

BAB II

LANDASAN TEORI

2.1 Teori yang digunakan

2.1.1 Rancang Bangun

Dalam kaitannya dengan perancangan sistem, rancang bangun merupakan tahapan yang menerjemahkan hasil analisis suatu permasalahan ke dalam sebuah sistem baik berupa aplikasi atau program yang diimplementasikan [13]. Tujuan rancang bangun adalah menggambarkan sistematisa suatu sistem sehingga sistem tersebut tidak hanya berguna untuk pembuatan sistem baru, melainkan berguna untuk memperbaiki sistem yang telah ada.

2.1.2 Sistem Pendukung Keputusan (SPK)

Sistem Pendukung Keputusan atau SPK merupakan suatu sistem berbasis komputer yang dapat membantu proses pengambilan keputusan atas masalah-masalah semi terstruktur [14]. SPK melibatkan *database* untuk menyimpan data yang diolah menjadi informasi yang digunakan untuk mempermudah dan mempercepat pengambilan keputusan yang akurat. Salah satu metode pengambilan keputusan yaitu *multiple criteria decision making* atau MCDM. MCDM merupakan suatu metode pengambilan keputusan yang melibatkan banyak kriteria dengan tujuan mendapatkan alternatif terbaik [14]. MCDM dibagi menjadi 2 model yaitu *multiple attribute decision making* atau MADM dan *multiple objective decision making* atau MODM. MADM umumnya digunakan pada ruang diskrit sehingga lebih cocok untuk menentukan alternatif yang dianggap paling baik dari beberapa pilihan alternatif dalam jumlah terbatas, sedangkan MODM umumnya digunakan pada ruang kontinu sehingga lebih cocok untuk membuat suatu alternatif terbaik karena jumlah alternatif pada model ini sangat besar. Beberapa metode yang termasuk dalam model MADM yaitu *simple additive weighting* (SAW), *technique for order preference by similarity to ideal solution* (TOPSIS), *analytical hierarchy process* (AHP), dan lain-lain.

2.1.3.1 Simple Additive Weighting (SAW)

Simple Additive Weighting atau SAW merupakan metode pengambilan keputusan dengan konsep menghitung nilai penjumlahan terbobot dari *rating* kinerja untuk setiap alternatif dan atribut [15]. Metode ini mewajibkan pengambil keputusan untuk memberikan bobot kepada seluruh atribut. Dalam proses penghitungannya, metode ini menggunakan matriks keputusan yang dinormalisasi ke skala tertentu sehingga dapat diperbandingkan dengan alternatif lain. Adapun rumus untuk menghitung nilai *rating* sebagai berikut:

$$r_{ij} = \frac{x_{ij}}{\text{Max } x_{ij}}, \text{ untuk } j \text{ sebagai atribut } \textit{benefit} \quad (2.1) [15]$$

$$r_{ij} = \frac{\text{Min } x_{ij}}{x_{ij}}, \text{ untuk } j \text{ sebagai atribut } \textit{cost} \quad (2.2) [15]$$

Keterangan:

r_{ij}	=	<i>Rating</i> kinerja yang dinormalisasi
$\text{Max } r_{ij}$	=	Nilai <i>rating</i> kinerja maksimum
$\text{Min } r_{ij}$	=	Nilai <i>rating</i> kinerja minimum
x_{ij}	=	Nilai pada baris dan kolom <i>rating</i> kinerja dari alternatif A_i dan kriteria C_j
<i>Benefit</i>	=	Nilai terbaik merupakan nilai terbesar
<i>Cost</i>	=	Nilai terbaik merupakan nilai terkecil

Adapun rumus untuk menghitung nilai preferensi setiap alternatif sebagai berikut:

$$V_i = \sum_{j=1}^n W_j \times r_{ij} \quad (2.3) [15]$$

Keterangan:

v_i	=	Nilai terbobot suatu alternatif
n	=	Jumlah kriteria
W_j	=	Bobot yang telah ditentukan
r_{ij}	=	Nilai <i>rating</i> kinerja yang dinormalisasi

2.1.3.2 *Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)*

Technique for Order Preference by Similarity to Ideal Solution atau TOPSIS merupakan metode pengambilan keputusan multikriteria dengan konsep penentuan alternatif terbaik yaitu alternatif yang memiliki jarak terpendek dari solusi ideal positif dan alternatif yang memiliki jarak terjauh dari solusi ideal negatif [16]. Metode ini tidak memperhatikan aspek prioritas pada kriteria. Adapun tahapan pada metode ini antara lain:

- 1) Menentukan kriteria sebagai dasar pengambilan keputusan
- 2) Menentukan matriks keputusan ternormalisasi

Tahapan ini dilakukan dengan menggunakan rumus berikut:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \quad (2.4) [16]$$

Keterangan:

r_{ij} = Nilai ternormalisasi

- 3) Menentukan matriks keputusan ternormalisasi berbobot

Tahapan ini dilakukan dengan menggunakan rumus berikut:

$$y_{ij} = w_i \times r_{ij} \quad (2.5) [16]$$

Keterangan:

y_{ij} = Nilai ternormalisasi berbobot

w_i = Bobot kriteria

- 4) Menentukan matriks solusi ideal positif dan negatif

Pada tahap ini ketentuan yang berlaku sebagai berikut:

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

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

Dengan ketentuan y_1^+ merupakan $\max_i y_{ij}$ apabila j merupakan atribut *benefit* dan $\min_i y_{ij}$ apabila j merupakan atribut *cost*, sebaliknya y_1^- merupakan $\min_i y_{ij}$ apabila j merupakan atribut *benefit* dan $\max_i y_{ij}$ apabila j merupakan atribut *cost*.

