

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Dalam penyusunan skripsi penulis mengambil referensi dari penelitian terdahulu yang berkaitan dengan judul yang diangkat oleh peneliti.

Purnomo Andang Wulan dan Andikos Fitra Adi (2018) dalam jurnal dengan berjudul "*Analisa Metode Promethee Dalam Menentukan Dosen Teladan Berprestasi Di Perguruan Tinggi*" Dalam hasil penelitian tersebut mendapatkan kesimpulan bahwa dengan menggunakan *Software Visual Promethee* menunjukkan bahwa hasil sesuai dengan yang diharapkan sehingga tujuan dari sistem ini yaitu terbentuknya sebuah sistem pendukung keputusan yang dapat menentukan perankingan prioritas dosen berprestasi pada Universitas Dharmas Indonesia.

Rika Rosnelly dan Retantyo Wardoyo (2011) dalam jurnal berjudul "*Penerapan Fuzzy Multi Criteria Decision Making (FMCDM) Untuk Diagnosis Penyakit Tropis*" Hasil dalam penelitian tersebut mendapatkan kesimpulan bahwa dengan menggunakan *Fuzzy Multi Criteria Decision Making* dapat ditetapkan penyakit demam berdarah sebagai diagnosis penyakit tropis dari 3 alternatif penyakit tropis. Penyakit demam berdarah merupakan hasil diagnosis penyakit optimal yang diperoleh, baik dengan drajat keoptimisan 0;0,5 dan 1

Priyanto Senja Febryan, Et al (2017) dalam jurnal Informatika Polinema (ISSN: 2407-070X) dengan judul “*Sistem Pendukung Keputusan Pemilihan Guru Berprestasi Menggunakan Metode Promethee*” Hasil dalam penelitian memiliki akurasi 100% dalam proses pemilihan guru berprestasi dan sistem ini bekerja sesuai dengan tujuan yang diharapkan.

Jorry Karim (2018) dalam jurnal ILKOM jurnal Ilmiah Volume 10 No.1 (ISSN cetak: 2087-1716, ISSN online: 2548-7779) dengan judul “*Sistem Pendukung Keputusan Penentuan Prioritas Pembangunan Menggunakan Metode Promethee Pada Desa Ayula Kecamatan Randangan Kabupaten Pohuwato Provinsi Gorontalo*” Dalam hasil penelitian disimpulkan bahwa Metode Promethee untuk sistem pendukung keputusan dapat memberikan hasil yang maksimal dalam hal pengambilan keputusan penentuan pembangunan desa.

Taufik Ichsan, Et al (2017) dengan judul “*Implementasi Metode Promethee Untuk Menentukan Penerima Beasiswa*” Dalam hasil penelitian mendapatkan kesimpulan bahwa telah berhasil menerapkan metode promethee untuk memberikan solusi berupa rekomendasi pilihan kandidat yang tepat untuk menerima beasiswa.

II.2. Sistem Informasi

Sistem Informasi adalah suatu sistem dimana suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi

operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu (Tata Sutabri, 2012)

II.3. Sistem Pendukung Keputusan

II.3.1. Pengertian Sistem Pendukung Keputusan

Sistem Pendukung Keputusan atau sering disebut Decision Support System (DSS) adalah sistem berbasis model yang terdiri dari prosedur-prosedur dalam pemrosesan data dan pertimbangannya untuk membantu manajer dalam mengambil keputusan. Agar berhasil mencapai tujuannya maka sistem tersebut harus sederhana, mudah untuk dikontrol, mudah beradaptasi lengkap pada hal-hal penting dan mudah berkomunikasi dengan penggunanya. Secara implisit spk berarti bahwa sistem ini harus berbasis komputer dan digunakan tambahan dari kemampuan penyelesaian masalah dari seseorang. (Miswan Gumanti, et al 2015).

