# BAB II

# TINJUAN PUSTAKA

## II.1 Penelitian terdahulu

Untuk mendukung penelitian ini, penulis menggunakan beberapa referensi jurnal-jurnal sebagai acuan yang penulis lakukan diantaranya sebagai berikut:

Berdasarkan penelitian yang telah dilakukan sebelumnya oleh Muklisufatih Latief, dkk. (2007) dengan judul "Metode Rational Unified Process Untuk Pengembangan Aplikasi Web dan Mobile (Studi Kasus Sistem Informasi Tanaman Obat Daerah Gorontalo)". Pengembangan aplikasi ini menggunakan metode RUP (*Rational Unified Process*). Aplikasi web yang dibangun menggunakan bahasa pemograman PHP, dimana aplikasi ini berfungsi sebagai pengolahan data tanaman obat. Sedangkan aplikasi *mobile* menggunakan *ionic framework* yang berfungsi untuk mencari data obat berdasarkan penyakit tertentu.

Pada penelitian yang dilakukan oleh Muklisufatih Latief, sistem kerja aplikasinya yaitu pada halaman utama terdapat tiga menu, daftar tanaman dan satu bar pencarian. Tiap daftar tanaman diberi label *view* yang mengizinkan kita untuk melihat dekripsi tanaman dan cara pengolahannya. Dengan bantuan aplikasi ini, pengguna hanya perlu mengetik nama penyakit akan muncul daftar tanaman obat yang berhubungan dengan penyakit tersebut.

Berdasarkan penelitian yang telah dilakukan sebelumnya oleh Ferdyani dan Sherly (2019) dengan judul "Perancangan Ensiklopedia Tanaman Obat Khas Kalimantan Tengah Berbasis Android". Aplikasi ini dirancang dengan

UML(*Unified Modeling Language*) yang berorientasi objek dengan basis data pada SQLite sebagai penyimpan data. Pengembangan aplikasi ini menggunakan metode *research and development* (Penelitian dan Pengembangan).

Untuk cara kerja sistem aplikasi ini yaitu terdapat tampilan menu yang berisi sub menu kategori obat, tentang dan keluar. Pada menu kategori obat terdapat dua sub menu yaitu jenis tanaman sebagai dekripsi dan keterangan khasiat. Pada seb menu jenis tanaman terdapat satu bar pencarian. Dengan bantuan beberapa tools diantaranya UML sebagai perancangan sistem , *SQLite* sebagai database, dan *Microsoft Visio* untuk perancangan antarmuka (*interface*) dapat dirancang aplikasi tanaman obat tradisional Kalimantan Tengah yang dapat membantu memudahkan pengguna khususnnya masyarakat Kalimantan Tengah mencari informasi tentang obat tradisionat Kalimantan Tengah yang disertai dengan panduan dalam pengolahannya.

Berdasarkan penelitian yang telah dilakukan sebelumnya oleh Dwi Eka Sari, dkk. (2019) pada penelitian yang berjudul "Rekayasa Aplikasi Ensiklopedia Tanaman Obat Berbasis Android". Pengembangan aplikasi ini menggunakan metode RUP (*Rational Unified Process*). Aplikasi ini dirancang dengan perangkat lunak Android Studio dan phpMyAdmin sebagai pengolah database.

Pada penelitian yang dilakukan oleh Dwi Eka Sari, sistem kerja aplikasi yaitu menu utama terdapat menu tanaman obat, resep obat dan info aplikasi. Dimana pada menu tanaman obat terdapat tiga sub menu yaitu buah, sayuran dan rempah-rempah. Tiap sub menu tersebut terdapat daftar tanaman obat dan satu bar pencarian. Pada menu resep obat terdapat sub menu daftar penyakit yang akan menampilkan nama penyakit dan cara pengolahan tanaman.

