



BAB II

TINJAUAN PUSTAKA

2.1 Teori Judul

2.1.1 Sistem Pendukung Keputusan

2.1.1.1 Pengertian Sistem Pendukung Keputusan

Menurut Latif, dkk. (2018:3), “Sistem Pendukung Keputusan adalah suatu sistem informasi spesifik yang ditujukan untuk membantu manajemen dalam mengambil keputusan yang berkaitan dengan persoalan yang bersifat semi terstruktur.”

Sedangkan menurut Bonczek yang dikutip dari buku *Sistem Pendukung Keputusan Teori dan Implementasi* (Latif, dkk., 2018:2) mengemukakan bahwa sistem pendukung keputusan merupakan sebuah sistem berbasis komputer yang terdiri atas tiga komponen yang saling berinteraksi. Komponen-komponen tersebut yaitu sistem bahasa (mekanisme untuk memberikan komunikasi antar pengguna dan komponen sistem pendukung keputusan lain), sistem pengetahuan (repositori pengetahuan domain masalah yang ada baik sebagai data atau sebagai prosedur), sistem pemrosesan masalah (hubungan antara komponen lainnya terdiri dari satu atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan).

Dari kedua pendapat di atas dapat disimpulkan bahwa sistem pendukung keputusan adalah suatu sistem informasi berbasis komputer yang ditujukan untuk membantu seseorang dalam proses pengambilan keputusan dari berbagai alternatif yang ada untuk memecahkan suatu persoalan yang bersifat tidak terstruktur.

2.1.1.2 Komponen Sistem Pendukung Keputusan

Menurut Latif, dkk. (2018:4) sistem pendukung keputusan terdiri atas empat komponen sebagai berikut.



1. *Data Management*

Termasuk *database*, yang mengandung data yang relevan untuk berbagai situasi dan diatur oleh *software* yang disebut *Database Management Systems* (DBMS).

2. *Model Management*

Melibatkan model finansial, statistik, *management science*, atau berbagai model kuantitatif lainnya, sehingga dapat memberikan ke sistem suatu kemampuan analitis, dan manajemen *software* yang diperlukan.

3. *Communication (dialog subsystem)*

User dapat berkomunikasi dan memberikan perintah pada *Decision Support System* (DSS) melalui subsistem ini. Subsistem ini berarti menyediakan antarmuka.

4. *Knowledge Management*

Subsistem *optional* ini dapat mendukung subsistem-subsistem lain atau bertindak sebagai komponen yang berdiri sendiri.

2.1.1.3 Tahapan Proses Pengambilan Keputusan

Menurut Simon dalam buku *Sistem Pendukung Keputusan Teori dan Implementasi* (Latif, dkk., 2018:5), mengemukakan bahwa terdapat tiga tahapan dalam proses pengambilan keputusan yaitu sebagai berikut.

1. *Intelligence*

Intelligence merupakan tahapan yang pertama dalam proses pengambilan keputusan yang mana pada tahapan ini dilakukan proses penelusuran dan pendeteksian dari ruang lingkup problematika secara proses pengenalan masalah. Data masukan diperoleh, kemudian diproses, dan diuji dalam rangka mengidentifikasi masalah.

2. *Design*

Pada tahapan ini dilakukan proses menemukan, mengembangkan, dan menganalisis alternatif tindakan yang bisa dilakukan. Tahap ini meliputi menguji kelayakan dan solusi.



3. Choice

Pada tahapan yang terakhir yaitu dilakukan proses pemilihan diantara berbagai alternatif tindakan yang mungkin dijalankan. Hasil penelitian tersebut kemudian diimplementasikan dalam proses pengambilan keputusan.