- 5) Menghitung jarak nilai terbobot setiap alternatif terhadap solusi ideal positif dan negatif

Tahapan ini dilakukan dengan menggunakan rumus berikut:

$$D_i^+ = \sqrt{\sum_{j=1}^n (y_1^+ - y_{ij})^2}, \text{ untuk solusi ideal positif} \quad (2.6) [16]$$

$$D_i^- = \sqrt{\sum_{j=1}^n (y_{ij} - y_1^-)^2}, \text{ untuk solusi ideal negatif} \quad (2.7) [16]$$

Keterangan:

D_i = Jarak setiap alternatif terhadap solusi ideal

- 6) Menghitung nilai preferensi setiap alternatif

Tahapan ini dilakukan dengan menggunakan rumus berikut:

$$V_i = \frac{D_i^-}{D_i^- + D_i^+} \quad (2.8) [16]$$

Keterangan:

V_i = Nilai preferensi alternatif

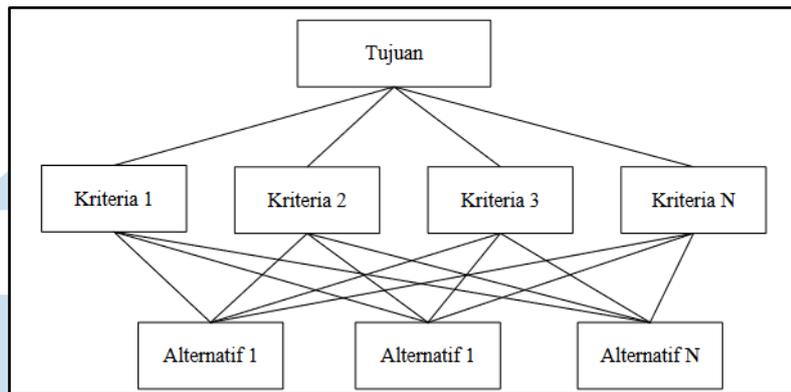
- 7) Menentukan *ranking* alternatif

2.1.3.3 Analytical Hierarchy Process (AHP)

Analytical Hierarchy Process atau AHP merupakan metode pengambilan keputusan yang bersifat kompleks, yaitu keputusan yang melibatkan multikriteria dan multialternatif, dengan menggolongkannya ke dalam kelompok yang diatur dalam hierarki [17]. Pada metode ini, baik hal kualitatif maupun hal kuantitatif diperhitungkan. Adapun dasar yang perlu dipahami pada metode AHP antara lain:

1. *Decomposition*

Pada metode AHP, tahapan pertama yang dilakukan adalah mendefinisikan permasalahan yang dihadapi kemudian menguraikannya menjadi beberapa unsur atau kelompok yang diatur dan digambarkan sebagai hierarki seperti yang terlihat pada gambar 2.1.



Gambar 2.1 Struktur Hierarki pada Metode AHP
Sumber: [17]

2. *Comparative Judgement*

Tahap selanjutnya adalah membuat matriks perbandingan berpasangan yang memuat nilai kepentingan seluruh elemen. Adapun skala yang diberikan mengacu pada skala dasar yang ditetapkan oleh metode AHP dituliskan pada tabel 2.1 berikut:

Tabel 2.1 Skala Penilaian pada Matriks Perbandingan Berpasangan dalam Metode AHP

Intensitas Kepentingan	Keterangan
1	Kedua elemen memiliki tingkat kepentingan yang sama
3	Salah satu elemen sedikit lebih penting dibandingkan elemen lainnya
5	Salah satu elemen memiliki tingkat kepentingan yang cukup lebih tinggi dibandingkan elemen lainnya
7	Salah satu elemen memiliki tingkat kepentingan yang lebih tinggi dibandingkan elemen lainnya
9	Salah satu elemen memiliki tingkat kepentingan yang ekstrim lebih tinggi dibandingkan elemen lainnya
2, 4, 6, 8	Nilai diantara dua penilaian yang berdekatan
Kebalikan	Apabila kriteria i memperoleh penilaian 1 angka dibandingkan dengan kriteria j, kriteria j dibandingkan kriteria i akan memiliki nilai kebalikannya.

