

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Telah ada beberapa penelitian terdahulu yang dilakukan terkait penempatan area kerja.

Berdasarkan penelitian yang dilakukan oleh Sulistyaningsih, dkk (2018) mengenai perbandingan metode penilaian sederhana dan kombinasi SPK-360 derajat dalam penilaian kinerja petugas kebersihan, Sulistyaningsih, dkk menggunakan metode AHP dan TOPSIS untuk menilai kinerja petugas kebersihan sedangkan penelitian ini untuk penempatan area kerja *cleaning service*. Keistimewaan sistem yang peneliti buat adalah peneliti menggunakan dua metode untuk menghasilkan satu keputusan.

Berdasarkan penelitian yang dilakukan oleh Sulistyaningsih (2019) mengenai penerapan metode AHP-TOPSIS pada penilaian petugas kebersihan di RSUD Banyumas, Sulistyaningsih menggunakan metode AHP dan TOPSIS untuk menilai kinerja petugas kebersihan di RSUD Banyumas sedangkan penelitian ini untuk penempatan karyawan baru di PT. Multi Jasa Cemara. Keistimewaan sistem yang peneliti buat adalah PT. Multi Jasa Cemara dapat menempatkan karyawan baru yang sesuai di area kerja.

Judul Penelitian terdahulu yang diteliti oleh Euis Sitinur Aisyah, Bayu Pramono, Haris Munandar dari Universitas Raharja dan tempat penelitiannya di Bandara pada tahun 2019. Dengan masalah penempatan area kerja *cleaning service* masih manual dan banyak memakan waktu dan melibatkan beberapa bagian pada prosesnya. Oleh karena itu tujuan yang ingin dicapai pada penelitian ini adalah penerapan sistem yang memudahkan dalam prosesnya yaitu dengan membuat sistem informasi penempatan area kerja *cleaning service* berbasis web yang akan dijadikan acuan dalam proses pengembangan sistem nantinya. Metode analisa yang digunakan dalam penelitian ini menggunakan metode pieces yang fungsinya membantu dalam proses analisa sistem yang telah berjalan(Euis dkk:Vol:5;No:2 Agustus 2019)

II.2. Landasan Teori

II.2.1. Penempatan Area Kerja

Penempatan Kerja adalah kebijaksanaan terhadap sumber daya manusia untuk menentukan posisi atau jabatan seseorang. Penugasan ini dapat berupa penugasan pertama untuk karyawan baru yang akan direkrut, tetapi dapat juga melalui promosi, pengalihan (transfer) dan penurunan jabatan (demosi) atau bahkan pemutusan hubungan kerja (Badriyah:2015).

Menurut Rivai dan Sagala dalam Priansa (2016:124) Penempatan adalah mengalokasikan para karyawan pada posisi kerja tertentu hal ini khusus terjadi pada karyawan baru.

Menurut Sastrohadiwiryono dalam Priansa (2016:14) menyatakan Penempatan adalah proses pemberian tugas dan pekerjaan kepada tenaga kerja yang lulus penempatan kerja untuk dilaksanakan sesuai dengan ruang lingkup yang telah ditetapkan, serta mampu mempertanggung jawabkan segala risiko dan kemungkinan – kemungkinan yang terjadi atas tugas dan pekerjaan, wewenang, serta tanggung jawab.

Menurut Tohardi dalam Priansa (2016:125) menyatakan bahwa Penempatan adalah menempatkan pegawai pada pekerjaan yang sesuai dengan keterampilan atau pengetahuannya atau dengan kata lain proses mengetahui karakter atau syarat – syarat yang diperlukan untuk mengerjakan suatu pekerjaan tugas.

Menurut Priansa (2016:129) faktor – faktor yang mempengaruhi penempatan area kerja yaitu :

1. Faktor Prestasi Akademik

Faktor akademis yang telah dicapai oleh karyawan yang bersangkutan selama mengikuti jenjang pendidikan tertentu harus dapat dijadikan sebagai pertimbangan. Melalui pertimbangan faktor prestasi akademis maka karyawan tersebut dapat ditempatkan sesuai dengan prestasinya tersebut.

2. Faktor Pengalaman

Pengalaman kerja yang dimiliki karyawan hendaknya perlu mendapatkan pertimbangan pada saat penempatan karyawan. Semakin banyak

pengalaman yang dimiliki oleh karyawan maka kecenderungan karyawan untuk menguasai tugas dan pekerjaannya semakin tinggi.