2.1.2 Bantuan Pangan Non Tunai (BPNT)

Bantuan Pangan Non Tunai (BPNT) adalah bantuan sosial pangan yang disalurkan dalam bentuk non tunai dari pemerintah kepada Keluarga Penerima Manfaat (KPM) setiap bulannya melalui uang elektronik yang digunakan hanya untuk membeli bahan pangan di pedagang bahan pangan atau disebut *E-warong* yang bekerjasama dengan Bank Penyalur. *E-warong* merupakan agen bank, pedagang dan/atau pihak lain yang telah bekerja sama dengan Bank Penyalur dan ditentukan sebagai tempat penarikan/pembelian Bantuan Sosial oleh KPM, yaitu usaha mikro, kecil, dan koperasi, pasar tradisional, warung, toko kelontong, *E-warung* KUBE, Warung Desa, Rumah Pangan Kita (RPK), Agen Laku Pandai, Agen Layanan Keuangan Digital (LKD) yang menjual bahan pangan, atau usaha eceran lainnya.

Program BPNT ini memiliki beberapa tujuan antara lain: 1) mengurangi beban pengeluaran KPM melalui pemenuhan sebagian kebutuhan pangan, 2) memberikan gizi yang lebih seimbang kepada KPM, 3) meningkatkan ketepatan sasaran dan waktu penerima bantuan pangan bagi KPM, 4) memberikan pilihan dan kendali kepada KPM dalam memenuhi kebutuhan pangan, 5) mendorong pencapaian tujuan pembangunan berkelanjutan (*Sustainable Development Goals* /SDGs).

2.1.3 Metode *Simple Multi Attribute Rating Technique* (SMART)

Menurut Latif (2018:13) mengatakan bahwa SMART merupakan sebuah metode pengambilan keputusan yang bersifat multi atribut yang dikembangkan oleh Edward pada tahun 1977. Teknik pembuatan keputusan multi atribut ini digunakan untuk mendukung pembuat keputusan dalam memilih diantara



berbagai alternatif yang ada. Setiap pembuat keputusan harus memilih sebuah alternatif yang sesuai dengan tujuan yang telah dirumuskan. Setiap alternatif terdiri dari sekumpulan atribut dan setiap atribut tersebut memiliki nilai tersendiri. Selanjutnya nilai ini dirata-rata dengan skala tertentu. Kemudian setiap atribut mempunyai bobot yang menggambarkan seberapa penting dibandingkan dengan atribut lain. Pembobotan dan pemberian peringkat ini digunakan untuk memberikan penilaian pada setiap alternatif agar diperoleh alternatif yang terbaik.

SMART menggunakan *linear additive* model untuk meramal nilai setiap alternatif. SMART merupakan metode pengambilan keputusan yang fleksibel. SMART lebih banyak digunakan karena kesederhanaannya dalam merespon kebutuhan pembuat keputusan dan caranya menganalisa respon.

Model yang digunakan dalam SMART adalah seperti berikut.

$$u(a_i) = \sum_{j=1}^k w_j \cdot u_j(a_i)$$

Keterangan:

1. $u(a_i)$ adalah nilai total alternatif
2. K adalah jumlah kriteria
3. W_j adalah hasil dari normalisasi bobot kriteria
4. $u_j(a_i)$ adalah hasil dari penentuan nilai *utility*

Adapun algoritma penyelesaian dari Metode SMART yaitu sebagai berikut:

- a. Menentukan jumlah kriteria dari keputusan yang akan diambil.
- b. Sistem secara *default* memberikan nilai 0-100 berdasarkan prioritas dengan melakukan normalisasi.

$$nw_j = \frac{w_j}{\sum w_j}$$

Keterangan:

- 1) nw_j adalah normalisasi bobot kriteria ke- j



- 2) w_j adalah nilai bobot kriteria ke- j
- c. Menghitung nilai *utility* untuk setiap kriteria masing-masing.

$$u_i(a_i) = 100 \frac{(C_{\max} - C_{\text{out } i})}{C_{\max} - C_{\min}}$$

Keterangan:

- 1) $u_i(a_i)$ adalah nilai *utility* kriteria ke-1 untuk kriteria ke-i
- 2) C_{\max} adalah nilai kriteria maksimal
- 3) C_{\min} adalah nilai kriteria minimal
- 4) $C_{\text{out } i}$ adalah nilai kriteria ke- i

- d. Menghitung nilai akhir dan melakukan perbandingan.

