

BAB II

LANDASAN TEORI

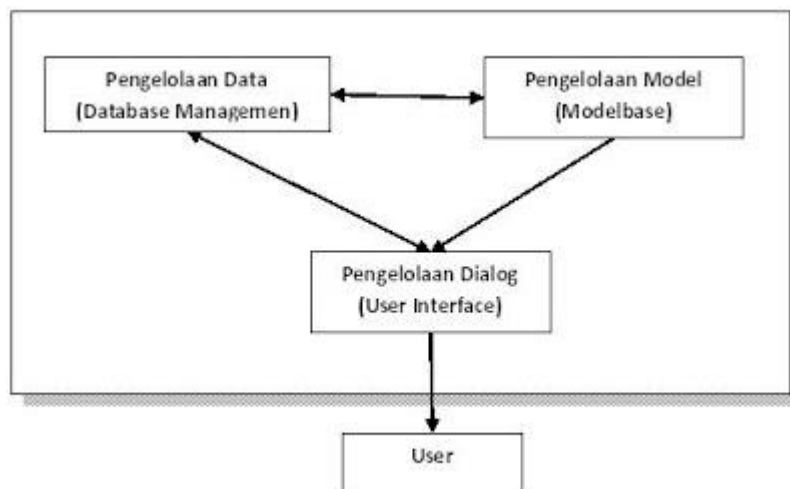
2.1 Sistem Pendukung Keputusan

2.1.1 Pengertian Sistem Pendukung Keputusan

Sistem pendukung keputusan merupakan sistem pemodelan dan pemanipulasian data yang digunakan untuk membantu mengambil keputusan dalam situasi yang semi terstruktur dan yang tidak terstruktur. Pemanfaatan sistem pendukung keputusan dalam penentuan penerima beasiswa termasuk dalam masalah semi terstruktur. Untuk membantu pengambil keputusan dalam memutuskan suatu masalah manajerial, maka dibutuhkan kualitas informasi yang Relevan (informasi tersebut terkait dengan keputusan yang akan diambil), akurat (kecocokan antara informasi dengan kejadiankejadian yang diwakili), lengkap (seberapa jauh informasi menyertakan kejadian-kejadian yang berhubungan), tepat waktu (Informasi sesuai waktu kejadiannya), dapat dipahami dan dapat dibandingkan antara dua obyek yang mirip (Hermiyanty, Wandira Ayu Bertin, 2017).

2.1.2 Komponen Sistem Pendukung Keputusan

Secara umum SPK terdiri oleh tiga komponen, (Turban and Aronso, 2001). Komponen Sistem Pendukung Keputusan tersebut dapat digambarkan seperti gambar di bawah ini.



Gambar 2.1 Komponen Sistem Pendukung Keputusan (SPK)

A. Pengelolaan Data (Database Management)

Merupakan subsistem data yang terorganisasi dalam suatu basis data. Untuk membangun sebuah Sistem Pendukung Keputusan, maka diperlukan data yang relevan dengan permasalahan akan dipecahkan.

B. Pengelolaan Model (Model Base)

Merupakan suatu model yang merepresentasikan permasalahan kedalam format kuantitatif sebagai dasar simulasi atau pengambilan keputusan. Pengelolaan Model memungkinkan pengambil keputusan menganalisa secara utuh dengan mengembangkan dan membandingkan solusi alternatif.

C. Pengelolaan Dialog (User Interface)

Merupakan gabungan antara kedua komponen sebelumnya yaitu Database Management dan Model Base yang disatukan dalam komponen ketiga (user interface), yang sebelumnya telah dipresentasikan dalam bentuk model yang dimengerti computer.

2.2 Logika *Fuzzy*

2.2.1 Definisi Logika *Fuzzy*

Logika fuzzy (fuzzy logic) adalah salah satu cabang dari AI (artificial intelligence). Logika fuzzy merupakan modifikasi dari teori himpunan dimana setiap anggotanya memiliki derajat keanggotaan yang bernilai kontinu antara 0 sampai 1. Sejak ditemukan pertama kali oleh Lotfi A. Zadeh pada tahun 1965, logika fuzzy telah digunakan pada lingkup domain permasalahan yang cukup luas, seperti kendali proses, klasifikasi dan pencocokan pola, manajemen dan pengambilan keputusan dan lain-lain (Anggraeni, Indarto and Kusumadewi, 2004).

