

## **BAB II**

### **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

Penelitian tentang sistem yang berkaitan dengan judul dari peneliti sebelumnya telah dilakukan oleh Rochman, Abdur, Achmad Sidik, dan Nada Nazahah, 2018 dengan judul “Perancangan Sistem Informasi Administrasi Pembayaran SPP Siswa Berbasis *Web* di SMK Al-Amanah”. Sistem ini dirancang dengan metode pendekatan *Object Oriented Analys and Design* (OOAD) menggunakan bahasa pemrograman *PHP* dan *MySQL* sebagai media databasenya. Dengan adanya sistem ini, pembayaran SPP di SMK Al-Amanah tidak lagi dilakukan secara manual melalui kertas SPP dan buku besar, sehingga dapat meningkatkan keakuratan, kecepatan, serta ketepatan dalam pembuatan laporan administrasi pembayaran SPP[3].

Tarmon, Rosaria Indah dan Rahmat Inggi, 2021 melakukan penelitian yang berjudul “Perancangan Sistem Informasi Pembayaran Tagihan Rekening Kota Kendari Secara *Online*”. Dalam penelitian ini digunakan metode *System Development Life Cycle* (SDLC). Tujuan dari penelitian ini adalah menghasilkan sebuah sistem pembayaran PDAM secara *online* sehingga pelanggan bisa dengan mudah melakukan pengecekan dan pembayaran rekening PDAM tanpa harus datang langsung dan antri di kasir pembayaran PDAM[4].

Penelitian yang dilakukan oleh Adamed, Shaded, Moontaha Siddika, Saiful Islam, Sadia Saima Anika, Anika Anjum, dan Milon Biswas, 2021 dengan judul penelitian mereka “*BPS : Blockchain Based Decentralized Secure and Versatile Light Payment Sistem*”. Sistem ini dibuat dengan menggunakan *blockchain* dengan tujuan untuk memudahkan konsumen supaya dapat melakukan transaksi secara *online* dengan mudah dan fleksibel. Dalam sistem ini konsumen dapat melihat riwayat transaksi yang telah mereka lakukan[5]. Penelitian lain dilakukan oleh Xu, Dongbo dan Hailin Cui, 2021 dengan judul “*Design of Real-Time Electricity Cost Informatization Calculation System Based on Big Data Mining Algorithm*”. Penelitian ini menggunakan algoritma *Big Data Mining* yang menghasilkan sebuah sistem yang dapat melakukan perhitungan pembayaran listrik dengan estimasi waktu yang jelas[6].

Pada penelitian ini, akan dibangun sebuah Sistem Informasi Layanan RT/RW Net Berbasis *Web* pada Unit Usaha *Wifi* BUMDes Arja Jaya Abadi. Dalam pembangunan sistem ini terdapat perbedaan dengan

penelitian sebelumnya, yaitu sistem ini dibangun dengan menggunakan *framework CodeIgniter* serta dilengkapi dengan fitur *reminder* melalui *E-mail* untuk mengingatkan jatuh tempo pembayaran kepada para pelanggan.

## 2.2 Landasan Teori

Penelitian ini diperlukan adanya teori-teori yang mendasar untuk menunjang proses penelitian ini. Teori-teori yang digunakan dalam penelitian ini adalah :

### 2.2.1 Sistem Informasi

Sistem merupakan suatu kerangka dari prosedur-prosedur yang saling berhubungan yang disusun sesuai dengan suatu skema yang menyeluruh dan sistematis. Sistem merupakan dasar pergerakan dalam seluruh kegiatan. Keberadaan sistem sangat diperlukan, karena tanpa adanya konsep dari sistem maka kegiatan atau pekerjaan akan berjalan tanpa kendali. Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Data merupakan kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata[7].