Sumber: [17]

3. *Synthesis of Priority*

Tahap selanjutnya adalah membuat *eigen vector* untuk menentukan prioritas elemen. Langkah dalam menentukan *eigen vector* yaitu:

- a) Menghitung jumlah nilai setiap kolom pada matriks

- b) Menghitung nilai normalisasi matriks dengan cara membagi nilai suatu kolom dengan nilai total kolom tersebut
- c) Menghitung nilai rata-rata dengan cara menghitung jumlah nilai setiap baris kemudian dibagi dengan jumlah elemen

4. Consistency

Tahap selanjutnya adalah mengukur konsistensi untuk menghasilkan keputusan yang lebih akurat. Nilai konsistensi maksimal yang dapat diterima adalah 0,1. Langkah dalam mengukur konsistensi yaitu:

- a) Mengalikan nilai pada kolom n dengan prioritas relatif elemen n
- b) Menghitung jumlah baris
- c) Membagikan jumlah baris dengan elemen prioritas relatifnya
- d) Menghitung nilai λ maksimal dengan cara menjumlahkan hasil pembagian pada langkah c dengan jumlah elemen
- e) Menghitung *consistency index* (CI)

CI dapat dihitung dengan rumus berikut:

$$CI = \frac{(\lambda_{maks} - n)}{(n - 1)} \quad (2.9) [17]$$

Keterangan:

n = Jumlah elemen

- f) Menghitung *consistency ratio* (CR)

CR dapat dihitung dengan rumus berikut:

$$CR = \frac{CI}{RI} \quad (2.10) [17]$$

Keterangan:

RI = Nilai indeks *random consistency*

Adapun nilai indeks *random consistency* dapat dilihat pada tabel 2.2 berikut:

Tabel 2.2 Nilai Indeks *Random Consistency*

Ukuran Matriks	Nilai IR
1,2	0
3	0,58
4	0,9
5	1,12
6	1,24
7	1,32
8	1,41
9	1,45
10	1,49
11	1,51
12	1,54
13	1,56
14	1,57
15	1,59

Sumber: [17]

2.1.3 Insentif

Insentif merupakan suatu bentuk pemberian diluar upah kepada karyawan yang ditujukan sebagai bentuk penghargaan atas kinerja yang melampaui standar sehingga dapat mendukung tercapainya tujuan perusahaan [18]. Kebijakan pemberian insentif bergantung pada masing-masing perusahaan, namun umumnya ditujukan untuk memotivasi karyawan dalam bekerja sehingga terjadi peningkatan kerja. Adapun bentuk pemberian insentif dapat berupa materi seperti uang atau jaminan sosial maupun non materi seperti pujian, gelar, piagam, promosi, dan sebagainya [7].

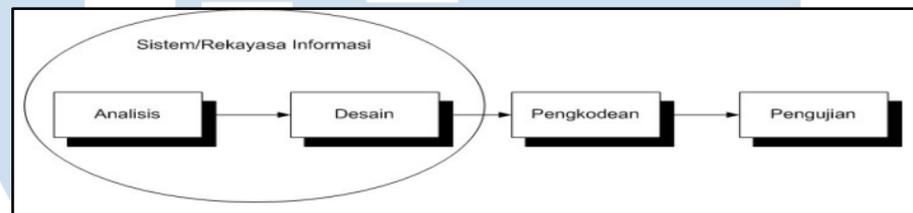
2.2 Framework yang digunakan

2.2.1 *Software Development Life Cycle (SDLC)*

Software Development Life Cycle atau SDLC merupakan alur kerja yang digunakan dalam proses pengembangan perangkat lunak [19]. Umumnya, tahapan-tahapan SDLC antara lain identifikasi kebutuhan sistem, perancangan sistem, validasi sistem, dan tahap akhir termasuk pelatihan dan penyerahan sistem kepada konsumen serta pemeliharaan sistem. Beberapa model SDLC yang sering digunakan antara lain model *waterfall*, *prototyping*, dan *rapid application development (RAD)*.

2.2.1.1 Model Waterfall

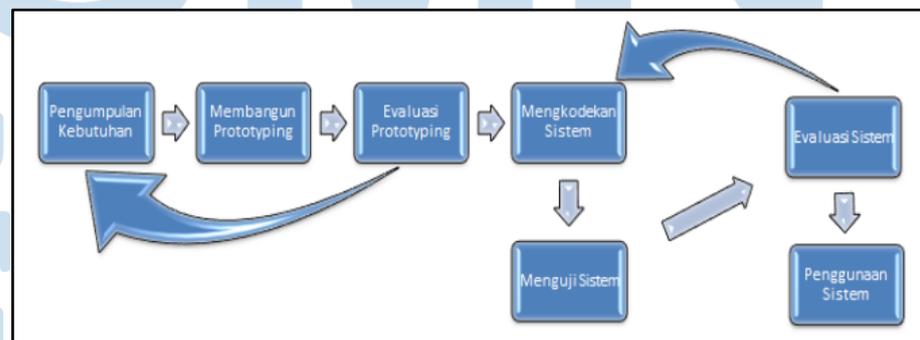
Model *waterfall* merupakan model pengembangan perangkat lunak dengan pendekatan alur hidup perangkat lunak secara sekuensial [20]. Artinya, suatu tahapan dalam alur hidup perangkat lunak tidak dapat dijalankan apabila tahapan sebelumnya belum selesai dilakukan. Pada model *waterfall* terdapat lima tahapan antara lain analisis kebutuhan sistem, desain sistem, pembuatan kode, pengujian sistem, dan pemeliharaan sistem. Gambar 2.2 menunjukkan alur pengembangan perangkat lunak menggunakan model *waterfall*.



