

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Penelitian Terkait**

Untuk mendukung keberhasilan penelitian ini, peneliti melakukan pendekatan teoritis melalui beberapa literatur yang berhubungan dengan penelitian yang dilakukan. Beberapa jurnal penelitian terdahulu terkait judul penelitian yang sedang dilaksanakan adalah sebagai berikut:

1. Dede Kurniadi, dkk, (2018) dengan judul “Aplikasi Simulasi Tes Buta Warna Berbasis Android Menggunakan Metode Ishihara” pada Jurnal Algoritma. Metode untuk pengembangan perangkat lunak menggunakan pendekatan *Rational Unified Process* (RUP). Pada penelitian ini terdapat beberapa tampilan pada aplikasi seperti menu utama, soal tes buta warna, hasil tes buta warna, informasi buta warna dan informasi pembuat aplikasi. Tampilan hasil yang dihasilkan setelah melakukan tes buta warna pada aplikasi berupa grafik batang yang menampilkan pengguna mengalami buta warna merah-hijau, buta warna total atau mata normal.
2. Yuni Herlina dan Marlindawati, (2020) dengan judul “Pembangunan Diagnosa Buta Warna Berbasis Android Dengan Metode *Hardy-Rand-Rittler*” pada jurnal Bina Darma *Conference on Computer Sciene*. Metode penelitian yang digunakan ialah penelitian deskriptif, metode penelitian ini bisa mendeskripsikan suatu keadaan saja, tetapi bisa juga mendeskripsikan keadaan dalam tahapan-tahapan perkembangannya, penelitian demikian

disebut penelitian perkembangan (*Developmental Studies*). Dalam penelitian perkembangan ini ada yang bersifat *longitudinal* atau sepanjang waktu dan ada yang bersifat *cross sectional* atau dalam potongan waktu. Pada aplikasi dipenelitian ini terdapat tampilan halaman buta warna, halaman soal, halaman hasil diagnosa yang menampilkan diagram batang serta presentase merah, hijau, biru dan normal yang didapat dari jawaban atas soal yang ada.

3. Alvino Octaviano dan Andri Umbari, (2017) dengan judul “Penerapan Metode Ishihara Untuk Mendeteksi Buta Warna Sejak Dini Berbasis Android” pada Jurnal Informatika Universitas Pamulang. Penelitian ini menghasilkan sebuah aplikasi tes buta warna berbasis android untuk mempermudah pengguna dalam tes buta warna. Aplikasi pada penelitian ini menggunakan android versi 2.2.3 (*Froyo*).
4. Mohamad Arif Suryawan dan Muhammad Safei, (2018) dengan judul “Implementasi Metode Ishihara Pada Aplikasi Tes Buta Warna Berbasis Android” pada Jurnal Informatika. Penelitian ini menghasilkan aplikasi tes buta warna berbasis android yang menyediakan tes sebanyak 24 plate dengan menggunakan logika perhitungan hasil dari nomor *plate* yang diujikan.
5. Dwi Wulandari Ningrum, dkk, (2019) dengan judul “Perancangan Sistem Aplikasi *Color And Blindness Detection* Dengan Metode Ishihara Berbasis Android” pada *Journal of Information System, Applied, Management, Accounting and Research*. Penelitian ini menghasilkan aplikasi tes buta warna berbasis android yang menyediakan pengujian tes buta warna serta memberikan keterangan jenis buta warna. Aplikasi ini berhasil mendeteksi

warna benda dengan perhitungan nilai RGB serta menampilkan hasil warna dalam bentuk teks dan suara.

## II.2. Pengertian Aplikasi

Aplikasi merupakan penerapan, menyimpan sesuatu hal, data, permasalahan, pekerjaan ke dalam suatu sarana atau media yang dapat digunakan untuk diterapkan menjadi sebuah bentuk yang baru. Pengertian aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya aplikasi merupakan suatu perangkat komputer yang siap pakai bagi *user* (Siregar, dkk, 2018).

Aplikasi adalah perangkat lunak yang digunakan untuk tujuan tertentu, seperti mengolah dokumen, mengatur *Windows &*, permainan (*game*), dan sebagainya. Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau *suite* aplikasi (*application suite*). Contohnya adalah *Microsoft Office* dan *OpenOffice.org*, Bahasa pemrograman yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi (Arifin dan Veza, 2019).

