

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Sistem merupakan kumpulan dari unsur atau elemen-elemen yang saling berkaitan/berinteraksi dan saling memengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu. (Asbon hendra : 2012 : 157).

II.2. Karakteristik Sistem

Menurut (Asbon hendra : 2012 : 158-160) Ada beberapa karakteristik sistem adalah sebagai berikut :

a. Komponen (*Component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

b. Batas Sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya.

c. Lingkungan Luar Sistem (*Environment*)

Environment merupakan segala sesuatu di luar batas sistem yang memengaruhi operasi dari suatu sistem.

d. Penghubung Sistem (*Interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya untuk membentuk satu kesatuan sehingga sumber-sumber daya mengalir dari subsistem yang satu ke subsistem yang lainnya.

e. Masukan Sistem (*Input*)

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa Masukan Perawatan (*Maintenance Input*) adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi.

f. Keluaran Sistem (*Output*)

Merupakan hasil dari energi yang diolah oleh sistem meliputi output yang berguna.

g. Pengolah Sistem (*Process*)

Merupakan bagian yang memproses masukan untuk menjadi keluaran yang di inginkan.

h. Tujuan Sistem (*Goal*)

Suatu sistem akan dikatakan berhasil kalau pengoperasian sistem itu mengenai sasaran atau tujuannya.

II.3. Informasi

Informasi merupakan data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat. (Asbon hendra : 2012 : 167).

II.4. Sistem informasi

Sistem informasi adalah sekumpulan prosedur manual atau terkomputerisasi yang mengumpulkan/mengambil, mengolah, menyimpan, dan menyebarkan informasi dalam mendukung pengambilan dan kendali keputusan. (Asbon hendra : 2012 : 168-169).

II.5. Sistem Informasi Geografis

Pada dasarnya, istilah sistem informasi geografis (SIG) merupakan gabungan tiga unsur pokok : sistem, informasi, dan geografis. Dengan demikian pengertian terhadap ketiga unsur pokok ini sangat membantu dalam memahami SIG.

Istilah “Geografis” merupakan bagian dari spasial. Istilah ini sering digunakan secara bergantian/tertukar satu sama lainnya hingga muncullah istilah geospasial. Istilah ini mengandung pengertian yang kurang lebih serupa di dalam konteks SIG. Penggunaan kata “Geografis” mengandung pengertian suatu hal mengenai bumi : baik permukaan dua dimensi maupun tiga dimensi.

Dengan pengertian sistem informasi, maka SIG juga dapat dikatakan sebagai suatu kesatuan formal yang terdiri dari berbagai sumber daya fisik dan logika yang berkenaan dengan objek – objek yang terapat di permukaan bumi. Jadi, SIG merupakan sejenis perangkat lunak, perangkat keras, manusia, prosedur, basis data, dan fasilitas jaringan komunikasi yang di gunakan untuk memfasilitasi proses – proses pemasukan, penyimpanan, manipulasi, menampilkan, dan keluaran ata/informasi geografis berikut atribut – atributnya. (Edy Prahasa : 2014)

II.6. MapInfo

Mapinfo adalah salah satu software pengolah sistem informasi geografis (SIG). Mapinfo diminati oleh pemakai SIG karena mempunyai karakteristik yang menarik, seperti mudah digunakan, harga relative murah, tampilan interaktif, user friendly, dan dapat dimodifikasi menggunakan bahasa skrip yang dimiliki.

Kemudahan lain MapInfo tidak memerlukan dukungan hardware yang terlalu tinggi, sehingga hampir semua spesifikasi computer dapat menggunakan MapInfo. Software ini memiliki kemampuan untuk mengorganisasi, memanipulasi, dan menganalisa data. (Wahana Komputer : 2015)

II.7. PHP (*Hypertext Preprocessor*)

