BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Tanzil Jiro Ahyana, dkk.;.2019 melakukan penelitian dengan judul "Perancangan Sistem Informasi Penjadwalan Mengajar menggunakan Metode *Algoritma* Genetika (Studi Kasus : SMK Satria Jakarta)". Penjadwalan merupakan kegiatan yang dibuat untuk dapat membantu dalam melakukan aktivitas sehari-hari. Pentingnya penjadwalan ini agar kegiatan dapat berjalan dengan sesuai yang telah di rencanakan dan di tata rapi. Proses dalam membuat jadwal di SMK Satria Serengseng Jakarta Barat masih menggunakan *Microsoft excel* yang berisi dari bagian komponen seperti guru, mata pelajaran dan kelas pada komponen waktu, sehingga masih sering terjadi bentrok antara salah satu jadwal dengan jadwal lainnya. Untuk mengatasi hal tersebut, peneliti memiliki upaya untuk merancang sistem penjadwalan menggunakan algoritma genetika. *Algoritma genetika* merupakan salah satu algoritma yang sangat tepat digunakan untuk menyelesaikan masalah optimasi kompleks yang sulit dilakukan oleh metode konvensional. Penelitian ini merupakan aplikasi penjadwalan mengajar menggunakan metode *Algoritma* Genetika. Adapun penelitian yang akan dibangun adalah suatu aplikasi yang tidak menerapkan metode *Algoritma* Genetika, namun aplikasi yang akan dibangun oleh peneliti akan menghasilkan suatu aplikasi penjadwalan teknisi berbasis *Android* menggunakan metode *Agile (model scrum).*

Irvan Azhary Chusna, dkk.;.2017 melakukan penelitian berjudul "Rancan Bangun Sistem Penjadwalan Guru Mengajar Berbasis *Web* (Studi Kasus : SMPN 2 Dawarblandong, Mojokerto)". SMP Negeri 2 Dawarblandong merupakan jenjang pendidikan dasar pada pendidikan formal di Indonesia setelah lulus Sekolah Dasar. Pada SMP Negeri 2 Dawarblandong, penjadwalan guru mengajar menjadi permasalahan yang sangat rumit apabila hanya ditangani menggunakan komputer saja dan masih menggunakan *microsoft excel* dan banyak keluhan yang di alami. Ketika ingin menyusun jadwal guru mengajar dan terjadi bentrokan data atau kesamaan jam, waktu, dan kelas solusi yang saat ini berjalan masih secara manual dengan cara memberi warna pada jadwal guru yang bentrokan data atau kesamaan jadwal. Hasil penelitian yang dibangun adalah sistem penjadwalan mengajar guru berbasis *Web*. Adapun perancangan yang akan dibangun oleh peneliti adalah aplikasi penjadwalan teknisi berbasis *Android* menggunakan metode *Agile (model scrum).*

Kurniawan Ikhwansyah, dkk.;.2019 melakukan penelitian dengan judul "Pemodelan *SCRUM* Dalam Pengembangan Sistem Informasi Kesehatan Pada Klinik Ar-Rokhim Sragen Kabupaten Sragen". Proses bisnis pada klinik Ar-Rokhim pada bagian pendaftaran, pencatatan rekam medis pasien, penjualan obat, pembuatan laporan dan penghitungan biaya rawat belum memaksimalkan aplikasi sistem yang ada. Hal ini membuat proses bisnis yang dilakukan oleh klinik kurang efektif. Salah satu metode pengembangan sistem yang sering digunakan untuk mengembangkan sistem yaitu *waterfall*, metode ini memiliki kekurangan karena

tidak diperkenankan adanya perubahan pada siklus pengembangan, membutuhkan tim yang banyak dan waktu pengembangan yang cukup lama. Untuk itu pada penelitian ini menggunakan salah satu *Agile sofware development* yaitu *SCRUM* model. Selain memiliki kelebihan *requiredment* yang fleksibel, jumlah anggota tim yang diperlukan tidak banyak. Tujuan dari penelitian ini ialah untuk menghasilkan pengembangan sistem informasi dengan cepat dalam penyesuaian perubahan dan sesuai dengan target yang diharapkan. Berdasarkan permasalahan tersebut, dipandang perlu untuk melakukan pengembangan sistem informasi kesehatan pada klinik Ar-Rokhim Sragen khususnya Pemodelan *Scrum*. Penelitian tersebut telah menghasilkan aplikasi berbasis *Web* untuk pengembangan sistem informasi kesehatan menggunakan Metode *Agile Model Scrum* sedangkan pada penelitian yang akan di buat oleh penulis mengenai penjadwalan teknisi lapangan berbasis *Android* menggunakan Metode *Agile Model Scrum*.