2.2.2 Himpunan *Fuzzy*

Himpunan fuzzy adalah pengelompokan sesuatu berdasarkan variabel bahasa (linguistik variable), yang dinyatakan dengan fungsi keanggotaan, dalam semesta U. Keanggotaan suatu nilai pada himpunan dinyatakan dengan derajat keanggotaan yang nilainya antara 0.0 sampai 1.0. Himpunan fuzzy didasarkan

pada gagasan untuk memperluas jangkauan fungsi karakteristik sedemikian hingga fungsi tersebut akan mencakup bilangan real pada interval $[0,1]$. Nilai keanggotaannya menunjukkan bahwa suatu item tidak hanya bernilai benar atau salah. Nilai 0 menunjukkan salah, nilai 1 menunjukkan benar, dan masih ada nilai-nilai yang terletak antara benar dan salah (Saelan, 2009).

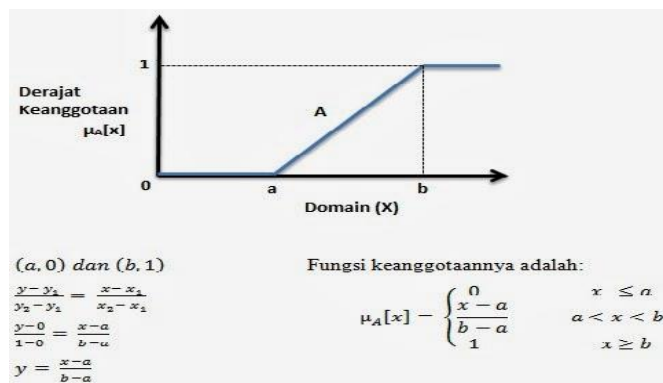
2.2.3 Fungsi Keanggotaan *Fuzzy*

Fungsi keanggotaan dari suatu himpunan fuzzy dinyatakan dengan derajat keanggotaan suatu nilai terhadap nilai tegasnya yang berkisar antara 0.0 sampai dengan 1.0. Fungsi keanggotaan adalah sebuah kurva yang menunjukkan titik input kedalam nilai keanggotaannya (Gultom, 2015). Ada beberapa fungsi yang digunakan yaitu:

1. Representasi Linier

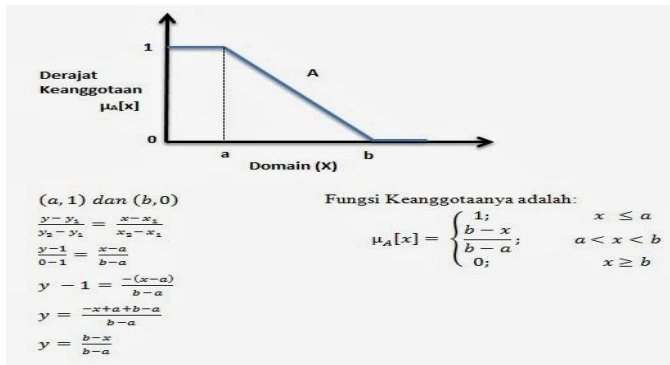
Pada representasi linear, pemetaan input ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Representasi linier terdiri dari:

a). Representasi linier naik, yaitu kenaikan himpunan dimulai dari nilai domain yang memiliki nilai keanggotaan nol $[0]$ bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan yang lebih tinggi.



Gambar 2.2 Representasi Linier Naik

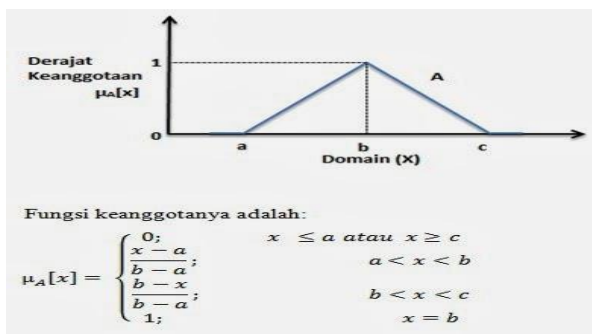
b). Representasi linier turun, yaitu garis lurus yang dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak turun ke nilai domain yang memiliki derajat keanggotaan lebih rendah.