Aplikasi yang dibangun oleh peneliti Dwi Eka Sari serinci dan sederhana mungkin guna dapat dipergunakan oleh masyarakat banyak. Dengan dibangunnya aplikasi ini peneliti mengharpkan aplikasi dapat menjadi media alternatif sebagai bahan baca untuk mengenal dan mengetahui informasi tentang tanaman obat serta khasiatnya dalam menyembuhkan penyakit secara tradisional ,

## II.2. Landasan Teori

Berikut ini merupakan landasan teori yang berkaitan dengan penelitian yang dilakukan oleh penulis :

### II.2.1. Rancang

Perancangan merupakan salah satu hal yang penting dalam membuat program. Adapun tujuan dari perancangan ialah untuk memberi gambaran yang jelas lengkap kepada pemrogram dan ahli teknik yang terlibat. Perancangan harus berguna dan mudah dipahami sehingga mudah digunakan.

Perancangan adalah sebuah proses untuk mendefinisikan sesuatu yang akan dikerjakan dengan menggunakan teknik yang bervariasi serta di dalamnya melibatkan deskripsi mengenai arsitektur serta detail komponen dan juga keterbatasan yang akan dialami dalam proses pengerjaanya.

### II.2.2. Bangun

Pengertian pembangunan atau bangun sistem adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada secara keseluruhan. Jadi dapat disimpulkan bahwa rancang bangun adalah penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang

terpisah kedalam suatu kesatuan yang utuh dan berfungsi. Dengan demikian pengertian rancang bangun merupakan kegiatan menerjemahkan hasil analisa ke dalam bentuk paket perangkat lunak kemudian menciptakan sistem tersebut atau memperbaiki sistem yang sudah ada.

### II.2.3. Aplikasi

Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus yang dapat memudahkan penggunanya dengan sesuai apa iang penggunanya inginkan. Aplikasi berasal dari kata *application* yaitu bentuk benda dari kata kerja *to apply* yang dalam bahasa Indonesia berarti pengolah.(Hasugian, 2014). Secara istilah, aplikasi komputer adalah suatu subkelas perangkat lunak komputer yang menggunakan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pemakai. Contoh utama perangkat lunak aplikasi adalah program pengolah kata, lembar kerja, dan pemutar media.

Kumpulan aplikasi komputer yang digabung menjadi suatu paket biasanya disebut paket atau *suite* aplikasi (*application suite*). Contohnya adalah *Microsoft Office* dan *OpenOffice.org,* 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. Umumnya aplikasi- aplikasi tersebut memiliki kemampuan untuk saling berinteraksi sehingga menguntungkan pemakai. Contohnya, suatu lembar kerja dapat dimasukkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

### II.2.4. Ensiklopedia

Ensiklopedia adalah sejumlah tulisan yang berisi penjelasan yang menyimpan informasi secara komprehensif dan cepat dipahami serta dimengerti mengenai keseluruhan cabang ilmu pengetahuan atau khusus dalam satu cabang ilmu.pengetahuan yang tersusun dalam bagian artikel-artikel dengan satu topik bahasan yang disusun berdasarkan abjad dan pada umumnya tercetak dalam bentuk rangkaian buku yang tergantung pada jumlah bahan yang disertakan.

Kata "ensiklopedia" diambil dari Bahasa Yunani; *enkyklios paideia* yang berarti sebuah lingkaran atau pengajaran yang lengkap. Yang mana dimaksud ensiklopedia itu sebuah pendidikan paripurna yang mencakup semua lingkaran ilmu pengetahuan. Seringkali ensiklopedia dicampurbaurkan kamus dan ensiklopedia-ensiklopedia awal memang berkembang dari kamus. Perbedaan utama antara kamus dan ensiklopedia ialah bahwa sebuah kamus hanya memberikan defenisi setiap entri yang dilihat dari sudut pandang linguistic atau hanya memberikan sinonim saja, sedangkan ensiklopedia memberikan penjelasan secara mendalam dari yang kita cari. Sebuah ensiklopedia mencoba menjelaskan setiap artikel sebagai sebuah fenomena. Atau lebih singkat, kamus adalah daftar kata-kata yang dijelaskan dengan kata-kata lainnya sedangkan sebuah ensiklopedia adalah sebuah daftar hal-hal yang kadang kala dilengkapi dengan gambar untuk lebih menjelaskan.