Sistem informasi merupakan kumpulan dari sub-sub sistem yang saling terintegrasi dan berkolaborasi untuk menyelesaikan masalah tertentu dengan cara mengolah data dengan alat yang namanya komputer sehingga memiliki nilai tambah dan bermanfaat bagi pengguna[8]. Menurut Jogiyanto, kualitas informasi ditentukan oleh tiga hal, yaitu informasi harus akurat (*accurate*), tepat waktu (*timelines*), serta relevan (*relevance*). Informasi harus akurat, artinya informasi harus bebas dari kesalahan dan tidak boleh menyesatkan. Informasi harus tepat waktu berarti informasi yang datang kepada penerimanya tidak boleh terlambat, karena sebuah informasi digunakan dalam proses pembuatan keputusan. Selain itu, informasi harus relevan karena informasi tersebut bermanfaat bagi pemakainya. Kegunaan dari informasi adalah untuk mengurangi ketidakpastian dalam proses pengambilan keputusan[9].

Kepuasan pengguna menjadi salah satu indikator suksesnya sebuah sistem informasi. Kepuasan pengguna berarti ada kesesuaian antara sesuatu yang diharapkan dengan hasil yang diperoleh. Tingkat kepuasan pengguna akan meningkatkan efisiensi dan efektivitas kerja penggunaan sistem informasi yang diterapkan[10]. Semakin tinggi suatu

sistem informasi yang menghasilkan kualitas informasi akan semakin meningkatkan pula kepuasan penggunaannya[11].

### **2.2.2 Website**

*Website* merupakan layanan yang didapati oleh pemakai komputer yang terhubung dengan jaringan komputer. *Website* adalah aplikasi tertentu yang berjalan di atas *platform* atau *operation system browser*. *Website* merupakan sebuah halaman informasi yang tersedia secara *online* dan dapat diakses di seluruh dunia selama tersambung dengan jaringan internet. *Website* dapat terdiri dari format teks, gambar, grafik, tabel, kutipan, musik, video, dan format visual lainnya yang menarik bagi pengunjung *website* tersebut[12].

### **2.2.3 CodeIgniter (CI)**

*CodeIgniter* (CI) merupakan *framework* pengembangan aplikasi yang menyediakan banyak librari dan fungsi[13]. Penggunaan *CodeIgniter* dapat memudahkan pengembang dalam membuat *web* dengan lebih cepat. *Framework* ini menggunakan model MVC (*Model, View, Controller*) yang dapat meningkatkan fleksibilitas kode[14]. Selain itu, *CodeIgniter* bersifat *open-source* sehingga bisa digunakan secara gratis dan memiliki konfigurasi yang sederhana.

### **2.2.4 Rekayasa Perangkat Lunak**

Perangkat lunak merupakan program yang berisi perintah-perintah untuk melakukan proses tertentu. Perangkat lunak digunakan untuk menjalankan komputer atau aplikasi tertentu pada komputer[15].

Rekayasa Perangkat Lunak merupakan terjemahan dari terminologi *Software Engineering*. Rekayasa Perangkat Lunak adalah suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal *requirement capturing* (analisa kebutuhan pengguna), *specification* (menentukan spesifikasi dari kebutuhan pengguna), desain, *coding*, *testing*, sampai pemeliharaan sistem setelah digunakan[16].

#### **2.2.4.1 Metode Pengembangan Sistem**

Tahap pengembangan sistem dalam penelitian ini menggunakan metode air terjun atau sering disebut dengan metode *waterfall*. Metode ini juga sering dinamakan siklus hidup klasik (*classic life cycle*). Metode *waterfall* dipilih karena proses pengembangan model *face one by one*,

sehingga dapat meminimalis kesalahan yang mungkin timbul. Tahapan metode *waterfall* yang diadopsi dari Ian Sommerville adalah sebagai berikut :

1) Analisis dan Definisi Persyaratan

Pada tahap ini dibutuhkan komunikasi pengembang sistem supaya bisa memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi ini biasanya diperoleh melalui wawancara maupun diskusi atau survei secara langsung. Kemudian informasi yang didapat dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

2) Perancangan Sistem dan Perangkat Lunak

Spesifikasi dari tahap sebelumnya akan dipelajari pada tahap ini dan merancang desain sistem. Perancangan sistem membantu dalam menentukan perangkat keras (*hardware*) dan sistem persyaratan serta membantu dalam mendefinisikan arsitektur secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan deskripsi abstraksi sistem perangkat lunak yang mendasar dan hubungan-hubungannya.