Secara keseluruhan, aplikasi berarti perangkat lunak yang dapat mengolah data atau menyimpan suatu data guna untuk mempermudah kinerja suatu pekerjaan. Kegunaan suatu aplikasi sangat beragam tergantung dari jenis

aplikasinya, seperti aplikasi permainan, aplikasi membuat gambar dan video, aplikasi membuat catatan harian, dan lain-lain.

### **II.3. Buta Warna**

Buta warna adalah suatu kelainan yang disebabkan ketidakmampuan sel-sel kerucut mata untuk menangkap suatu spektrum warna tertentu yang disebabkan oleh faktor genetis. Buta warna merupakan kelainan genetika yang diturunkan dari orang tua kepada anaknya, kelainan ini sering juga disebut *sex-linked*, karena kelainan ini dibawa oleh kromosom X. Artinya kromosom Y tidak membawa faktor buta warna.

Seorang perempuan terdapat istilah ‘pembawa sifat’, hal ini menunjukkan ada satu kromosom X yang membawa sifat buta warna. Perempuan dengan pembawa sifat, secara fisik tidak mengalami kelainan buta warna sebagaimana wanita normal pada umumnya, tetapi wanita dengan pembawa sifat berpotensi menurunkan faktor buta warna kepada anaknya kelak. Apabila kedua kromosom X mengandung faktor buta warna maka seorang wanita tersebut menderita buta warna (Octaviano dan Umbari, 2017).

### **II.4. Tes Buta Warna**

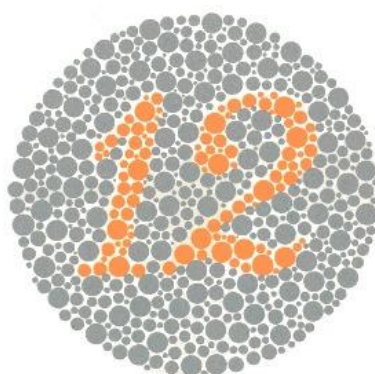
Tes buta adalah pemeriksaan yang dilakukan untuk mendeteksi penyakit buta warna, yaitu kondisi yang menyebabkan kualitas penglihatan terhadap warna terganggu. Pada kondisi ini, seseorang akan merasa kesulitan membedakan warna tertentu maupun warna secara keseluruhan. Selain sebagai syarat masuk kerja atau

sekolah, nyatanya pemeriksaan buta warna sangat penting untuk dilakukan sedini mungkin ([www.halodoc.com/artikel/sering-dilakukan-mengapa-tes-buta-warna-penting](http://www.halodoc.com/artikel/sering-dilakukan-mengapa-tes-buta-warna-penting)).

## II.5. Metode Ishihara

Tes Ishihara dikembangkan oleh Dr. Shinobu Ishihara pada tahun 1917, hingga saat ini metode Ishihara masih menjadi salah satu pilihan utama hamper di semua negara untuk mengidentifikasi seseorang yang mengalami buta warna. Tes metode Ishihara adalah tes yang digunakan untuk mendeteksi gangguan persepsi warna, berupa tabel warna khusus berupa lembaran *pseudoisochromatic (plate)* yang disusun oleh titik-titik dengan kepadatan warna berbeda yang dilihat dengan mata normal, tapi tidak bisa dilihat oleh mata yang mengalami defisiensi sebagian warna.

*Plate* adalah warna primer dengan dasar warna yang hampir sama atau abu-abu. Tes Ishihara secara relatif dapat dipercaya dalam membedakan antara defisit (lemah) warna merah dan defisit (lemah) warna hijau (Octaviano dan Umbari, 2017).



