

BAB II

LANDASAN TEORI

2.1 Sistem Pakar

2.1.1 Pengertian Sistem Pakar

Menurut Muhamad Fuad Satria Putra, Boko Susilo, dan Diah Puspitaningrum Jurnal Rekursif, Vol. 4 No. 2 Juni 2016, Sistem pakar adalah aplikasi berbasis komputer yang digunakan untuk menyelesaikan masalah sebagaimana yang diperoleh pakar. Pakar yang dimaksud disini adalah orang yang mempunyai keahlian, pengetahuan dan pengalaman khusus yang dapat menyelesaikan masalah yang tidak dapat diselesaikan oleh orang awam.

Secara umum, sistem pakar adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer yang dirancang untuk memodelkan kemampuan menyelesaikan masalah seperti layaknya seorang pakar. Dengan sistem pakar ini, orang awam pun dapat menyelesaikan masalahnya atau hanya sekedar mencari suatu informasi berkualitas yang sebenarnya hanya dapat diperoleh dengan bantuan para ahli di bidangnya. Sistem pakar ini juga akan dapat membantu aktivitas para pakar sebagai asisten yang berpengalaman dan mempunyai asisten yang berpengalaman dan mempunyai pengetahuan yang dibutuhkan.

Dalam penyusunannya, sistem pakar mengkombinasikan kaidah-kaidah penarikan kesimpulan (inference rules) dengan basis pengetahuan tertentu yang diberikan oleh satu atau lebih pakar dalam bidang tertentu. Kombinasi dari kedua

hal tersebut disimpan dalam komputer, yang selanjutnya digunakan dalam proses pengambilan keputusan untuk penyelesaian masalah tertentu.

Pada dasarnya sistem pakar diterapkan untuk mendukung aktifitas pemecahan masalah. Beberapa aktifitas pemecahan masalah yang dimaksud seperti :

a. Interpretasi

Membuat kesimpulan atau deskripsi dari sekumpulan data mentah.
Pengambilan keputusan dari hasil observasi.

b. Prediksi

Memproyeksikan akibat-akibat yang dimungkinkan dari situasi-situasi tertentu.

c. Diagnosis

Menentukan sebab malfungsi dalam situasi kompleks yang didasarkan pada gejala-gejala yang teramati diagnosis medis.

d. Perancangan (desain)

Menentukan konfigurasi komponen-komponen sistem yang cocok dengan tujuan-tujuan kinerja tertentu yang memenuhi kendala-kendala tertentu.

e. Perencanaan

Merencanakan serangkaian tindakan yang akan dapat mencapai sejumlah tujuan dengan kondisi awal tertentu.

f. Monitoring

Membandingkan hasil pengamatan dengan kondisi yang diharapkan.

g. Debugging

Menentukan dan menginterpretasikan cara-cara untuk mengatasi malfungsi.

h. Instruksi

Mendeteksi dan mengoreksi defisiensi dalam pemahaman domain subyek.

i. Kontrol

Mengatur tingkah laku suatu environment yang kompleks.

Dengan sistem pakar, pemakai dapat memperoleh informasi yang berkualitas dengan mudah seperti halnya memperoleh dari para ahli di bidangnya. Selain itu, sistem pakar juga dapat membantu aktifitas para pakar sebagai asisten yang mempunyai pengetahuan yang dibutuhkan.

2.1.2 Kelebihan Sistem Pakar

Sistem pakar memiliki beberapa kelebihan atau keuntungan antara lain seperti :

- a. Dengan adanya sistem pakar di sebuah tempat atau perusahaan, probabilitas dan frekuensi pengambilan keputusan yang baik menjadi tinggi. Sistem pakar dapat mendukung konsistensi dalam pengambilan keputusan. Pengembangan sistem pakar untuk memecahkan masalah dalam kehidupan nyata dengan cara yang berbeda telah memungkinkan untuk menolong seorang pakar (human expert).
- b. Dalam kebanyakan kasus, pengembangan sistem pakar akan mengurangi biaya pengambilan keputusan karena ketersediaan sistem pakar membuat penggunaan yang tepat dan efektif dari data yang tersedia.
- c. Sistem pakar memungkinkan pemeriksaan secara objektivitas dengan cara mempertimbangkan keterangan data dari pengguna tanpa memperhatikan kepribadian pengguna atau reaksi emosional pengguna.