Salah satu penelitian mengenai sistem pendukung keputusan adalah “Implementasi Analytic Hierarchy Process Dan Simple Additive Wighted Dalam Pemilihan Karyawan Berprestasi“. Dalam penelitian tersebut dilakukan untuk mendapatkan karyawan yang berprestasi karena dengan melakukan metode tersebut akan menumbuhkan semangat kerja karyawan agar berkualitas dan berproduktif dengan bagus. Ketepatan penilaian prestasi juga mampu meminimalisir kecemburuan antar karyawan terhadap prestasi kerjanya. Sehingga dibutuhkan suatu sistem pendukung keputusan agar penentuan karyawan terbaik dapat dilakukan dengan mudah. (Abdul,M. 2018).

II.3.2. Konsep Pengambilan Keputusan

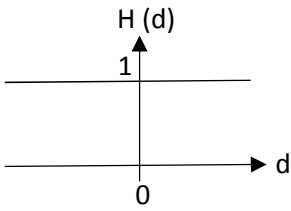
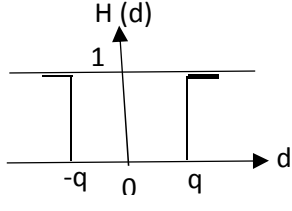
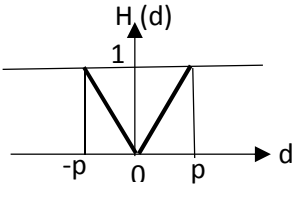
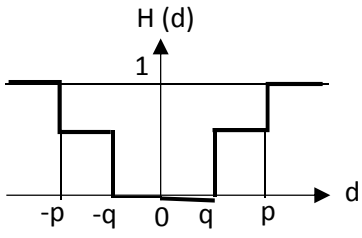
Konsep SPK pertama kali diperkenalkan pada awal tahun 1970-an oleh Michael Scott Morton (Turban, 2001: 13), yang selanjutnya dikenal dengan istilah “*Management Decision System*”. Konsep SPK merupakan sebuah sistem interaktif berbasis komputer yang membantu pembuatan keputusan memanfaatkan data dan model untuk menyelesaikan masalah-masalah yang bersifat tidak terstruktur dan semi terstruktur. SPK dirancang untuk menunjang seluruh tahapan pembuatan keputusan, yang dimulai dari tahapan mengidentifikasi masalah, memilih data yang relevan, menentukan pendekatan yang digunakan dalam proses pembuatan keputusan sampai pada kegiatan mengevaluasi pemilihan alternatif. (Murni, et al, 2018)

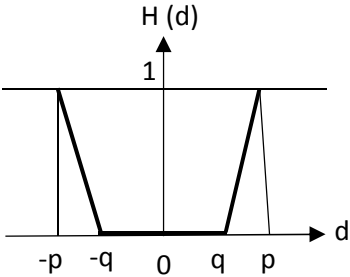
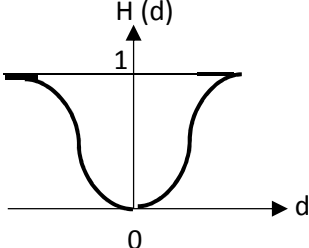
II.4. Metode Promethee

Promethee adalah suatu metode penentuan urutan (prioritas) dalam analisis multikriteria. Masalah pokoknya adalah kesederhanaan, kejelasan, dan kestabilan. Dugaan dari dominasi kriteria yang digunakan dalam *Promethee* adalah penggunaan nilai dalam hubungan *outranking*. Metode ini termasuk metode peringkat yang cukup sederhana dalam konsep dan aplikasi dibandingkan dengan metode lain untuk analisis multikriteria. Untuk setiap kriteria, fungsi preferensi menerjemahkan perbedaan antara dua alternatif menjadi derajat preferensi mulai dari nol sampai satu. Struktur preferensi *Promethee* berdasarkan perbandingan berpasangan. Semakin kecil nilai deviasi maka semakin kecil nilai preferensinya, semakin besar deviasi semakin besar preferensinya.

Dalam promethee disajikan enak bentuk fungsi preferensi kriteria. Hal ini tentu saja tidak mutlak, tetapi bentuk cukup baik untuk beberapa kasus, untuk memberikan gambaran yang lebih baik terhadap area yang tidak sama, digunakan gungsi selisih nilai kriteria antara alternative $H(d)$ dimana hal ini mempunyai hubungan langsung pada fungsi preferensi P .

