BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Sistem dapat didefinisikan sebagai kumpulan dari elemen – elemen berupa data, jaringan kerja dari prosedur – prosedur yang saling berhubungan, sumber daya manusia, teknologi baik *hardware* maupun *software* yang saling berinteraksi sebagai satu kesatuan untuk mencapai suatu tujuan/ sasaran tertentu yang sama (Maniah, 2017).

A. Model Pembelajaran Berbasis Web

Secara terminologi, Web atau *website* adalah kumulan dari halaman situs dan dokumen yang tersebar di beberapa komputer server yang berada di seluruh penjuru dunia dan terhubung menjadi satu jaringan melalui jaringan yang di sebut internet.

Menurut Batubara (2018), "Pembelajaran berbasis web adalah pembelajaran yang pelaksanaannya di dukung oleh bantuan teknologi internet". Dari berbagai pengalaman dan hasil penelitian telah diperoleh beberapa keunggulan atau kelebihan pembelajaran berbasis web, yaitu sebagai berikut:

1. Memungkinkan setiap orang untuk belajar sesuai dengan waktu dan tempat yang disukainya sehingga proses pembelajaran tidak membosankan.
2. Peserta didik dapat belajar sesuai dengan karakteristik dan caranya sendiri karena pembelajaran berbasis web membuat kegiatan pembelajrana menjadi bersifat individual.
3. Dapat mendorong peserta didik untuk lebih aktif dan mandiri dalam belajar.

4. Materi pelajaran dapat diperbaharui dengan mudah dan murah disbanding dengan penggunaan materi buku cetak.
5. Menyediakan sumber belajar tambahan dalam bentuk tautan yang dapat digunakan untuk mengakses informasi dari berbagai sumber, baik di dalam maupun luar lingkungan belajar.
6. Tersedianya fitur *chatting, forum* dan *web conference* yang dapat digunakan untuk berkomunikasi dan berdiskusi melaluo jaringan internet dengan tanpa dibatasi oleh jarak, tempat dan waktu (Batubara, 2018).

Menurut Darmawan dalam (Batubara, 2018), terdapat tiga fungsi pemanfaatan media elektronik atau web dalam kegiatan pembelajaran, yaitu:

1. Suplemen (tambahan)

   Fungsi ini menjadikan *e-learning* sebagai sumber belajar tambahan yang dapat memperkaya khasanah pengetahuan pelajar.

2. Komplemen (pelengkap)

   Fugsi ini mengharuskan situs web *e-learning* memiliki konten yang sesuai dengan kurikulum dan kebutuhan pembelajaran agar konten tersebut dapat dijadikan materi penguatan, remedial, media latihan, atau alat bantu dalam memberikan penugasan secara online bagi pelajar yang mengikuti pembelajaran di kelas.

3. Subtitusi (pengganti)

   Fungsi ini mengharuskan situs web *e-learning* memiliki konten pembelajaran yang mengacu pada kurikulum, lengkap dengan metode yang terintegrasi dalam materi, dan berbagai fitur pengelolaan kegiatan pembelajaran sehingga sistem

pembelajaran berbasis web tersebut dapat digunakan untuk menggantikan sebagaian dari pembelajaran tatap muka.

B. Sistem

Secara sederhana suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen atau variabel-variabel yang terorganisir, saling berinteraksi, saling ketergantungan satu sama lainnya dan terpadu.

Menurut (Pratama, 2014) “Sistem mendefinisikan sebagai sekumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan suatu tugas bersama-sama”.