PHP merupakan singkatan dari “*Hypertext Preprocessor*” yang merupakan kependekan dari *Personal Home Page* (Situs Personal) dan dikembangkan pertama kali tahun 1995 oleh Rasmus Lerdoff dan pada saat PHP masih bernama F1 (*Form Interpreter*), yang wujudnya berupa sekumpulan *script* yang digunakan untuk mengolah data form dari web. PHP adalah sebuah bahasa *scripting* server-side yang terpasang pada HTML. Sebagian besar syntaknya mirip dengan bahasa pemrograman C, java, asp, dan perl, ditambah dengan beberapa fungsi PHP yang spesifik dan mudah di mengerti. PHP digunakan untuk membuat tampilan web menjadi dinamis, dengan PHP kita bias menampilkan atau menjalankan beberapa file dalam 1 file dengan cara di *include* atau *require*. (Riesda Ganevi : 2011 : 39)

II.8. Macromedia Dreamweaver

Macromedia Dreamweaver merupakan aplikasi yang digunakan sebagai HTML editor professional untuk mendesain web secara visual. Aplikasi ini juga biasa dikenal dengan istilah WYSIWYG (*What You See Is What You Get*), yang intinya adalah anda tidak harus berurusan dengan tag – tag HTML untuk membuat sebuah site dan dapat melihat hasil desainnya secara langsung.

Dengan kemampuan fasilitas yang optimal dalam jendela Design akan memberikan kemudahan untuk para web desainer pemula sekalipun. Kemampuan *Adobe Dreamweaver* untuk berinteraksi dengan beberapa bahasa pemrograman seperti *PHP*, *ASP*, *JavaScript*, dan yang lainnya juga memberikan fasilitas maksimal kepada desainer web dengan menyertakan bahasa pemrograman di desainnya. (Madcoms : 2013 : 2)

II.9. MySQL

MySQL adalah suatu database server yang sangat terkenal di dunia an merupakan open source SQL database (databse SQL yang open source). *MySQL* merupakan database server dimana pemrosesan data terjadi di server dan *client* hanya mengirim data dan memindah data. Oleh karena pemrosesan terjadi di server sehingga pengaksesan data tidak terbatas. Pengaksesan dapat dilakukan dimana saja dan oleh siapa saja dengan catatan komputer telah terhubung ke server. Lain halnya dengan database desktop dimana segala pemrosesan data seperti penambahan data ataupun penghapusan data harus dilakukan pada komputer yang bersangkutan. (Riesda Ganevi : 2014 : 39)

II.10. Entity Relationship Diagram

Pada dasarnya ERD (Entity Relationship Diagram) adalah sebuah diagram yang secara konseptual memetakan hubungan antar penyimpanan pada diagram DFD diatas, ERD ini digunakan untuk melakukan permodelan terhadap struktur data dan hubungannya. Penggunaan ERD ini dilakukan untuk mengurangi tingkat kerumitan penyusunan sebuah database yang baik.

Entity dapat berarti sebuah obyek yang dapat dibedakan dengan obyek lainnya. Obyek tersebut dapat memiliki komponen-komponen data (atribut atau field) yang membuatnya dapat dibedakan dari obyek yang lain. Dalam dunia database entity memiliki atribut yang menjelaskan karakteristik dari entity tersebut. Wahana Komputer (2010 : 30)

II.11. Normalisasi Data

Wahana Komputer (2010 : 32-35) Normalisasi data adalah proses dimana tabel-tabel pada database dites dalam hal kesalingtergantungan di antara field-field pada sebuah tabel. Misalnya jika pada sebuah tabel terdapat ketergantungan terhadap lebih dari satu field dalam tabel tersebut, maka tabel tersebut harus dipecah menjadi banyak tabel. Banyaknya tabel pecahannya tergantung pada seberapa banyak ketergantungannya. Tiap tabel hanya boleh memiliki sbuah field kunci yang menjadi ketergantungan dari field lainnya dalam tabel tersebut.

Pada proses normalisasi data, aturan yang dijadikan acuan adalah metode ketergantungan fungsional. Teorinya adalah bahwa tiap kolom dalam sebuah tabel selalu memiliki hubungan yang unik dengan sebuah kolom kunci.