3. Faktor Kesehatan Fisik dan Mental

Faktor fisik dan mental perlu dipertimbangkan dalam penempatan karyawan karena tanpa pertimbangan yang matang maka hal – hal yang bakal merugikan organisasi akan terjadi. Penempatan karyawan pada tugas dan pekerjaan tertentu harus disesuaikan dengan kondisi fisik dan mental karyawan yang bersangkutan.

4. Faktor Usia

Faktor usia merupakan salah satu pertimbangan dalam penempatan karyawan, karyawan dengan usia lebih muda relatif memiliki produktivitas dan kinerja yang tinggi dibandingkan dengan karyawan dengan usia yang lebih tua.

II.2.2. Cleaning Service

Cleaning service adalah pelayanan yang diberikan terhadap kebersihan suatu gedung atau bangunan lainnya yang dilakukan secara seksama dan menyeluruh dengan bantuan alat – alat kebersihan mesin non mesin serta bahan kimia (*chemical*) yang dilakukan oleh seorang petugas kebersihan. Tujuan *cleaning service* adalah menciptakan 5K yaitu Kebersihan, Keindahan, Kerapian, Keamanan, dan Kenyamanan pada gedung atau bangunan lainnya yang dilakukan oleh seorang petugas kebersihan.

Cleaning service sangat dibutuhkan karena kebersihan merupakan salah satu prioritas suatu perusahaan karena beberapa perusahaan banyak tempat yang harus diperhatikan lebih terutama bagian – bagian yang susah dijangkau terutama semisal diperkantoran yang mempunyai lantai yang tinggi dan harus dibersihkan kacanya dari luar.

II.2.3. Sistem Pendukung Keputusan

Konsep Sistem Pendukung Keputusan pertama kali diungkapkan pada awal tahun 1970-an oleh Michael S.Scott Morton dengan istilah Management Decision System. Sistem tersebut adalah suatu system yang berbasis computer yang ditunjukan untuk membantu pengambil keputusan dengan memanfaatkan data dan model tertentu untuk memecahkan berbagai persoalan yang tidak terstruktur (Dewanto,2015).

Sistem Pendukung Keputusan adalah suatu sistem informasi yang menggunakan model – model keputusan, basis data, dan pemikiran manajer sendiri, proses *modelling* interaktif dengan komputer untuk mencapai pengambilan keputusan oleh manajer tertentu . Dengan adanya system pendukung keputusan dapat memberikan perangkat interaktif yang memungkinkan *decission maker* melakukan berbagai analisis dari model yang tersedia (Utomo, et, al. 2015).

Ada beberapa karakteristik umum dari sebuah sistem pendukung keputusan yang membantu kita dalam memahami definisi yang ideal yaitu :

1. Sistem pendukung keputusan adalah sebuah sistem berbasis komputer dengan antarmuka antara mesin komputer dengan pengguna.
2. Sistem pendukung keputusan ditujukan untuk membantu pembuat keputusan dalam menyelesaikan suatu masalah dalam berbagai level manajemen dan bukan untuk mengganti posisi manusia sebagai pembuat keputusan.
3. Sistem pendukung keputusan mampu memberi alternatif solusi bagi masalah semi / tidak terstruktur baik bagi perseorangan atau kelompok dan dalam berbagai macam proses dan gaya pengambilan keputusan.
4. Sistem pendukung keputusan menggunakan data, basis data, dan analisa model – model keputusan.
5. Sistem pendukung keputusan bersifat adaptif, efektif, interaktif, *easy to use* dan fleksibel.
6. Sistem pendukung keputusan menyediakan akses terhadap berbagai macam format dan tipe sumber data.

Selanjutnya didalam sistem pendukung keputusan terdapat tiga tujuan yang harus dicapai yaitu :

1. Membantu manajer dalam pembuatan keputusan untuk memecahkan masalah semi terstruktur.
2. Mendukung keputusan manajer dan bukannya mengubah atau mengganti.
3. Meningkatkan efektivitas manajer dalam pembuatan keputusan dan bukannya peningkatan efisiensi.

Saat ini sistem pendukung keputusan salah satu jenis aplikasi teknologi informasi yang mendominasi perusahaan – perusahaan modern yang ingin meningkatkan kualitas manajemen dalam menunjang proses pengambilan keputusan.

Beberapa alasan mengapa perusahaan – perusahaan utama memulai pengembangan sistem pendukung keputusan dalam skala besar diuraikan sebagai berikut :

1. Kebutuhan akan informasi yang akurat.
2. Sistem pendukung keputusan dipandang sebagai pemenang secara organisasi.
3. Kebutuhan akan informasi yang baru.
4. Penyediaan informasi yang tepat waktu.
5. Pencapaian pengurangan biaya.