3) Implementasi dan Pengujian Unit

Pada tahap ini perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai unit *testing*.

4) Integrasi dan Pengujian Sistem

Seluruh unit program atau program individual diintegrasikan dan diuji sebagai sistem yang lengkap untuk menjamin persyaratan sistem telah terpenuhi. Setelah integrasi seluruh sistem diuji untuk mengecek setiap kegagalan maupun kesalahan[17].

### 2.2.4.2 Metode Pengujian Sistem

1) Pengujian *Black Box*

Pengujian *black box* merupakan teknik pengujian tanpa mengacu pada struktur internal dari komponen atau sistem. Pengujian *black box* menangani kebutuhan pelanggan dari *input* yang valid maupun tidak valid. Pengujian ini digunakan untuk memvalidasi fungsi keseluruhan sistem apakah telah berfungsi dengan benar[18]. Tujuan dari pengujian *black box* adalah untuk menemukan kesalahan dalam kategori :

- a. Fungsi yang tidak benar atau hilang.
- b. Kesalahan *interface*.
- c. Kesalahan dalam struktur data atau akses *database* eksternal.



- d. Kesalahan kinerja.
- e. Inisialisasi dan kesalahan terminasi.
  - 2) Pengujian *White Box*

Pengujian *white box* digunakan untuk menguji dan menganalisis kode program apabila terjadi kesalahan. Pengujian ini dilakukan pada bagian *code* tanpa melihat tampilan *interface* dari halaman aplikasi[19].

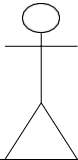
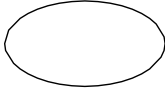


### 2.2.5 UML (Unified Modelling Language)

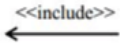

*Unified Modelling Language* (UML) merupakan sebuah bahasa yang sudah menjadi standar dalam industri visualisasi, merancang dan mendokumentasikan sebuah sistem perangkat lunak[20].

#### 1) *Use Case Diagram*

*Use case diagram* merupakan pemodelan untuk menyatakan visualisasi interaksi yang terjadi antara pengguna (aktor) dengan sistem. *Use case diagram* digunakan untuk menjelaskan konteks dari sebuah sistem sehingga terlihat jelas batasan dari sebuah sistem[21]. Simbol-simbol yang ada di dalam *use case diagram* antara lain :

**Tabel 2. 1** *Use Case Diagram*



No.	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Pengguna sistem, sistem yang berhubungan dengan sistem lain, atau waktu.
2.		<i>Use Case</i>	Abstraksi dan interaksi antara sistem dengan <i>actor</i> .
3.		<i>Association</i>	Untuk menghubungkan antara <i>actor</i> dengan <i>use case</i> .
4.		<i>Generalization</i>	Menunjukkan spesialisasi <i>actor</i> untuk dapat

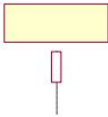
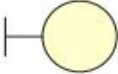






			berpartisipasi dengan <i>use case</i> .
5.		<i>Include</i>	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> yang lain.
6.		<i>Extend</i>	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsionalitas <i>use case</i> lainnya.