Ada beberapa langkah dalam normalisasi table, yaitu :

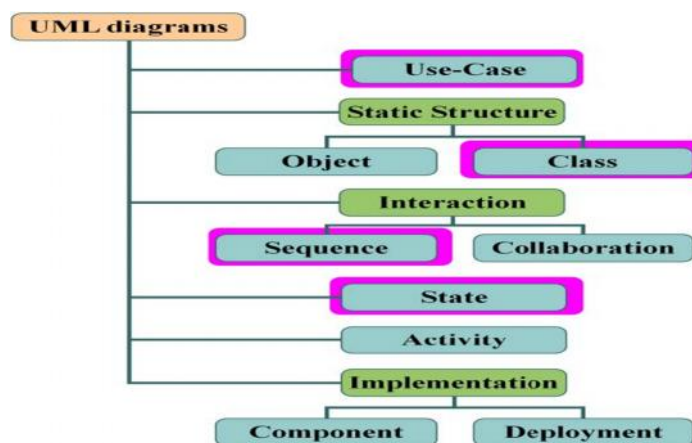
1. Decomposition, dekomposisi adalah proses mengubah bentuk table supaya memenuhi syarat tertentu sebagai table yang baik.
2. Bentuk tidak normal, pada bentuk ini data yang ada Pada tiap entry (diambil atributnya) masih ditampung dalam satu table besar. Data yang ada pada table ini masih ada yang redundansi dan ada juga yang kosong.
3. Normal form pertama, pada tahapan ini table di-ekomposisi dari table tidak normal yang kemudian dipisahkan menjadi table - table kecil yang memiliki criteria tidak memiliki atribut yang bernilai ganda dan komposit.
4. Normal form kedua, pada tahapan ini table dianggap memenuhi normal kedua jika pada table tersebut semua atribut yang bukan kunci primer bergantung penuh terhadap kunci primer table tersebut.
5. Normal form ketiga, setiap atribut pada table selain kunci primer atau kunci utama harus bergantung penuh pada kunci utama. (Wahana Komputer 2010 : 32-35)

II.12. UML (*Unified Modeling Language*)

Unified Modelling Language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual. Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek.

Secara filosofi UML diilhami oleh konsep yang telah ada yaitu konsep permodelan *Object Oriented* karena konsep ini menganalogikan sistem seperti kehidupan nyata yang didominasi oleh obyek dan digambarkan atau dinotasikan dalam simbol-simbol yang cukup spesifik. (Haviluddin; 2011: 01).

Berikut gambar dari diagram UML



Gambar II.1. Diagram UML

Sumber : Haviluddin (2011 : 02)

II.12.1. Tujuan Pemanfaatan UML

Tujuan dari penggunaan diagram seperti diungkapkan oleh Schmuller J. (2004), *“The purpose of the diagrams is to present multiple views of a system; this set of multiple views is called a model”*.

Berikut tujuan utama dalam desain UML adalah (Haviluddin; 2011: 02) :

1. Menyediakan bagi pengguna (analisis dan desain sistem) suatu bahasa pemodelan visual yang ekspresif sehingga mereka dapat mengembangkan dan melakukan pertukaran model data yang bermakna.
2. Menyediakan mekanisme yang spesialisasi untuk memperluas konsep inti.

3. Karena merupakan bahasa pemodelan visual dalam proses pembangunannya maka UML bersifat independen terhadap bahasa pemrograman tertentu.
4. Memberikan dasar formal untuk pemahaman bahasa pemodelan.
5. Mendorong pertumbuhan pasar terhadap penggunaan alat desain sistem yang berorientasi objek (OO).
6. Mendukung konsep pembangunan tingkat yang lebih tinggi seperti kolaborasi, kerangka, pola dan komponen terhadap suatu sistem.
7. Memiliki integrasi praktik terbaik.

II.12.2. Komponen-komponen UML

Sejauh ini para pakar merasa lebih mudah dalam menganalisa dan mendesain atau memodelkan suatu sistem karena UML memiliki seperangkat aturan dan notasi dalam bentuk grafis yang cukup spesifik.