Tabel II.1. Tipe dari Fungsi Preferensi Kriteria

Tipe Preferensi Kriteria	Parameter
1. Kriteria Umum <i>(Usual Criterion)</i>	
2. Kriteria Quasi <i>(Quasi Criterion)</i>	
3. Kriteria Preferensi Linier <i>(Criterion with Linear Preference)</i>	
4. Kriteria Level <i>(Level Criterion)</i>	

<p>5. Kriteria Dengan Preferensi Linier dan Area yang tidak Berbeda <i>(Criterion with Linear Preference and Independence Area)</i></p>		<p>q,p</p>
<p>6. Kriteria Gaussian <i>(Gaussian Criterion)</i></p>		<p>σ</p>

(Sumber: Safrizal, dan Lili Tanti, 2015, hal 871)

II.5. Visual Studio 2010

Visual Studio 2010 merupakan suatu perangkat lunak yang dapat digunakan untuk mengembangkan berbagai macam aplikasi yang memiliki berbagai macam tipe antara lain aplikasi desktop (*Windows Form, CommandLine (Console)*), Aplikasi Web, *Windows Mobile (Pocket PC)*. Visual Studio 2010 memiliki lebih dari satu kompiler, SDK (Software Development Kit), dan Dokumentasi Tutorial (MSDN Library). Kompiler yang dimasukkan kedalam Visual Studio 2010 antara lain *Visual Basic, Visual C#, Visual C++, Visual InterDev, Visual J++, Visual F#, dan Visual Source Safe*, dan banyak yang lainnya. (Rolly Yesputra 2017)

II.6. SQL Server

Microsoft *SQL Server* merupakan salah satu produk RDBMS (*Relation Database Management System*). Salah satu produk andalan yang dibuat oleh Microsoft yang berfungsi sebagai relation database. Microsoft *SQL Server* mendukung SQL sebagai bahasa pemroses *query*. SQL merupakan bahasa standart internasional untuk proses *query database* dan SQL ini sudah banyak digunakan pada hampir semua aplikasi, baik itu *e-commerce*, pendidikan, organisasi, pemerintah, atau bahkan personal sekalipun. (Agus Saputra, 2018)

Sebagai software, fungsi utama SQL Server adalah menampung dan menggunakan data yang terintegrasi dengan aplikasi, baik pada komputer yang sama ataupun dari komputer lain dalam jaringan. Dilengkapi dengan fitur *In-Memory OLTP* memungkinkan untuk memindahkan suatu tabel secara individual ke dalam struktur data *in-memory* khusus, *Managed Backup* ke dalam Azure *memback-up* secara otomatis berdasarkan interval *requery*, Azure VM untuk replika ketersediaan. (Jubilee, 2018)