## 2) Sequence Diagram

*Sequence* diagram adalah UML yang menggambarkan interaksi antar objek yang berada di dalam dan sekitar sistem yang meliputi pengguna, *display*, dan sebagainya berupa *message* yang digambarkan terhadap waktu. *Sequence* diagram biasa digunakan untuk menggambarkan sebuah skenario atau rangkaian langkah-langkah yang dilakukan sebagai bentuk respons dari sebuah *event* untuk menghasilkan *output* tertentu[22]. Berikut ini adalah simbol-simbol yang terdapat pada *sequence* diagram :

**Tabel 2. 2** *Sequence* Diagram

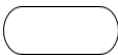


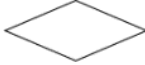


No.	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Menggambarkan pengguna yang berinteraksi dengan sistem, baik di dalam maupun di luar sistem.
2.		<i>Lifeline</i>	Menghubungkan objek selama <i>message</i> dikirim atau diterima dan aktifasinya.
3.		<i>General</i>	Menggambarkan entitas tunggal

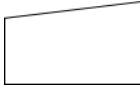

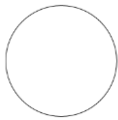
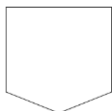
			dalam <i>sequence diagram</i> .
4.		<i>Boundary</i>	Merupakan tepi dari sistem, biasanya dapat berupa <i>user interface</i> .
5.		<i>Control</i>	Untuk mengatur aliran dari informasi dalam sebuah skenario sistem.
6.		<i>Entity</i>	Menyatakan elemen yang bertugas untuk menyimpan data atau informasi.
7.		<i>Activation</i>	Mempresentasikan waktu yang dibutuhkan oleh suatu objek untuk menyelesaikan tugas tertentu.
8.		<i>Message</i>	Sebagai komunikasi antar objek terhadap aksi yang dilakukannya.
9.		<i>Message Entri</i>	Untuk menunjukkan pesan antar objek yang menunjukkan urutan kejadian yang terjadi.
10.		<i>Message to Self</i>	Untuk menunjukkan pesan objek itu sendiri.
11.		<i>Message Return</i>	Untuk menunjukkan hasil dari pengiriman <i>message</i> yang digambarkan menggunakan arah dari kanan ke kiri.

### 2.2.6 Flowchart

*Flowchart* merupakan langkah-langkah untuk memecahkan masalah yang ditulis atau dilambangkan dengan simbol-simbol tertentu. *Flowchart* akan menunjukkan alur program secara realistis dan logis[23]. Berikut adalah simbol-simbol yang terdapat di dalam *flowchart* beserta fungsinya :

**Tabel 2. 3 Flowchart**

No.	Simbol	Nama	Keterangan
1.		<i>Terminal</i>	Untuk memulai atau mengakhiri suatu program.
2.		<i>Input/Output</i>	Untuk memasukkan data dan menunjukkan hasil dari proses tanpa tergantung dengan jenis peralatannya.
3.		<i>Flow</i>	Untuk menghubungkan simbol satu dengan simbol yang lain, atau menyatakan jalan arus dalam suatu proses. Simbol ini sering disebut juga dengan <i>connecting line</i> .
4.		<i>Decision</i>	Untuk memilih proses berdasarkan kondisi.
5.		<i>Document</i>	Simbol ini digunakan untuk data yang berbentuk informasi.
6.		<i>Manual Operation</i>	Menunjukkan pengolahan yang tidak dilakukan oleh <i>pc</i> /komputer.

7.		<i>Manual Input</i>	Untuk memasukkan data secara manual.
8.		<i>Proses</i>	Digunakan untuk menyatakan suatu proses yang dilakukan oleh komputer.
9.		<i>On-Page Connector</i>	Untuk menghubungkan proses data dengan proses lainnya yang berada dalam satu halaman.
10.		<i>Off-Page Connector</i>	Untuk menghubungkan proses dalam halaman yang berbeda.

### 2.2.7 Pengertian Basis Data

Basis data merupakan kumpulan data yang menjabarkan suatu aktivitas dari berbagai entitas yang berhubungan. Basis data dikelola melalui perangkat lunak yang disebut dengan *Database Managements System (DBMS)*[24]. Saat ini banyak DBMS yang beredar di pasaran, misalnya *Oracle, SQL Server, Sybase*, dan *Acces*. Ada tiga tipe basis data (*database*) yang dikenal luas saat ini, yaitu :

#### 1) *Relational Database*

Tipe ini menyimpan data dalam susunan baris dan kolom. Jenis data yang disimpan meliputi mata uang, angka riil, *integer*, dan *string* (karakter data).

