

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Penelitian Terdahulu**

Berdasarkan penelitian yang dilakukan oleh Salmon dan Harpad (2018) mengenai Penerapan Metode *Analytical Hierarchy Process* (AHP) Pada Pemilihan Staf Laboratorium Komputer Stmik Widya Cipta Dharma Samarinda, Salmon dan Harpad menyimpulkan hasil perhitungan metode *Analytical Hierartical Process* (AHP) dikembangkan perangkat lunaknya dengan *Macromedia Dreamweaver* dan database *MySQL* mendapatkan hasil yang dapat dijadikan rekomendasi kepada pihak pimpinan dalam menentukan staf yang layak diterima.

Berdasarkan penelitian yang dilakukan oleh Eka Safitri Nasution, Mesran, Natalia Silalahi (2018) mengenai Pengembangan Aplikasi Sistem Pendukung Keputusan Dalam Pemilihan Tempat Pemakaman Umum (TPU) Dengan Metode AHP, Salmon dan Harpad menyimpulkan sistem pendukung keputusan pemilihan lokasi tempat pemakaman umum (TPU), menampilkan metode *AHP* yang menggunakan kriteria-kriteria dapat menghasilkan suatu keputusan untuk menentukan lokasi tempat pemakaman umum yang baik dan tepat.

Berdasarkan penelitian yang dilakukan oleh Rafhael Stevanus, Rani Irma Handayani, Dinar Ajeng Kristiyanti (2018) mengenai Sistem Pendukung Keputusan Pemberian Bonus Karyawan Menggunakan Metode Ahp pada Rumah Sakit Buah Hati Ciputat, Stevanus, dkk Secara keseluruhan kriteria keahlian,sikap & perilaku,

loyalitas dan tanggung jawab mempengaruhi sistem pendukung keputusan pemberian bonus karyawan.

Berdasarkan penelitian yang dilakukan oleh Novica Irawati, Hommy D.E. Sinaga, Adyanata Lubis (2018) mengenai Sistem Pendukung Keputusan Penerimaan Dana Bos Untuk Sekolah Dasar Dengan Metode AHP (Dinas Pendidikan Kec. Sei Kepayang), Irawati, dkk menyimpulkan Sistem yang telah di kembangkan dengan metode AHP ini, dapat digunakan dengan jumlah faktor kriteria yang ditentukan oleh user sendiri, sehingga dapat disesuaikan dengan kebutuhan dalam pengambilan keputusan.

## **II.2. Landasan Teori**

Berikut ini adalah beberaa landasan teori yang diambil dari beberapa referensi-referensi yang terkait dengan penelitian :

### **II.2.1. Sistem**

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran yang tertentu. Sistem adalah sekelompok sistem yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan. (Setiawan dan Permadani, 2016 : 53).

### **II.2.2. Sistem Pendukung Keputusan**

Sistem Pendukung Keputusan (SPK) merupakan merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data. Sistem

itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semi terstruktur dan situasi tidak terstruktur, di mana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat. (Rohmat Taufiq dan Niswan Fiqih Fahrozi 2016 : 95).

Sistem pendukung keputusan/*Decision support system* (DSS) menurut Kusri dalam buku Konsep dan aplikasi sistem pendukung keputusan. Sistem pendukung keputusan merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data. sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semiterstruktural dan situasi yang tidak terstruktur, dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat. (Resky kestar Sitorus, 2017 : 52).

*Decision Support System* atau Sistem Pendukung Keputusan secara umum didefinisikan sebagai sebuah sistem yang mampu memberikan kemampuan, baik kemampuan pemecahan masalah maupun kemampuan mengkomunikasikan untuk masalah semi-terstruktur. Secara khusus, menurut Hermawan (2005) SPK didefinisikan sebagai sebuah sistem yang mendukung kerja seorang manajer maupun sekelompok manajer dalam memecahkan masalah semi-terstruktur dengan cara memberikan informasi ataupun usulan menuju pada keputusan tertentu. (Maria Sari, Linda Wahyuni, dan Lili Tanti, 2015 : 2).

