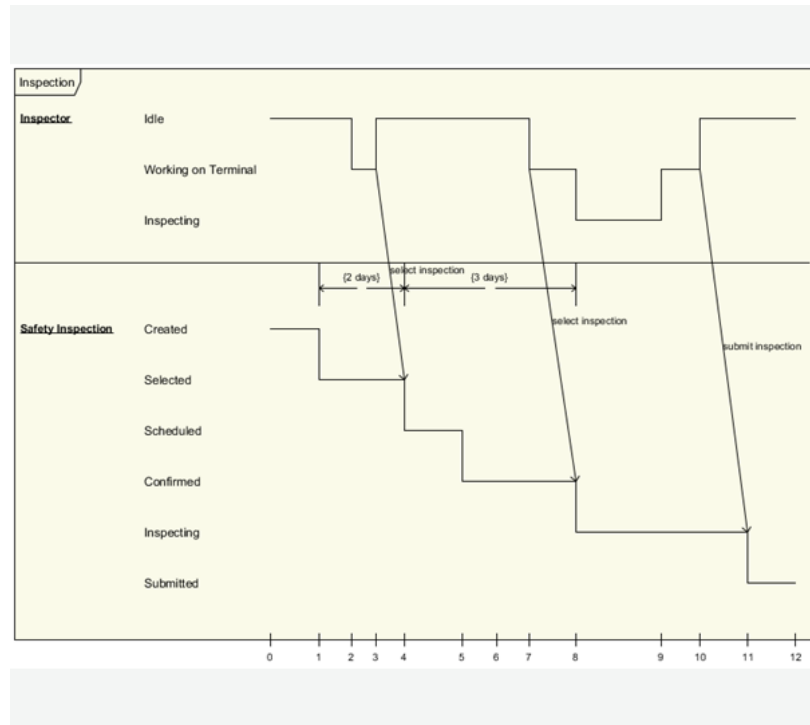


Sumber : Havaluddin (2011:5)

Gambar II.12. Notasi *Sequence Diagram*

d. *Timing Diagram*

Timing diagram di UML didasarkan pada diagram waktu *hardware* awalnya dikembangkan oleh para insinyur listrik.

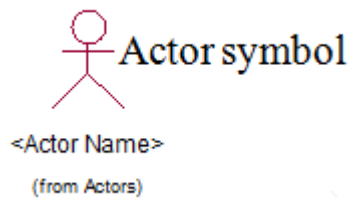


Sumber : Haviluddin (2011:6)

Gambar II.13. Notasi *Timing Diagram*

Untuk menggambarkan analisa dan desain diagram, UML memiliki seperangkat notasi yang akan digunakan ke dalam tiga kategori diatas yaitu *struktur diagram*, *behaviour diagram* dan *interaction diagram*. Berikut beberapa notasi dalam UML diantaranya :

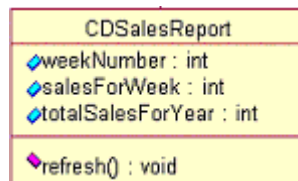
1. *Actor*; menentukan peran yang dimainkan oleh *user* atau sistem lain yang berinteraksi dengan subjek. *Actor* adalah segala sesuatu yang berinteraksi langsung dengan sistem aplikasi komputer, seperti orang, benda atau lainnya. Tugas *actor* adalah memberikan informasi kepada sistem dan dapat memerintahkan sistem untuk melakukan sesuatu tugas.



Sumber : Haviluddin (2011:6)

Gambar II.14. Notasi *Actor*

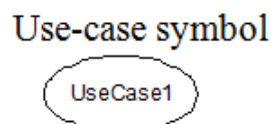
2. *Class* diagram; Notasi utama dan yang paling mendasar pada diagram UML adalah notasi untuk mempresentasikan suatu *class* beserta dengan atribut dan operasinya. *Class* adalah pembentuk utama dari sistem berorientasi objek.



Sumber : Haviluddin (2011:6)

Gambar II.15. Notasi *Class*

3. *Use Case* dan *use case spesification*; *Use case* adalah deskripsi fungsi dari sebuah sistem perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai.