Irfan Mahendra, dkk.;.2018 melakukan penelitian dengan judul "*Agile Development Methods* Dalam Pengembangan Sistem informasi Pengajuan Kredit Berbasis *Web* (Studi Kasus : Bank BRI Unit Kolonel Sugiono)". Proses pengajuan kredit pada Bank BRI Unit Kolonel Sugiono masih dilakukan secara manual. Hal ini menyebabkan proses pengajuan kredit tidak efektif dan membutuhkan waktu yang cukup lama, sehingga dapat berpengaruh negatif terhadap pertumbuhan bisnis perusahaan ditengah persaingan bisnis yang semakin ketat dan kompleks. Berdasarkan permasalahan tersebut, dipandang perlu untuk melakukan pengembangan sistem informasi pengajuan kredit berbasis *Web* menggunakan

Agile Development Methods, khususnya *Model Scrum*. Penerapan *Agile Development Methods (Model Scrum)* yang iteratif, cepat, adaptif, dan secara aktif melibatkan pengguna di dalam kegiatan pengembangan sistem informasi, telah terbukti dapat menghasilkan sistem informasi yang sesuai dengan kebutuhan pengguna dalam waktu yang singkat. Penelitian tersebut telah menghasilkan aplikasi berbasis *Web* untuk sistem informasi pengajuan kredit berbasis *Web* menggunakan Metode *Agile* sedangkan pada penelitian yang akan di buat oleh penulis mengenai penjadwalan teknisi lapangan berbasis *Android* menggunakan Metode *Agile*.

II.2. Bab Studi Literatur

II.2.1. Aplikasi

Secara istilah pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Menurut kamus *computer* eksekutif, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu tehnik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang di harapkan. Pengertian aplikasi menurut Kamus Besar Bahasa Indonesia, “Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. (Andi Juansyah ; 2015).

II.2.2. Penjadwalan

Penjadwalan kegiatan perbaikan/pemasangan jaringan dan *spareparts* dilapangan dimaksudkan sebagai pengaturan perencanaan jadwal yang terdiri atas jumlah jadwal yang sudah dijadwalkan per teknisi. Secara umum penjadwalan kegiatan perbaikan/pemasangan disajikan dalam sebuah tabel yang memuat id jadwal, tujuan, keterangan, tanggal jadwal dan status yang sesuai dengan penjadwalan yang dilakukan. Penjadwalan teknisi yang dipilih yakni penjadwalan perbaikan, pemasangan dimana dalam setiap jadwal adanya status jadwal yang menyatakan apakah jadwal tersebut sudah selesai dilaksanakan oleh teknisi lapangan.

II.2.3. Bahasa Pemrograman Java

Java adalah bahasa pemrograman yang dapat dijalankan 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. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi kedalam *p-code (bytecode)* dan dapat dijalankan pada berbagai Mesin *Virtual Java (JVM)*. Java merupakan bahasa pemrograman yang bersifat umum dan secara khusus di desain untuk memanfaatkan dependensi implementasi seminimal mungkin.

(Maria. W. H Barri, dkk ; 2015).

II.2.4. *Android*

Android adalah sebuah *toolkit software* yang baru untuk perangkat bergerak yang dibuat oleh *Google* dan *Open Handset Alliance*. Dalam beberapa tahun, *Android* diharapkan dapat ditemukan dalam jutaan *handphone* dan berbagai perangkat bergerak membuat *Android* menjadi *platform* utama untuk pengembang *aplikasi.* Sudah ada banyak *platform mobile* di pasar saat ini, termasuk *Symbian, iPhone, Windows Mobile, Blackberry, Java Mobile Edition, Linux Mobile (Limo).* Meskipun beberapa fitur telah muncul sebelumnya, *Android* adalah *platform* yang menggabungkan beberapa hal berikut :

1. *Android* merupakan sebuah *platform* yang berbasis *linux* dan *open source*. Pembuat *handset* menyukai hal ini karena mereka dapat menggunakan dan menyesuaikan *platform* tanpa membayar *royalti.*
2. Sebuah *Arsitektur* berbasis komponen. Bagian dari *aplikasi Android* dapat digunakan sebagai bahan lain oleh *developer.*
3. Banyak *built-in service* yang tidak biasa. Servis berdasarkan lokasi menggunakan *GPS* atau *cell tower triangulation* yang membuat pengalaman pemakai terjadi bergantung lokasi. (Ahmad, dkk ; 2015).