Gambar 2.3 Representasi Linier Turun

2. Representasi Kurva Segitiga

Representasi kurva segitiga adalah gabungan antara representasi linear naik dan representasi linear turun.



Gambar 2.4 Representasi Kurva Segitiga

2.3 Metode SAW (Simple Additive Weighting)

Metode *Simple Additive Weighting* (SAW) merupakan metode penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut. Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua alternatif yang ada (Ruskan, Ibrahim and Hartini, 2013).

$$R_{ij} = \begin{cases} \frac{X_{ij}}{\text{Max } X_{ij}} & \text{Jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\text{Min } X_{ij}}{X_{ij}} & \text{Jika } j \text{ adalah atribut biaya (cost)} \end{cases} \dots\dots (1)$$

Dimana :

R_{ij} = rating kinerja ternormalisasi

Max_{ij} = nilai maksimum dari setiap baris dan kolom

Min_{ij} = nilai minimum dari setiap baris dan kolom

X_{ij} = baris dan kolom dari matriks

Dengan R_{ij} adalah rating kinerja ternormalisasi dari alternatif A_i pada atribut C_j ; $i = 1, 2, \dots, m$ dan $j = 1, 2, \dots, n$.

Nilai preferensi untuk setiap alternatif (V_i) diberikan sebagai :

$$V_i = \sum_{j=1}^n W_j R_{ij} \quad \dots\dots\dots (2)$$

Dimana :

V_i = Nilai akhir dari alternatif

w_j = Bobot yang telah ditentukan

r_{ij} = Normalisasi matriks

Berikut adalah tahapan metode SAW:

1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pendukung keputusan yaitu C_i .
2. Menentukan rating kecocokan dari setiap alternatif pada setiap kriteria.
3. Membuat matriks keputusan berdasarkan kriteria (C_i).
4. Kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut *benefit* maupun atribut *cost*) sehingga diperoleh matriks yang ternormalisasi R .
5. Hasil akhir diperoleh dari proses perankingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan vector bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatis terbaik (A_i) sebagai solusi.

2.4 Lomba Kompetensi Siswa (LKS)

2.4.1 Pengertian Lomba Kompetensi Siswa (LKS)

LKS adalah suatu kegiatan lomba yang dilaksanakan melalui kompetisi siswa/siswi SMK yang dititikberatkan pada bidang keterampilan praktik, didukung oleh pemahaman teori yang relevan serta sikap kerja dalam menyelesaikan pekerjaan sesuai standar industri. Pengetahuan, keterampilan dan sikap kerja yang dimaksud adalah manifestasi kemampuan yang dilandasi oleh daya pikir, daya qolbu, dan daya fisik, untuk siap menjadi tenaga kerja tingkat menengah yang handal dalam bidang masing-masing sesuai tuntutan kebutuhan dunia kerja. LKS merupakan ajang promosi bursa tenaga kerja SMK yang potensial, sebagai salah satu upaya agar keberadaan SMK dan tamatannya lebih dikenal secara luas oleh masyarakat (Sutopo, 2007).

2.5 Android

2.5.1 Pengertian Android

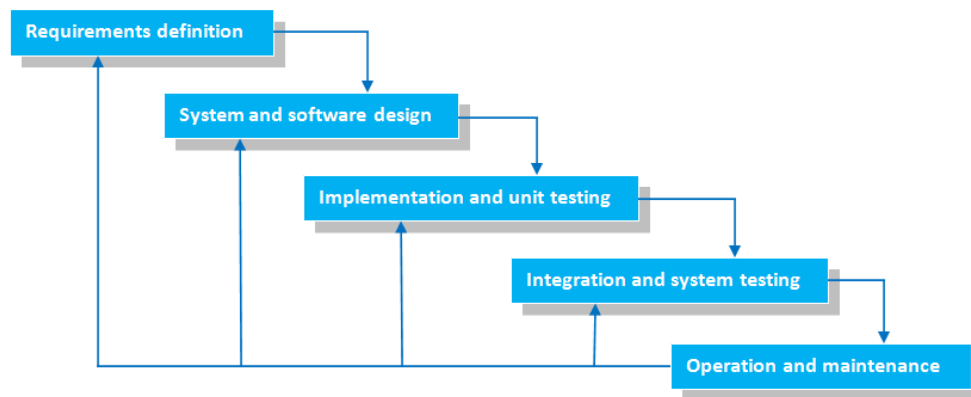
Android adalah sistem operasi dan platform pemrograman yang dikembangkan oleh Google untuk ponsel cerdas dan perangkat seluler lainnya (seperti tablet). Android bisa berjalan di beberapa macam perangkat dari banyak produsen yang berbeda. Android menyertakan kit development perangkat lunak untuk penulisan kode asli dan perakitan modul perangkat lunak untuk membuat aplikasi bagi pengguna Android (Developer, 2016)