Komponen atau notasi UML diturunkan dari 3 (tiga) notasi yang telah ada sebelumnya yaitu Grady Booch, OOD (*Object-Oriented Design*), Jim Rumbaugh, OMT (*Object Modelling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

Pada UML versi 2 terdiri atas tiga kategori dan memiliki 13 jenis diagram yaitu (Haviluddin; 2011: 03) :

1. Struktur Diagram

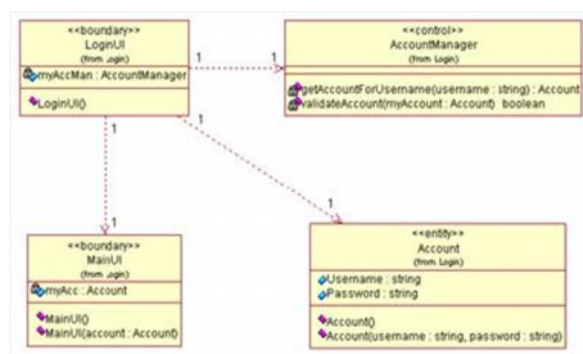
Menggambarakan elemen dari spesifikasi dimulai dengan kelas, obyek, dan hubungan mereka, dan beralih ke dokumen arsitektur logis dari suatu sistem. Struktur diagram dalam UML terdiri atas :

1) *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



Gambar II.2. Notasi *class diagram*

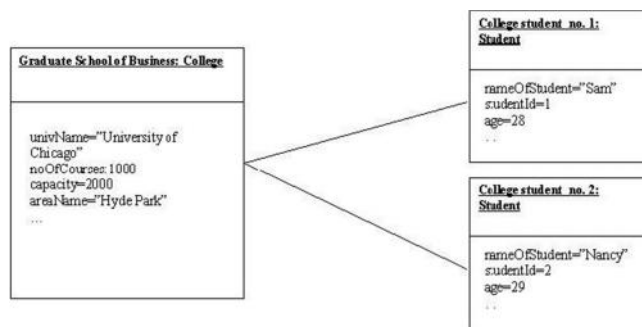
Sumber : Haviluddin (2011 : 03)

2) *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.

Berikut notasi *object diagram*.

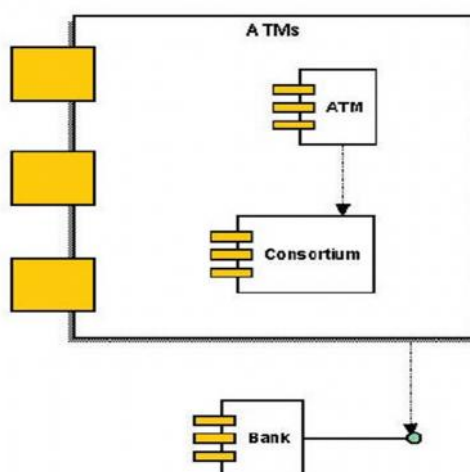


Gambar II.3. Notasi *object diagram*

Sumber : Haviluddin (2011 : 03)

3) *Component diagram*

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

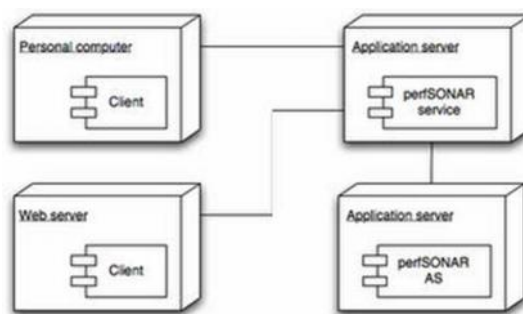


Gambar II.4. Notasi *component diagram*

Sumber : Haviluddin (2011 : 03)

4) *Deployment diagram (Collaboration diagram in version 1.x)*

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.

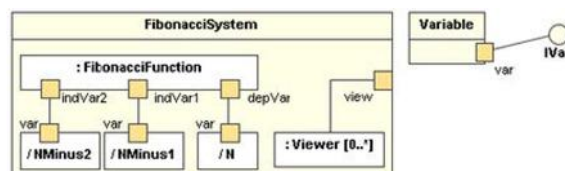


Gambar II.5. Notasi *deployment diagram*

Sumber : Haviluddin (2011 : 04)

5) *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.