Sistem operasi *Android* atau *OS Android* terdiri dari beberapa versi, setiap versi *Android* terbaru memiliki nama-nama unik tersendiri, berikut daftar nama 10 *OS Android* tahun terakhir *:*

1. Android 3.0 & 3.2 : Honeycomb

Untuk para pengguna *smartphone* mungkin akan agak asing dengan versi

Android yang satu ini. *Android* 3.0 & 3.2 *Honeycomb* yang menggunakan ikon lebah ini memang diperuntukkan penggunaannya khusus untuk perangkat *tablet*. Tentu perilisan *Android* 3.0 & 3.2 *Honeycomb* pada 10 Mei 2011 ini untuk mendukung *Samsung* yang mulai merilis perangkat *tablet, Samsung Galaxy Tab Series* untuk menyaingi *Apple iPad*.

2. ***Android 4.0 : Ice Cream Sandwich***

   Selain itu, *Android* pun juga merilis versi *Android* 4.0 *Ice Cream Sandwich* yang kembali diperuntukkan untuk perangkat *smartphone*. *Android* 4.0 *Ice Cream Sandwich* sendiri dirilis pada 19 Oktober 2011 silam. Punya versi nama paling panjang hingga saat ini, *Android* 4.0 *Ice Cream Sandwich* memberikan banyak pembaruan. Mulai dari animasi yang semakin halus, sederhana, dan mudah digunakan.

3. ***Android* 4.1 & 4.3 : *Jelly Bean***

   Peningkatan signifikan terasa saat menggunakan *Android* 4.1 & 4.3 *Jelly Bean.* Sistem operasi ini sendiri pertama kali dirilis pada Juni 2012 dengan membawa sejumlah peningkatan terutama di sektor pengolahan grafis. Dengan begini, tentu *Android* 4.1 & 4.3 *Jelly Bean* bisa memberikan peningkatan fungsi pada *user interface* dan *teknologi Vsync* yang digunakannya.

4. ***Android* 4.4 : *KitKat***

   Menggunakan nama *brand* cemilan terkenal, *Android* 4.4 *KitKat* pertama kali dirilis pada Oktober 2013. Versi *Android* terbaik ini pun bisa dikatakan menjadi favorit karena mendukung hampir seluruh perangkat *smartphone* di

dunia. Hal ini dikarenakan *Android* 4.4 *KitKat* dapat memberikan optimalisasi yang baik termasuk pada perangkat *smartphone* yang memiliki *spesifikasi* kurang mumpuni alias cukup rendah saat itu.

5. ***Android* 5.0 & 5.1: *Lollipop***

   Mulai beberapa versi ke belakang, *Android* dan *Google* pun mulai secara rutin memperbarui sistem operasi mereka dalam selang waktu setahun. Termasuk *Android* 5.0 & 5.1 *Lollipop* yang dirilis dan diresmikan pada Juni 2014. Bisa dibilang *Android* 5.0 & 5.1 *Lollipop* menjadi *pionir* dibuatnya *smartphone flagship* dengan *spesifikasi* cukup mumpuni. Versi *Android* ini sudah mendukung arsitektur 64-bit yang sudah memungkinan penggunaan RAM di atas 3GB. Salah satunya *ASUS Zenfone* 2 yang sudah mengusung RAM 4GB saat itu.

6. ***Android* 6.0 : *Marshmallow***

   Android 6.0 *Marshmallow* menjadi suksesor dari versi *Android* sebelumnya. Sistem operasi ini sendiri pertama kali diperkenalkan pada Mei 2015 dan mulai dirilis pada Oktober 2015 silam. Sistem operasi ini secara jelas memberikan peningkatan pada sistem keamanan dengan dihadirkannya *fingerprint sensor* sebagai sistem keamanan *biometrik* yang digunakan. Selain digunakan untuk mengunci layar, *fingerprint sensor* ini dapat digunakan untuk otentikasi *Google Play Store* dan pembelian dengan menggunakan *Android Pay*.