II.2.4. AHP (*Analitycal Hierarchy Process*)

Menurut Nugeraha (2017:114) mengemukakan bahwa AHP adalah sebuah konsep untuk pembuatan keputusan berbasis *multicriteria* (kriteria yang banyak).

Dan menurut (Firdaun, Abdillah, dan Renaldi, 2016) yang mengemukakan bahwa “*Analytic Hierarchy Process* (AHP) dapat menyelesaikan masalah multikriteria yang kompleks menjadi suatu hirarki. Masalah yang kompleks dapat diartikan bahwa kriteria dari suatu masalah yang begitu banyak (multi kriteria), struktur masalah yang belum jelas,

ketidakpastian pendapat dari pengambil keputusan, pengambila keputusan lebih dari satu orang, serta ketidak akuratan data yang tersedia”.

Tahapan – tahapan pengambilan keputusan dengan Metode AHP :

- a. Mengidentifikasi permasalahan kemudian membuat struktur hirarki dari permasalahan tersebut.
- b. Membandingkan elemen secara berpasangan menurut kriteria yang ditentukan.
- c. Matriks perbandingan berpasangan diisi bilangan yang menggambarkan tingkat kepentingan relatif dari setiap elemen.
- d. Menjumlahkan nilai-nilai dari setiap kolom pada matriks.
- e. Mencari matriks normalisasi dengan cara membagi setiap nilai dari kolom dengan total kolom.
- f. Menjumlahkan nilai–nilaidari setiap baris kemudian membaginya dengan jumlah elemen untuk memperoleh nilai rata-rata.

Setelah melakukan langkah – langkah diatas, selanjutnya adalah mengukur konsistensi dengan langkah–langkah sebagai berikut:

- a. Mengalikan nilai pada kolom pertama dengan prioritas relatif elemen pertama.
- b. Menjumlahkan setiap baris.
- c. Membagi hasil penjumlahan baris dengan elemen prioritas relative.
- d. Menjumlahkan hasil pembagian di atas dengan jumlah elemen yang ada. Hasilnya disebut λ maks.

- e. Menghitung *Consistency Indeks CI* menggunakan rumus:

$$CI = \frac{\lambda_{\max} - n}{n}$$

dimana : n = jumlah elemen

- f. Menghitung Rasio Konsistensi / *Consistency Ratio (CR)* menggunakan rumus:

$$CR = \frac{CI}{RI}$$

dimana : *CR = Consistency Ratio*

CI = Consistency Index

- g. Memeriksa Konsistensi hierarki, suatu data dikatakan benar apabila memiliki nilai rasio konsistensi kurang atau sama dengan 0,1.

Berikut ini adalah beberapa kelebihan AHP adalah :

1. Struktur yang berhirarki sebagai konsekuensi dari kriteria yang dipilih sampai pada sub – sub kriteria yang paling dalam.
2. Memperhitungkan validitas sampai dengan batas toleransi inkonsistensi berbagai kriteria dan alternatif yang dipilih oleh para pengambil keputusan.
3. Memperhitungkan daya tahan atau ketahanan output analisis sensitifitas pengambil keputusan.

II.2.5. TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution).

Metode TOPSIS pertama kali diperkenalkan oleh Hwang dan Yoon, TOPSIS digunakan sebagai salah satu metode dalam memecahkan masalah multi kriteria. Dalam bukunya menurut Marbun dan Sinaga (2018) metode TOPSIS memberikan sebuah solusi dari sejumlah alternatif dengan alternatif terbaik dan alternatif terburuk yang ada dalam alternatif – alternatif masalah. Metode ini menggunakan jarak untuk melakukan perbandingan tersebut.

TOPSIS akan merangking alternatif berdasarkan prioritas nilai kedekatan relatif suatu alternatif terhadap solusi ideal positif. Alternatif – alternatif yang telah dirangking kemudian dijadikan sebagai referensi bagi pengambil keputusan untuk memilih solusi terbaik yang diinginkan. Metode ini banyak digunakan untuk menyelesaikan pengambilan keputusan secara praktis. Hal ini disebabkan konsepnya sederhana dan mudah dipahami, komputasinya efisien dan memiliki kemampuan mengukur kinerja relatif dari alternatif – alternatif keputusan (Marbun dan Sinaga, 2018).