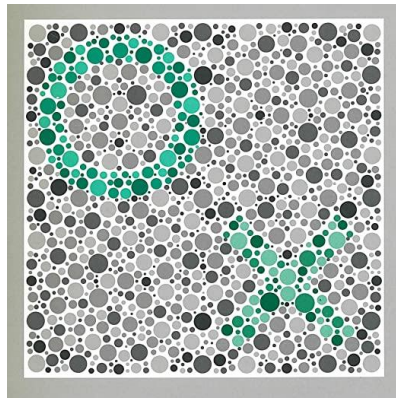
**Gambar II.1. *Plate* Ishihara**

Sumber : ([www.colorblindtest.net/wp-content/gallery/color-blind-test/Ishihara1-0000.jpg](http://www.colorblindtest.net/wp-content/gallery/color-blind-test/Ishihara1-0000.jpg))

## II.6. Metode *Hardy-Rand-Rittler*

Metode *Hardy-Rand-Rittler* (HRR) pertama kali ditemukan pada tahun 1945 dan dapat digunakan untuk mendeteksi keseluruhan jenis buta warna (merah, hijau, dan biru). Sejak saat itu, tes dengan metode ini terus digunakan di seluruh dunia. Diagnosa dengan metode HRR terdiri dari 4 bagian besar dan hasil dari masing-masing bagian akan digunakan untuk menentukan jenis kelainan warna yang dimiliki.

Pada tes dengan metode HRR, penderita akan diminta untuk melihat beberapa bentuk gambar, seperti segita ataupun lingkaran. Warna titik itu dibuat sedemikian rupa sehingga orang buta warna tidak akan melihat perbedaan warna seperti yang dilihat orang normal (Herlina dan Marlindawati, 2020).



**Gambar II.2. *Plate Hardy-Rand-Rittler***

Sumber : ([www.amazon.com/Laminated-HRR-Standard-Pseudoisochromatic-Test/dp/B07TCCCN2Q](http://www.amazon.com/Laminated-HRR-Standard-Pseudoisochromatic-Test/dp/B07TCCCN2Q))

## II.7. Android

Android adalah sistem operasi (*Operating System*) yang umumnya digunakan pada perangkat dengan navigasi *full touch screen* yang biasa dimiliki oleh *smartphone* dan komputer tablet. Android sudah diambil alih oleh perusahaan *Google Inc.* yang telah membelinya pada tahun 2005 dari *Android Inc.* *Google* menyediakan *software/tools* yang dikembangkan khusus untuk dijadikan alat pengembang aplikasi android (Yusfrizal 2019).



**Gambar II.3. Android**

Sumber : ([www.liputan6.com/tekno/read/2058940/android-akan-ganti-logo](http://www.liputan6.com/tekno/read/2058940/android-akan-ganti-logo))

### II.7.1. Pengenalan Android

Sistem Operasi Android merupakan sebuah sistem operasi yang berbasis *Linux* untuk telepon seluler seperti telepon pintar dan komputer tablet. Sistem operasi ini bersifat *open source* (terbuka) sehingga para *programmer* dapat membuat aplikasi secara mudah untuk digunakan oleh bermacam perangkat bergerak (misalnya telepon seluler). Karena sistem operasi Android ini merupakan aplikasi *open source* maka dapat dilakukan modifikasi dan penyebaran secara bebas..

Pengembang memiliki beberapa pilihan ketika membuat aplikasi yang berbasis Android. Kebanyakan pengembang menggunakan *Eclipse* yang tersedia secara bebas untuk merancang dan mengembangkan aplikasi Android (Yusfrizal, 2018).

### II.7.2. Versi Android

Android memiliki versi-versi sebagai berikut:

#### 1. Android 1.1

Android memang diluncurkan pertama kali pada tahun 2007, namun sistem operasi ini mulai dirilis dan diterapkan ke berbagai *gadget* pada tanggal 9 Maret 2009 silam. Android versi 1.1 merupakan android awal yang dimana versi ini baru memberikan sentuhan di beberapa aplikasinya seperti sistem antar muka bagi pengguna (*user interface*) yang lebih baik, serta beberapa aplikasi yang lain (Kusniyati dan Sitanggang, 2016).

#### 2. Android 1.5 (*Cupcake*)

Pada bulan Mei 2009 Android kembali mengalami perubahan versi. Android versi 1.1 kemudian disempurnakan dengan Android versi 1.5 atau yang dikenal sebagai android *cupcake* (Kusniyati dan Sitanggang, 2016).

#### 3. Android 1.6 (*Donut*)

*Donut* (versi 1.6) diluncurkan dalam tempo kurang dari 4 bulan semenjak peluncuran perdana android *cupcake*, yaitu pada bulan September 2009 (Kusniyati dan Sitanggang, 2016).