2.2. Teori Program

2.2.1 Basis Data (*Database*)

Menurut Winarno (2014:102) dalam buku *Pemrograman Web Berbasis HTML5, PHP, dan JavaScript* mengatakan bahwa basis data merupakan sebuah tempat penyimpanan data yang jenisnya beraneka ragam.

Sedangkan menurut Pamungkas (2017:2) mengemukakan bahwa basis data merupakan suatu kumpulan data yang saling berhubungan disimpan secara bersama-sama pada suatu media, yang mana diorganisasikan berdasarkan sebuah skema atau struktur tertentu, dan menggunakan *software* untuk melakukan proses manipulasi untuk suatu kegunaan tertentu.

Selain itu menurut Jubilee (2017:1) dalam buku *Otodidak MySQL untuk Pemula* mengatakan bahwa basis data adalah sebuah aplikasi yang menyimpan sekumpulan data. Dan setiap basis data tersebut memiliki perintah tertentu untuk dapat menciptakan, mengakses, mengatur, mencari, serta menyalin data yang ada di dalamnya.



Dari ketiga definisi di atas dapat disimpulkan bahwa basis data adalah sekumpulan data yang saling berhubungan disimpan dalam sebuah media tertentu guna memperoleh informasi dari basis data tersebut.

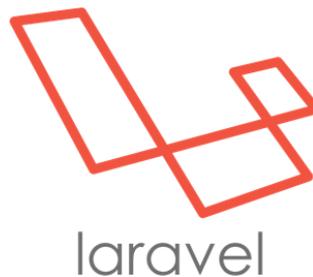
2.2.2 MySQL

Menurut Mundzir (2018:217) dalam buku *Pemrograman Web Seri PHP* mengatakan bahwa MySQL merupakan sistem manajemen *database* SQL yang bersifat *open source* (terbuka) dan paling banyak digunakan saat ini. Sistem *database* MySQL mampu mendukung beberapa fitur seperti *multithreaded*, *multi-user*, dan *SQL database management system (DBMS)*.

Sedangkan menurut Winarno (2014:102) mengatakan bahwa MySQL merupakan tipe data relasional yang berarti MySQL dapat menyimpan datanya dalam bentuk tabel-tabel yang saling berelasi.

Dari kedua pengertian di atas dapat disimpulkan bahwa MySQL merupakan sebuah aplikasi yang bersifat *open source* (terbuka) digunakan untuk manajemen data dalam bentuk tabel-tabel yang saling berhubungan.

2.2.3 Framework Laravel



Gambar 2.1 Logo Laravel

(Sumber: www.surround-bg.com)

Menurut Naista (2017) mengemukakan bahwa *framework* adalah suatu struktur konseptual dasar digunakan untuk memecahkan atau menangani suatu masalah yang bersifat kompleks. Singkatnya, *framework* merupakan suatu kerangka kerja dari sebuah *website* yang akan dibangun. Dengan menggunakan



kerangka tersebut, waktu yang diperlukan dalam membangun sebuah *website* menjadi lebih singkat dan memudahkan dalam proses perbaikan.

Sedangkan definisi Laravel menurut Naista (2017) mengatakan bahwa Laravel merupakan salah satu *framework* berbasis *PHP* bersifat *open source* (terbuka), dan menggunakan konsep MVC (*model – view – controller*). Laravel berada di bawah lisensi MIT *License* dengan menggunakan Github sebagai tempat berbagi *code* menjalankannya.

Menurut Abdulloh (2017:3) mengatakan bahwa terdapat beberapa keunggulan yang dimiliki Laravel yaitu sebagai berikut.

