

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Perancangan**

Pengertian perancangan menurut J.C Jones dalam bukunya "*Design Methods*" yaitu : melakukan suatu pengkajian proses desain sebagai suatu penyelidikan untuk metode-metode yang akan memperbaiki mutu desain. Pendekatan perancangan dimulai dari Pendekatan terstruktur dilengkapi dengan alatalat (*tools*) dan teknik-teknik (*techniques*) yang dibutuhkan dalam pengembangan sistem, sehingga hasil akhir dari sistem yang dikembangkan akan diperoleh sistem yang strukturnya didefinisikan dengan baik dan jelas. Melalui pendekatan terstruktur, permasalahan yang kompleks di organisasi dapat dipecahkan dan hasil dari sistem akan mudah untuk dipelihara, *fleksibel*, lebih memuaskan pemakainya, mempunyai dokumentasi yang baik, tepat waktu, sesuai dengan anggaran biaya pengembangan, dapat meningkatkan produktivitas dan kualitasnya akan lebih baik (bebas kesalahan). (Andreas Andoyo, dkk:2014;40)

#### **II.2. Aplikasi**

Aplikasi berasal dari kata *application* yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju. Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer

langsung untuk melakukan tugas yang diinginkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media (Fricles Ariwisanto Sianturi; 2013: 43).

### **II.3. IP Address**

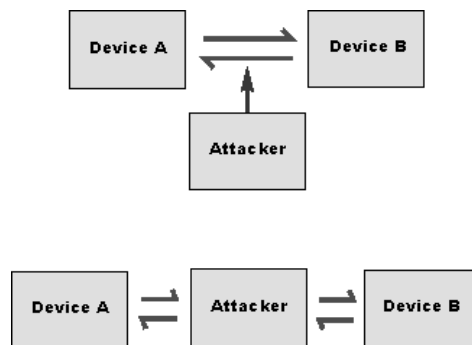
*IP (Internet Protocol) address* merupakan alamat yang diberikan kepada komputer-komputer yang terhubung dalam suatu jaringan. *IP Address* terdiri dari dua bagian, yaitu *Network ID* dan *Host ID*. *Network ID* menentukan alamat dalam jaringan (*Network address*), sedangkan *Host ID* menentukan alamat dari peralatan jaringan yang sifatnya unik untuk membedakan antara satu mesin dengan mesin yang lain. Ibarat sebuah alamat rumah, *Network ID* seperti alamat rumah dan *Host ID* seperti nomor rumah. (Madcoms:2013;21)

### **II.4. Man in the middle (MITM) attack**

*Man in the middle Attack (MITM)* dalam kriptografi, penyerangan dengan metode *man in the middle* disingkat MITM adalah sebuah bentuk penyadapan dimana sang penyerang membuat sebuah koneksi yang independen antara korban dan mengirimkan pesan diantara para korban yang mengira mereka sedang berkomunikasi pada sebuah koneksi privat dimana sebenarnya semua percakapan tersebut diatur oleh sang penyerang. Pada metode ini, sang penyerang diharuskan untuk bisa menyadap semua pesan yang dikomunikasikan antara kedua korban dan memasukkan pesan baru. Penyerangan ini hanya bisa sukses jika dan hanya jika sang penyerang bisa menyamar menjadi setiap *end point* dari korban dengan persetujuan yang lainnya. (Kurnia Ramadhan:2011;2)

Istilah *man in the middle attack* seseorang bisa membaca, menyisipkan dan memasukan data antara dua komputer yang saling terhubung. Pada dasarnya teknis implementasinya adalah sebagai berikut :

1. Sewaktu target membuka sebuah *website*, misalnya sebagai contoh disini adalah paypal. Pada alamat <https://www.paypal.com>, yang terlihat pada komputer target adalah sebuah halaman *website* paypal yang normal dan baik-baik saja.
2. Namun, akan muncul sebuah pesan ada masalah dengan sertifikat keamanan *website* tersebut.(Efv Zam:2013;110)



**Gambar II.1 Man In The Middle Attack Diagram**

Sumber : (Efv Zam;2013;110)

## **II.5. Address Resolution Protocol (ARP)**

*Address Resolution Protocol* adalah sebuah protokol pada jaringan komputer yang digunakan untuk menentukan alamat *hardware* (pada *Link Layer*) dari host jaringan hanya ketika alamat pada *Internet Layer* atau *Network Layer* diketahui. Fungsi ini sangat penting pada jaringan *local area network* dan juga lainnya ketika *next-hop* dari *router* harus bisa ditentukan.