#### 4. Android 2.0/2.1 (*Éclair*)

Masih ditahun yang sama, android kembali merilis *operating sistem* versi terbarunya, yaitu android versi 2.0/2.1 *Eclair*. Android *eclair* diluncurkan oleh *Google* 3 bulan setelah peluncuran (Kusniyati dan Sitanggang, 2016).

5. Android 2.2 (*Froyo: Frozen Yoghurt*)

Butuh 5 bulan bagi *Google* untuk melakukan regenerasi dari android *eclair* versi sebelumnya ke versi *froyo: frozen yoghurt*. Pada tanggal 20 Mei 2010, android versi 2.2 alias android *froyo* ini dirilis (Kusniyati dan Sitanggang, 2016).

6. Android 2.3 (*Gingerbread*)

7 bulan kemudian android kembali melakukan gebrakan dengan merilis kembali android versi 2.3 atau yang dikenal sebagai android *gingerbread* (Kusniyati dan Sitanggang, 2016).

7. Android 3.0/3.1 (*Honeycomb*)

Pada bulan Mei 2011, android versi 3.0/3.1 atau android *Honeycomb* dirilis. android *Honeycomb* merupakan sebuah sistem operasi android yang tujuannya memang dikhususkan bagi penggunaan tablet berbasis android (Kusniyati dan Sitanggang, 2016).

8. Android 4.0 (ICS: *Ice Cream Sandwich*)

Android ICS atau *Ice Cream Sandwich* juga dirilis pada tahun yang sama dengan *honeycomb*, yaitu pada bulan Oktober 2011 (Kusniyati dan Sitanggang, 2016).

9. Android 4.1 (*Jelly Bean*)

Android *Jelly Bean* merupakan versi android yang terbaru pada saat ini. Salah satu *gadget* yang menggunakan sistem operasi *Jelly Bean* adalah *Google Nexus 7*

yang diprakarsai oleh ASUS, vendor asal Taiwan yang juga menjadi teman satu kampung halaman dengan Acer (Kusniyati dan Sitanggang, 2016).

#### 10. Android 4.4 (*Kit Kat*)

Kehadiran android kit kat merupakan peluncuran produk OS yang diluncurkan pada 4 september 2013, sebelumnya banyak kabar beredar jikalau android akan meluncurkan OS baru yang bernama Android *Key Lime Pie* namun setelah di analisa tidak sesuai dengan ejaan orang umum, sehingga namanya diganti dengan OS Android *Kit Kat* yang sebagian besar orang sudah familiar dengan itu (Kusniyati dan Sitanggang, 2016).

#### 11. Android 5.0.2 (*Lollipop*)

Android *Lollipop* merupakan keberadaan OS android yang memang saat ini sudah menjadi trend baru di industri *smartphone*, hal ini tak lepas dari keunikan dan kelebihan yang banyak dimiliki dari OS tersebut. Kehadiran android versi ini amat di nanti oleh sekian banyak orang karna diharapkan sistem operasi *lollipop* ini bias lebih baik dibandingkan versi-versi sebelumnya (Kusniyati dan Sitanggang, 2016).

#### 12. Android 6.0 (*Marshmallow*)

Android 6.0 *Marshmallow* adalah versi dari sistem operasi mobile android. Pertama kali diperkenalkan Mei 2015 di *Google I / O* di bawah kode nama Android M, secara resmi dirilis pada Oktober 2015. Android *marshmallow* memperkenalkan model izin aplikasi didesain ulang sekarang ada hanya delapan kategori izin, dan aplikasi yang tidak lagi secara otomatis diberikan semua hak akses mereka ditentukan pada waktu instalasi (Kusniyati dan Sitanggang, 2016).

## II.8. Eclipse

*Eclipse* adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform* (*platform-independent*). Berikut ini adalah sifat dari *Eclipse*:

1. *Multi-platform*: Target sistem operasi *Eclipse* adalah *Microsoft Windows*, *Linux*, *Solaris*, *AIX*, *HP-UX* dan *Mac OS X*.
2. *Mult-language*: *Eclipse* dikembangkan dengan bahasa pemrograman *Java*, akan tetapi *Eclipse* mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti *C/C++*, *Cobol*, *Python*, *Perl*, *PHP*, dan lain sebagainya.
3. *Multi-role*: Selain sebagai IDE untuk pengembangan aplikasi, *Eclipse* pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, tes perangkat lunak, pengembangan *web*, dan lain sebagainya (Pratama dan Hermawan, 2016).