- d. Sistem pakar memungkinkan seorang pakar untuk memiliki waktu luang dan lebih berkonsentrasi pada beberapa kegiatan yang bermakna lainnya.

2.1.3 Kekurangan Sistem Pakar

Tidak hanya memiliki banyak kelebihan, sistem pakar juga memiliki beberapa kekurangan, berikut beberapa kekurangan sistem pakar :

- a. Sistem pakar tidak banyak digunakan di perusahaan-perusahaan bisnis atau organisasi. Karena penggunaan yang terbatas, perusahaan masih ragu-ragu tentang kemampuan sistem pakar dan yang jelas biaya yang diperlukan cukup tinggi dalam membangun sistem pakar.
- b. Menggunakan sistem pakar sangat sulit dan belajar dan menguasai itu membutuhkan waktu yang lama. Hingga membuat seorang manajer enggan menggunakan sistem pakar. Dalam satu aspek, mengembangkan sistem pakar yang user-friendly adalah tantangan terbesar bagi pengembang sistem pakar.
- c. Ruang lingkup yang terbatas. Ini adalah kelemahan paling jelas dalam sistem pakar; ruang lingkup sangat terbatas. Dalam aspek pengembangan, sistem pakar yang dibangun adalah yang terbaik yang dikembangkan karena nilai akurasi yang tinggi. Namun, pengambil keputusan dapat menghadapi masalah yang terus berubah yang melibatkan berbagai bidang yang saling terkait.
- d. Sumber utama dari pengetahuan yang didapat dalam sistem pakar adalah seorang ahli/pakar. Manusia pasti bisa membuat kesalahan, jika seorang ahli memasukan informasi yang salah ke dalam sistem pakar, maka sistem

pakar tersebut dapat memiliki dampak negatif dari hasil yang di dikeluarkan.

2.1.4 Elemen Manusia Pada Sistem Pakar

Pengembangan sistem pakar dari awal hingga menghasilkan solusi akhir melibatkan peran serta 4 kelompok diantaranya :

a. Pakar (expert)

Pakar adalah individu yang memiliki pengetahuan khusus, pemahaman, pengalaman, dan metode-metode yang digunakan untuk memecahkan persoalan dalam bidang tertentu. Selain itu seorang pakar, juga memiliki kemampuan untuk mengaplikasikan pengetahuannya dan memberikan saran serta pemecahan masalah pada domain tertentu. Pada skripsi ini saya mengambil pakar Dr. Erlina Burhan, Sp.P(K). Seorang dokter spesialis paru dari Divisi Infeksi Departemen Pulmonologi dan Kedokteran Respirasi Fakultas Kedokteran Universitas Indonesia.

b. Pembangun pengetahuan (*knowledge engineer*)

Pembangun pengetahuan adalah individu yang memiliki tugas menerjemahkan dan mempresentasikan pengetahuan yang diperoleh dari pakar, baik berupa pengalaman pakar dalam menyelesaikan masalah maupun sumber terdokumentasi lainnya ke dalam bentuk yang diterima oleh sistem. Dalam hal ini, pembangun pengetahuan mengintrepetasikan dan merepresentasikan pengetahuan dalam bentuk jawaban atas pertanyaan – pertanyaan yang diajukan pada pakar atau pemahaman, penggambaran analogis, sistematis, konseptual yang diperoleh dari

membaca beberapa dokumen cetak seperti text book, jurnal, makalah, dan sebagainya.

c. **Pembangun Sistem (*system engineer*)**

Pembangun sistem adalah individu yang bertugas untuk merancang antar muka pemakai sistem pakar, merancang pengetahuan yang sudah diterjemahkan oleh pembangun pengetahuan ke dalam bentuk yang sesuai dan dapat diterima oleh sistem pakar dan mengimplementasikan ke dalam mesin inferensi. Selain itu, pembangun sistem juga bertanggung jawab apabila sistem pakar akan diintegrasikan dengan sistem komputerisasi lain.

d. **Pemakai (*user*)**

Banyak sistem berbasis komputer mempunyai susunan pemakai tunggal.