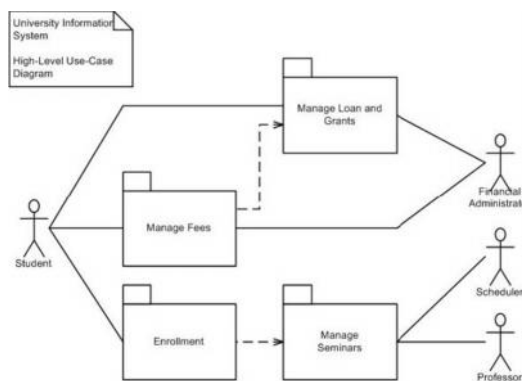


Gambar II.6. Notasi *composite diagram*

Sumber : Haviluddin (2011 : 04)

6) *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.



Gambar II.7. Notasi *Package diagram*

Sumber : Haviluddin (2011 : 04)

2. *Behavior Diagram*

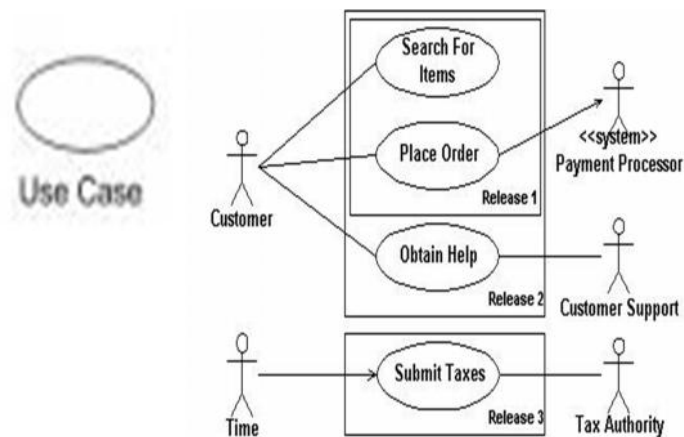
Menggambarkan ciri-ciri *behavior/metode/* fungsi dari sebuah sistem atau *business process*. *Behavior diagram* dalam UML terdiri atas

7) *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

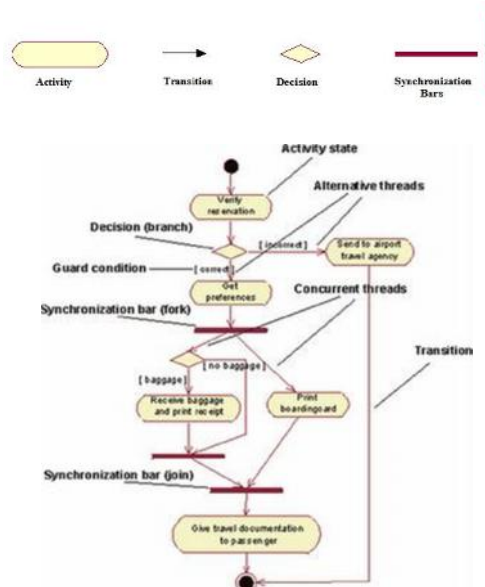


Gambar II.8. Notasi Use case diagram

Sumber : Haviluddin (2011 : 04)

8) Activity diagram

Menggambarkan aktifitas-aktifitas, objek, state, transisi state dan event. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas.

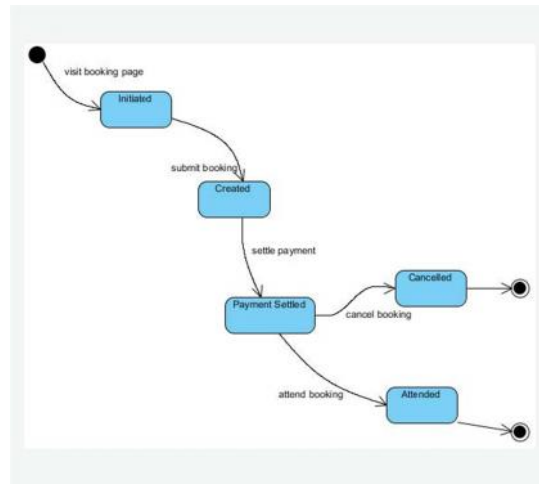


Gambar II.9. Notasi Activity diagram

Sumber : Haviluddin (2011 : 04)

9) *State Machine diagram (State chart diagram in version 1.x)*

Menggambarkan *state*, transisi *state* dan *event*.