Langkah – langkah metode TOPSIS sebagai berikut:

- a. Menentukan matriks keputusan yang ternormalisasi

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{j=1}^m x_{ij}^2}}, (i = 1, 2, \dots, n; j = 1, 2, \dots, m)$$

Keterangan: X_{ij} merupakan rating kinerja alternatif ke- i terhadap atribut ke- j

r_{ij} adalah elemen dari matriks keputusan yang ternormalisasi.

- b. Menentukan matriks keputusan yang terbobot

$$y = \begin{bmatrix} y_{11} & y_{12} & \dots & y_{1j} \\ y_{21} & y_{22} & \dots & y_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ y_{i1} & y_{i2} & \dots & y_{ij} \end{bmatrix} \text{ untuk } y_{ij} = w_j r_{ij}$$

Keterangan: w_j adalah bobot dari kriteria ke- j

y_{ij} adalah elemen dari matriks keputusan yang ternormalisasi terbobot

- c. Menentukan matriks solusi ideal positif (A^+) dan matriks solusi ideal negatif (A^-)

$$A^+ = (y_1^+, y_2^+, \dots, y_j^+)$$

$$A^- = (y_1^-, y_2^-, \dots, y_j^-)$$

dengan

$$y_j^+ = \begin{cases} \max_i y_{ij}, & \text{jika } j = \text{keuntungan} \\ \min_i y_{ij}, & \text{jika } j = \text{biaya} \end{cases}$$

$$y_j^- = \begin{cases} \max_i y_{ij}, & \text{jika } j = \text{keuntungan} \\ \min_i y_{ij}, & \text{jika } j = \text{biaya} \end{cases}$$

- d. Menentukan jarak nilai alternatif dari matriks solusi ideal positif (d_i^+) dan matriks solusi ideal negatif (d_i^-), jarak solusi ideal positif (d_i^+)

$$d_i^+ = \sqrt{\sum_{j=1}^m (y_{ij} - y_j^+)^2}$$

Keterangan: y_j^+ adalah elemen dari matriks solusi ideal positif

$$d_i^- = \sqrt{\sum_{j=1}^m (y_{ij} - y_j^-)^2}$$

Keterangan: y_j^- adalah elemen dari matriks solusi ideal negatif

- e. Menentukan nilai preferensi (c_i) untuk setiap alternative, nilai preferensi merupakan kedekatan suatu alternatif terhadap solusi ideal

$$c_i = \frac{d_i^-}{d_i^- + d_i^+}$$

Keterangan: nilai C_i yang lebih besar menunjukkan prioritas alternatif.

II.2.6. Database

Menurut One Yunita, *database* diartikan sebagai gudang, tempat bersarang atau berkumpul. Basis data merupakan suatu media penyimpanan yang digunakan untuk menyimpan data – data penunjang sebagai inputan sistem dan kemudian diolah menjadi data output sistem. Prinsip utama *database* adalah pengaturan data dengan tujuan utama kecepatan dalam pengambilan data kembali. Adapun tujuan *database* adalah untuk menangani data dalam jumlah besar, meniadakan duplikasi dan dapat dicari secara cepat.

Database yang dibuat pada sistem informasi ini menggunakan *SQL Server* dan bahasa pemrogramannya memakai *VB. Net 2010*.

II.2.7. Visual Basic Net 2010

Visual Basic adalah sebuah bahasa pemrograman yang berpusat pada object (*Object Oriented Programming*) digunakan dalam pembuatan aplikasi windows yang berbasis *Graphical User Interface*. Beberapa karakteristik objek tidak dapat dilakukan oleh Visual Basic misalnya seperti *Inheritance* tidak bisa module dan *Polymorphism* secara terbatas bisa dilakukan dengan deklarasi *class module* yang mempunyai interface tertentu, sifat Visual Basic tidak *case sensitif*.

Sejarah Visual Basic berawal dari perkembangan bahasa Basic di Amerika Serikat pada awal tahun 1960 an. Pada tahun 1982 IBM/PC diperkenalkan pada masyarakat yang didalamnya disertakan pula bahasa BASIC yang dikenal juga QuickBasic / QBASIC. Pada tahun 1990-an DOS digantikan oleh Windows. Microsoft akhirnya membuat BASIC versi Windows yang dikenal sebagai Microsoft Visual Basic. Awal perkembangan Visual Basic yaitu Dirintis proyek “Thunder”.