2.1.5 Metode *Forward Chaining*

Menurut Lukman Abdul Hafiz, Ernawati, Desi Andreswari Jurnal Rekursif, Vol. 6 No. 1 Maret 2018, Metode *Forward Chaining* adalah metode pencarian atau teknik pelacakan ke depan yang dimulai dengan informasi yang ada dan penggabungan rule untuk menghasilkan suatu kesimpulan atau tujuan. Pelacakan maju ini sangat baik jika bekerja dengan permasalahan yang dimulai dengan rekaman informasi awal dan ingin dicapai penyelesaian akhir, karena seluruh proses akan dikerjakan secara berurutan maju. Berikut adalah diagram *Forward Chaining* secara umum untuk menghasilkan sebuah *goal*. *Forward Chaining* merupakan metode inferensi yang melakukan penalaran dari suatu masalah kepada solusinya. Jika klausa premis sesuai dengan situasi (bernilai TRUE), maka proses akan menyatakan konklusi. *Forward Chaining* adalah *data driven* karena

inferensi dimulai dengan informasi yang tersedia dan baru konklusi diperoleh. Jika suatu aplikasi menghasilkan tree yang lebar dan tidak dalam, maka gunakan *Forward Chaining*. Tipe sistem yang dapat dicari dengan *Forward Chaining* :

1. Sistem yang dipersentasikan dengan satu atau beberapa kondisi.
2. Untuk setiap kondisi, sistem mencari rule-rule dalam *knowledgebase* untuk rule-rule yang berkorespondensi dengan kondisi dalam bagian *IF*
3. Setiap rule dapat menghasilkan kondisi baru dari konklusi yang diminta pada bagian *THEN*. Kondisi baru ini ditambahkan ke kondisi lain yang sudah ada.
4. Setiap kondisi yang ditambahkan ke sistem akan diproses. Jika ditemui suatu kondisi baru dari konklusi yang diminta, sistem akan kembali ke langkah 2 dan mencari rule-rule dalam *knowledgebase* kembali. Jika tidak ada konklusi baru, sesi ini berakhir.

2.1.6 Kelebihan Metode Forward Chaining

Berikut adalah kelebihan metode forward chaining :

- a. Kelebihan utama dari forward chaining yaitu metode ini akan bekerja dengan baik ketika problem bermula dari mengumpulkan/menyatukan informasi lalu kemudian mencari kesimpulan apa yang dapat diambil dari informasi tersebut.
- b. Metode ini mampu menyediakan banyak sekali informasi dari hanya sejumlah kecil data.

- c. Merupakan pendekatan paling sempurna untuk beberapa tipe dari problem solving task, yaitu planning, monitoring, control, dan interpretation.

2.1.7 Kekurangan Metode Forward Chaining

Berikut adalah kekurangan metode forward chaining :

- a. Kelemahan utama metode ini yaitu kemungkinan tidak adanya cara untuk mengenali dimana beberapa fakta lebih penting dari fakta lainnya.
- b. Sistem bisa saja menanyakan pertanyaan yang tidak berhubungan, walaupun jawaban dari pertanyaan tersebut penting, namun hal ini akan membingungkan user untuk menjawab pada subjek yang tidak berhubungan pada metode forward chaining.

2.2 Coronavirus

Coronavirus merupakan keluarga besar virus yang menyebabkan penyakit pada manusia dan hewan. Pada manusia biasanya menyebabkan infeksi saluran pernapasan, mulai flu biasa hingga penyakit serius seperti Middle East Respiratory Syndrome (MERS) dan Sindrom Pernapasan Akut Berat/ Severe Acute Respiratory Syndrome (SARS). Penyakit ini terutama menyebar di antara orang-orang melalui tetesan pernapasan dari batuk dan bersin . Virus ini dapat tetap bertahan hingga tiga hari dengan plastik dan stainless steel SARS CoV-2 dapat bertahan hingga tiga hari, atau dalam aerosol selama tiga jam . Virus ini juga telah ditemukan di feses, tetapi hingga Maret 2020 tidak diketahui apakah penularan melalui feses mungkin, dan risikonya diperkirakan rendah.