1. Laravel memiliki banyak fitur yang tidak dimiliki oleh *framework* lain.
2. Laravel merupakan *framework* *PHP* yang ekspresif, artinya sintaks pada Laravel menggunakan bahasa yang mudah dimengerti sehingga *programmer* pemula sekalipun akan mudah memahami kegunaan dari suatu sintaks meskipun *programmer* tersebut belum mempelajarinya.
3. Laravel memiliki dokumentasi yang cukup lengkap, bahkan setiap versinya memiliki dokumentasi tersendiri mulai dari cara instalasi hingga penggunaan fitur-fiturnya.
4. Laravel digunakan oleh banyak *programmer* sehingga banyak *library* yang mendukung Laravel yang diciptakan para *programmer* pecinta Laravel.
5. Laravel didukung oleh Composer sehingga *library-library* diperoleh dengan mudah dari internet menggunakan Composer.
6. Laravel memiliki *template engine* tersendiri yang diberi nama *blade* yang memudahkan dalam menampilkan data pada *template* HTML.

Adapun fitur-fitur yang dimiliki Laravel antara lain sebagai berikut (Aminudin, 2015:5).

- a) **Bundles** yaitu sebuah fitur dengan sistem pengemasan modular dan berbagai *bundle* telah tersedia untuk digunakan dalam aplikasi.



- b) **Eloquent ORM** merupakan penerapan PHP lanjutan dari pola *active record* menyediakan metode internal untuk mengatasi kendala hubungan antara objek *database*.
- c) **Application Logic** merupakan bagian dari aplikasi yang dikembangkan, baik menggunakan *controllers* maupun sebagai bagian dari deklarasi *route*. Sintaks yang digunakan untuk mendefinisikannya mirip dengan yang digunakan oleh *framework Sinatra*.
- d) **Reverse Routing**, mendefinisikan hubungan antara *link* dan *route*. Sehingga jika suatu saat ada perubahan pada *route* secara otomatis akan tersambung dengan *link* yang relevan. Ketika *link* yang dibuat dengan menggunakan nama-nama dari *route* yang ada, secara otomatis Laravel akan membuat URI yang sesuai.
- e) **Restful Controller**, memberikan sebuah pilihan untuk memisahkan logika dalam melayani HTTP GET dan permintaan POST.
- f) **Class Auto Loading**, menyediakan otomatis *loading* untuk kelas-kelas PHP, tanpa membutuhkan pemeriksaan manual terhadap jalur masuknya. Fitur ini mencegah *loading* yang tidak perlu.
- g) **View Composers** adalah kode unit *logical* yang dapat dijalankan ketika sebuah *view* di *load*.
- h) **IoC Container** memungkinkan untuk objek baru yang dihasilkan dengan mengikuti prinsip *control* pembalik, dengan pilihan contoh dan referensi dari objek baru sebagai Singletons.
- i) **Migrations** menyediakan versi sistem *control* untuk skema *database*, sehingga memungkinkan untuk menghubungkan perubahan adalah basis kode aplikasi dan keperluan yang dibutuhkan dalam merubah tata letak *database*, mempermudah dalam penempatan dan memperbarui aplikasi.
- j) **Unit Testing** mempunyai peran penting dalam *framework* Laravel, dimana *unit testing* ini mempunyai banyak tes untuk mendeteksi dan mencegah regresi. *Unit testing* dapat dijalankan melalui fitur “*artisan command-line*”.



- k) ***Automatic Pagination*** menyederhanakan tugas dari penerapan halaman, menggantikan penerapan yang manual dengan metode otomatis yang terintegrasi ke Laravel.