II.7. Normalisasi

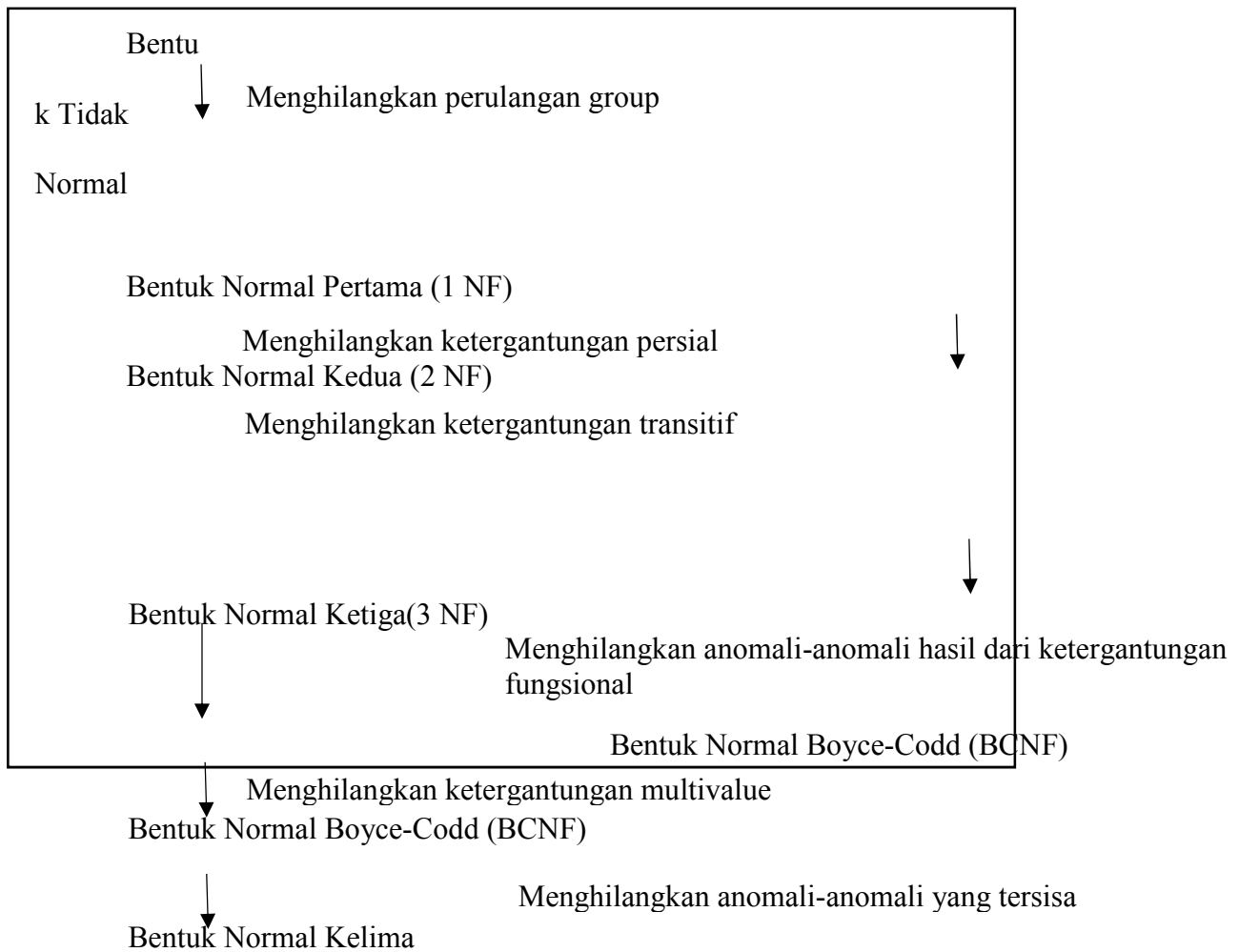
Normalisasi adalah proses pembentukan struktur basis data sehingga sebagian besar *ambiguity* bisa dihilangkan. Normalisasi merupakan sebuah teknik dalam *logical* desain sebuah basis data relasional yang mengelompokkan atribut dari suatutabel sehingga membentuk struktur tabel yang normal. Adapun kriteria tabel dikatakan normal adalah ketika tidak ada kerangkapan data (redudansi data).

Tujuan dari normalisasi adalah untuk :

1. Untuk menghilangkan kerangkapan data sehingga meminimumkan pemakaian *storage* yang dipakai oleh base relations(*file*).
2. Untuk mengurangi kompleksitas
3. Untuk mempermudah pemodifikasian data

Adapun proses normalisasi sebagai berikut:

1. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat kemudian.
2. Apabila tabel yang diuji belum memenuhi persyaratan tertentu, maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal. Untuk melakukan proses tersebut dibutuhkan beberapa tahapan. Tahapan dalam normalisasi dimulai dari tahap paling ringan (1NF) hingga paling ketat (5NF). Biasanya hanya sampai pada tingkat 3NF atau BCNF karena sudah cukup memadai untuk menghasilkan tabel-tabel yang berkualitas baik. Urutan tahapan normalisasi tampak seperti gambar II.1.



Gambar II.I. Tahapan Normalisasi
(Sumber : Dwi Puspitasari, et al ; 2016)

Adapun menurut Dwi Puspitasari, et al (2016) dalam jurnalnya aturan dalam normalisasi adalah suatu tabel dikatakan baik (efisien) atau normal jika memenuhi 3 kriteria sbb:

1. Jika ada dekomposisi (penguraian) tabel, maka dekomposisinya harus dijamin aman (*Lossless-Join Decomposition*). Artinya, setelah tabel tersebut diuraikan

/ didekomposisi menjadi tabel-tabel baru, tabel-tabel baru tersebut bisa menghasilkan tabel semula dengan sama persis.