Gambar 2.2 Model *Waterfall*
Sumber: [20]

2.2.1.2 Model Prototyping

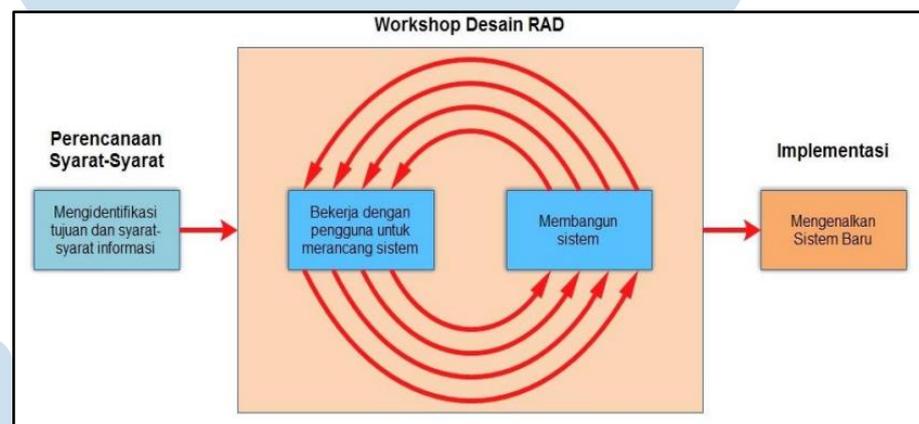
Model *prototyping* merupakan model pengembangan perangkat lunak yang memungkinkan *user* untuk memiliki gambaran dasar terkait sistem yang akan dirancang dalam bentuk *prototype* [21]. Pada model *prototyping* terdapat tujuh tahapan antara lain pengidentifikasian dan pengumpulan kebutuhan sistem, perancangan *prototype*, evaluasi *prototype*, pembuatan kode sistem, pengujian sistem, evaluasi sistem, dan penggunaan sistem. Gambar 2.3 menunjukkan alur pengembangan perangkat lunak menggunakan model *prototyping*.



Gambar 2.3 Model *Prototyping*
Sumber: [21]

2.2.1.3 *Rapid Application Development (RAD)*

Rapid Application Development atau RAD merupakan metode pengembangan perangkat lunak dengan karakteristik perkembangan yang teratur dimana waktu pengerjaan dalam pengembangan sistemnya relatif singkat [22]. Metode RAD mengadaptasi model *waterfall* namun dalam versi lebih cepat karena berbasis komponen. Umumnya, pengembangan perangkat lunak dengan metode RAD dilakukan oleh sekelompok orang yang terbagi menjadi beberapa tim sehingga setiap komponen dapat dikerjakan secara paralel oleh beberapa tim. Pada model RAD terdapat tiga tahapan antara lain perencanaan kebutuhan sistem, perancangan atau desain sistem, dan implementasi sistem. Gambar 2.4 menunjukkan alur pengembangan perangkat lunak menggunakan metode RAD.



Gambar 2.4 Metode *Rapid Application Development (RAD)*
Sumber: [22]

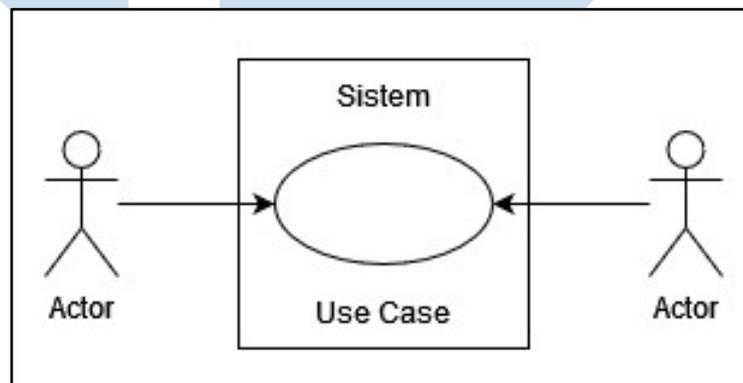
2.2.2 *Unified Modelling Language (UML)*

Unified Modelling Language atau UML merupakan alat bantu dalam pengembangan sistem berorientasi obyek yang menyediakan bahasa pemodelan visual untuk membentuk *blueprint* suatu sistem [23]. UML dibuat dengan bahasa baku yang mudah dipahami sehingga dapat digunakan untuk mengomunikasikan rancangan sistem kepada pihak terkait dalam tim pengembang sistem tersebut. UML dibentuk berdasarkan model 4+1 *view* yaitu *scenario* atau *use case view*, *development view* atau *implementation view*, *logical view*, *physical view* atau *deployment view*, dan *process view*. Masing-

masing *view* memiliki beberapa jenis diagram yang dapat digunakan. Beberapa contoh diagram pada UML yaitu *use case diagram*, *class diagram*, dan *activity diagram*.