#### 2) *Object-oriented Database*

Tipe ini menyimpan objek sebagai data. Objek dapat berupa sesuatu yang nyata ataupun abstrak. Objek data ini dapat dikembangkan menjadi tipe data tingkat lanjut, seperti gambar, film, dan suara.

#### 3) *Object-relational Database*

Tipe ini merupakan gabungan dari tipe *relational* dan *object-oriented database*. Tipe ini menyimpan dan memanfaatkan data dalam bentuk

tradisional (mata uang, angka riil, *integer*, *string*) maupun data tingkat lanjut seperti gambar, film, dan suara.


Semua DBMS setidaknya memiliki dua jenis perintah yang digunakan untuk mengelola dan mengorganisasikan data, yaitu :




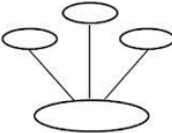



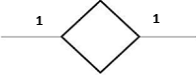
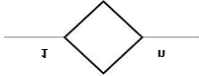
- 1) Perintah DDL (*Data Defination Language*), merupakan suatu perintah yang ditujukan pada *database* secara keseluruhan, biasanya hanya dapat diakses oleh admin *database*. Contoh perintah DDL :
  - a. *Create*, digunakan untuk membuat *database* atau tabel baru. Contohnya : `CREATE TABLE pegawai (id INTEGER PRIMARY KEY, nama_pegawai VARCHAR(25) NOT NULL);`.
  - b. *Alter*, berfungsi untuk mengubah struktur tabel. Contohnya : `Alter TABEL pegawai ADD nama_pegawai(25);`.
  - c. *Drop*, digunakan untuk menghilangkan suatu objek. Contohnya : `DROP TABLE pegawai;`
- 2) Perintah DML (*Data Manipulation Language*), merupakan suatu perintah yang diberikan kepada isi datanya dan dapat diakses secara umum. Contoh perintah DML :
  - a. *Select*, digunakan untuk mengambil atau menampilkan data dari tabel. Contohnya : `SELECT*FROM pegawai;`
  - b. *Insert*, digunakan untuk menyimpan data ke dalam tabel. Contohnya : `INSERT*INTO pegawai (nama, alamat) values (“Eva”, “Sidareja”);`.
  - c. *Update*, digunakan untuk mengubah data di dalam tabel. Contohnya : `UPDATE pegawai SET alamat = “Cilacap” WHERE id_pegawai = 001;`
  - d. *Delete*, digunakan untuk menghapus data dari tabel. Contohnya : `DELETE FROM pegawai WHERE id_pegawai = 001;`

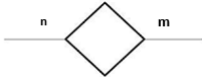
### 2.2.4.3 ERD (*Entity Relationship Diagram*)

*Entity Relationship Diagram* (ERD) merupakan diagram yang menggambarkan *relationship* antar entitas yang relevan dari *system interest*[25]. Simbol-simbol yang terdapat dalam ERD antara lain :

**Tabel 2. 4** *Entity Relationship Diagram*

No.	Simbol	Nama	Keterangan
1.		<i>Himpunan Entity</i>	Menjelaskan sesuatu apa saja yang terdapat dalam sistem.

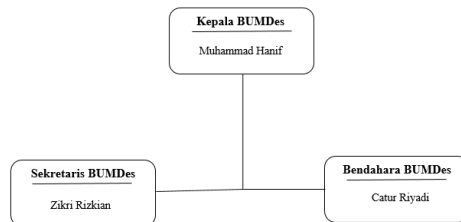
2.		<i>Atribut Key</i>	Digunakan untuk menentukan entitas secara unik.
3.		<i>Atribut Simple</i>	Merupakan atribut yang memiliki nilai tunggal dan tidak dapat dipecah.
4.		<i>Atribut Multivalued</i>	Atribut yang memiliki sekelompok nilai untuk setiap entitas.
5.		<i>Atribut Composite</i>	Merupakan atribut yang terdiri dari beberapa atribut yang lebih kecil.
6.		<i>Atribut Derivatif</i>	Merupakan atribut yang dihasilkan dari atribut lain.
7.		<i>Relationship</i>	Hubungan antar entitas dari himpunan entitas yang berbeda.
8.		<i>Link</i>	Penghubung antar atribut untuk memperlihatkan adanya hubungan entitas.
9.		<i>Relasi One to One</i>	Setiap entitas hanya memiliki relasi dengan satu entitas yang lain.
10.		<i>Relasi One to Many</i>	Satu entitas memiliki relasi dengan beberapa entitas lain.