Gambar II.10. Notasi *State Machine diagram*

Sumber : Haviluddin (2011 : 05)

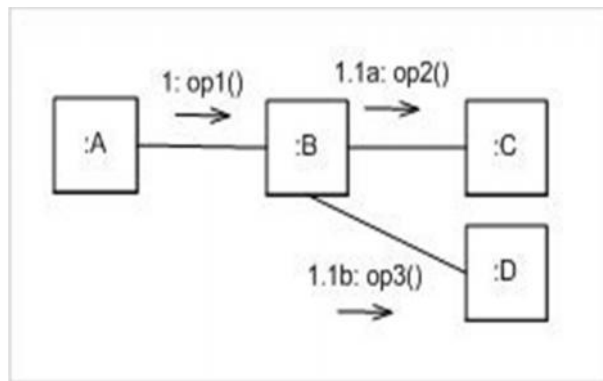
3. Interaction diagram

Bagian dari behavior diagram yang menggambarkan interaksi objek.

Interaction diagram dalam UML terdiri atas :

10) *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.

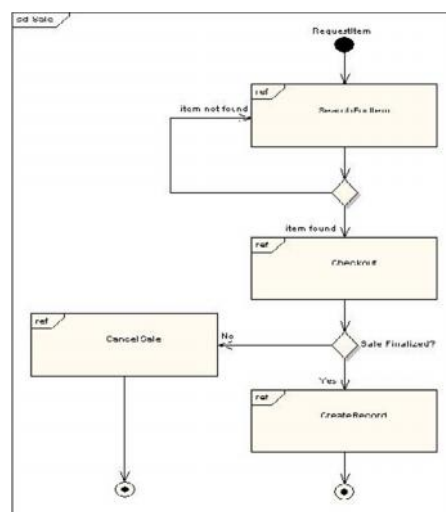


Gambar II.11. Notasi *communication diagram*

Sumber : Haviluddin (2011 : 05)

11) *Interaction Overview diagram*

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

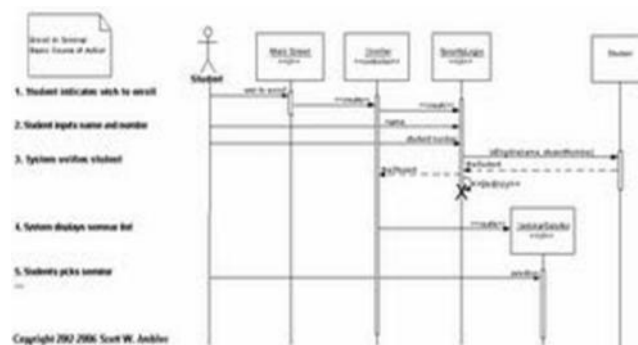


Gambar II.12. Notasi *overview diagram*

Sumber : Haviluddin (2011 : 05)

12) 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*.

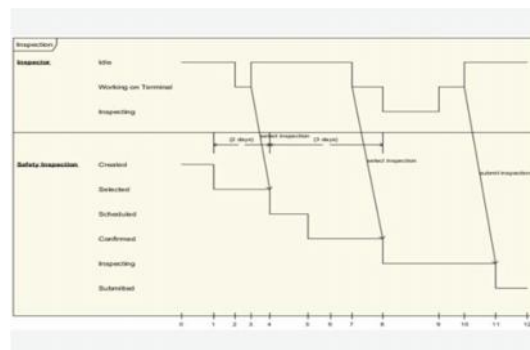


Gambar II.13. Notasi Sequence diagram

Sumber : (Haviluddin 2011 : 05)

13) Timing diagram

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



Gambar II.14. Notasi Timing diagram

Sumber : Haviluddin (2011 : 05)