7. ***Android* 7.0 & 7.1 : *Nougat***

   Untuk saat ini, sistem operasi *Android* ini masih digunakan

beberapa *smartphone* yang baru dirilis belakangan ini. *Android* 7.0 & 7.1 *Nougat* pertama kali diperkenalkan pada Juni 2016 dengan menampilkan ikon robot *Android* dengan batangan *Nougat*. Sistem operasi *Android* 7.0 & 7.1 *Nougat* mengalami perubahan dari segi tampilan antarmuka. Selain itu ada juga *fitur splitscreen* untuk membagi tampilan layar untuk dua aplikasi sekaligus.

8. Android* 8.0 & 8.1 : *Oreo

Android 8.0 & 8.1 *Oreo* menjadi sistem operasi *Android* paling terbaru yang banyak digunakan saat ini. Sistem operasi ini dirilis secara stabil mulai Agustus 2017 sudah mengalami pembaruan lewat versi *Android* 8.1 *Oreo*. *Android* 8.0 & 8.1 *Oreo* menjadi versi *Android* kedua yang menggunakan makanan manis dari nama *brand* terkenal setelah *Android* 4.4 *KitKat*. Sistem operasi ini menawarkan pengalaman *multitasking* yang makin mumpuni dibanding versi sebelumnya.

9. Android 9.0 : Pie

Kemudian ada *Android* 9.0 *Pie* yang secara resmi diperkenalkan pada Agustus 2018. Sistem operasi *Android* ini memberi banyak perubahan, terutama untuk HP dengan desain baru. Misal *Android* 9.0 *Pie* memberikan navigasi berupa *gesture* yang menggantikan tombol fisik *Home, Back,* dan *Recent Apps*.

Fitur lainnya yang cukup berguna adalah sistem notifikasi, pengatur kecerahan, hingga sistem *screenshoot* terbaru yang lebih memudahkan.

10. Android Q Beta

Terakhir ada *Android Q Beta* yang baru diluncurkan pada 13 Maret 2019 dan saat ini masih terbatas pada beberapa perangkat HP *Android* saja. Seperti pada seri *smartphone Google*, yakni *Google Pixel, Google Pixel XL, Google Pixel 2, Google Pixel 2 XL, Google Pixel 3, Google Pixel 3 XL,* dan *Google Pixel 3 Lite.* Salah satu fitur *Android Q Beta* adalah *Dark Mode* alias mode gelap yang diklim mampu meningkatkan performa baterai. (https://jalantikus.com/tips/urutan-versi-android/).

II.2.5. *Android Studio*

Android Studio merupakan sebuah *IDE (Integrated Development Environment)* untuk pengembangan aplikasi android, aplikasi ini dipublikasikan oleh *Google* pada tanggal 16 mei 2013 dan tersedia secara gratis dibawah *lisensi Apache 2.0, Android Studio* ini menggantikan *software* pengembangan *android* sebelumnya yaitu *Eclipse*. (Efmi Maiyana ; 2018)

II.2.6. *Android SDK* (*Software Development Kit*)

Android-SDK adalah *tools API (Application Programming Interface)* yang diperlukan untuk memulai pengembangan aplikasi pada *Platform Android* menggunakan bahasa pemrograman *Java.* Pada *Android SDK* ini terdiri dari *debugger, libraries, handset emulator*, dokumentasi, kode contoh dan tutorial. *SDK* memungkinkan pengembang membuat aplikasi untuk *platform Android SDK, Android* mencakup proyek sampel dengan kode sumber, perangkat

pengembangan, *emulator* dan perpustakaan yang diperlukan untuk membangun aplikasi *Android.* Aplikasi yang ditulis dengan bahasa pemrograman *Java* dan berjalan di *Dalvik*, mesin *virtual* yang dirancang khusus untuk penggunaan *embedded* yang berjalan di atas *kernel Linux.* (Efmi Maiyana ; 2018)

II.2.7. *JDK* (*Java Development Kit*)

JDK (Java Development Kit) adalah paket fungsi *API* untuk bahasa pemrograman *JAVA*, meliputi *Java Runtime Environment (JRE)* dan *Java Virtual Machine (JVM)*. (Efmi Maiyana ; 2018).