Sumber : Haviluddin (2011:6)

Gambar II.16. Notasi *Use Case*

4. *Realization*; *Realization* menunjukkan hubungan bahwa elemen yang ada di bagian tanpa panah akan merealisasikan apa yang dinyatakan oleh elemen yang ada di bagian dengan panah.



Sumber : Haviluddin (2011:6)

Gambar II.17. Notasi *Realization*

5. *Interaction*; *Interaction* digunakan untuk menunjukkan baik aliran pesan atau informasi antar obyek maupun hubungan antar obyek.



Sumber : Haviluddin (2011:6)

Gambar II.18. Notasi *Interaction*

6. *Dependency*; *Dependency* merupakan relasi yang menunjukkan bahwa perubahan pada salah satu elemen memberi pengaruh pada elemen lain. Terdapat 2 stereotype dari *dependency*, yaitu *include* dan *extend*. *Include* menunjukkan bahwa suatu bagian dari elemen (yang ada digaris tanpa panah) memicu eksekusi bagian dari elemen lain (yang ada di garis dengan panah). *Extend* menunjukkan bahwa suatu bagian dari elemen di garis tanpa panah bisa disisipkan ke dalam elemen yang ada di garis dengan panah.



Sumber : Haviluddin (2011:7)

Gambar II.19. Notasi *Dependency*

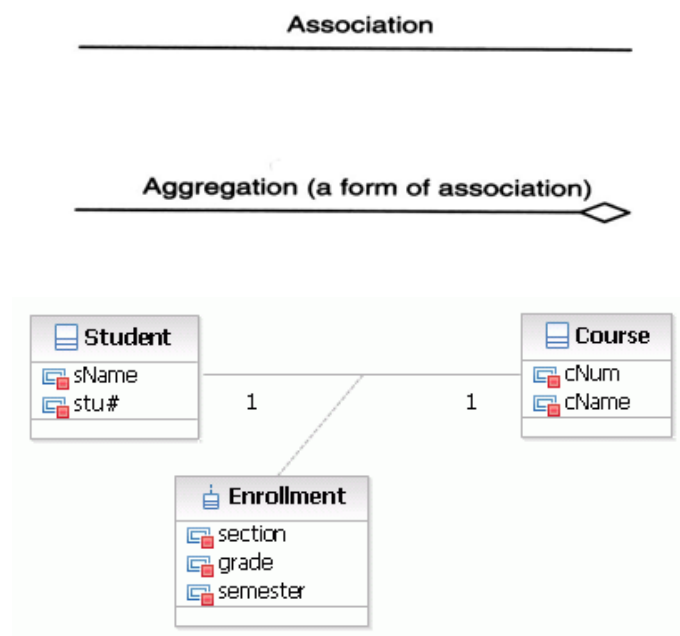
7. *Note*; *Note* digunakan untuk memberikan keterangan atau komentar tambahan dari suatu elemen sehingga bisa langsung terlampir dalam model. *Note* ini bisa disertakan ke semua elemen notasi yang lain.



Sumber : Haviluddin (2011:7)

Gambar II.20. Notasi *Note*

8. *Association*; *Association* menggambarkan navigasi antar class (*navigation*), berapa banyak obyek lain yang bisa berhubungan dengan satu obyek (*multiplicity* antar *class*) dan apakah suatu *class* menjadi bagian dari *class* lainnya (*aggregation*).



Sumber : Haviluddin (2011:7)

Gambar II.21. Notasi *Association*

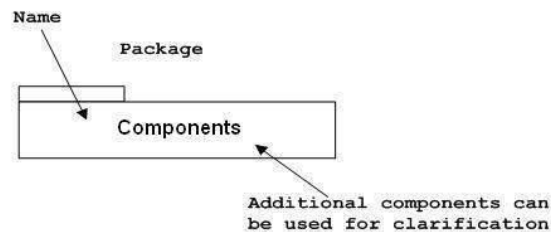
9. *Generalization*; *Generalization* menunjukkan hubungan antara elemen yang lebih umum ke elemen yang lebih spesifik.