**Gambar II.4. Eclipse**

Sumber : ([www.eclipse.org/artwork/](http://www.eclipse.org/artwork/))

## II.9. Java

*Java* adalah bahasa pemrograman yang dapat di jalankan 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 (Pratama dan Hermawan, 2016).

*Java* merupakan bahasa berorientasi objek untuk pengembangan aplikasi mandiri, aplikasi berbasis *internet*, aplikasi untuk perangkat cerdas yang dapat berkomunikasi lewat *internet*/jaringan komunikasi. Melalui teknologi *java*, dimungkinkan perangkat *audio stereo* dirumah terhubung jaringan komputer. *Java* tidak lagi hanya untuk membuat *applet* yang memerintah halaman web tapi *java* telah menjadi bahasa untuk pengembangan aplikasi skala *interprise* berbasis jaringan besar (Fajarianto, dkk, 2018).

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


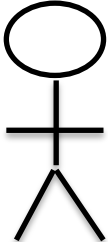

UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem (Urva dan Siregar, 2015).




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

*Use case diagram* merupakan permodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat

dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram yaitu:

**Tabel II.1. Simbol-simbol *Use Case Diagram***

Gambar	Keterangan
	<p><i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja diawal nama <i>use case</i>.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i>, tetapi tidak memiliki kontrol terhadap <i>use case</i>.</p>
	<p>Asosiasi antara aktor dan <i>use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran</p>

	data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem
	<i>Include</i> , merupakan di dalam <i>use case</i> lain ( <i>required</i> ) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.




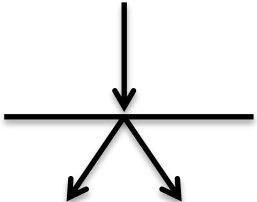
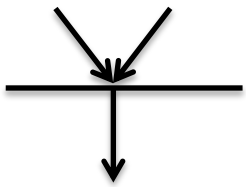
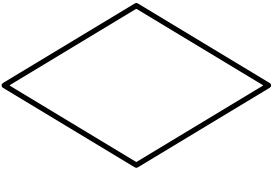
Sumber : (Urva dan Siregar, 2015)


### II.10.2. Activity Diagram

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* yaitu:

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

Gambar	Keterangan
--------	------------

	<p><i>Start Point</i>, diletakkan pada pojok kiri atas dan merupakan awal aktivitas</p>
	<p><i>End Point</i>, akhir aktivitas</p>
	<p><i>Activities</i>, menggambarkan suatu proses/kegiatan bisnis.</p>
	<p><i>Fork</i> (percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.</p>
	<p><i>Join</i> (penggabungan) atau <i>rake</i>, digunakan untuk menunjukkan adanya dekomposisi.</p>
	<p><i>Decision Points</i>, menggambar kan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>fals</i>.</p>

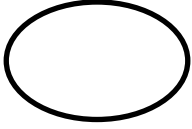
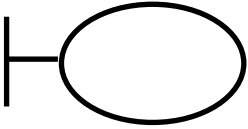
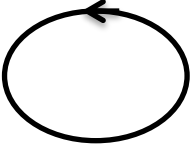
	<p><i>Swimlane</i>, pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.</p>
---	--


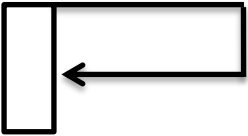
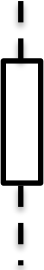

Sumber : (Urva dan Siregar, 2015)

### II.10.3. *Sequence Diagram*

*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* yaitu:

**Tabel II.3. Simbol-simbol *Sequence Diagram***

Gambar	Keterangan
	<p><i>Entity Class</i>, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.</p>
	<p><i>Boundary Class</i>, berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form entry</i> dan <i>form cetak</i>.</p>
	<p><i>Control class</i>, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan</p>

	aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>

Sumber : (Urva dan Siregar, 2015)