### II.2.5. Tanaman Obat

Tanaman obat merupakan spesies tumbuhan yang diketahui mempunyai khasiat baik dalam membantu memelihara kesehatan maupun pengobatan suatu

penyakit (Harmida dan Yuni, 2011). Tanaman yang dimanfaatkan sebagai obat-obatan dipercaya masyarakat mempunyai khasiat yang dapat menyembuhkan penyakit dan menjadi bahan dasar obat tradisional (Yuniati dan Alwi, 2010).

Tanaman obat yang ditemukan dalam berbagai jenis seperti rempah-rempah, tanaman pagar, buah, sayur, atau tanaman liar sekalipun dapat dijadikan sebagai tanaman obat. Tanaman obat umumnya memiliki efek samping yang jauh lebih rendah dibandingkan dengan obat yang dibuat secara kimiawi karena obat yang dibuat secara kimiawi lebih berbahaya dibandingkan dengan obat dari tanaman obat (Muhlisah, 2001).

Pada hutan tropika sekitar 30.000 spesies tumbuhan berbunga yang dapat memberi manfaat bagi manusia baik ekologis maupun ekonomis. Hutan tropis Indonesia mempunyai banyak kekayaan alam, dan memiliki berbagai tumbuhan yang berkhasiat obat. Dari beberapa hasil penelitian yang dilakukan, di hutan tropis Indonesia terdapat sekitar 1.300 jenis tanaman yang berkhasiat obat dan sekitar 300 jenis telah dimanfaatkan sebagai pengobatan tradisional, sedangkan menurut WHO lebih dari 20.000 spesies tumbuhan berkhasiat obat yang digunakan oleh penduduk di seluruh dunia (Zahud dkk, 2012).

### II.2.6. *Rational Unified Process*

*Rational Unified Process* (RUP) merupakan suatu metode rekayasa perangkat lunak yang dikembangkan dengan mengumpulkan berbagai *best practices* yang terdapat dalam industry pengembangan perangkat lunak. RUP menggunakan konsep objek oriented, dengan akifitas yang berfokus pada pengembangan dengan menggunakan *Unified Modeling Language* (UML).

Metode *Rational Unified Process* adalah suatu kerangka kerja proses pengembangan perangkat lunak iteratif yang dibuat oleh Rational Software, suatu divisi dari IBM sejak 2003. Motode RUP bukanlah suatu proses tunggal dengan aturan konkret, melainkan suatu kerangka proses yang dapat diadaptasi dan dimaksudkan untuk disesuaikan oleh organisasi pengembang dan tim proyek perangkat lunak yang akan memilih elemen proses sesuai dengan kebutuhan mereka.

Adapun tahap pengembangan metode *Rational Unified Process* yaitu :

1. *Inception* (permulaan)

   Pada tahap ini pengembangan mendefenisikan batasan kegiatan, melakukan analisis kebutuhan user, dan melakukan perancangan awal perangkat lunak (perancangan arsitektural dan user case). Pada fase ini, prototype perangkat lunak versi Alpa harus sudah rilis.

2. *Elaboration* (perluasaan atau perencanaan)

   Pada tahap ini dilakukan perancangan perangkat lunak mulai dari menspesifikan fitur perangkat lunak hingga perilisan prototype versi Beta dari perangkat lunak

3. *Construction* (kontruksi)

   Pengimpleentasian rancangan perangkat lunak yang telah dibuat dilakukan pada tahap ini. Pada akhir tahap ini, perangkat lunak versi akhir yang sudah disetujui administrator dirilis beserta dokumentasi perangkat lunak.