2. Terpeliharanya ketergantungan fungsional pada saat perubahan data (*Dependency Preservation*).
3. Tidak melanggar *Boyce-Codd* Normal Form (BCNF) Jika kriteria ketiga (BCNF) tidak dapat terpenuhi, maka paling tidak tabel tersebut tidak melanggar Bentuk Normal tahap ketiga (3rd Normal Form / 3NF). Pada penelitian ini formula yang dibuat sampai memenuhi bentuk normal ke 3 yaitu 3NF.

mahasiswa.

II.8. UML (*Unified Modelling Language*)

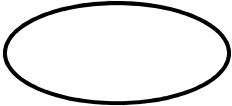
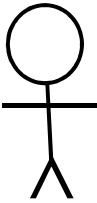


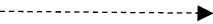
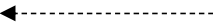
Unified Modeling Language (UML) merupakan kumpulan diagram-diagram yang sudah memiliki standart untuk pembangunan perangkat lunak berbasis objek. UML memiliki banyak diagram, empat diantaranya akan dibahas pada bab ini yaitu, *use case* diagram, *class* diagram, *sequence* diagram dan *state diagram/activity* diagram. (Feri Sulianta, 2017)

II.8.1. *Use Case Diagram*

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi siapa saja yang ada di dalam sistem informasi dan siapa saja yang berhak

menggunakan fungsi-fungsi tersebut. Berikut simbol-simbol yang digunakan dalam *Usecase diagram* pada tabel II.2.

Tabel II.2. Simbol-simbol *Use case Diagram*

Gambar	Keterangan
	<i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja.
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang 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 kontrol terhadap <i>use case</i> .
	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 mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila <i>actor</i> berinteraksi secara pasif dengan sistem.
	<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 sebuah fungsi program
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber: Ade Hendini, 2015, hal 108)

II.8.2. Class Diagram

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap 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 *Aggregations*, Atribut (*Attributes*), Operasi (*Operation/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.


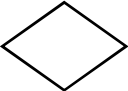
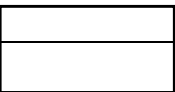

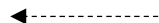
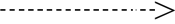

Tabel II.3. *Multiplicity Class Diagram*

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

(Sumber: Ade Hendini, 2015, hal 111)

Simbol-simbol yang digunakan dalam *class diagram* dapat dilihat pada tabel II.3.

Tabel II.4. Simbol-simbol *class Diagram*

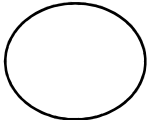
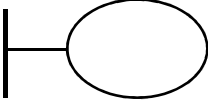
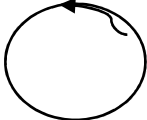

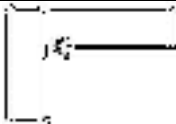


Gambar	Nama	Keterangan
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk.
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Kelas pada struktur sistem.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>Independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

(Sumber: Rossa.A.S. dan M.shalaluddin, 2015, hal 165)

II.8.3. Sequence Diagram

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima objek. Simbol-simbol yang digunakan dalam *sequence diagram* berikut gambarnya pada tabel II.5.

Tabel II.5. Simbol-simbol *Sequence Diagram*




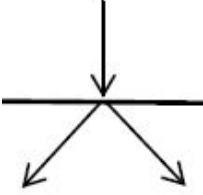
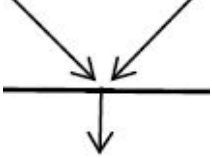
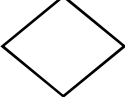

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi interface atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>formentry</i> dan form 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 objek.
	<i>Message</i> , Simbol pengiriman antar pesan.
	<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 aktiivtas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Ade Hendini, 2015, hal 110)

II.8.4. Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis. Berikut simbol-simbol yang digunakan dalam *activity diagram* pada tabel II.6.

Tabel II.6. Simbol-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>Activities</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan parallel 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> , Pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

(Sumber: Ade Hendini, 2015, hal 110)