2.2.4 PHP Hypertext Preprocessor (PHP)

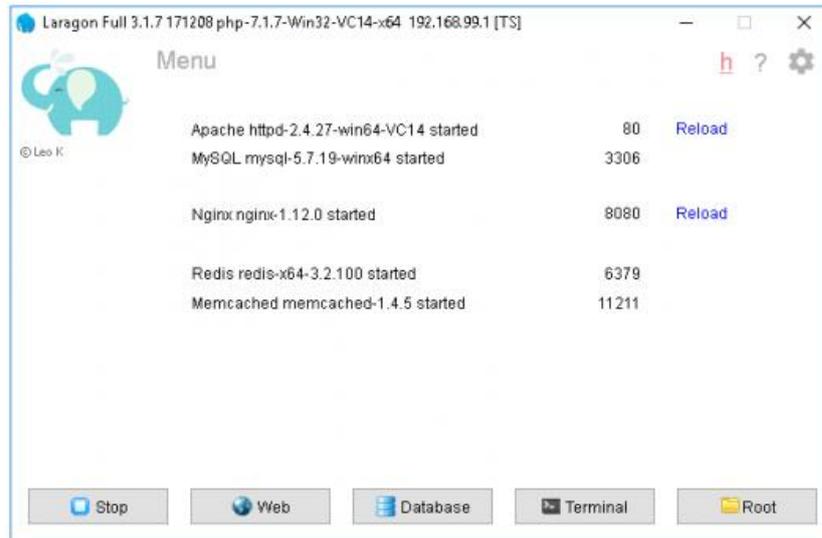
Menurut Yudhanto dan Prasetyo (2018:7) mengemukakan bahwa PHP adalah bahasa pemrograman *script server side* yang dirancang untuk membuat dan mengembangkan sebuah *website*. Bahasa pemrograman ini dirancang untuk para pengembang web agar dapat menciptakan suatu halaman web yang bersifat dinamis.

Menurut Jubilee (2017:1) mengatakan bahwa PHP adalah sebuah bahasa pemrograman yang digunakan untuk membangun sebuah *website* yang bersifat dinamis dan interaktif. Dinamis memiliki arti bahwa *website* tersebut bisa berubah-ubah tampilan kontennya sesuai dengan kondisi tertentu, dan interaktif berarti *website* tersebut dapat memberikan *feedback* bagi *user*.

Sedangkan menurut Winarno (2014:49) mengemukakan bahwa PHP merupakan sebuah bahasa pemrograman web yang berbasis *server* dan mampu mem-*parsing* kode PHP dari kode web dengan ekstensi *.php*, sehingga menghasilkan tampilan sebuah *website* yang bersifat dinamis disisi *client* (*browser*).

Dari ketiga definisi di atas dapat disimpulkan bahwa PHP adalah bahasa pemrograman web berbasis *server side* yang digunakan untuk membuat *website* yang bersifat dinamis.

2.2.5 Laragon



Gambar 2.2 Laragon

(Sumber: www.laragon.org)

Laragon adalah *universal development environment* yang portabel, terisolasi, cepat, ringan, dan mudah digunakan untuk bahasa pemrograman PHP, Node.js, Python, Java, Go, dan Ruby. Aplikasi ini sama halnya seperti XAMPP, namun didesain untuk kebutuhan *developer PHP* yang menggunakan *framework* Laravel. Laravel lebih berfokus pada kinerja, dirancang dengan kesederhanaan, fleksibilitas, dan kebebasan. Sehingga sangat bagus untuk membangun dan mengelola *modern web applications*.

2.2.6 Sublime Text



Gambar 2.3 Logo Sublime Text

(Sumber: www.MacUpdate.com)



Sublime Text Editor adalah *text editor* yang digunakan untuk mempermudah pekerjaan *programmer*. selain itu juga *sublime* digunakan untuk banyak sekali bahasa pemrograman dan bahasa *markup* serta mendukung penambahan *plugin*. *Sublime* dibangun dengan menggunakan *python*.

Sublime Text memiliki banyak kelebihan diantaranya:

1. *Multiple Selection*, mempunyai fungsi untuk melakukan perubahan pada sebuah kode dalam waktu yang sama dan dalam baris yang berbeda.
2. *Command Pallete*, berguna untuk mengakses file *shortcut* dengan mudah dengan menekan CTRL+SHIFT+P.
3. *Distraction free mode*, fitur ini sangat dibutuhkan oleh pengguna yang sedang fokus dalam pekerjaan, yaitu dapat merubah tampilan layar menjadi penuh dengan menekan SHIFT+F11.
4. *Find in project*, fungsi ini memberikan kemudahan dalam mencari dan memilih file dalam *project*, dengan menekan SHIFT+P.
5. *Multi platform*, *Sublime Text* sudah tersedia dalam berbagai *platform* sistem operasi seperti Windows, Linux, Mac os.