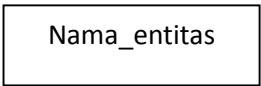
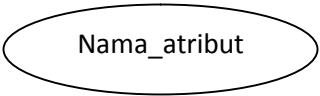
II.10. ERD (*Entity Relationship Diagram*)

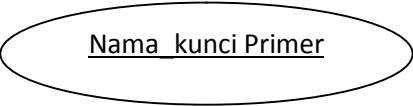
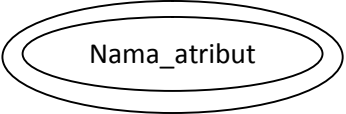
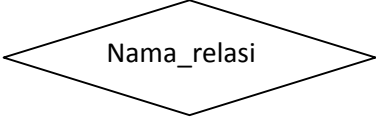
ERD (*Entity Relationship Diagram*) dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD (*Entity Relationship Diagram*) digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data perlu menggunakan ERD (*Entity Relationship Diagram*) (Rosa A.S-M. Shalahuddin;2011:49).

II.10.1. Simbol-simbol ERD (*Entity Relationship Diagram*)

Adapaun simbol-simbol ERD (*Entity Relationship Diagram*) ditunjukkan pada tabel II.3

Tabel II.3. Simbol-simbol ERD (*Entity Relationship Diagram*)

Simbol	Deskripsi
Entitas/ <i>entity</i> 	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data
Atribut 	<i>Field</i> atau kolom data yang perlu disimpan dalam suatu entitas

Atribut kunci primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id
Atribut multivalued/ <i>multivalued</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu
Relasi 	Relasi yang menghubungkan antarentitas; relasi biasanya diawali dengan kata kerja
Asosiasi/ <i>Association</i> 	Penghubung antar relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian

Sumber: (Rossa A.S-M. Shalahuddin;2011:49-50)

II.11. Normalisasi

Normalisasi merupakan bentuk transformasi tinjauan pemakai yang kompleks dimana data tersimpan ke dalam sekumpulan bagian struktur data yang kecil dan stabil. Normalisasi merupakan kegiatan perlakuan data untuk

menyederhanakan sebuah tabel data agar lebih terstruktur dan mudah digunakan (Idris Asmuni;2005:4).

Pemahaman normalisasi merupakan keterampilan yang harus diperhatikan oleh programmer sistem dengan mengadakan pengamatan dan analisis yang memadai pada formulir atau dokumen masukan menjadi laporan utama (Idris Asmuni;2005:4).

Tahapan normalisasi terdiri dari beberapa bentuk yaitu sebagai berikut:

1. Bentuk Normal Pertama (1NF/ *First Normal Form*)

Bentuk normal pertama memiliki ciri yaitu data berbentuk *flat file* (file datar), *record* disusun sesuai kedatangan, masih mungkin terjadi penyimpangan data (*anomali data*). *Anomali data* dapat berupa *insert anomali*, *delete anomali*, *update anomali* dan *redudancy data* (data duplikat).

2. Bentuk Normal Kedua (2NF/ *Second Normal Form*)

Bentuk normal kedua memiliki ciri yaitu tidak terjadi *anomali data*, setiap *field/ atribut* bukan kunci harus tergantung fungsi (*functional dependency*) terhadap *field/ atribut* kunci, masih mungkin terjadi *transitive dependency* (*field* bukan kunci tergantung pada *field* bukan kunci dalam satu tabel).

3. Bentuk Normal Ketiga (3NF/ *Third Normal Form*)

Bentuk normal ketiga memiliki syarat yaitu tabel harus tidak terdapat *transitive dependency* (*field* bukan kunci tergantung pada *field* bukan kunci dalam satu tabel).

4. Bentuk Normal Boyce Codd (BCNF/ *Boyce Codd Normal Form*)

Pada tahap ini menghilangkan ketergantungan *field* bukan kunci secara persial (bagian) kunci dalam satu tabel. Apabila pada normal ketiga tidak lagi ditemukan *field* bukan kunci tergantung secara persial dalam satu tabel, maka normal ketiga juga merupakan bentuk BCNF (Ir. Yuniar Supardi;2008:10-12).