2.2.2.1 Use Case Diagram

Use case diagram merupakan bagian dari model *scenario* yang bertujuan untuk menjelaskan fungsi-fungsi pada sebuah sistem berdasarkan sudut pandang pengguna [23]. Dengan kata lain, *use case diagram* dapat menjelaskan bagaimana sebuah sistem digunakan oleh penggunanya, yang disebut *actor*, secara berurutan. *Use case diagram* umumnya dapat mengumpulkan dan menggambarkan kebutuhan sistem yang akan dirancang. Diagram ini dapat digunakan kembali dengan menggunakan <<*include*>> atau <<*extend*>>. Adapun notasi pada *use case diagram* melibatkan tiga aspek yaitu *actor*, *use case*, dan *system boundary* yang ditunjukkan pada gambar 2.5.

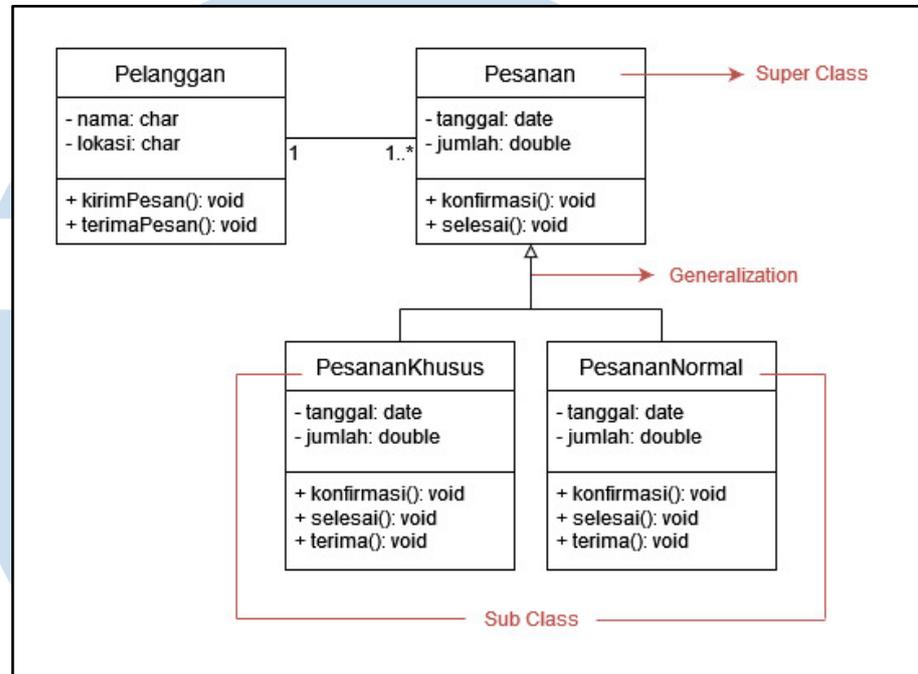


Gambar 2.5 Notasi pada *Use Case Diagram*
Sumber: [23]

2.2.2.2 Class Diagram

Class diagram merupakan bagian dari model *logical view* yang bertujuan untuk menggambarkan berbagai aspek dalam sistem yang dapat diimplementasikan pada bahasa pemrograman berorientasi objek secara langsung [23]. Beberapa aspek penting yang digunakan dalam penggambaran *class diagram* antara lain nama *class diagram*, elemen pada diagram beserta hubungannya, dan atribut dan *operation* serta

properti pada diagram. Gambar 2.6 menunjukkan salah satu contoh *class diagram*.



Gambar 2.6 Contoh *Class Diagram*

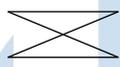
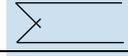
Sumber: [23]

2.2.2.3 Activity Diagram

Activity diagram merupakan bagian dari model *process view* yang bertujuan untuk menggambarkan alur kerja suatu aktivitas dalam sistem [23]. Dengan kata lain, diagram ini mampu menggambarkan tingkah laku sistem secara dinamis. *Activity diagram* memiliki kemiripan fungsi dengan *flowchart*, akan tetapi *activity diagram* mampu menggambarkan aktivitas secara paralel sedangkan *flowchart* tidak memiliki kemampuan tersebut. Diagram ini menggunakan beberapa simbol yang dijelaskan pada tabel 2.3.