Nama	Versi	Peluncuran
Cupcake	1.5	27 April 2009
Donut	1.6	15 September 2009
Eclair	2.0 – 2.1	26 Oktober 2009
Froyo	2.2 – 2.2.3	20 Mei 2010
Gingerbread	2.3 – 2.3.7	6 Desember 2010
Honeycomb	3.0–3.2.6	22 Pebruari 2011
Ice Cream Sandwich	4.0 – 4.0.4	18 Oktober 2011
Jelly Bean	4.1 – 4.3.1	9 Juli 2012
KitKat	4.4 – 4.4.4	31 Oktobe 2013
Lollipop	5.0 – 5.1.1	12 November 2014
Marshmallow	6.0 – 6.0.1	5 Oktober 2015
Nougat	7.0	Agustus / September 2016
Oreo	8.0	Agustus 2017
Pie	9.0	Agustus 2018

Gambar 2.5 Versi Android

2.6 Metode Waterfall

Model waterfall adalah proses pengembangan perangkat lunak tradisional yang umum digunakan dalam proyek-proyek perangkat lunak yang paling pembangunan. Ini adalah model sekuensial, sehingga penyelesaian satu set kegiatan menyebabkan dimulainya aktivitas berikutnya. Hal ini disebut waterfall karena proses mengalir "secara sistematis dari satu tahap ke tahap lainnya dalam mode ke bawah (Fahrurrozi and SN, 2015).



Gambar 2.6 Metode Waterfall

Berikut adalah deskripsi dari tahap metode waterfall (Safitri and Supriyadi, 2015):

A. Analisis Persyaratan Seluruh kebutuhan software harus bisa didapatkan dalam fase ini, termasuk didalamnya kegunaan software yang diharapkan pengguna dan batasan software. Informasi ini biasanya dapat diperoleh melalui wawancara, survey atau diskusi. Informasi tersebut dianalisis untuk mendapatkan dokumentasi kebutuhan pengguna untuk digunakan pada tahap selanjutnya.

B. Desain Sistem Tahap ini dilakukan sebelum melakukan coding. Tahap ini bertujuan untuk memberikan gambaran apa yang seharusnya dikerjakan dan bagaimana tampilannya. Tahap ini membantu dalam menspesifikasikan kebutuhan hardware dan sistem serta mendefinisikan arsitektur sistem secara keseluruhan.

C. Implementasi Dalam tahap ini dilakukan pemrograman. Pembuatan software dipecah menjadi modul-modul kecil yang nantinya akan digabungkan dalam tahap berikutnya. Selain itu dalam tahap ini juga dilakukan pemeriksaan terhadap modul yang dibuat, apakah sudah memenuhi fungsi yang diinginkan atau belum.

D. Integrasi dan Testing Di tahap ini dilakukan penggabungan modul-modul yang sudah dibuat dan dilakukan pengujian ini dilakukan untuk mengetahui apakah software yang dibuat telah sesuai dengan desainnya dan masih terdapat kesalahan atau tidak.

E. Operasi dan Maintenance Ini merupakan tahap terakhir dalam model waterfall. Software yang sudah jadi dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi unit sistem dan peningkatan jasa system sebagai kebutuhan baru.

2.7 UML (*Unified Modeling Language*)

Unified Modelling Language (UML) adalah sebuah bahasa yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan class dan operation dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasabahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C (Fahrurrozi and SN, 2015).