### **II.2.2.1. Tujuan Sistem Pendukung Keputusan**

Sistem pendukung keputusan tidak dimaksudkan untuk mengotomatiskan pengambilan keputusan, tetapi memberikan perangkat interaktif yang memungkinkan pengambilan keputusan untuk melakukan berbagai analisis menggunakan model-model yang tersedia. Tujuan dari DSS adalah :

1. Membantu manajer dalam pengambilan keputusan atas masalah semistruktur.
2. Memberikan dukungan atas pertimbangan manajer dan bukannya dimaksudkan untuk menggantikan fungsi manajer.
3. Meningkatkan efektifitas keputusan yang diambil lebih daripada perbaikan efisiensinya.
4. Kecepatan komputasi. Komputer memungkinkan para pengambil keputusan untuk melakukan banyak komputasi secara cepat dengan biaya rendah.
5. Peningkatan produktivitas.
6. Dukungan kualitas.
7. Berdaya saing.
8. Mengatasi keterbatasan kognitif dalam pemrosesan dan penyimpanan. (Sylvia Saragih, 2015 : 83).

### **II.2.2.2. Landasan Utama Sistem Pendukung Keputusan**

Landasan utama dalam pengembangan SPK adalah konsepsi model. Konsepsi model ini menggambarkan hubungan abstrak antara 3 komponen utama dalam penunjang keputusan, yaitu pengambil keputusan atau pengguna, model dan

data. Masing-masing komponen tersebut dikelola oleh sebuah sistem manajemen. Ada empat tahap yang harus dilakukan dalam memecahkan suatu masalah. Tahap-tahap ini antara lain :

1. Tahap *Intelligence*
2. Tahap *Design*
3. Tahap *Choice*
4. Tahap *Implementation*. (Muhammad, Anwar, Syarifuddin, Amrullah, 2016 : 6).

### **II.2.3. Metode *Analytical Hierarchy Process* (AHP)**

Metode *Analytical Hierarchy Process* (AHP) dikembangkan oleh Prof. Thomas Lorie Saaty dari Wharton Business School di awal tahun 1970, yang digunakan untuk mencari ranking atau urutan prioritas dari berbagai alternatif dalam pemecahan suatu permasalahan. Dalam kehidupan sehari-hari, seseorang senantiasa dihadapkan untuk melakukan pilihan dari berbagai alternatif. Disini diperlukan penentuan prioritas dan uji konsistensi terhadap pilihan-pilihan yang telah dilakukan. (Resky Kestar Sitorus, 2017 : 52).

Peralatan utama AHP adalah sebuah hirarki fungsional dengan input utamanya persepsi manusia. Dengan hirarki, suatu masalah kompleks dan tidak terstruktur dipecahkan ke dalam kelompok-kelompok. Kemudian kelompok-kelompok tersebut diatur menjadi suatu bentuk hirarki. Proses yang terdapat pada langkah AHP adalah input kriteria dan sub kriteria disini menggunakan *predefined*

*process* yang maksudnya input yang prosesnya berada dalam tempat lain. Rohmat Taufiq dan Niswan Fiqih Fahrozi, 2016 : 96).

Berikut ini adalah tahapan metode AHP :

1. Mendefinisikan masalah dan menentukan solusi yang diinginkan.

Dalam tahap ini penulis berusaha menentukan masalah yang akan penulis pecahkan secara jelas, detail dan mudah dipahami. Dari masalah yang ada penulis coba tentukan solusi yang mungkin cocok bagi masalah tersebut. Solusi dari masalah mungkin berjumlah lebih dari satu. Solusi tersebut nantinya penulis kembangkan lebih lanjut dalam tahap berikutnya.

2. Membuat struktur hierarki yang diawali dengan tujuan utama.

Setelah menyusun tujuan utama sebagai level teratas akan disusun level hirarki yang berada di bawahnya yaitu kriteria-kriteria yang cocok untuk mempertimbangkan atau menilai alternatif yang penulis berikan dan menentukan alternatif tersebut. Tiap kriteria mempunyai intensitas yang berbeda-beda. Hirarki dilanjutkan dengan subkriteria (jika mungkin diperlukan).

3. Membuat matrik perbandingan berpasangan yang menggambarkan kontribusi relatif atau pengaruh setiap elemen terhadap tujuan atau kriteria yang setingkat di atasnya. Matriks yang digunakan bersifat sederhana, memiliki kedudukan kuat untuk kerangka konsistensi, mendapatkan informasi lain yang mungkin dibutuhkan dengan semua perbandingan yang mungkin dan mampu menganalisis kepekaan prioritas secara keseluruhan untuk perubahan pertimbangan. Pendekatan dengan matriks mencerminkan aspek ganda dalam prioritas yaitu mendominasi

dan didominasi. Perbandingan dilakukan berdasarkan *judgment* dari pengambil keputusan dengan menilai tingkat kepentingan suatu elemen dibandingkan elemen lainnya. Untuk memulai proses perbandingan berpasangan dipilih sebuah kriteria dari level paling atas hirarki misalnya K dan kemudian dari level di bawahnya diambil elemen yang akan dibandingkan misalnya E1,E2,E3,E4,E5.