“Sistem adalah kumpulan dari elemen – elemen yang berinteraksi untuk mencapai suatu tujuan tertentu “(Trisyanto, 2017). Dari beberapa pengertian diatas dapat dijelaskan mengenai karakteristik sistem sebagai berikut:

1. Karakteristik Sistem

Menurut Trisyanto (2017)” suatu sistem mempunyai karakteristik atau sifat – sifat yang tertentu, yaitu mempunyai komponen – komponen (*components*), batas sistem (*boundary*), linkungan luar sistem (*environments*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolah (*process*) dan sasaran (*objectives*) atau tujuan (*goal*)”. Dari pengertian karakteristik sistem diatas dapat diuraikan sebagai berikut:

a. Komponen Sistem (*Component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan.

b. Batas Sistem (*Boundary*)

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

c. Lingkungan Luar Sistem (*Environment*)

Adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

d. Penghubung Sistem (*Interface*)

Merupakan media penghubung antara satu subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya.

e. Masukan Sistem (*Input*)

Energi yang dimasukan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*) energi yang diproses untuk didapatkan keluaran.

f. Keluaran Sistem (*Output*)

Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

g. Pengolahan Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran.

h. Sasaran Sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem tidak mempuanyai sasaran, maka operasi sistem tidak akan ada gunanya.

C. Website

Awalnya web hanya dibuat untuk mempermudah tukar menukar dan memperbarui informasi. Jaman berkembang dan teknologi web pun semakin banyak digunakan untuk pembuatan *website* hingga *web application*. Jenis – jenis *website* barupun mulai bermunculan dan di kembangkan oleh para *developer* web. Untuk merancang sebuah web di butuhkan pemahaman tentang internet, web dan aplikasi berbasis web.

1. Internet

Internet (*Interconnected Network*) adalah sebutan untuk jaringan komputer global yang menghubungkan satu komputer dengan komputer lain yang ada di seluruh dunia (Wijaya, 2015).

2. Web / website

Menurut Rerung (2018), " Definisi web adalah jaringan komputer yang terdiri dari kumpulan situs internet yang menawarkan teks dan grafik dan suara dan sumber daya animasi melalui *hypertext transfer protocol* ".

3. Aplikasi Berbasis Web

Aplikasi berbasis web merupakan sebuah aplikasi yang diakses melalui web browser dengan menggunakan jaringan sebagai media transmisi. Pengguna yang menggunakan browser dibutuhkan web server untuk melakukan atau mentransfer berkas permintaan melalui protokol komunikasi yang telah ditentukan.

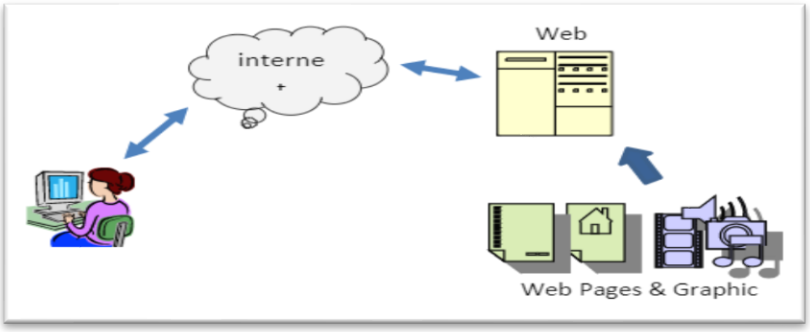
a. Web Browser

Sebelum memulai aktivitas internet, kunci awal yang harus anda pegang adalah penguasaan program penjelajah internet yang di sebut web browser. Program ini

merupakan jendela pemandu yang akan mengantarkan anda saat ber-internet (Wijaya, 2015).

b. Web Server

Web server adalah *software* yang menjadi tulang punggung dari *World Wide Web (www).* (Rerung, 2018).



Sumber: (Rerung, 2018).

Gambar II. 1. Web Server

D. Basis Data (*Database*)

Database adalah termasuk ke dalam salah satu komponen sistem informasi yang berfungsi sebagai media penyimpanan dan pengolahan data menjadi informasi. Basis data dapat dijadikan sebagai bank data atau gudang data.

Menurut Yanto (2016), basis data merupakan "himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa agar dapat dimanfaatkan kembali dengan cepat dan mudah".