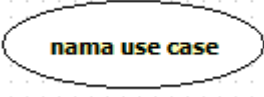



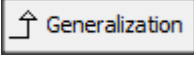
UML mendefinisikan diagram-diagram sebagai berikut:

1) Use Case Diagram

Use Case Diagram merupakan pemodelan untuk melakukan (*behavior*) sistem informai yang akan dibuat. Use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Berikut adalah simbol-simbol diagram use case pada Tabel 2.1:


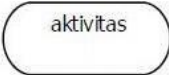
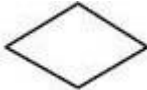


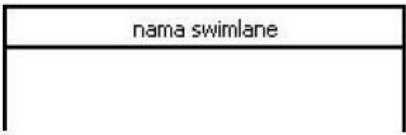
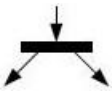
Tabel 2.1 Simbol *Use Case*


Simbol	Deksripsi
<p data-bbox="528 374 652 405">Use Case</p> 	<p data-bbox="836 374 1406 562">Fungsionalitas yang disediakan <i>system</i> sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>.</p>
<p data-bbox="555 598 625 629">Actor</p> 	<p data-bbox="836 598 1406 902">Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan di buat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang. biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p data-bbox="445 938 735 969">Asosiasi (Association)</p> 	<p data-bbox="836 938 1406 1046">Komunikasi anatara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
<p data-bbox="475 1146 705 1178">Ekstensi (Extend)</p> 	<p data-bbox="836 1146 1406 1451">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 di tambahkannya.</p>
<p data-bbox="395 1534 785 1565">Generalisaasi (Generalization)</p> 	<p data-bbox="836 1534 1406 1682">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>

2) Activity Diagram

Activity Diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Perlu diperhatikan bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut adalah simbol-simbol diagram aktivitas pada Tabel 2.2 sebagai berikut:

Tabel 2.2 Simbol *Activity Diagram*





Simbol	Deksripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan / <i>decision</i> 	Asosiasi percabangan pilihan jika ada aktivitas lebih dari satu.
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Sebuah <i>activity diagram</i> memiliki sebuah status akhir.
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
Fork 	Fork sebagai penunjuk kegiatan yang dilakukan secara paralel


<p>Join</p> 	Untuk menunjukkan kegiatan yang digabungkan.
---	--

3) Class Diagram

Class Diagram menggambarkan struktur sebuah sistem dari segi pendefinisian class yang akan dibuat untuk membangun sistem. Berikut adalah simbol-simbol diagram Kelas (Rosa dan Shalahuddin, 2019 : 141), pada Tabel 2.3 sebagai berikut:

Table 2.3 Simbol *Class Diagram*



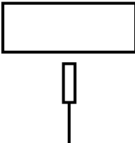


Simbol	Deksripsi
<p>Kelas</p> 	Kelas pada sturktur sistem
<p>Antarmuka / <i>Interface</i></p>  <p>Nama_interface</p>	Sama dengan konsep interface dalam pemograman berorientasi objek
<p>Asosiasi / <i>Association</i></p> 	Relasi antar kelas, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
<p>Asosiasi berarah / <i>Directed Association</i></p> 	Relasi antar kelas dengan makna kelas satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
<p>Generalisasi</p> 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
<p>Kebergantungan / <i>Dependency</i></p>	Relasi antar kelas dengan makna kebergantungan antarkelas

Agregasi / Aggregation 	Relas antarkelas dengan makna semuabagian (whole-part)
---	--



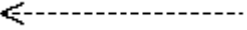

4) Sequence Diagram

Untuk menggambarkan diagram sekuen maka harus diketahui objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Berikut adalah simbol-simbol diagram sekuen (Rosa dan Shalahuddin, 2019 : 165), pada Tabel 2.4 sebagai berikut:

Table 2.4 Simbol *Sequence Diagram*

Simbol	Deskripsi
Actor 	Aktor merepresentasikan entitas yang berada di luar sistem dan berinteraksi dengan sistem. Mereka bisa berupa manusia, perangkat keras ataupun sistem yang lain.
Lifeline 	Fungsi dari simbol ini adalah Mengeksekusi objek selama <i>sequence</i> (<i>message</i> dikirim atau diterima dan aktifasinya).
General 	Merepresentasikan entitas tunggal dalam sequence diagram, Entitas ini memiliki nama, <i>stereotype</i> atau berupa <i>instance</i> (<i>class</i>).
Boundary 	Boundary biasanya berupa tepi dari sistem, seperti <i>user interface</i> atau suatu alat yang berinteraksi dengan sistem yang lain.
Control 	<i>Control element</i> mengatur aliran dari informasi untuk sebuah skenario. Objek ini umumnya mengatur perilaku dan perilaku bisnis.