Sebagai contoh, bila satu komputer pada sebuah jaringan LAN ingin mengirimkan paket kepada mesin dengan IP 192.168.0.23, komputer itu memerlukan alamat MAC dari komputer

dengan IP tersebut. Pertama, komputer akan melihat tabel ARP miliknya dan mencari apakah ada alamat MAC dari IP tersebut. Bila ada, dia bisa mengirim paket dan paket tersebut akan berjalan dari (misal) ethernet adaptornya ke kabel, switch, kabel, dan akhirnya sampai ke ethernet adapter sang tujuan, semua dituntun oleh alamat MAC yang didapat tadi. Bila sebelumnya dia tidak menemukan alamat MAC pada tabel ARP miliknya, maka sebelumnya dia akan mengirimkan paket secara broadcast kepada semua mesin di jaringan LAN menanyakan siapa pemilik IP 192.158.0.23. Ini adalah protokol ARP yang sebenarnya. (Kurnia Ramadhan:2011;2)

## **II.6. Pengertian Jaringan**

Jaringan komputer adalah sistem yang terdiri dari komputer-komputer, serta piranti-piranti yang saling terhubung sebagai satu kesatuan. Dengan dihubungkan piranti-piranti tersebut, alhasil dapat saling berbagi sumber daya antar satu piranti dengan piranti lainnya (Wahana Komputer:2013;2)

Dalam istilah komputer, jaringan merupakan penghubung antara dua komputer atau lebih yang tujuan utamanya adalah berbagi data. Betulkah jaringan komputer itu *hardware* dan *software*? Jawabnya adalah betul. Jaringan komputer adalah gabungan antara *hardware* dan *software*.

## **II.7. Data**

Dalam *Webster's New World's Dictionary* tertulis bahwa datum: *something known or assumed*. Artinya, datum (bentuk tunggal data) merupakan suatu yang diketahui/dianggap. Dengan demikian, data dapat memberi gambaran tentang suatu keadaan atau persoalan. Sedangkan, data menurut kamus *Oxford Dictionary* adalah *The Facts*. Jadi, dapat

disimpulkan bahwa data adalah sesuatu yang nyata diketahui atau dianggap yang dipakai untuk keperluan suatu analisa, diskusi, presentasi ilmiah atau tes statistik.

Jenis-jenis data dapat dibagi menurut sifatnya, menurut sumbernya, menurut cara memperolehnya dan menurut waktu pengumpulannya. Menurut sifatnya data dapat terbagi menjadi dua jenis, yaitu data kualitatif (non-metrik) dan data kuantitatif (metrik). Kemudian jenis data kualitatif terbagi lagi menjadi dua jenis, yaitu data nominal dan data ordinal. Begitu pula dengan jenis data kuantitatif terbagi menjadi dua jenis, yaitu data interval dan data rasio. (Johan Oscar Ong: 2013;11)

## **II.8. Pesan**

Pesan merupakan komunikasi pengiriman surat secara elektronik yang terhubung melalui jaringan yang saling terkoneksi. Internet yang merupakan media *global* yang dapat dengan mudah diakses oleh siapa saja, hal tersebut menjadikan penyampaian pesan menggunakan Pesan menjadi kurang aman. Kurang amannya pengiriman Pesan dapat terjadi karena ketika dilakukan pengiriman, Pesan yang dikirim berupa pesan asli sehingga apabila dilakukan pengambilan Pesan dari jalur pengiriman maka pesan akan dapat diketahui oleh pihak yang tidak berhak. Maka diperlukan sebuah keamanan pada sebuah data Pesan..( Sugeng Sutrisno:2012;2)