2.2.1 Karakteristik Epidemiologi

A. Orang Dalam Pemantauan (ODP)

Seseorang yang mengalami gejala demam ($\geq 38^{\circ}\text{C}$) atau memiliki riwayat demam atau ISPA tanpa pneumonia. Selain itu seseorang yang memiliki riwayat perjalanan ke negara yang terjangkit pada 14 hari terakhir sebelum timbul gejala juga dikategorikan sebagai dalam pemantauan.

B. Pasien Dalam Pengawasan (PDP)

- 1) Seseorang yang mengalami memiliki riwayat perjalanan ke negara yang terjangkit pada 14 hari terakhir sebelum timbul gejala-gejala covid-19 dan seseorang yang mengalami gejala-gejala, antara lain: demam ($>38^{\circ}\text{C}$), batuk, pilek, dan radang tenggorokan, pneumonia ringan hingga berat berdasarkan gejala klinis dan gambaran radiologis, serta pasien dengan gangguan sistem kekebalan tubuh (*immunocompromised*) karena gejala dan tanda menjadi tidak jelas.
- 2) Seseorang dengan demam $>38^{\circ}\text{C}$ atau ada riwayat demam atau ISPA ringan sampai berat dan pada 14 hari terakhir sebelum timbul gejala, memiliki salah satu dari paparan berikut. Riwayat kontak dengan kasus konfirmasi covid-19, bekerja atau mengunjungi fasilitas kesehatan yang berhubungan dengan pasien konfirmasi covid-19, memiliki riwayat perjalanan ke Provinsi Hubei, memiliki sejarah kontak dengan orang yang memiliki riwayat perjalanan pada 14 hari terakhir ke Provinsi Hubei.

C. Orang Tanpa Gejala (OTG)

Menurut Pedoman Pencegahan dan Pengendalian *Coronavirus Disease* (COVID-19) yang dikeluarkan oleh Kemenkes, OTG adalah seseorang yang tidak bergejala tapi berisiko tertular Virus Corona dari pasien COVID-19.

D. Suspect

Suspect corona adalah orang yang diduga kuat terjangkit infeksi COVID-19 karena pernah melakukan kontak dekat dengan pasien positif dan menunjukkan gejala terinfeksi.

E. Negatif

Pasien negatif adalah pasien yang tidak terbukti terinfeksi SARS-CoV-2 melalui hasil pengujian laboratorium. Istilah ini juga bisa diberikan kepada pasien yang sebelumnya dinyatakan positif corona, namun sudah sembuh dan terbebas dari virus itu.

F. Mekanisme Penularan

Covid-19 paling utama ditransmisikan oleh tetesan aerosol penderita dan melalui kontak langsung. Aerosol kemungkinan ditransmisikan ketika orang memiliki kontak langsung dengan penderita dalam jangka waktu yang terlalu lama. Konsentrasi aerosol di ruang yang relatif tertutup akan semakin tinggi sehingga penularan akan semakin mudah.

G. Karakteristik Kliniks

Berdasarkan penyelidikan epidemiologi saat ini, masa inkubasi covid-19 berkisar antara 1 hingga 14 hari, dan umumnya akan terjadi dalam 3 hingga 7 hari. Demam, kelelahan dan batuk kering dianggap sebagai manifestasi klinis utama. Gejala seperti hidung tersumbat, pilek, pharyngalgia, mialgia dan diare relatif jarang terjadi pada kasus yang parah, dispnea dan / atau hipoksemia biasanya terjadi setelah satu minggu setelah onset penyakit, dan yang lebih buruk dapat dengan cepat berkembang menjadi sindrom gangguan pernapasan akut, syok septik, asidosis metabolik sulit untuk dikoreksi dan disfungsi perdarahan dan batuk serta kegagalan banyak organ, dan lain-lain. Pasien dengan penyakit parah atau kritis mungkin mengalami demam sedang hingga rendah, atau tidak ada demam sama sekali. Kasus ringan hanya hadir dengan sedikit demam, kelelahan ringan dan sebagainya tanpa manifestasi pneumonia.