Tabel 2.3 Simbol pada *Activity Diagram*

Simbol	Keterangan
●	Titik awal
⦿	Titik akhir
□	<i>Activity</i>
◇	Pilihan untuk pengambilan keputusan

Simbol	Keterangan
	<i>Fork</i> yang digunakan menunjukkan <i>activity</i> yang dilakukan secara paralel
	<i>Rake</i> yang digunakan untuk menunjukkan dekomposisi
	Tanda waktu
	Tanda pengiriman
	Tanda penerimaan
	Aliran akhir

Sumber: [23]

2.2.3 *HyperText Markup Language (HTML)*

HyperText Mark Up Language atau HTML merupakan bahasa yang digunakan untuk membentuk struktur halaman *website* [24]. Pada dokumen HTML digunakan *tag* yang umumnya berpasangan untuk menyatakan elemen. HTML dapat digunakan dengan bahasa pemrograman seperti PHP, JavaScript, dan sebagainya.

2.2.4 *Cascading Style Sheets (CSS)*

Cascading Style Sheets atau CSS merupakan bahasa yang digunakan dalam desain *website* sehingga dapat mengatur format tampilan suatu halaman *website* baik warna, ukuran, *font*, dan *formatting* lainnya [25]. CSS digunakan untuk memisahkan konten halaman dengan tampilan dokumen sehingga mengurangi kerumitan dalam penulisan *code* serta memungkinkan untuk digunakan ulang pada halaman *website* yang berbeda sehingga mempermudah dan mempercepat dalam pembuatan *website* yang memiliki banyak halaman.

2.2.5 JavaScript

JavaScript merupakan bahasa pemrograman berbentuk *script* yang digunakan untuk membuat dokumen HTML yang tampil pada *browser* sebagai halaman *website* menjadi lebih interaktif [26]. Bahasa pemrograman ini dijalankan dengan *interpreter* yang umumnya terdapat pada *browser*. Beberapa contoh implementasi bahasa JavaScript pada *website* antara lain menampilkan halaman *pop up*, mengubah tampilan kursor saat melintasi objek tertentu pada

halaman *website*, menjalankan fungsi validasi pada *form* sebelum dikirimkan ke *server*, dan sebagainya [27].

2.2.6 Bootstrap

Bootstrap merupakan sebuah *front-end framework* dengan tujuan untuk membuat tampilan *website* yang responsif yaitu tampilan yang dapat menyesuaikan layar perangkat yang digunakan [28]. *Framework* ini menyediakan HTML, CSS, dan JavaScript yang dapat dikembangkan dengan tujuan untuk mempermudah dan mempercepat pengembangan *website*.

2.2.7 Hypertext Preprocessor (PHP)

Hypertext Preprocessor atau PHP merupakan bahasa pemrograman *open source* berbentuk *script* yang ditujukan untuk pengembangan *website* dimana eksekusinya dilakukan pada sisi *server* [29]. Artinya, kode program yang dituliskan tidak dapat diketahui oleh *client* dan untuk menjalankan kode program diperlukan proses pengiriman *file* dari *client* ke *server*. Bahasa pemrograman PHP dapat digabungkan ke dalam HTML dengan menyisipkan tag *script* PHP yaitu “<?” atau “<?php” dan “?>”.

2.2.8 MySQL

MySQL merupakan program *database* bersifat *open-source* yang dapat menangani *database* relasional baik mengirim maupun menerima data dengan cepat [30]. Program *database* ini menggunakan perintah *structured query language* (SQL) dan dapat digunakan secara *multi-user*. Salah satu keunggulan *database* MySQL yang membuat *database* ini cukup populer digunakan yaitu mampu menampung data dalam ukuran besar [31].

2.2.9 CodeIgniter

CodeIgniter merupakan sebuah *framework* pengembangan *website* menggunakan bahasa pemrograman PHP yang bersifat *open source* [32]. *Framework* ini menggunakan konsep *model-view-controller* atau MVC yang mampu memisahkan pengembangan fitur dalam *layer application-logic* dengan pengembangan tampilan dalam *layer presentation*. Sebagai sebuah *framework*, CodeIgniter ditujukan untuk mempermudah dan mempercepat pengembangan

website dengan menyediakan beberapa *library* yang dapat digunakan untuk mengerjakan berbagai aktivitas umum pada *website* seperti menghubungkan dengan *database*, melakukan validasi *form*, menangani *session*, dan sebagainya.

2.2.10 User Acceptance Test

User Acceptance Test atau UAT merupakan sebuah proses untuk memverifikasi bahwa solusi yang diimplementasikan dalam sistem sesuai dengan kebutuhan pengguna [33]. Proses verifikasi dengan UAT tidak hanya berfokus untuk memastikan sistem yang berjalan lancar, melainkan berfokus pada penerimaan pengguna terhadap solusi pada sistem yang dirancang. Oleh karena itu, verifikasi dengan UAT umumnya dilakukan oleh pengguna akhir sistem. Beberapa jenis UAT antara lain *alpha & beta testing*, *contract acceptance testing*, *regulation acceptance testing*, *operational acceptance testing*, dan *black box testing*.