2.3 Teori Khusus

2.3.1 *Unified Modeling Language (UML)*

Menurut Shalahuddin (2014:137) mengatakan bahwa UML merupakan bahasa visual digunakan untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

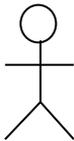
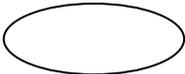
Pada perkembangan teknologi perangkat lunak, diperlukan adanya bahasa yang digunakan untuk memodelkan sebuah perangkat lunak yang akan dibangun dan diperlukan adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak tersebut. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak.



2.3.2 Use Case Diagram

Menurut Rosa dan Shalahuddin (2018:155) mengatakan bahwa *use case diagram* digunakan untuk memodelkan kelakuan (*behavior*) dari sistem informasi yang akan dibuat. *Use case* akan mendeskripsikan sebuah interaksi yang terjadi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Atau dapat dikatakan bahwa *use case* digunakan untuk mengetahui fungsi-fungsi yang terdapat di dalam sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Tabel 2.1 Simbol *Use Case Diagram*

| No | Simbol | Nama | Keterangan |
|----|---|--------------------------------|--|
| 1. |  | Aktor/ <i>Actor</i> | Aktor menggambarkan pengguna sistem, dapat berupa manusia atau sistem <i>terotomatisasi</i> lain yang berinteraksi dengan sistem lain untuk berbagi, mengirim, dan menerima informasi. |
| 2. |  | <i>Use Case</i> | Simbol ini menggambarkan interaksi antara aktor dengan <i>software</i> aplikasi tersebut. |
| 3. |  | <i>System Boundary</i> | Menggambarkan batasan anatara sistem dengan aktor. |
| 4. |  | Asosiasi/ <i>Asosiation</i> | Menggambarkan hubungan antar aktor dan <i>use case</i> . |

(Sumber : Indrajani, 2015:46)

2.3.3 Activity Diagram (Diagram Aktivitas)

Menurut Rosa dan Shalahuddin (2018:161) mengatakan bahwa *activity diagram* menggambarkan aktivitas dari sebuah sistem atau proses bisnis yang ada



pada perangkat lunak. Berikut adalah simbol-simbol yang ada pada diagram aktivitas.

Tabel 2.2 Simbol Diagram Aktivitas

| No | Simbol | Nama | Keterangan |
|----|--------|-------------------------------|--|
| 1. | | Status awal | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal. |
| 2. | | Aktivitas | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
| 3. | | Percabangan / <i>decision</i> | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu |
| 4. | | Penggabungan / <i>join</i> | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu |
| 5. | | Status akhir | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |
| 6. | | <i>Swimlane</i> | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |

(Sumber : Rosa dan Shalahuddin, 2018:162)

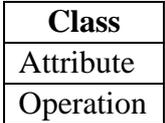
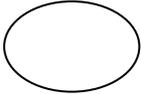
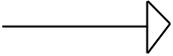
2.3.4 Class Diagram (Diagram Kelas)

Menurut Rosa dan Shalahuddin (2018:141) mengemukakan bahwa diagram kelas atau *class diagram* menggambarkan struktur sistem dari pendefinisian kelas-



kelas yang akan digunakan untuk membangun sebuah sistem sistem. Setiap kelas memiliki atribut dan *method*. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas. Sedangkan operasi atau *method* merupakan fungsi-fungsi yang dimiliki oleh suatu kelas.

Tabel 2.3 Simbol Diagram Kelas

| No | Simbol | Nama | Deskripsi |
|----|---|--------------------------------------|---|
| 1. |  | Kelas | Kelas pada struktur sistem. |
| 2. |  | Antarmuka/ <i>Interface</i> | Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi obyek. |
| 3. |  | Asosiasi/ <i>Association</i> | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> . |
| 4. |  | Asosiasi berarah/ <i>directed</i> | Asosiasi antarkelas dengan makna kelas yang satu digunakan oleh kelas lain, asosiasi biasanya juga disertain dengan <i>multiplicity</i> . |
| 5. |  | Generalisasi | Asosiasi antar kelas dengan makna generelisasi spesialisasi (umum – khusus). |
| 6. |  | Kebergantungan/ <i>dependency</i> | Relasi antar kelas dengan makna kebergantungan antarkelas. |
| 7. |  | Agregasi / <i>aggregation</i> | Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>). |