4. *Transition* (transisi)

   Instalasi, deployment dan sosialisasi perangkat lunak dilakukan pada tahap ini.

### II.2.7. Sejarah Android

Android pada mulanya berada dibawah naungan Android, Inc. Sebuah perusahaan yang didirikan pada tahun 2003 silam oleh Andy Rubin, Rich Miner, Nick Sears dan Chris White. Markas besarnya berlokasi di Palo Alto, California. Lucunya, Android pada awalnya tidak dibuat untuk ponsel, melainkan untuk kamera digital. Tujuan itu bergeser ketika perusahaan menyadari peluang mereka lebih besar untuk berkembang jika menyasar perangkat *mobile.* Jadilah Android diplot untuk menyaingi *Symbian* dan *Windows Mobile* yang kala itu menjadi pemain utama. Tapi saat mulai dikerjakan, karena suatu alasan, Android Inc menjalani misi itu secara tersembunyi. Lalu di tahun 2005, tepatnya pada tanggal

17 bulan Agustus, Google meminang Android Inc yang artinya Google juga memboyong para pendiri Android ke dalam naungan mereka. Tapi sampai di penghujung tahun 2006, Google belum melahirkan apa-apa, Android menghilang dan barulah pada 22 Oktober 2008 ponsel seluler komersil pertama berbasis Android diluncurkan dengan nama HTC Dream.

### II.2.8. Android

Aplikasi *Android* ditulis dalam bahasa pemrograman *java*, yaitu kode *java* yang terkompilasi bersama-sama dengan data dan *file-file* sumber yang dibutuhkan oleh aplikasi yang digabungkan oleh *app tools* menjadi paket aplikasi Android, sebuah file yang ditandai dengan akhiran .*apk.* file inilah yang didistribusikan sebagai aplikasi dan diinstal pada *handset Android*. (Ahmad : 2015 : 190-200).

Android merupakan sistem operasi yang digunakan pada perangkat *mobile* berbasis *Linux*. Awal mulanya sistem operasi ini dikembangkan oleh Android Inc, kemudian dibeli oleh Google pada tahun 2005.

Penamaan Sistem Operasi android sangat unik karena menggunakan nama-nama makanan dan diawali dengan urutan alfabet. Berikut merupakan perkembangan versi-versi android dari masa ke masa mulai dari versi *1.0* hingga versi 9.0 *Pie*

1. Android v1.0 *Beta* (28 September 2008)
2. Android v1.1 *Beta* (9 Februari 2009)
3. Android v1.5 *Cupcake* (27 April 2009)
4. Android v1.6 *Donut* (15 September 2009)
5. Android v2.0-v2.5 *Éclair* (26 Oktober 2009)
6. Android v2.2-v2.2.3 *Froyo* (20 Mei 2010)
7. Android v2.3-v.2.3.7 *Gingerbread* (6 Desember 2010)
8. Android v3.0-v3.2.6 *Honeycomb* (22 Februari 2011)
9. Android v4.0-v4.0.4 *Ice Cream Sandwich* (18 Oktober 2011)
10. Android v4.1-v4.3.1 *Jelly Bean* (9 Juli 2012)
11. Android v4.4-v4.4.4 *KitKat* (31 Oktober 2013)
12. Android v5.0-5.1.1 *Lollipop* (12 November 2014)
13. Android v6.0-v6.0.1 *Marshmallow* (5 Oktober 2015)
14. Android v7.0-7.1.2 *Nugget* (22 Agustus 2016)
15. Android v8.0-v8.1 *Oreo* (21 Agustus 2017)
16. Android v9.0 *Pie* (6 Agustus 2018) (*Android.com*, 2019)
17. Android v10 *Queen Tart* sering disebut android Q (3 September 2019)
18. Android v11 *Red Velvet Cake* (8 September 2020)