Sedangkan menurut Lubis (2016) "basis data merupakan gabungan file data yang dibentuk dengan hubungan atau relasi yang logis dan dapat diungkapkan dengan catatan serta bersifat *independent*".

Berdasarkan pendapat dari para ahli di atas, dapat ditarik kesimpulan bahwa basis data merupakan kumpulan data yang tersimpan di tempat penyimpanan data, dapat diolah, dimanipulasi menjadi informasi dan digunakan kembali.

E. Model Pengembangan Perangkat Lunak

Metode pengembangan perangkat lunak dikenal sebagai suatu kerangka kerja yang digunakan untuk menstrukturkan, merencanakan, dan mengendalikan proses pengembangan suatu sistem informasi.

Menurut Rosa & Shalahuddin (2015), "model *prototype* cocok digunakan untuk menggali spesifikasi kebutuhan pelanggan secara lebih detail tetapi beresiko tinggi terhadap membengkaknya biaya dan waktu proyek".

Mock-up disebut sebagai prototipe apabila perangkat lunak tersebut menyediakan atau mampu mendemonstrasikan sebagian besar fungsi sistem perangkat lunak dan memungkinkan pengujian desain sistem perangkat lunak (Rosa & Shalahuddin, 2015).

Tahapan dari model *prototype* (Rosa dan Shalahuddin, 2015) dijelaskan sebagai berikut:

1. Mendengarkan Pelanggan

   Model *Prototype* dimulai dari mengumpulkan kebutuhan pelanggan terhadap perangkat lunak yang akan di buat.

2. Membangun atau Memperbaiki *Mock-up*

   Setelah mendengarkan kebutuhan pelanggan, maka dibuatlah program prototipe agar pelanggan lebih terbayang dengan apa yang sebenarnya diinginkan. Program

prototipe biasanya merupakan program yang belum jadi dan menyediakan tampilan dengan simulasi alur perangkat lunak sehingga tampak seperti perangkat lunak yang sudah jadi.

3. Pelanggan Melihat atau Menguji *Mock-up*

Program *prototype* ini dievaluasi oleh pelanggan atau user sampai ditemukan spesifikasi yang sesuai dengan keinginan perusahaan.

2.2. Teori Pendukung

Didalam merancang sebuah sistem diperlukan suatu peralatan yang dapat mendukung terciptanya sebuah rancangan. Peralatan pendukung (*Tools Sistem*) merupakan alat yang digunakan untuk menggambarkan bentuk logika model dari suatu sistem dengan menggunakan simbol, lambang, diagram yang menunjukan secara tepat arti fisiknya.

A. *Entity – Relationship* Diagram (Diagram E-R)

Model *entity – Relationship* (ER) pada awalnya disampaikan oleh Peter di tahun 1976 sebagai suatu cara untuk menyatukan jaringan dan menggambarkan *relational* database.

1. *Entity Relationship* Diagram (ERD)

Entity Relationship Diagram adalah gambar atau diagram yang menunjukan informasi di buat, di simpan, dan digunakan dalam sistem bisnis (Trisyanto, 2017).

Menurut Lubis (2016) “ERD menjadi salah satu pemodelan data konseptual yang paling sering digunakan dalam proses pengembangan basis data bertipe relasional “.

Adapun simbol-simbol yang digunakan pada ERD (Rosa & Shalahuddin, 2015) adalah sebagai berikut:

a. Entitas/entity

Entitas merupakan data inti yang akan disimpan, bakal tabel pada basis data, benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer, penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.

b. Atribut

Field atau kolom data yang butuh disimpan dalam suatu entitas.

c. Atribut kunci primer

Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).

d. Atribut multinilai/multivalue

Field atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.

e. Relasi

Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.

f. Asosiasi/association

Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki multipliciy kemungkinan jumlah pemakaian.