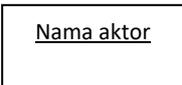
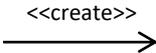
(Sumber : Rosa dan Shalahuddin, 2018:146)



2.3.5 Sequence Diagram (Diagram Sekuen)

Menurut Rosa dan Shalahuddin (2018:165) mengatakan bahwa *sequence diagram* menggambarkan kelakuan objek yang ada pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirim dan diterima antar objek tersebut. Maka dari itu dalam menggambarkan diagram sekuen, perlu diketahui terlebih dahulu objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki oleh setiap kelas yang diinstansiasi menjadi objek tersebut. Selain itu membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

Tabel 2.4 Simbol Diagram Sekuen

| No | Simbol | Nama | Deskripsi |
|----|--|-------------------------------|---|
| 1. |  | Garis hidup / <i>lifeline</i> | Menyatakan kehidupan suatu objek. |
| 2. | <p>Atau</p>   <p>Nama aktor</p> | Aktor | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat 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. |
| 3. |  | Waktu aktif | Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya. |
| 4. |  | Pesan tipe <i>create</i> | Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat. |

Lanjutan **Tabel 2.4** Simbol Diagram Sekuen

| No | Simbol | Nama | Deskripsi |
|----|--------|------------------------|---|
| 5. | | Pesan tipe <i>call</i> | Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi/metode. |
| 6. | | Pesan tipe <i>send</i> | Menyatakan bahwa suatu objek mengirimkan data/masukan/infor-masi ke objek lainnya, arah panah mengarah pada objek yang dikirim. |
| 7. | | Pesan tipe keluaran | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian. |

(Sumber : Rosa dan Shalahuddin, 2018:165)

2.4 Referensi Penelitian Sebelumnya

Penelitian dengan judul **Analisis Penentuan Bantuan Raskin Dengan Metode Fuzzy Analytic Hierarchy Process (Studi Kasus Kotamadya Palembang)** tahun 2017 oleh Leni Novianti, Isnaini Azro, dan Robinson, menggunakan metode AHP untuk menganalisis dan membantu pemerintah dalam menentukan keputusan pendistribusian beras keluarga miskin secara tepat dan optimal. Peneliti menyatakan bahwa metode AHP mampu mensimulasikan proses pengambilan keputusan yang berlangsung di dalam pikiran manusia dengan lebih baik dan terdiri dari berbagai tingkatan atau jentang pengambilan keputusan. Dalam penelitian ini, peneliti merumuskan empat kriteria penentu keputusan pemberian raskin yaitu: pendapatan, papan, sandang, pangan, dan pendidikan. Hasil yang didapat yaitu ditolak dirumuskan sebagai Alternatif 1 (A1),



dipertimbangkan dirumuskan sebagai Alternatif 2 (A2), dan disetujui dirumuskan sebagai Alternatif 3 (A3).

Menurut penelitian yang berjudul **Pendukung Keputusan Penentuan Penerima Beasiswa Menggunakan Metode SMART (*Simple Multi Attribute Rating Technique*)** tahun 2017 oleh Nurhasanah, faktor penentuan penerima beasiswa berdasarkan pada empat kriteria yaitu nilai rata-rata ujian semester, nilai absensi, nilai tugas harian, dan prestasi ekstrakurikuler yang diberi bobot untuk masing-masing kriteria dan perhitungan pembobotan menggunakan metode SMART. Hasil yang didapatkan ialah nilai terbaik dari setiap alternatif yang dapat dijadikan sebagai rekomendasi dalam seleksi calon penerima beasiswa.