4. Melakukan Mendefinisikan perbandingan berpasangan sehingga diperoleh jumlah penilaian seluruhnya sebanyak  $n \times [(n-1)/2]$  buah, dengan n adalah banyaknya elemen yang dibandingkan. Hasil perbandingan dari masing-masing elemen akan berupa angka dari 1 sampai 9 yang menunjukkan perbandingan tingkat kepentingan suatu elemen. Apabila suatu elemen dalam matriks dibandingkan dengan dirinya sendiri maka hasil perbandingan diberi nilai 1. Skala 9 telah terbukti dapat diterima dan bisa membedakan intensitas antar elemen. Hasil perbandingan tersebut diisikan pada sel yang bersesuaian dengan elemen yang dibandingkan. Skala perbandingan perbandingan berpasangan dan maknanya yang diperkenalkan oleh Saaty bisa dilihat di bawah. Intensitas Kepentingan:

- 1) 1 berarti kedua elemen sama pentingnya, Dua elemen mempunyai pengaruh yang sama besar.
- 2) 3 berarti elemen yang satu sedikit lebih penting daripada elemen yang lainnya, Pengalaman dan penilaian sedikit menyokong satu elemen dibandingkan elemen yang lainnya

- 3) 5 berarti elemen yang satu lebih penting daripada yang lainnya, Pengalaman dan penilaian sangat kuat menyokong satu elemen dibandingkan elemen yang lainnya
- 4) 7 berarti satu elemen jelas lebih mutlak penting daripada elemen lainnya, Satu elemen yang kuat disokong dan dominan terlihat dalam praktek.
- 5) 9 berarti satu elemen mutlak penting daripada elemen lainnya, Bukti yang mendukung elemen yang satu terhadap elemen lain memiliki tingkat penegasan tertinggi yang mungkin menguatkan.
- 6) 2,4,6,8 berarti nilai-nilai antara dua nilai pertimbangan-pertimbangan yang berdekatan, Nilai ini diberikan bila ada dua kompromi di antara 2 pilihan  
Kebalikan = Jika untuk aktivitas  $i$  mendapat satu angka dibanding dengan aktivitas  $j$ , maka  $j$  mempunyai nilai kebalikannya dibanding dengan  $i$
5. Menghitung nilai eigen dan menguji konsistensinya. Jika tidak konsisten maka pengambilan data diulangi.
6. Mengulangi langkah 3,4, dan 5 untuk seluruh tingkat hirarki.
7. Menghitung vektor eigen dari setiap matriks perbandingan berpasangan yang merupakan bobot setiap elemen untuk penentuan prioritas elemen-elemen pada tingkat hirarki terendah sampai mencapai tujuan. Penghitungan dilakukan lewat cara menjumlahkan nilai setiap kolom dari matriks, membagi setiap nilai dari kolom dengan total kolom yang bersangkutan untuk memperoleh normalisasi matriks, dan menjumlahkan nilai-nilai dari setiap baris dan membaginya dengan jumlah elemen untuk mendapatkan rata-rata.

8. Memeriksa konsistensi hirarki. Adapun yang diukur dalam *Analytical Hierarchy Process* adalah rasio konsistensi dengan melihat *index* konsistensi. Konsistensi yang diharapkan adalah yang mendekati sempurna agar menghasilkan keputusan yang mendekati valid. Walaupun sulit untuk mencapai yang sempurna, rasio konsistensi diharapkan kurang dari atau sama dengan 10 %.

Rumus Untuk Menentukan Rasio Konsistensi (CR) Indeks konsistensi dari matriks berordo  $n$  dapat diperoleh dengan rumus :

Dimana :

CI = Indeks konsistensi (*Consistency Index*)

$\lambda$  maksimum = Nilai eigen terbesar dari matrik berordo  $n$

$\lambda$  maksimum didapat dengan menjumlahkan hasil perkalian jumlah kolom dengan eigen vektor utama.