2. *Logical Record Structure* (LRS)

Logical record structure (LRS) merupakan representasi dari struktur *record-record* pada tabel dimana tabel-tabel tersebut terbentuk dari hasil himpunan antar entitas pada *entity relationship diagram* yang telah ditransformasikan menjadi bentuk LRS (Pratama, 2014).

Sebuah model sistem yang digambarkan dengan sebuah diagram-ER akan mengikuti pola/ aturan pemodelan tertentu dalam kaitannya dengan konversi ke LRS, maka perubahan yang terjadi (Puspitasari, 2016) adalah mengikuti aturan - aturan berikut ini:

a. Setiap entitas akan diubah kebentuk kotak, Sebuah atribut relasi disatukan dalam sebuah kotak bersama entitas jika hubungan yang terjadi pada diagram-ER 1:M (relasi bersatu dengan *cardinality* M) atau tingkat hubungan 1:1 (relasi bersatu dengan *cardinality* yang paling membutuhkan referensi)

b. sebuah relasi dipisah dalam sebuah kotak tersendiri (menjadi entitas baru) jika tingkat hubungannya M:M (*many to many*) dan memiliki *foreign key* sebagai *primary key* yang diambil dari kedua entitas yang sebelumnya saling berhubungan.

B. *Unified Modeling Language* (UML)

UML adalah Bahasa pemodlan visual yang digunakan untuk menspesifikasikan, menvisualisasikan, membangun, dan mendokumentasikan rancangan dari satu sistem perangkat lunak (Akil, 2018). UML menyediakan beberapa diagram visual yang menunjukkan berbagai aspek dalam sistem yaitu *Use Case* Diagram, *Activity* Diagram, *Class* Diagram dan *Sequence* Diagram.

1. Diagram *Use Case* (*Use Case Diagram*)

Menurut Tohari (2014) "*Use Case* adalah rangkain atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang di lakukan atau diawasi oleh sebuah *actor*. *Use Case* di gunakan untuk membentuk tingkah laku benda dalam sebuah model serta direalisasikan oleh sebuah kolaborasi."

Sedangkan Menurut Akil (2018), "*Use Case* diagram berguna dalam menangkap dan mendefinisikan kebutuhan *sistem*". Pada diagram *use case* ini terdapat beberapa simbol yang dikemukakan oleh Rosa & Shalahuddin (2015) yaitu:

a. *Use case*

Sebuah use case adalah sebuah unit eksternal dari sistem (berupa antar muka) yang akan menerima perintah dari seorang *actor* berupa sebuah *event* (Akil, 2018).

b. Aktor/*actor*

Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri.

c. Asosiasi/*association*

Komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan *actor*

d. Ekstensi/*extend*

Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walaupun tanpa *use case* tambahan itu.

e. Generalisasi/*generalization*

Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.

f. Menggunakan / *include* / *uses*

Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini.

2. Diagram *Activity* (*Activity Diagram*)

Activity diagram atau disebut diagram aktivitas adalah menggambarkan arus dari satu aktivitas ke aktivitas lainya dari suatu fungsional sistem (Akil, 2018).

Menurut Rosa & Shalahuddin (2015) "Diagram aktivitas atau *activity* diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Berikut adalah fungsi dan komponen diagram *activity*:

a. Fungsi *Activity* Diagram

1. Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses
2. Memperlihatkan urutan aktifitas proses pada sistem
3. *Activity* diagram dibuat berdasarkan sebuah atau beberapa *use case* pada *use case* diagram

b. Komponen *Activity* Diagram

Komponen yang ada pada *Activity* Diagram antara lain:

1. *Start Point/Initial State* (Titik Mulai)

   Start Point merupakan lingkaran hitam kecil, yang menandakan tindakan awal atau titik awal aktivitas untuk setiap diagram aktivitas.

2. *Activitiy* (Aktivitas)

   Activity menunjukan aktivitas yang dilakukan atau yang sedang terjadi dalam *activity* diagram.