## **II.9. Visual Basic .Net**

Visual basic 2010 merupakan salah satu bagian dari produk pemrograman terbaru yang dikeluarkan oleh Microsoft, yaitu microsoft Visual Studio 2010. Sebagai produk lingkungan pengembangan terintegrasi atau IDE andalan yang di keluarkan oleh microsoft, visual studio 2010 menambahkan perbaikan-perbaikan fitur dan fitur baru yang lebih lengkap visual studio

pendahulunya, yaitu mirosoft visual studio 2008. (Wahana Komputer;2010;2)

Sedangkan menurut Aswan (2012 : 1) Visual basic 2010 adalah salah satu bagian dari microsoft visual studio 2010. Sebuah alat yang digunakan oleh pengembang windows dari berbagai level untuk mengembangkan dan membangun aplikasi yang bergerak diatas sistem .NET Framework, dengan menggunakan bahasa BASIC. Visual Basic menyediakan cara cepat dan mudah untuk membuat aplikasi.

Setiap generasi baru dari perangkat lunak bahasa pemrograman datang karena adanya keterbatasan dari generasi sebelumnya. Teknologi device, *hardware*, *Network* dan internet baru yang muncul menyebabkan bahasa pemrograman yang ada tidak lagi menjadi alat yang ideal untuk mengembangkan perangkat lunak yang dapat bekerja dengan teknologi baru tersebut (WAH[12]). Sekarang untuk pertama kalinya, platform pengembang perangkat lunak yang lengkap, Microsoft .NET telah didesain dari dasar dengan internet sebagai fokus utamanya (walaupun tidak secara eksklusif hanya untuk pengembang internet saja). Banyak inovasi baru yang berada dalam platform ini akan mengatasi keterbatasan dari tool-tool dan teknologi lama. Visual Basic .NET adalah pengembangan dari Visual basic sebelumnya. Kelebihan VB .NET 2010 terletak pada tampilannya yang lebih canggih dibandingkan dengan edisi Visual Basic sebelumnya. Selain memiliki kelebihan, VB .NET 2005 memiliki kekurangan. Kekurangan VB .NET 2005 yang terlihat jelas adalah beratnya aplikasi ini apabila dijalankan pada komputer yang memiliki spesifikasi sederhana.

## **II.10. UML (*Unified Modeling Language*)**

UML(*Unified Modeling Language*) yang merupakan metodologi kolaborasi antara metoda booch, OMT (*Object Modeling Technique*), serta OOSE (*Oriented Software Engineering*)

dan beberapa metoda lainnya, merupakan metodologi yang paling sering digunakan saat ini untuk mengadaptasi maraknya penggunaan bahasa “pemrograman berorientasi objek” (OOP). (Adi Nugroho;2009;4)

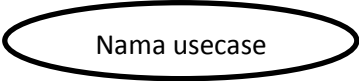


UML (*Unified Modeling Language*) adalah sebuah ”bahasa” yang telah menjadi standar dalam industry untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax/semantic*. 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. *Unified Modeling Language* biasa digunakan untuk :

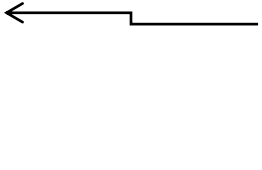
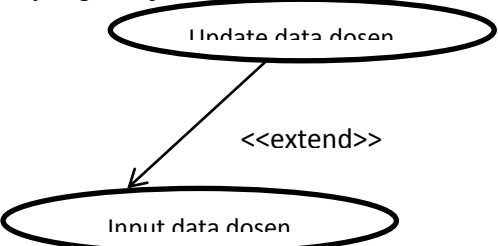
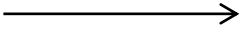
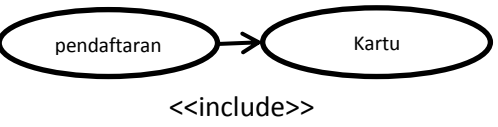
1. Menggambarkan batasan sistem dan fungsi–fungsi sistem secara umum, di buat dengan *use case* dan *actor*.
2. Menggambarkan kegiatan atau proses bisnis yang di laksanakan secara umum, di buat dengan *interaction diagrams*.
3. Menggambarkan representasi struktur *static* sebuah sistem dalam bentuk *class diagrams*.
4. Membuat model behavior “yang menggambarkan kebiasaan atau sifat sebuah sistem” dengan *state transition diagrams*.
5. Menyatakan arsitektur implementasi fisik menggunakan *component and development diagrams*.
6. Menyampaikan atau memperluas *functionality* dengan *stereotypes*. (Yuni Sugiarti; 2013 :36)