Sumber : Haviluddin (2011:7)

Gambar II.22. Notasi *Generalization*

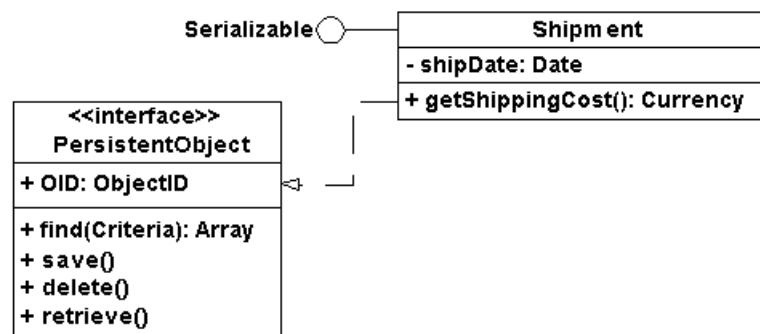
10. *Package*; *Package* adalah mekanisme pengelompokkan yang digunakan untuk menandakan pengelompokkan elemen-elemen model.



Sumber : Haviluddin (2011:7)

Gambar II.23. Notasi *Package*

11. *Interface*; *Interface* merupakan kumpulan operasi berupa implementasi dari suatu *class*. Atau dengan kata lain implementasi operasi dalam *interface* dijabarkan oleh operasi di dalam *class*.



Sumber : Haviluddin (2011:7)

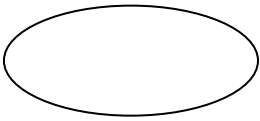
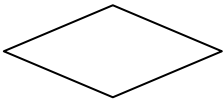
Gambar II.24. Notasi *Interface*

2.1.5 ERD (*Entity Relationship Diagram*)

Menurut Pratama (2014:49) mengemukakan bahwa “ERD (*Entity Relationship Diagram*) adalah diagram yang menggambarkan keterkaitan antar tabel beserta dengan *field-field* di dalamnya pada suatu database sistem”..

Komponen-komponen yang digunakan dalam ERD adalah sebagai berikut :

Tabel II.1.
Simbol *Entity Realtionship Diagram*

No	Simbol	Keterangan Fungsi
1.	Entitas 	Persegi panjang menyatakan himpunan entitas adalah orang, kejadian atau berada dimana data akan dikumpulkan.
2.	Atribut 	Atribut merupakan informasi yang diambil tentang sebuah entitas.
3.	Relasi 	Belah ketupat menyatakan himpunan relasi merupakan hubungan antar entitas.
4.	Link 	Garis sebagai penghubung antara himpunan, relasi dan himpunan entitas dengan atributnya.

Sumber : Aqil (2010:6)

2.2 Penelitian Terkait

Perpustakaan sekolah menengah pertama Negeri 1 Donorojo dalam pengelolaan data administrasi perpustakaan masih banyak menemukan masalah yang bisa menghambat proses pelayanan kepada siswa-siswi. Dengan masih ditemukan masalah dalam pengelolaan data administrasi pada perpustakaan Sekolah Menengah Pertama Negeri 1 Donorojo maka dibuatlah suatu perangkat lunak yang bisa memberi solusi untuk membantu petugas perpustakaan dalam mengelola data administrasi perpustakaan Sekolah Menengah Pertama Negeri 1 Donorojo. Hasil penelitian ini dapat memberi solusi untuk meningkatkan pelayanan dan kinerja petugas perpustakaan dalam hal pengelolaan data administrasi perpustakaan serta mempercepat transaksi peminjaman dan pengembalian buku oleh siswa (Hendrianto, 2014:57).

Perpustakaan mempunyai tanggung jawab yang besar terhadap peningkatan dan pengembangan minat dan kegemaran membaca. Teknologi informasi dan komunikasi atau ICT (*Information and Communication Technology*) telah menjadi bagian yang tidak terpisahkan dari kehidupan global. Pada penelitian ini dibangun Sistem Perpustakaan Digital Buku Elektronik yang di mana buku elektronik yang diunggah sudah dibuat dengan format PDF dan hanya dapat diakses lokal di lingkungan Universitas Stikubank Semarang (Zuliarso dan Februariyanti, 2013:46).