11.		<i>Relasi Many to Many</i>	Setiap entitas bisa memiliki relasi dengan entitas yang lain.
-----	---	----------------------------	---

### 2.2.8 BUMDes

BUMDes atau Badan Usaha Milik Desa merupakan suatu lembaga/badan perekonomian di desa yang berbadan hukum dibentuk dan dimiliki oleh Pemerintah Desa, dikelola secara ekonomis mandiri dan profesional dengan modal sebagian besar atau seluruhnya dari kekayaan desa yang dipisahkan. Tujuan dari BUMDes adalah mengoptimalkan pengelolaan aset-aset yang dimiliki desa, memajukan perekonomian desa, serta meningkatkan kesejahteraan masyarakat desa. BUMDes juga berfungsi sebagai motor penggerak perekonomian desa serta sebagai sarana untuk mempercepat peningkatan kesejahteraan masyarakat desa. Dengan adanya BUMDes diharapkan suatu desa dapat menjadi lebih mandiri dan masyarakatnya lebih sejahtera[26].

BUMDes Arja Jaya Abadi merupakan Badan Usaha Milik Desa Tinggarjaya, Kecamatan Sidareja, Kabupaten Cilacap, Jawa Tengah. BUMDes Arja Jaya Abadi berperan penting dalam meningkatkan perekonomian desa. Salah satu bentuk peran serta BUMDes Arja Jaya Abadi dalam memajukan perekonomian Desa Tinggarjaya adalah dengan mendirikan Unit Usaha *Wifi*. Unit usaha ini menawarkan pemasangan *wifi* untuk masyarakat Tinggarjaya dengan tujuan dapat membantu para warga memiliki kualitas internet yang lebih baik dengan harga yang bervariasi dan cukup terjangkau. Unit usaha ini berdiri pada bulan Maret 2020 dan masih berjalan hingga sekarang. Struktur kepengurusan BUMDes Arja Jaya Abadi saat ini tercantum pada **Gambar 2.1**.



**Gambar 2. 1** Struktur Kepengurusan BUMDes Arja Jaya Abadi



Berikut ini adalah penjelasan mengenai jabatan yang ada di BUMDes Arja Jaya Abadi beserta deskripsi tugasnya :

- 1) Ketua BUMDes, merupakan orang yang memimpin, mengendalikan, serta bertanggungjawab atas seluruh aktivitas yang dijalankan oleh BUMDes. Tugas dari Ketua BUMDes antara lain :
  - a. Memimpin, mengelola, serta mengurus BUMDes dan seluruh unit usaha yang dijalankan oleh BUMDes.
  - b. Merumuskan SOP untuk unit usaha yang dijalankan oleh BUMDes.
  - c. Merumuskan kebijakan operasional dalam mengelola BUMDes.
  - d. Mengendalikan kegiatan-kegiatan yang dilakukan oleh BUMDes.
  - e. Mengangkat dan memberhentikan pegawai BUMDes sesuai dengan peraturan yang berlaku.
  - f. Mengadakan perjanjian kerjasama dengan pihak lain dalam upaya mengembangkan unit usaha yang dikelola oleh BUMDes.
- 2) Sekretaris BUMDes, adalah orang yang melaksanakan fungsi pengelolaan administrasi BUMDes. Tugas dari sekretaris BUMDes antara lain :
  - a. Melaksanakan tugas sekretaris untuk mendukung kegiatan Kepala BUMDes.
  - b. Melaksanakan administrasi umum kegiatan operasional BUMDes.
  - c. Mengelola surat menyurat secara umum.
  - d. Melaksanakan kearsipan di BUMDes.
  - e. Mengelola data serta informasi seluruh unit usaha yang dijalankan oleh BUMDes.
- 3) Bendahara BUMDes, merupakan orang yang bertanggungjawab atas pengelolaan keuangan BUMDes. Tugas dari bendahara BUMDes antara lain :
  - a. Melaksanakan kebijakan operasional pengelolaan fungsi keuangan BUMDes.
  - b. Melaksanakan strategi pengelolaan unit usaha BUMDes.
  - c. Mengelola gaji dan insentif pegawai BUMDes.
  - d. Menyusun laporan pengelolaan keuangan BUMDes.
  - e. Mengeluarkan uang berdasarkan bukti-bukti yang sah.
  - f. Menyetorkan uang ke bank setelah mendapat persetujuan dari Kepala BUMDes.