Apabila C.I = 0, berarti matriks konsisten.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0,700	0,700	0,700	0,700	0,700	0,700	0,700	0,700	0,700	0,700	0,700	0,700	0,700	0,700	0,700

Batas ketidakkonsistenan yang ditetapkan Saaty diukur dengan menggunakan rasio konsistensi (CR), yakni perbandingan indeks konsistensi dengan nilai pembangkit random (RI). Nilai RI bergantung pada ordo matrik  $n$ . (Sylvia Hartati Saragih, 2015 : 84).

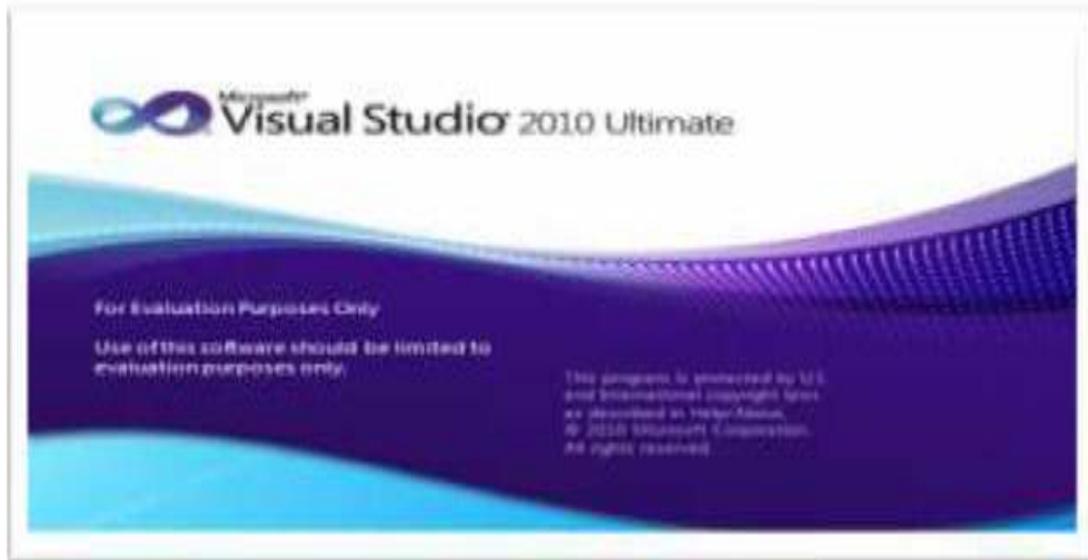
#### II.2.4. Bahasa Pemrograman

Bahasa pemrograman adalah perintah-perintah yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu, *visual basic* adalah salah satu *development tool*, yaitu alat untuk membantu berbagai macam program komputer yang khususnya menggunakan Sistem operasi *windows*. (Setiawan dan Permadani, 2016 : 54).

#### **II.2.5. *Visual Basic 2010***

*Visual Basic 2010* merupakan versi perbaikan dan pengembangan dari versi pendahulunya, yaitu *Visual Basic 2008*. Beberapa pengembangan yang terdapat didalamnya antara lain dukungan terhadap *library* terbaru dari *Microsoft*, yaitu *.Net Framework 4.0*, dukungan terhadap aplikasi berbasis *Cloud Computing*, serta perluasan dukungan terhadap *database-database*, baik *standalone* maupun *database server*. (Maria Sari, Linda Wahyuni, dan Lili Tanti, 2015 : 2).

Pada saat pertama kali dijalankan *Visual Basic 2010 Ultimate*, akan menampilkan sebuah jendela *Splash Visual Basic 2010 Ultimate* pada gambar II.1.



**Gambar II.1. Jendela *Splash Visual Basic 2010***

**(Sumber : Maria Sari, Linda Wahyuni, dan Lili Tanti, 2015 : 2)**

### **II.2.6. SQL Server 2008**

SQL Server 2008 adalah sebuah terobosan baru dari *Microsoft* dalam bidang database. SQL Server adalah DBMS (*Database Management System*) yang dibuat oleh *Microsoft* untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan *Oracle*. SQL Server 2008 dibuat pada saat kemajuan dalam bidang *hardware* sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa SQL Server 2008 membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data. *Microsoft* merilis SQL Server 2008 dalam beberapa versi yang disesuaikan dengan segment-segment pasar yang dituju. (Maria Sari, Linda Wahyuni, dan Lili Tanti, 2015 : 3).