Berdasarkan penelitian yang dilakukan oleh Faizal, Fatma Agus Setyaningsih, dan Muhammad Diponegoro dengan judul **Implementasi Sistem Pendukung Keputusan dengan Metode SMART untuk Merangking Kemiskinan dalam Proses Penentuan Penerima Bantuan PKH** pada tahun 2017, proses penentuan penerima bantuan Program Keluarga Harapan (PKH) dilakukan dengan melihat beberapa kriteria yang telah ditetapkan. Kriteria yang digunakan yaitu penghasilan, jumlah tabungan, luas bangunan, jenis lantai, jenis dinding, biaya pengobatan, makan sehari-hari, konsumsi daging/susu/ayam, fasilitas buang air besar, beli pakaian dalam 1 (satu) tahun, sumber penerangan, sumber air bersih, bahan bakar memasak, dan pendidikan. Setiap kriteria diberi bobot dan dilakukan perhitungan normalisasi dimana bobot setiap kriteria yang diperoleh akan dibagikan dengan jumlah setiap bobot kriteria. Selanjutnya menghitung nilai *utility* setiap kriteria dengan rumus untuk mendapatkan nilai akhir. Nilai akhir perhitungan tersebut menunjukkan ada 20 data yang muncul dengan nilai tertinggi dari pengujian 200 peserta yaitu lima peserta yang menjadi rekomendasi calon penerima bantuan PKH dan 15 peserta lulus seleksi kemiskinan dan tidak lulus seleksi PKH.

Menurut penelitian dengan judul **Sistem Pendukung Keputusan Menentukan Penerimaan Bansos Rastra Menggunakan Metode SMART di Desa Tawing Kecamatan Gondang Kabupaten Tulungagung** oleh Alvin Dwi



Andriansah pada tahun 2018, kriteria yang digunakan adalah luas lantai, jenis lantai, jenis dinding, kamar mandi, sumber penerangan, sumber air minum, bahan bakar, pengobatan, penghasilan, dan pendidikan. Masing-masing kriteria diberi bobot dan dilakukan normalisasi perhitungan dengan membandingkan antar nilai bobot kriteria dan jumlah bobot kriteria. Kemudian menghitung nilai *utility* masing-masing kriteria dan nilai akhir masing-masing alternatif. Selain itu peneliti melakukan perbandingan terhadap hasil perhitungan menggunakan sistem aplikasi dengan perhitungan manual. Analisa dilakukan menggunakan metode perhitungan presentase keakuratan. Hasil perbandingan berdasarkan presentase akurasi menunjukkan bahwa nilai 80% termasuk ke dalam tingkat korelasi tinggi. Artinya sistem aplikasi yang dibangun menggunakan metode *Simple Multi Attribute Rating Technique* (SMART) dapat digunakan untuk menentukan prioritas rumah tangga miskin penerima Bansos Rastra dari Pemerintah.

Berdasarkan penelitian yang telah dilakukan oleh Andri Andreas, Antonius Wiryadinata, dan Halim Agung pada tahun 2019 dengan judul **Penerapan Algoritma *Simple Additive Weighting* untuk Membantu dalam Menentukan Calon Penerima Bantuan Pangan Non Tunai**, penentuan calon penerima BPNT didasarkan pada 5 (Lima) kriteria dan beberapa sub kriteria, seperti luas lahan (lebih dari sama dengan 16 m²/orang, kurang dari 16 m²/orang), dinding rumah (bambu, triplek, papan, tembok), lantai (tanah, kayu, semen, keramik), penghasilan (lebih dari sama dengan Rp5000.000, kurang dari Rp5000000), dan aset (lebih dari sama dengan Rp5000.000, kurang dari Rp5000.000) setiap kriteria tersebut diberi bobot tersendiri. Perhitungan pembobotan kriteria menggunakan metode SAW yaitu dengan melakukan normalisasi sesuai dengan nilai himpunan setiap kriteria yang telah ditentukan. Kemudian mencari nilai dari setiap alternatif dengan mengalikan nilai normalisasi dari setiap kriteria dengan bobot kriteria tersebut dan selanjutnya menjumlahkan semua kriterianya. Hasil yang didapat berupa perangkungan nilai akhir yang menyeleksi alternatif terbaik dari sejumlah alternatif.