II.2.8. *IDE* (*Integrated Development Environment)*

Integrated Development Environment adalah aplikasi pengembang perangkat lunak dengan fungsi-fungsi terintegrasi yang dibutuhkan untuk membangun sebuah perangkat lunak seperti *code editor, debugger, compiler,* dan sebagainya. (Efmi Maiyana ; 2018).

II.2.9. *AVD (Android Virtual Device)*

Android Virtual Device adalah merupakan *emulator* untuk menjalankan program aplikasi *Android* yang kita buat. *AVD* ini selanjutnya digunakan sebagai tempat untuk test dan menjalankan aplikasi *Android* tanpa harus menggunakan perangkat *Android* yang sebenarnya. Sebelum menggunakan *AVD* harus menentukan karakteristiknya, misalkan dalam menentukan versi *Android*, jenis

dan ukuran layar dan besarnya memori. *AVD* bisa dibuat sebanyak yang kita inginkan. (Efmi Maiyana ; 2018).

II.2.10. *PHP*

PHP atau kependekan dari *Hypertext Preprocessor* adalah salah satu bahasa pemrograman *open source* yang sangat cocok atau dikhususkan untuk pengembangan *Web* dan dapat ditanamkan pada sebuah skripsi HTML. Bahasa *PHP* dapat dikatakan menggambarkan beberapa bahasa pemrograman seperti C, *Java*, dan *Perl* serta mudah untuk dipelajari. *PHP* merupakan *bahasa scripting server - side,* dimana pemrosesan datanya dilakukan pada sisi *server*. Sederhananya, serverlah yang akan menerjemahkan *script program*, baru kemudian hasilnya akan dikirim kepada *client* yang melakukan permintaan. Adapun pengertian lain *PHP* adalah *akronim* dari *Hypertext Preprocessor*, yaitu suatu bahasa pemrograman berbasiskan kode - kode (*script*) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke *Web browser* menjadi kode HTML". (Astria, dkk ; 2016).

II.2.11. *Agile Software Development Methods*

Konsep *Agile Software Development* dicetuskan oleh Kent Beck dan 16 rekannya dengan menyatakan bahwa *Agile Software Development* adalah cara membangun *software* dengan melakukannya dan membantu orang lain membangunnya sekaligus. *Agile Software Development Methods atau Agile Methodology* merupakan sekumpulan metodologi pengembangan perangkat lunak

yang berbasis pada pengembangan iteratif, dimana persyaratan dan solusi berkembang melalui kolaborasi antar tim yang terorganisir.

Ada beberapa model pengembangan perangkat lunak yang termasuk *Agile Software Development Methods*, yaitu :

1. *Extreme Programming.*
2. *Adaptive Software Development.*
3. *Dynamic Systems Development Method.*
4. *Model Scrum.*
5. *Agile Modeling.*

Di dalam penelitian ini, model yang akan digunakan adalah *model scrum.* Menurut Pressman (2015) model *scrum* adalah metode pengembangan peranti lunak secara cepat *(Agile).* Prinsip *scrum* sesuai dengan prinsip-prinsip yang terdapat pada metode pengembangan peranti secara cepat yang digunakan untuk menuntun kegiatan pengembangan peranti lunak, seperti pemenuhan kebutuhan, analisa, desain dan penyampaian *(delivery).* Rangkaian kegiatan dalam model *scrum* terdiri dari :

1. Aktivitas *Backlog.*
2. Aktivitas *Sprints.*
3. Aktivitas *Scrum Meeting.*
4. Demo.

II.2.12. *MySQL*

MySQL adalah database *server open source* yang cukup popular keberadaannya. Dengan berbagai keunggulan yang dimiliki, membuat *software* database ini banyak digunakan oleh praktisi untuk membangun suatu *project*. Adanya fasilitas *API (Application Programming Interface)* yang dimiliki oleh *MySQL*, memungkinkan bermacam - macam aplikasi komputer yang ditulis dengan berbagai bahasa pemrograman dapat mengakses basis data *MySQL*. (Maria. W. H Barri, dkk ; 2015).

II.2.13. *Unified Modeling Language* (*UML*)

UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. *UML* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan *UML* tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya *UML* paling banyak digunakan pada metodologi berorientasi objek. (Rosa A.S, M. Shalahuddin, 2015:133).