3. *Action Flow*

   Action Flow digunakan untuk transisi dari suatu tindakan ke tindakan yang lain atau menunjukan aktivitas selanjutnya setelah aktivitas sebelumnya.

4. *Decision* (Keputusan) adalah suatu titik atau *point* pada *activity* diagram yang mengindikasikan suatu kondisi dimana ada kemungkinan perbedaan transisi.

5. *Synchornization*

   Synchornization dibagi menjadi 2 yaitu *fork* dan *join*.

   a. *Fork* (percabangan) digunakan untuk memecah *behaviour* menjadi *activity* atau *action* yang paralel.

   b. *Join* (penggabungan) untuk menggabungkan kembali *activity* atau *action* yang paralel.

6. *Merge Event* (Menggabungkan)

   Merge Event berfungsi untuk menggabungkan *flow* yang dipecah oleh decision.

7. Swimlanes

   Swimlanes berfungsi untuk memecah *activity* diagram menjadi baris dan kolom untuk membagi tangung jawab obyek-obyek yang melakukan aktivitas. Ada dua *construct* utama swimlanes (Maniah, 2017) yaitu:

a. Pool: mempresentasikan partisipan dalam sebuah proses. Pool juga bertindak sebagai grafis *container* untuk memisahkan sekumpulan aktivitas dan pool – pool yang lainnya.
b. Lane: sebuah lane merupakan sub-partisi di dalam pool dan akan memperpanjang seluruh panjang pool, baik secara vertical atau horizontal.

8. *Final State* atau *End Point* (Titik Akhir)

   Final State menunjukan bagian akhir dari aktivitas.

3. *Class* Diagram

Class Diagram adalah diagram struktur statis yang menjelaskan struktur dari sistem pada level *classifiers* (*classes, interfaces* dan lain - lain). Class diagram menunjukan beberapa *classifier* dari sistem, sub sistem atau komponen, relasi antar *classifier*, atribut dan operasi, serta batasan (Akil, 2018).

Sebuah kelas diagram adalah diagram yang menunjukan satu set kelas – kelas, antarmuka dan hubungan – hubungannya. Ada tiga pemodelan umum class diagram (Akil, 2018) diantaranya:

a. Memodelkan kosakata dari sistem

   Memodelkan kosakata dari sistem melibatkan pengambilan keputusan mengenai abstaksi – abstraksi (objek – objek) apa saja yang menjadi bagian dari sistem dan yang berada di luar sistem.

b. Memodelkan kolaborasi yang sederhana

Sebuah kolaborasi adalah kumpulan dari kelas – kelas, antarmuka – antarmuka dan elemen – elemen lain yang bekerja sama untuk menyediakan beberapa tingkah laku yang koperatif yang lebih besar dari jumlah semua elemen – elemen.

c. Memodelkan skema logis database

Skema adalah sebuah cetak biru untuk konsep desain sebuah database. Dalam banyak domain, anda akan menyimpan beberapa informasi secara tetap dalam database relasional atau dalam database berorientasi objek.

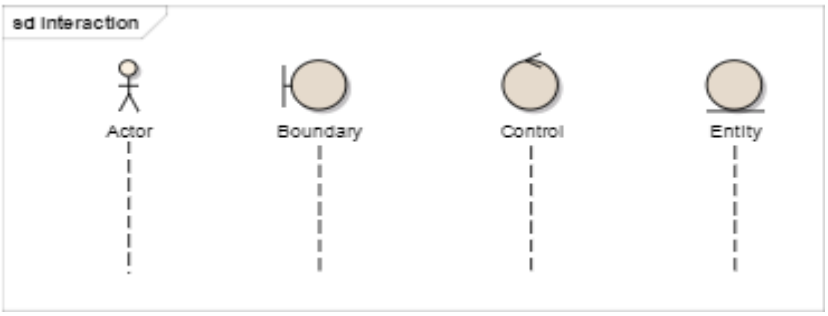
4. *Sequence* Diagram

Menurut Rosa & Shalahuddin (2015) "diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek".