2.2.11 Black Box Testing

Black box testing merupakan salah satu bentuk pengujian sistem secara fungsional yang dilakukan berdasarkan sudut pandang pengguna akhir [34]. Pada pengujian dengan *black box testing* diperlukan kasus uji dengan dua perbandingan kondisi yaitu kondisi benar dan kondisi salah [35]. Tujuannya untuk memvalidasi apakah fungsi pada sistem dapat berjalan dengan baik. Bentuk pengujian ini dapat dilakukan oleh orang yang tidak berasal dari bidang pemrograman karena untuk melakukan bentuk pengujian ini tidak diperlukan pengetahuan khusus mengenai bahasa pemrograman.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

2.3 Tools yang digunakan

2.3.1 Figma

Figma merupakan suatu aplikasi desain yang banyak dimanfaatkan untuk membuat tampilan antarmuka aplikasi *mobile*, *desktop*, *website*, dan sebagainya [36]. Figma cukup banyak digunakan khususnya oleh UI/UX *designer* karena memiliki fitur yang cukup lengkap dan dapat dikerjakan oleh beberapa orang pada waktu yang sama. Kelebihan ini dipengaruhi oleh penggunaan aplikasi figma yang memanfaatkan jaringan internet.

2.3.2 SpiceLogic Analytical Hierarchy Process

SpiceLogic Analytical Hierarchy Process merupakan suatu *software* pendukung keputusan berbasis metode *analytical hierarchy process* (AHP) yang dikembangkan oleh *SpiceLogic*. *Software* ini dirancang untuk dapat digunakan secara *offline* pada sistem operasi Windows yang mendukung teknologi *Microsoft .NET Framework 4.8* [37].

2.3.3 Visual Studio Code

Visual Studio Code merupakan suatu aplikasi *code editor* bersifat *open source* yang dikembangkan oleh Microsoft dan dapat digunakan pada berbagai sistem operasi baik Windows, Linux, maupun MacOS [38]. Aplikasi *code editor* ini dapat mengenali bahasa pemrograman yang dituliskan serta mendukung cukup banyak bahasa pemrograman termasuk C++, C#, Java, Python, PHP, HTML, CSS, JavaScript, GO, dan lain-lain.

2.4 Penelitian Terdahulu

Tabel 2.4 Penelitian Terdahulu

No	Judul Penelitian	Penulis	Jurnal	Hasil Penelitian
1	Sistem Pendukung Keputusan Pemberian Bonus Karyawan Menggunakan Metode AHP pada Rumah Sakit Buah Hati Ciputat [39]	Rafhael Stevanus, Rani Irma Handayani, Dinar Ajeng Kristiyanti	Jurnal PILAR Nusa Mandiri, Vol. 14(2), September 2018	Penelitian ini dilakukan terhadap Rumah Sakit Buah Hati Ciputat untuk menentukan pemberian bonus kepada karyawan dengan menggunakan metode pengambilan keputusan yaitu <i>analytical hierarchy process</i> . Kriteria yang digunakan pada penelitian ini yaitu keahlian, sikap dan perilaku, loyalitas, serta tanggung jawab, sedangkan alternatif yang digunakan pada penelitian ini

No	Judul Penelitian	Penulis	Jurnal	Hasil Penelitian
				adalah Zuliana, Tri Sea, dan Pitriana yang merupakan perawat rumah sakit tersebut. Hasil dari penelitian ini yaitu keempat kriteria mempengaruhi pemberian bonus dengan bobot 32% untuk keahlian, 12% untuk sikap dan perilaku, 31% untuk loyalitas, dan 25% untuk tanggung jawab. Adapun alternatif yang diprioritaskan untuk mendapatkan bonus berdasarkan kriteria dan bobot yang telah dihitung adalah karyawan A atau Zuliana dengan nilai bobot 59%.
2	Sistem Penunjang Keputusan Penilaian Kinerja Karyawan Berprestasi Menggunakan Acuan MBO dan Metode AHP Menggunakan Aplikasi <i>Expert Choice</i> [40]	Rudi Sutomo, Johny Hizkia Siringo Ringo	ULTIMATICS, Vol. 10(1), Juni 2018	Penelitian ini dilakukan terhadap Universitas Tanri Abeng untuk membantu proses penilaian kinerja karyawan sehingga dapat menentukan pemberian <i>reward</i> atau <i>punishment</i> di masa yang akan datang. Kriteria yang digunakan pada penelitian ini yaitu kepuasan konsumen, prestasi, kinerja operasional, kedisiplinan, dan kepribadian yang merupakan kriteria dari metode <i>management by objectives</i> (MBO) yang disesuaikan dengan ukuran sasaran kinerja individu, sedangkan alternatif yang digunakan pada penelitian ini adalah <i>staff</i> pada divisi BAAK dan <i>staff</i> pada divisi UPT. Hasil dari penelitian ini berupa prioritas kriteria berdasarkan bobot kriteria yaitu kepuasan konsumen sebesar 0,434, kedisiplinan sebesar 0,285, kinerja operasional sebesar 0,152, kepribadian sebesar 0,071, dan prestasi sebesar 0,058 dengan tingkat inkonsistensi sebesar 0,03. Hasil ini dapat diterapkan untuk mengetahui nilai kinerja dan prioritas setiap karyawan.
3	Pemanfaatan <i>Framework CodeIgniter</i> dalam Membangun Aplikasi Penentuan Besaran Insentif <i>Canvasser</i> Studi Kasus: PT. Arga Boga Cemerlang	Sri Murni, Medi Zulham	Jurnal Ilmu Teknik dan Komputer, Vol. 6(1), Januari 2022	Penelitian ini dilakukan terhadap PT Arga Boga Cemerlang cabang Pontianak yang bergerak dibidang distribusi makanan dan minuman produk OT untuk menghitung besar nominal insentif <i>canvasser</i> berdasarkan hasil penjualannya. Insentif dihitung setiap hari dengan mempertimbangkan kriteria nominal penjualan harian minimal sebesar Rp2.200.000, kunjungan efektif minimal 36 kunjungan