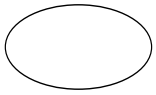
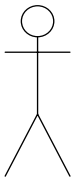
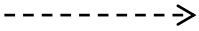
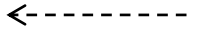
II.2.13.1 *Use Case* Diagram

Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan

siapa saja yang berhak menggunakan fungsi-fungsi itu. (Rosa A.S, M. Shalahuddin, 2015:155).

Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.1 dibawah ini :

Tabel II.1. Simbol *Use Case* Diagram

Gambar	Keterangan
	*Use case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama *use case*.
	Aktor adalah orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan di buat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal fase nama aktor.
	Asosiasi, komunikasi antara aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor.
	Asosiasi antara aktor dan *use case* yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	*Include*, merupakan di dalam *use case* lain (*required*) atau pemanggilan *use case* oleh *use case* lain, contohnya adalah pemanggilan sebuah fungsi program.
	*Extend*, merupakan perluasan dari *use case* lain jika kondisi atau syarat terpenuhi.

(*Sumber : Rosa.A.S. dan M.Shalahuddin, 2015, Hal : 156*)

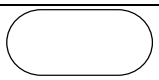
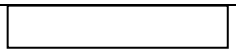
II.2.13.2 *Activity* Diagram

Diagram aktivitas atau *activity* diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem

bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. (Rosa A.S, M. Shalahuddin, 2015:161)

Simbol-simbol yang digunakan dalam *activity* diagram dapat dilihat pada tabel II.2 dibawah ini :

Tabel II.2. Simbol *Activity* Diagram

Gambar	Keterangan
	*Start point*, status awal aktifitas.
	*End point*, akhir aktifitas.
	*Activites*, menggambarkan suatu proses/kegiatan yang dilakukan sistem.
	*Decision* (percabangan), digunakan dimana jika ada pilihan aktivitas lebih dari satu.
	Join (penggabungan) dimana lebih dari satu aktivitas digabungkan menjadi satu.
	*Swimlane*, pembagian *activity* diagram untuk menunjukkan siapa melakukan apa.

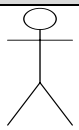
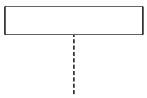
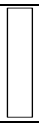
(*Sumber : Rosa.A.S. dan M.Shalahuddin, 2015, Hal : 162*)

II.2.13.3 *Sequence* Diagram

Diagram *sequence* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. (Rosa A.S, M. Shalahuddin, 2015:165).

Simbol-simbol yang digunakan dalam *sequence* diagram dapat dilihat pada tabel II.3 dibawah ini :

Tabel II.3. Simbol *Sequence* Diagram

Gambar	Keterangan
	*Aktor* adalah orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
	Lifeline, menyatakan kehidupan suatu objek.
	*Objek*, menyatakan objek yang berinteraksi pesan.
	*Activation*, menyatakan objek dalam keadaan aktif dan berinteraksi.
	Pesan tipe *create*, menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
	Pesan tipe *call*, menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
	Pesan tipe *return*, menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu.
	Pesan tipe *destroy*, menyatakan suatu objek mengakhiri hidup objek yang lain.

(*Sumber : Rosa.A.S. dan M.Shalahuddin, 2015, Hal : 165-167*)

II.2.13.4 *Class* Diagram

Diagram kelas atau *class* diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. (Rosa A.S, M. Shalahuddin, 2015:141).

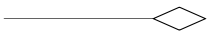
Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti yang dapat dilihat pada tabel II.4 dibawah ini :

Tabel II.4. *Multiplicity Class* Diagram

Multiplicity	Penjelasan
1	Satu dan hanya satu.
0..*	Boleh tidak ada atau 1 atau lebih.
1..*	1 atau lebih.
0..1	Boleh tidak ada, maksimal 1.
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4.

Simbol-simbol yang digunakan dalam *class* diagram dapat dilihat pada tabel II.5 dibawah ini :

Tabel II.5. Simbol *Class* Diagram

Gambar	Keterangan
	Kelas pada struktur sistem.
	*Interface,* sama dengan konsep *interface* dalam pemrograman berorientasi objek.
	Association, relasiantar kelas dengan makna umum, biasanya disertai dengan *multiplicity.*
	Asosiasi berarah, relasi antarkelas dengan makna kelas yang satu digunakan kelas yang lain.
	Generalisasi, relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).
	Agregasi, relai antarkelas dengan makna semua bagian (*whole-part*).

(*Sumber : Rosa.A.S. dan M.Shalahuddin, 2015, Hal : 146-147*)