Dari kasus yang ditangani saat ini, sebagian besar pasien memiliki prognosis yang baik. Orang tua dan orang-orang dengan penyakit kronis yang mendasari biasanya memiliki prognosis buruk sedangkan kasus dengan gejala yang relatif ringan sering terjadi pada anak-anak.

2.3 Database

Menurut (Achmad Solichin, 2018) dalam E-Book " *MySQL 5 Dari Pemula Hingga Mahir*". *Basis data (atau database)* adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data

tersebut. *Database* digunakan untuk menyimpan informasi atau data yang terintegrasi dengan baik di dalam komputer.

2.4 Website

Menurut (Febrin Aulia Batubara, 2017) dalam Jurnal “*REINTEK. Vol.7, No.1.Tahun 2017. ISSN 1907-5030*” yang berjudul “Perancangan Webiste”. Website adalah halaman informasi yang disediakan melalui jalur internet sehingga bisa diakses di seluruh dunia selama terkoneksi dengan jaringan internet. Website merupakan komponen atau kumpulan komponen yang terdiri dari teks, gambar, suara animasi sehingga lebih merupakan media informasi yang menarik untuk dikunjungi.

2.5 Hypertext Preprocessor (PHP)

Menurut MADCOMS (2016) “*PHP (Hypertext Preprocessor)* adalah bahasa script yang dapat ditanamkan atau disisipkan ke dalam *HTML*. *PHP* banyak dipakai untuk membuat program situs *web* dinamis”.

2.6 MySQL Database

Menurut MADCOMS (2016) “*MySQL* adalah sistem manajemen *Database SQL* yang bersifat Open Source dan paling populer saat ini. Sistem *Database MySQL* mendukung beberapa fitur seperti multithreaded, multiuser dan *SQL Database management system (DBMS)*”.

2.7 Xampp

Menurut (Fitri Ayu , Nia Permatasari, 2018) dalam Jurnal “*Intra-Tech. Volume 2, No.2 Oktober 2018 ISSN. 2549-0222*” yang berjudul “perancangan sistem informasi pengolahan data”. *XAMPP* adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. Nama *XAMPP* merupakan singkatan dari X (empat sistem operasi apapun), *Apache*, *MySQL*, *PHP* dan *Perl*. Program ini tersedia dalam *GNU General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis.

2.8 HyperText Markup Language (HTML)

Menurut Jubilee Enterprise (Penerbit PT Elex Media Komputindo, 2016) dalam Buku “*Pengenalan HTML dan CSS*”. *HTML (Hyper Text Markup Language)* Adalah singkatan dari *HyperText Markup Language* yaitu bahasa pemrograman standar yang digunakan untuk membuat sebuah halaman *web*, yang kemudian dapat diakses untuk menampilkan berbagai informasi di dalam sebuah penjelajah *web Internet (Browser)*. *HTML* dapat juga digunakan sebagai *link* antara *file* dalam situs atau dalam komputer dengan menggunakan *localhost*, atau *link* yang menghubungkan antar situs dalam dunia internet.

2.9 JavaScript

Menurut (IpanRipai, M.Kom, 2017) dalam Jurnal “*ICT Learning Vol. 3 No. 1 Mei 2017 ISSN. 2569-0256*” yang berjudul “ rancangan bangun media pembelajaran menggunakan adroid”. *JavaScript* adalah bahasa pemrograman web yang bersifat *Client Side Programming Language*. *Client Side Programming Language* adalah tipe bahasa pemrograman yang pemrosesannya dilakukan oleh client. Aplikasi client yang dimaksud merujuk kepada *web browser*.

2.10 UML (Unified Modelling Language)

Menurut Rosa dan Shalahuddin (2018, h.137), *UML* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.