### **II.10.1. Use Case Diagram**

*Use case diagrams* merupakan pemodelan untuk menggambarkan kelakuan (*behavior*) sistem yang akan dibuat. Diagram *use case* mendeskripsikan sebuah interaksi antara satu atau lebih *actor* dengan sistem yang akan dibuat. Dengan pengertian yang cepat, diagram *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi–fungsi tersebut. Terdapat beberapa simbol dalam menggambarkan diagram *use case*, yaitu *use case*, *actor* dan relasi. Berikut adalah simbol–simbol yang ada pada diagram *use case*. (Yuni Sugiarti; 2013: 42)

**Tabel II.1 Simbol–simbol pada *Use Case Diagram***

Simbol	Deskripsi
Use case 	Fungsionalitas yang disediakan sistem sebagai unit–unit yang saling bertukar pesan antar unit atau <i>actor</i> ; biasanya ditanyakan dengan menggunakan kata kerja di awal frase nama <i>use case</i> .
Aktor 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari <i>actor</i> adalah gambar orang, tapi <i>actor</i> belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama <i>actor</i> .
Asosiasi/ <i>association</i> 	Komunikasi antara actor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
Extend	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i>

 <p data-bbox="191 491 347 520">&lt;&lt;extend&gt;&gt;</p>	<p data-bbox="571 197 1247 667">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, arah panah menunjuk pada <i>use case</i> yang dituju. Contoh :</p> 
<p data-bbox="191 932 289 961">Include</p>  <p data-bbox="191 1003 354 1033">&lt;&lt;include&gt;&gt;</p>	<p data-bbox="571 932 1247 1478">Relasi <i>use case</i> tambahan kesebuah <i>use case</i> dimana <i>use case</i> yang yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini. Ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i>, <i>include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, contoh :</p> 

Sumber: (Yuni Sugiarti; 2013)

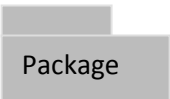
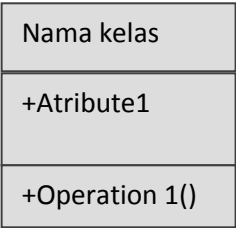

**II.10.2. Class Diagram**

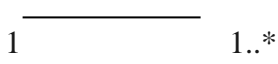
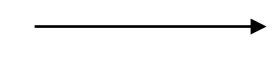
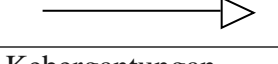
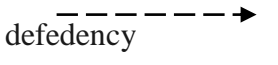
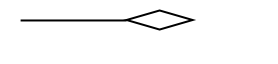
Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan di buat untuk membangun sistem. Kelas memiliki apa yang di sebut atribut dan metode atau operasi.

1. Atribut merupakan variabel-variabel yang di miliki oleh suatu kelas.
2. Atribut mendeskripsikan properti dengan sebaris teks di dalam kotak kelas tersebut.
3. Operasi atau metode adalah fungsi-fungsi yang di miliki oleh suatu kelas.

Diagram kelas mendeskripsikan jenis-jenis objek dalam sistem dan berbagai hubungan statis yang terdapat di antara mereka. Diagram kelas juga menunjukkan properti dan operasi sebuah kelas dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut.(Yuni Sugiarti; 2013: 57)

**Tabel II.2 Simbol-simbol *Class Diagram***

Simbol	Deskripsi
Package 	Package merupakan sebuah bungkusan dari satu atau lebih kelas
Operasi 	Kelas pada struktur sistem
Antarmuka / interface 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek

Asosiasi 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarah/directed asosiasi 	Relasi antar kelas dengan makna kelas yang satu di gunakan oleh kelas yang lain, asosiasi biasanya juga di sertai dengan <i>multiplicity</i> .
Generalisasi 	Relasi antar kelas dengan makna generalisasi – spesialisasi (umum khusus).
Kebergantungan defedency 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi 	Relasi antar kelas dengan makna semua bagian ( <i>whole-part</i> )