No	Judul Penelitian	Penulis	Jurnal	Hasil Penelitian
	Cabang Pontianak [41]			untuk dapat memperoleh insentif sebesar Rp100.000. Selain itu terdapat pula insentif yang diberikan atas penjualan produk tertentu. Penelitian ini dilakukan dengan menggunakan metode pengembangan perangkat lunak model <i>waterfall</i> dan diuji dengan metode <i>black box testing</i> . Hasil dari penelitian ini berupa aplikasi berbasis <i>web</i> untuk perhitungan insentif <i>canvasser</i> dengan memanfaatkan <i>framework CodeIgniter</i> .
4	Analisa dan Perancangan Sistem Perhitungan Insentif Menggunakan Sistem Remunerasi [42]	Siti Khoiriyah, Ratna Mutu Manikam	EDUMATIC: Jurnal Pendidikan Informatika, Vol. 3(2), Desember 2019	Penelitian ini dilakukan terhadap PT Kobe Boga Utama yang bergerak dibidang manufaktur makanan untuk mempermudah perhitungan dan pelaporan insentif karyawan dengan sebuah sistem. Perhitungan insentif dilakukan dengan sistem remunerasi dengan mempertimbangkan total penjualan, target penjualan, dan periode penjualan. Penelitian ini dilakukan dengan menggunakan metode pengembangan perangkat lunak model <i>prototyping</i> . Hasil dari penelitian ini berupa rancangan sistem perhitungan dan pelaporan insentif karyawan khususnya <i>salesman</i> .
5	Perancangan Sistem Informasi Lelang Elektronik Kendaraan Menggunakan Metode <i>Prototype</i> [43]	Mochamad Febryan Putra Herawan, Rizal Rachman	JIKA (Jurnal Informatika), Vol. 7 (1), Februari 2023	Penelitian ini dilakukan terhadap PT Asuransi Ramayana Tbk. sebagai salah satu perusahaan yang bergerak dibidang asuransi kendaraan dan secara aktif mengadakan lelang kendaraan. Pada penelitian ini dilakukan untuk mendigitalisasi proses pelelangan sehingga dapat memudahkan dalam pengelolaan data pelelangan, mempercepat proses <i>review</i> berkas yang diunggah, dan meningkatkan transparansi dokumen. Digitalisasi proses pelelangan dilakukan dengan mengembangkan <i>website</i> e-lelang dengan metode pengembangan sistem <i>software development life cycle</i> (SDLC) model <i>prototyping</i> . Metode pengembangan sistem ini dipilih untuk mengurangi kesalahpahaman dalam perancangan sistem akibat ketidakmampuan pengguna dalam

No	Judul Penelitian	Penulis	Jurnal	Hasil Penelitian
				mendeskrripsikan keinginan mereka. Sistem berbasis <i>web</i> yang dihasilkan kemudian diuji dengan menggunakan metode <i>white box testing</i> oleh peneliti dan metode <i>black box testing</i> oleh tim TI dari perusahaan.

Tabel 2.3 menjelaskan lima penelitian terdahulu berkaitan dengan topik penelitian sistem perhitungan insentif karyawan berdasarkan kriteria yang diperoleh dari perhitungan menggunakan metode pendukung keputusan. Perbedaan antara penelitian ini dengan kelima penelitian terdahulu tersebut yaitu pada penelitian ini sistem yang dirancang adalah sistem untuk menghitung besar insentif karyawan dimana kriteria yang digunakan beserta bobotnya merupakan implementasi dari perhitungan menggunakan metode pendukung keputusan yaitu *analytical hierarchy process* (AHP) dan data yang digunakan berasal dari *database* sistem. Penelitian ini memanfaatkan metode pengembangan perangkat lunak *software development life cycle* (SDLC) model *prototyping* dan *framework* CodeIgniter.