Selain itu, *UML* juga dikatakan sebagai metodologi kolaborasi antara metode-metode Booch, *OMT (Object Modeling Technique)*, serta *OOSE (Object Oriented Software Engineering)* dan beberapa metoda lainnya, merupakan metodologi yang paling sering digunakan saat ini untuk analisa dan perancangan sistem dengan metodologi berorientasi objek mengadaptasi maraknya penggunaan bahasa “pemrograman berorientasi objek” (*OOP*).



2.11 Use Case Diagram

Menurut Rosa dan Shalahuddin (2018, h.155), “*use case* merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat”. *Use case* digunakan untuk membentuk tingkah laku

benda dalam sebuah model serta direalisasikan oleh sebuah kolaborasi. Adapun simbol dari use case diagram antara lain :

Tabel 2.1 Simbol-simbol *Use Case Diagram*



Simbol	Keterangan
<p data-bbox="507 562 624 591"><i>Use Case</i></p> 	<p data-bbox="826 555 1374 808">Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya di-nyatakan dengan meng-gunakan kata kerja di awal frase nama <i>use case</i>.</p>
<p data-bbox="480 869 639 898">Aktor / <i>Actor</i></p>  <p data-bbox="480 1099 639 1128">Nama Aktor</p>	<p data-bbox="826 860 1374 1227">Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya menggunakan kata benda di awal frase nama aktor.</p>
<p data-bbox="437 1339 699 1368"><i>Asosiasi / Association</i></p> 	<p data-bbox="826 1272 1374 1413">Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor.</p>
<p data-bbox="459 1473 667 1503"><i>Ekstensi / Extend</i></p>  <p data-bbox="496 1697 639 1727"><<extend>></p>	<p data-bbox="826 1464 1374 1944">Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.</p>



<p>Generalisasi / <i>Generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum- khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>
<p>Menggunakan / <i>Include / uses</i></p> <p><<include>></p> 	<p>Relasi use case tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>usecase</i> tambahan dijalankan</p>

2.12 Activity Diagram

Menurut Rosa dan Shalahuddin (2018, h.161) “*activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Tabel 2.2 Simbol-simbol *Activity Diagram*



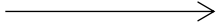
Gambar	Keterangan
 <p><i>Activity</i></p>	<p>Aktivitas yang dilakukan sistem, aktifitas biasanya diawali dengan kata kerja</p>
	<p>Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.</p>

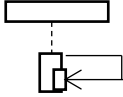
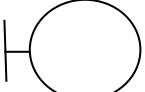


<i>Decision</i>	
 Status Awal	Status awal aktiviatas sistem, sebuah diagram aktivitas memiliki status awal.
 Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

2.13 Sequence Diagram

Menurut Rosa dan Shalahuddin (2018, h.165), “diagram sekuen menggambarkan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek”.

Tabel 2.3 Simbol-simbol *Sequence Diagram*

Gambar	Keterangan
 Aktor	Aktor adalah pengguna sistem, pengguna dapat berarti manusia, mesin atau sistem lain atau subsistem dari model apapun yang berinteraksi dengan sistem dari <i>boundary</i> sistem
 <i>Lifeline</i>	Peserta individu dalam interaksi (yaitu jalur hidup yang tidak dapat memiliki multiplisitas)
 <i>Message</i>	Menunjukkan aliran informasi atau kendali transaksi antar elemen

 <p><i>Sel-Message</i></p>	<p>Mencerminkan proses baru atau metode pemanggilan operasi <i>lifeline</i>. Ini adalah spesifikasi pesan biasanya dalam <i>sequence diagram</i></p>
 <p><i>Boundary</i></p>	<p><i>Boundary</i> adalah objek stereotip yang memodelkan batasan sistem. Biayanya layar <i>user interface</i></p>
 <p><i>Control</i></p>	<p><i>Control</i> adalah objek stereotip yang mengontrol atau mengatur entitas</p>
 <p><i>Entity</i></p>	<p>Merupakan table pada <i>database</i> yang merupakan model penyimpanan data yang menangkap dan menyimpan informasi dalam sistem</p>