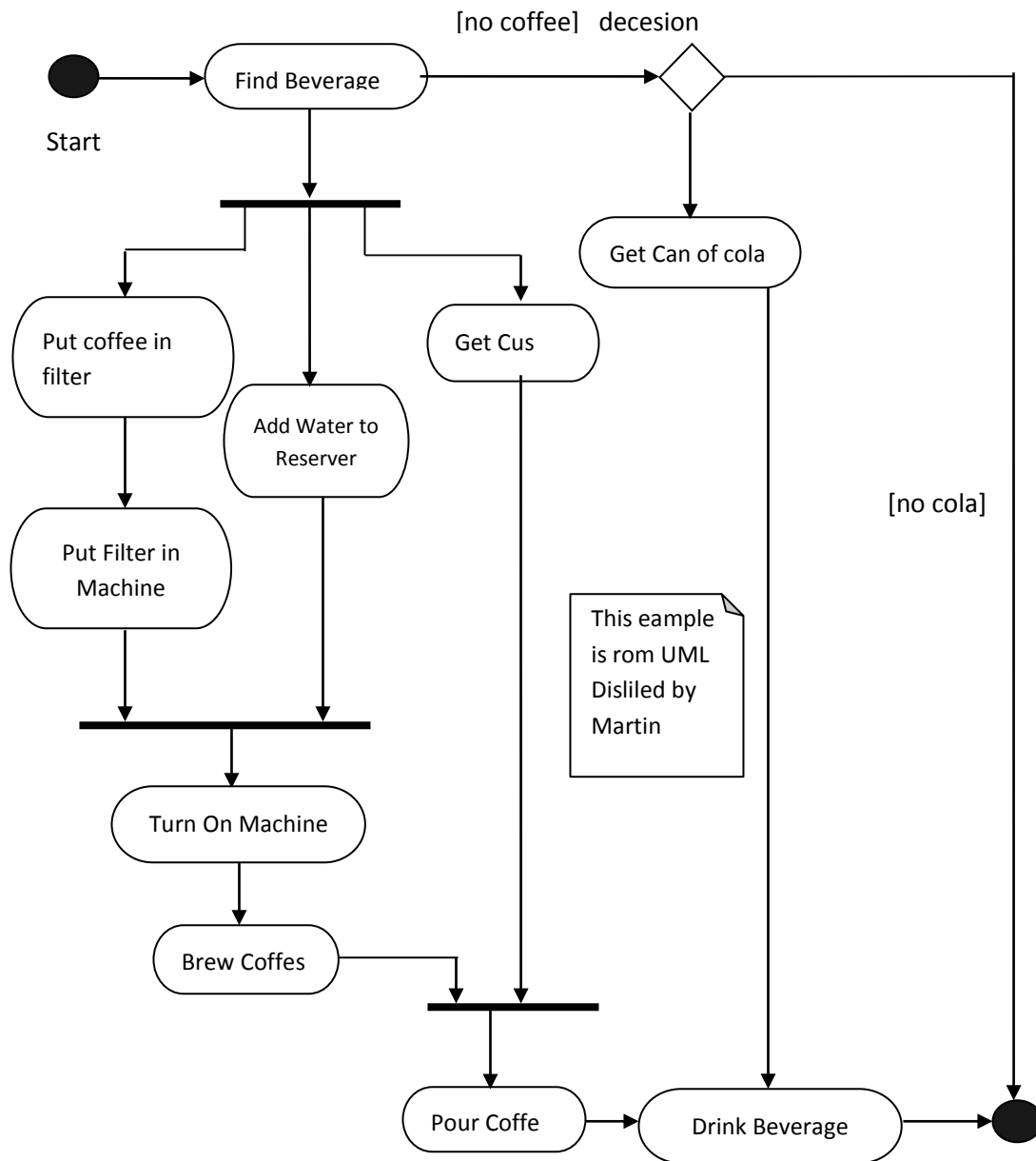
Sumber : (Yuni Sugiarti ; 2013 )

### II.10.3. Activity Diagram

Diagram aktivitas atau *activity* diagram menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis.