II.2.8. SQL Server

SQL adalah singkatan dari (*Structured Query Language*)SQL server adalah sistem manajemen database relasional (RDBMS) yang dirancang untuk aplikasi dengan arsitektur client / server. Istilah client / server dapat

digunakan untuk merujuk kepada konsep yang sangat umum, jadi secara umum SQL Server adalah sebuah software yang dibuat oleh perusahaan microsoft yang digunakan untuk membuat database yang dapat diimplementasikan untuk Client Server. Client adalah setiap komponen dari sebuah sistem yang meminta layanan atau sumber daya (*resource*) dari komponen sistem lainnya. Server adalah setiap komponen sistem yang menyediakan layanan atau sumber daya ke komponen sistem lainnya.

II.2.9. Microsoft Visual Studio

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi bisnis, aplikasi personal atau pun komponen aplikasinya dalam bentuk aplikasi console, aplikasi windows, atau pun aplikasi web.

II.2.10. Unified Modeling Language (UML)

Unified Modelling Language (UML) merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah system software yang terkait dengan objek. UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan bahasa pemodelan visual yang memungkinkan pengembang sistem membuat blue print atas visinya dalam bentuk yang baku. UML berfungsi sebagai jembatan

dalam mengkomunikasikan beberapa aspek dalam sistem melalui jumlah elemen grafis yang bisa dikombinasikan menjadi diagram (Sirait, dkk:2015).

Unified Modelling Language (UML) juga merupakan sebuah bahasa yang sudah menjadi standar didunia industri untuk visualisasi, merancang dan mendokumentasi sistem piranti lunak.UML menawarkan sebuah standar untuk merancang model sebuah sistem (Sentosa Pohan, 2015:14).


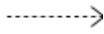





Pada tahap ini dilakukan desain terhadap sistem yang diusulkan secara keseluruhan. Desain sistem secara umum dapat digambarkan dengan, *Use Case Diagram*, *Class Diagram*, *Activity Diagram* dan *Sequence Diagram*.



1. *Use case Diagram*

Dalam penyusunan suatu program diperlukan suatu model data yang berbentuk diagram yang dapat menjelaskan suatu alur proses sistem yang akan di bangun. Dalam penulisan tugas akhir ini penulis menggunakan metode UML yang dalam metode itu penulis menerapkan Diagram *Use Case*. Maka digambarlah suatu bentuk Diagram *Use Case* yang dapat dilihat pada tabel II.1 dibawah ini.

Adapun simbol – simbol yang dibuat dalam *Use Case Diagram* adalah sebagai berikut :

Tabel II.1. Simbol *Use Case Diagram*

Gambar	Nama	Keterangan
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya.
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya



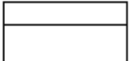
Sumber: Aprianti 2016


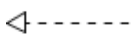
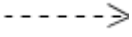

2. Class Diagram

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas – kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi (Aprianti,2016:2).

Class Diagram dapat dilihat pada Tabel II.2 berikut :

Tabel II.2.Simbol *Class Diagram*

No	Gambar	Nama	Keterangan
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.



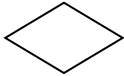


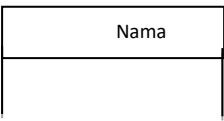
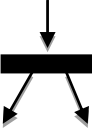
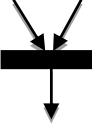
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

Sumber: Aprianti 2016

3. *Activity Diagram*

Activity Diagram menggambarkan *workflow* (aliran kerja) dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak (Aprianti,2016:3). *Activity diagram* dapat dilihat pada Tabel II.3 berikut :

Tabel II.3. Simbol *Activity Diagram*

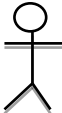



Simbol	Fungsi	Keterangan
	Status awal	Status awal aktivitas system, sebuah diagram aktivitas memiliki status awal
	Aktivitas	Aktivitas yang dilakukan system, aktivitas biasanya diawali dengan kata kerja
	Percabangan / <i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
	Penggabungan / <i>join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
	Status akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
	<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi
	<i>Fork</i>	Digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel
	<i>Join</i>	Digunakan untuk menunjukkan kegiatan yang digabungkan

Sumber: Aprianti 2016

4. *Sequence Diagram*

Sequence Diagram menggambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup objek dan *message* yang di kirimkan dan di terima antar objek. *Sequence Diagram* dapat di lihat pada Tabel II.4 berikut :

Tabel II.4 Simbol *Sequence Diagram*

Simbol	Fungsi	Keterangan
	<i>Actor</i>	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi dan mendapat manfaat dari sistem ditempatkan di bagian atas diagram
	<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

Sumber: Aprianti 2016