Sequence Diagram adalah menggambarkan urutan penyampaian pesan atau pemanggilan metode antar objek dalam suatu *event* atau *scenario* (Akil, 2018). Notasi yang terdapat pada s*equence diagram* (Akil, 2018) yaitu:

a. *Lifeline Stereotype Icon*

Biasanya garis hidup memiliki stereotip *icon* pada bagian kepalanya untuk mengklasifikasikan jenis objek. Ini biasanya digunakan jika *sequence* diagramdimiliki oleh satu *use case*. Stereotip tersebut antara lain; *actor, boundary, control*, dan *entity*.



Sumber: (Akil, 2018).

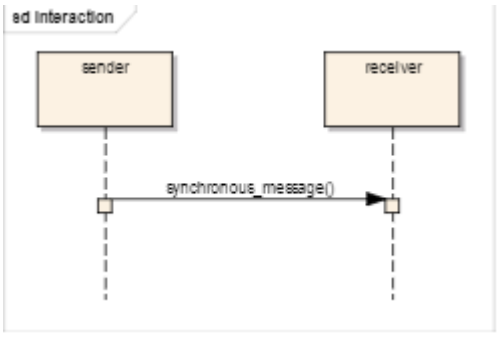
Gambar II. 2. Lifeline Stereotype Icon

b. *Messages* (Pesan)

Ketika suatu objek menyampaikan pesan kepada objek lain, hal ini digambarkan sebagai sebuah panah di antara garis hidup mereka. Panah tersebut dimulai dari sipengirim dan di akhiri pada sipenerima.

c. *Synchronous Message*

Sebuah pesan *synchronous* digunakan ketika sipengirim menunggu sampai sipenerima selesai memproses pesan, setelah itu baru sipengirim melanjutkan. Sebagian besar pemanggilam metode dalam pemograman berorientasi objek adalah bersifat sinkron.

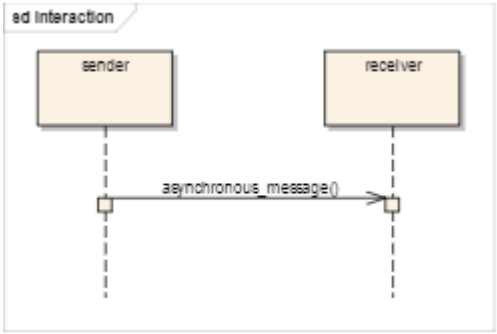


Sumber: (Akil, 2018).

Gambar II. 3. Synchronous Message

d. *Asyinchronous Message*

Dengan pesan *asynchronous,* si pengirim tidak menunggu sipenerima selesai memproses pesan, tetapi langsung lanjut. Pesan – pesan yang dikirim kepada penerima didalam proses yang lain atau pemanggilan yang memulai *thread* (rangkaian) baru adalah contoh pesaan tidak bersifat sinkron.

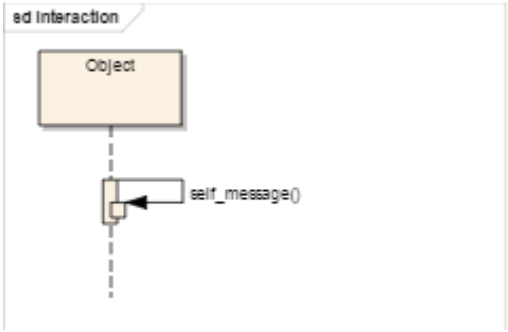


Sumber: (Akil, 2018).

Gambar II. 4. Asynchronous Message

e. *Self Message*

Sebuah pesan yang di kirim kepada objek itu sendiri disebut *self message*. Objek tersebut memanggil metode yang ada pada dirinya sendiri.



Sumber: (Akil, 2018).

Gambar II. 5. Self Message