*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*). Oleh karena itu *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.

Sebuah aktivitas dapat direalisasikan oleh satu use case atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.(Yuni Sugiarti; 2013: 75)



**Gambar II.2 Activity Diagram**

Sumber : (Yuni Sugiarti ; 2013)

#### II.10.4. Sequence Diagram

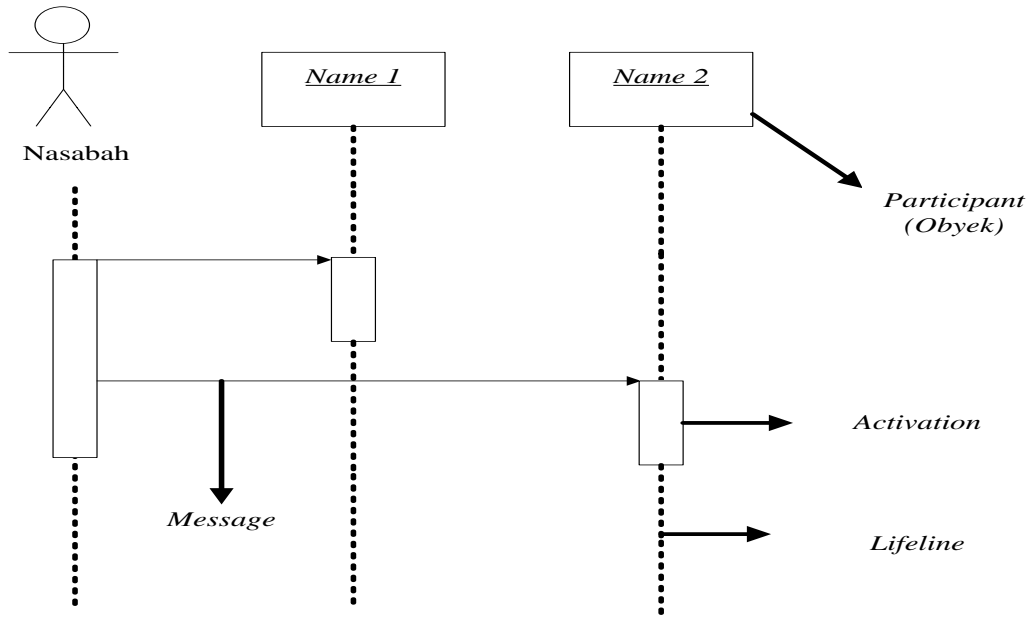
Diagram *sequence* menggambarkan kelakuan/pelaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek.

Oleh karena itu untuk menggambarkan diagram sequence maka harus diketahui objek–objek yang terlibat dalam sebuah *use case* beserta metode–metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Diagram *sequence* memiliki ciri yang berbeda dengan diagram interaksi pada diagram kolaborasi sebagai berikut :

1. Pada diagram *sequence* terdapat garis hidup objek. Garis hidup objek adalah garis vertical yang mencerminkan eksistensi sebuah objek sepanjang periode waktu. Sebagian besar objek–objek yang tercakup dalam diagram interaksi akan eksis sepanjang durasi tertentu dari interaksi, sehingga objek–objek itu diletakkan dibagian atas diagram dengan garis hidup tergambar dari atas hingga bagian bawah diagram. Suatu objek lain dapat saja diciptakan, dalam hal ini garis hidup dimulai saat pesan *destroy*,
2. jika kasus ini terjadi, maka garis hidupnya juga berakhir.
3. Terdapat focus kendali (*Focus Of Control*), berupa empat persegi panjang ramping dan tinggi yang menampilkan aksi suatu objek secara langsung atau sepanjang sub ordinat. Puncak dari empat persegi panjang adalah permulaan aksi, bagian dasar adalah akhir dari suatu aksi. Pada diagram ini mungkin juga memperhatikan penyaringan (*nesting*) dan *focus* kendali yang disebabkan oleh proses rekursif dengan menumpuk *focus* kendali yang lain pada induknya. (Yuni Sugiarti; 2013: 70)

Berikut simbol–simbol yang ada pada sequence diagram.



**Gambar II.3 Simbol Squence**

Sumber : (Yuni Sugiarti ; 2013)