### II.2.9. *Android SDK (Software Development Kit)*

*Android SDK* adalah *tools* bagi para *programmer* yang ingin mengembangkan aplikasi berbasis Google Android. Android SDK mencakup seperangkat alat pengembangan yang komprehensif. Android SDK terdiri dari debugger, libraries, handset emulator, dokumentasi, contoh kode, dan tutorial. IDE yang didukung secara resmi adalah Eclipse 2.3 atau lebih yang menggunakan *plugin Android Development Tools*, dengan ini pengembang dapat menggunakan teks editor untuk mengedit file Java dan XML serta menggunakan peralatan *command line* untuk menciptakan, membangun, melakukan *debug* aplikasiandroid dan pengendalian perangkat android, misalnya *reboot* , memginstal paket perangkat lunak dengan jarak jauh.

### II.2.10. *Android Studio*

*Android Studio* adalah *Integrated Development Enviroment* (IDE) untuk sistem operasi Android, yang dibangung diatas perangkat lunak *JetBrains IntelliJ IDEA* dan didesain khusus untuk pengembangan Android. IDE ini merupakan pengganti dari *Eclipse Android Development Tools* (ADT) yang sebelumnya merupakan IDE utama untuk pengembangan aplikasi android.

Android Studio sendiri pertama kali diumumkan di Google I/O conference pada tanggal 16 Mei 2013. Ini merupakan tahap preview dari versi 0.1 pada Mei 2013, dan memasuki tahap beta sejak versi 0.8 dan mulai diliris pada Juni 2014. Versi liris stabil yang pertama diliris pada December 2014, dimulai sejak versi 1.0. Sedangkan versi stabil yang sekarang adalah versi 3.13 yang diliris pada Juni 2018. Fitur Fitur yang tersedia saat ini dalam *stable version*.

*a.* Dukungan *Gradle-based build*

*b.* *Android-specific refactoring* dan perbaikan cepat

*c.* *Lint tools* untuk menangkap kinerja, kegunaan, kompatibilitas versi, dan masalah lainnya

*d.* *Integrasi proguard* dan kemampuan penanda tanganan aplikasi

*e.* *Template-based wizards* untuk membuat *template design* umum seperti *drawer* atau *empty activity*.

*f.* Mendukung untuk perkembangan aplikasi *Android Wear*

*g.* Editor tata letak yang memungkinkan pengguna untuk menyeret dan menjatuhkan (drag-and-drop) komponen UI, opsi untuk melihat tata letak pada beberapa konfigurasi layar

*h.* Dukungan bawaan untuk *Google Cloud Platform*, memungkinkan integrasi dengan *Firebase Cloud Messaging* (‘Perpesanan *Google Cloud’* Sebelumnya) dan *Google App Engine*

*i.* *Android Virtual Device* (*Emulator*) untuk menjalankan dan men-debug aplikasi di studio Android

### II.2.11. UML *(Unified Modelling Language)*

Menurut Rosa dan Shalahuddin (2014:133), berpendapat bahwa UML (*Unified Modeling Language*) adalah “Salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requerement, membuat analisa & desain, serta menggambarkan arsitektur dalam pemrograman berorintasi objek”. Sedangkan Mulyani (2016:48) mengatakan UML (*Unified Modeling Language*) adalah “Sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem”.

Dari beberapa penjelasan teori tersebut dapat disimpulkan bahwa UML (*Unified Modeling Language*) adalah bahasa yang sering digunakan untuk

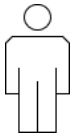
membangun sebuah sistem perangkat lunak dengan melakukan penganalisaan desain dan spesifikasi dalam pemrograman berorintasi objek. UML (*Unified Modeling Language*) memiliki diagram-diagram yang digunakan dalam pembuatan aplikasi berorintasi objek, diantaranya (Rosa dan Shalahuddin, 2014:155):

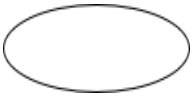
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan *Unified Modelling Language* adalah sebagai berikut:

a. *Use Case Diagram*

*Use Case Diagram* merupakan pemodelan untuk melakukan atau merancang sebuah system yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut adalah simbol-simbol yang ada pada diagram *use case*:

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

| Gambar | Nama | Arti/Keterangan |
|---|---|---|
|  | Aktor | Aktor adalah *abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan 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 Use Case, tapi tidak memiliki kontrol terhadap Use Case. |

| | | |
|---|---|---|
|  | *Use Case* | Use Case menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bentukan pesan antar unit dengan Aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama Use Case. |
|  | *Association / Dependency* | Asosiasi antara Aktor dan Use Case, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data. |
|  | *Generalization* | Asosiasi antara Aktor dan Use Case yang menggunakan panah terbuka untuk mengindikasikan bila Aktor berinteraksi secara pasif dengan sistem. |
|  | *Include* | Include, merupakan di dalam Use Case lain (required) atau pemanggilan Use Case oleh Use Case lain jika kondisi atau syarat terpenuhi. |

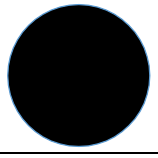
| | | |
|---|---|---|
|  | *Extend* | Extend, merupakan perluasan dari Use Case lain jika kondisi atau syarat terpenuhi. |

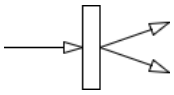
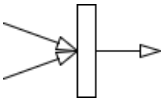
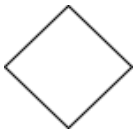
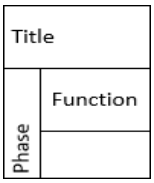
**(Sumber: Urva dan Siregar (2015:94)**

*b. Activity Diagram*

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Perlu diperhatikan bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut adalah simbol-simbol yang ada pada diagram aktivitas:

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

| Gambar | Nama | Arti/Keterangan |
|---|---|---|
|  | *Start Point* | *Start Point*, diletakkan pada pojok kiri atas dan merupakan awal aktivitas. |
|  | *End Point* | *End Point*¸ akhir dari aktivitas. |

| | | |
|---|---|---|
|  | *Action* | *Action/Activities*, menggambarkan suatu proses kegiatan/bisnis. |
|  | *Fork Node* | *Fork Node* (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu. |
|  | *Join Node* | *Join Node (Penggabungan), digunakan untuk menunjukkan adanya dekomposisi.* |
|  | *Decision Points* | *Decision Points, menggambarkan pilihan untuk pengambilan keputusan (true/false).* |
|  | *Swimline* | *Swimline, pembagian activity diagram untuk menunjukkan siapa melakukan apa.* |

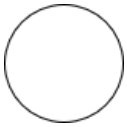
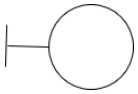
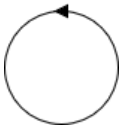
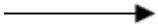
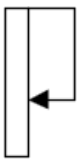
**(Sumber: Urva dan Siregar (2015:94))**

c. *Sequence Diagram*

*Sequence Diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Untuk menggambarkan *sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah

*use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

**Tabel II.3. *Sequence Diagram***

| Gambar | Nama | Arti/Keterangan |
|---|---|---|
|  | *Entity Class* | *Entity Class*, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data. |
|  | *Boundary Class* | *Boundary Class*, berisi kumpulan kelas yang menjadi *interface* atau interaksi antara satu atau lebih actor dengan sistem, seperti tampilan formentry dan *form* cetak. |
|  | *Control Class* | *Control Class*, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contoh nya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek. |
|  | *Message* | *Message*, simbol mengirim pesan antar *class*. |
|  | *Recursive* | *Recursive*, menggambarkan pengiriman pesan yang dikirimkan untuk dirinya sendiri. |

| | | |
|---|---|---|
|  | *Activation* | *Activation*, *activation* mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi. |
|  | *Lifeline* | *Lifeline, garis titik-titik yang terhubung dengan objek, sepanjang lifeline terdapat activation.* |

**(Sumber: Urva dan Siregar (2015:95))**