### 2.2.9 Uji Usability

Uji *usability* merupakan metode pengujian sistem yang digunakan untuk mengetahui seberapa mudah perangkat lunak digunakan oleh pengguna ketika berinteraksi dengan sistem yang telah dikembangkan. Dalam proses pengujian *usability* dapat disesuaikan dengan pilihan penguji perangkat lunak serta keadaan objek. Uji *usability* ini dapat dilakukan pada perangkat lunak *desktop*, *web*, dan *mobile*.

Uji *usability* terdiri atas lima aspek, yaitu *learnability*, *efficiency*, *memorability*, *errors*, dan *satisfaction*. *Learnability* digunakan untuk mengetahui tingkat kemudahan perangkat lunak untuk dipelajari. *Efficiency* digunakan untuk mengetahui tingkat kecepatan perangkat lunak dalam menyelesaikan tugasnya. *Memorability* digunakan untuk mengetahui tingkat kemudahan tampilan perangkat lunak dapat diingat oleh pengguna. *Errors* digunakan untuk mengetahui cara perangkat lunak mengatasi kesalahan yang muncul. Dan *satisfaction* digunakan untuk mengetahui seberapa besar tingkat kepuasan pengguna ketika menggunakan perangkat lunak [27]. Perhitungan nilai *usability* dapat dihitung menggunakan rumus sebagai berikut :

$$Usability = \frac{Nilai\ Perolehan}{Nilai\ Maximal} \times 100\%$$

Perolehan nilai uji *usability* dapat dikelompokkan ke dalam interval dan interpretasi persen untuk mengetahui besar penilaian setiap komponen. Kategori yang digunakan untuk mengelompokkan nilai uji *usability* dapat dilihat pada **Tabel 2.5**.

**Tabel 2. 5** Interval Predikat Sistem

Persentase Hasil Pengujian	Predikat
81% - 100 %	Sangat Baik
61% - 80%	Baik
41% - 60%	Cukup
21% - 40%	Kurang
0% - 20%	Sangat Kurang

### 2.2.10 Skala Likert

Skala likert digunakan untuk mengukur sikap dan pendapat yang digunakan dalam penelitian. Skala likert digunakan untuk melengkapi

kuesioner yang mengharuskan para responden menunjukkan tingkat persetujuan terhadap sederet pertanyaan.

Dengan menggunakan skala likert maka jawaban dari setiap butir pertanyaan memiliki interval jawaban sebanyak lima. Jawaban setiap butir pertanyaan yang menggunakan skala likert memiliki tingkatan dari sangat positif sampai sangat negatif [28]. Interval jawaban dengan skala likert dapat dilihat pada **Tabel 2.6**.

**Tabel 2. 6** Interval Jawaban Skala Likert

<b>Jawaban</b>	<b>Skor</b>
Sangat Kurang	1
Kurang	2
Cukup	3
Baik	4
Sangat Baik	5

**~Halaman ini sengaja dikosongkan~**