Tabel 2.4 Lanjutan

<p>Entity</p> 	<p>Entitas biasanya elemen yang bertanggung jawab menyimpan data atau informasi. Ini dapat berupa <i>beans</i> atau <i>model object</i>.</p>
<p>Activation</p> 	<p>Suatu titik dimana suatu objek mulai berpartisipasi dalam sebuah sequence yang menunjukkan kapan objek itu mengirim atau menerima objek.</p>
<p>Message</p>	<p>Message berfungsi sebagai komunikasi antar objek yang menggambarkan aksi yang akan dilakukan. Message terjadi antara dua buah objek dimana satu objek (<i>client</i>) dan meminta objek (<i>supplier</i>) untuk melakukan sesuatu.</p>
<p>Message Entry</p> 	<p>Message berfungsi sebagai komunikasi antar objek yang menggambarkan aksi yang akan dilakukan . Message terjadi antara dua buah objek dimana satu objek (<i>client</i>) dan meminta objek (<i>supplier</i>) untuk melakukan sesuatu.</p>
<p>Message to Self</p> 	<p>Simbol ini menggambarkan pesan atau hubungan objek itu sendiri, menunjukkan urutan kejadian yang terjadi.</p>
<p>Message Return</p> 	<p>Simbol ini menggambarkan hasil dari pengiriman message dan digambarkan dengan arah dari kanan ke kiri.</p>

2.8 Penelitian Terdahulu

Penelitian ini dilakukan berdasarkan dari hasil penelitian-penelitian sebelumnya yang pernah dilakukan sebagai bahan perbandingan. Berikut merupakan penelitian sebelumnya yang terkait dengan sistem pendukung keputusan pemilihan peserta lomba LKS menggunakan metode *fuzzy SAW*.

Tabel 2.5 Penelitian Terdahulu

No	Judul	Penulis	Tahun	Uraian
1	Sistem Pendukung Keputusan (SPK) Pemilihan Peserta Lomba Sains Dengan Metode <i>Simple Additive Weighting</i> (SAW) Di SMA N Kebakkramat Karanganyar	Alvin Agus Budi Prasetyo	2018	SMA N Kebakkramat melakukan seleksi dengan menjaring siswa terbaik untuk mengikuti lomba Sains. Sistem Pendukung Keputusan (SPK) berbasis web ini menjadi alternatif dari perhitungan manual dan mempercepat proses pengambilan keputusan dalam pemilihan peserta lomba Sains. Metode yang digunakan yaitu metode perhitungan <i>Simple Additive Weighting</i> (SAW).
2	Rekayasa Aplikasi Pemilihan Anggota Peserta Lomba Cerdas Cermat	Muhammad Fadlan, Deby Kurniawan	2014	Terkadang muncul masalah pada saat proses penentuan peserta yang akan diikuti sertakan untuk lomba, siapa saja yang pantas untuk masuk

	Menggunakan Metode <i>Simple Additive Weighting</i> (Saw)			ke dalam tim cerdas cermat dari sekian banyak siswa. Penelitian ini akan membuat sebuah aplikasi sistem pendukung keputusan pemilihan peserta lomba cerdas cermat menggunakan metode <i>simple additive weighting</i> .
3	Penentuan Peserta Lomba Kompetensi Siswa Menggunakan <i>Simple Additive Weighting</i> (Saw)	Maura Widyaningsih, Leo Giovanni	2016	Sulitnya menyeleksi siswa dan siswi terbaik untuk dipilih dan di ikut sertakan lomba. Dengan melibatkan sebuah metode, suatu sistem akan menghasilkan sebuah keputusan yang sesuai untuk penyeleksian data.
4	Sistem Pendukung Keputusan Pemilihan Calon Peserta Lomba Siswa Berprestasi	Veny Cahya Hardita I Wayan Pandu Swardiana, Kusri	2019	Banyaknya siswa dari berbagai jurusan yang mengikuti seleksi lomba, sehingga dalam proses seleksi pelaksanaan menjadi kurang optimal dalam segi proses penilaian dan penentuan keputusan akhir. Sehingga diputuskan menggunakan SPK.