SQL Server 2008 adalah sebuah terobosan baru dari *Microsoft* dalam bidang

*database*. *SQL Server* adalah DBMS (*Database Management System*) yang dibuat oleh Microsoft untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan Oracle. *SQL Server* 2008 dibuat pada saat kemajuan dalam bidang *hardware* sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa *SQL Server* 2008 membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data. Microsoft merilis *SQL Server* 2008 dalam beberapa versi yang disesuaikan dengan segment-segment pasar yang dituju. (Setiawan dan Permadani, 2016 : 54).

Versi-versi tersebut adalah sebagai berikut. Menurut cara pemrosesan data pada prosesor maka Microsoft mengelompokkan produk ini berdasarkan 2 jenis yaitu :

1. Versi 32-bit(x86), yang biasanya digunakan untuk komputer dengan single prosesor (Pentium 4) atau lebih tepatnya prosesor 32 bit dan sistem operasi Windows XP.
2. Versi 64-bit(x64), yang biasanya digunakan untuk komputer dengan lebih dari satu prosesor (Misalnya Core 2 Duo) dan system operasi 64 bit seperti Windows XP 64, Vista, dan Windows 7. Sedangkan secara keseluruhan terdapat versi-versi seperti berikut ini:
3. Versi Compact, ini adalah versi “Tipis” dari semua versi yang ada. Versi ini seperti versi desktop pada *SQL Server* 2000. Versi ini juga digunakan pada handheld device seperti Pocket PC, PDA, SmartPhone, Tablet PC.
4. Versi Express, ini adalah versi “Ringan” dari semua versi yang ada (tetapi versi ini

berbeda dengan versi compact) dan paling cocok untuk latihan. Express Manager standar, integrasi dengan CLR dan XML. (Setiawan dan Permadani, 2016 : 55).



**Gambar II.2. Tampilan Awal SQL Server 2008**

**(Sumber : Setiawan dan Permadani, 2016 : 55)**

### **II.2.7. Unified Modeling Language (UML)**

Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language (UML)*. UML adalah bahasa spesifik standar yang dipergunakan untuk mendokumentasikan, menspesifikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak digunakan dalam dunia industri yang merupakan standar bahasa pemodelan

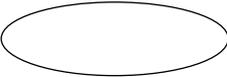
umum dalam industri perangkat lunak dan pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berdasarkan UML yaitu *Use Case Diagram*, *Activity Diagram*, *Class Diagram* dan *Sequence Diagram*. (Urva dan Siregar, 2012 : 95).

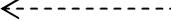
Alat bantu yang digunakan dalam perancangan berorientasi objek berdasarkan UML adalah sebagai berikut :

#### 1. *Use Case Diagram*

*Use case* diagram 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. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.1 dibawah ini :

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

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, dan dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang</p>

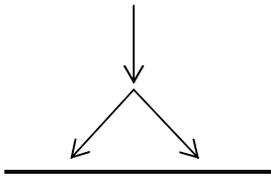
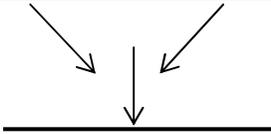
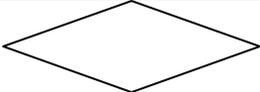
	<p>lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i>, tetapi tidak memiliki <i>control</i> terhadap <i>use case</i>.</p>
	<p>Asosiasi antara aktor dan <i>use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasi aliran data.</p>
	<p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidikasi bila aktor berinteraksi secara pasif dengan sistem.</p>
	<p><i>Include</i>, merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.</p>
	<p><i>Extend</i>, merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.</p>

(Sumber : Urva dan Siregar, 2015 : 94)

## 2. Diagram Aktivitas (*Activity Diagram*)

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.2 dibawah ini :

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

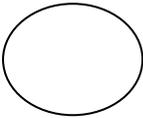
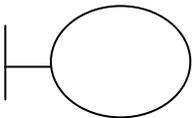
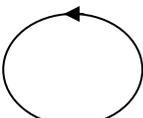
Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true, false</i> .
	<i>Swimlane</i> , untuk menunjukkan siapa melakukan apa.

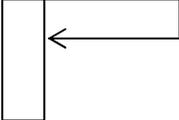
(Sumber : Urva dan Siregar, 2015 : 94)

### 3. Diagram Urutan (*Sequence Diagram*)

*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.3 dibawah ini :

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

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

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

(Sumber : Urva dan Siregar, 2015 : 95)

#### 4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. 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**

<b>Multiplicity</b>	<b>Penjelasan</b>
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

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