

BAB II LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam penelitian ini akan digunakan lima tinjauan pustaka yang nantinya dapat mendukung penelitian, berikut ini merupakan tinjauan pustaka yang diambil yaitu:

Tabel 2.1 Tinjauan Pustaka

No	Judul	Nama (Tahun)	Hasil
1	Sistem Informasi Monitoring Perjanjian Kerja Sama Berbasis Web Pada PT Dayamitra Telekomunikasi Jakarta	(Marsuyitno <i>et al.</i> , 2020)	Sistem informasi monitoring yang akan dibuat ini berbasis web yang akan memudahkan user dalam monitoring, update status monetize kuota titik yang tercantum dokumen perjanjian kerja dengan governtemnt.
2	Sistem Informasi Monitoring Kontrak Pada Pengadaan Area PT PLN (Persero) Palembang	(Agustina and Pirza, 2019)	Dari hasil implementasi dapat dilihat bahwa Sistem Informasi Monitoring Kontrak Pada Pengadaan Area PT PLN (Persero) Palembang dapat diterapkan dengan baik
3	Pengembangan Sistem Informasi Monitoring Dan Evaluasi Perjanjian Kerja Waktu Tertentu	(Akbar and Kamisut, 2021)	Berdasarkan demo dan pengujian sistem yang telah dilakukan, dapat disimpulkan bahwa pengembangan sistem monitoring dan evaluasi, yang telah dilakukan, dapat membantu karyawan dalam proses bisnis perusahaan outsourcing.

Tabel 2.1 Tinjauan Pustaka (Lanjutan)

No	Judul	Nama (Tahun)	Hasil
4	Sistem Informasi Monitoring Perkembangan Proyek Berbasis Web Pada Pt. Wahana Reka Tekindo Jakarta	(Saputro, Sukendar and Hidayat, 2018)	Hasil dari penelitian ini adalah sebuah sistem informasi monitoring dalam sebuah website yang menampilkan informasi perkembangan proyek dari mulai terkontrak hingga proyek selesai.
5	Rancangan Sistem Informasi Monitoring Progres Multiproyek	(Kusuf, Nuh and Indrayadi, 2018)	Hasil pengolahan data pada sistem akan ditampilkan dalam bentuk tabel dan grafik. Sistem akan dibuat dalam bentuk website sehingga pengguna dapat memonitoring menggunakan teknologi yang terhubung dengan internet.

Berdasarkan uraian tinjauan pustaka diatas, maka pembahasan yang dilakukan belum pernah dilakukan oleh peneliti-peneliti sebelumnya terutama pada perusahaan angkatan sepuluh, serta dapat dilihat perbedaannya yaitu :

1. Metode pengembangan yang digunakan yaitu *extreme programming*.
2. Monitoring yang dilakukan pada jadwal pelaksanaan
3. Menampilkan informasi progres hasil pelaksanaan.

2.2 Monitoring

Menurut Lestari and Setiyadi (2019) monitoring adalah proses pengumpulan dan analisis informasi berdasarkan indikator yang ditetapkan secara sistematis dan kontinu tentang suatu kegiatan atau program sehingga mampu dilaksanakan tindakan koreksi untuk penyempurnaan kegiatan itu selanjutnya.

Menurut Prayitno and Noe'man (2020) monitoring adalah monitoring akan memberikan informasi tentang status dan kecenderungan bahwa pengukuran dan evaluasi yang diselesaikan berulang dari waktu ke waktu, pemantauan umumnya dilakukan untuk tujuan tertentu, untuk memeriksa terhadap proses berikut objek atau untuk mengevaluasi kondisi maupun kemajuan menuju tujuan hasil manajemen atas efek tindakan dari beberapa jenis antara lain tindakan untuk mempertahankan manajemen yang sedang berjalan.

Output monitoring berupa *progressreport* proses. *Output* tersebut diikut secara deskriptif maupun *non-deskriptif*, *output* monitoring bertujuan untuk mengetahui kesesuaian proses telah berjalan. *Output monitoring* berguna pada perbaikan mekanisme proses kegiatan dimana monitoring dilakukan.

2.2.1 Proses Monitoring

Menurut Lestari and Setiyadi (2019) Proses Monitoring adalah proses monitoring adalah proses dimana dilakukannya pengumpulan data serta memantau hasil dari kemajuan object program. Proses monitoring berfokus memantau perubahan yang menitik beratkan pada proses dan hasil keluaran. Monitoring memiliki beberapa tujuanyaitu :

1. Mengkaji apakah kegiatan monitoring yang telah direncanakan sesuai dengan rencana awal.
2. Mengamati dan memantau setiap aktivitas proses monitoring terhadap object program.
3. Mengidentifikasi setiap permasalahan yang timbul supaya dapat teratasi dengan cepat.

4. Menyesuaikan kegiatan dengan lingkungan yang berubah tanpa menyimpang dari tujuan awal melakukan penilaian dan manajemen apakah pola kerja yang digunakan sesuai dengan rencana dan mampu mencapai tujuan kegiatan.

2.3 Operasional

Menurut Wahjono (2021) Operasional adalah suatu konsep yang bersifat abstrak guna memudahkan pengukuran suatu variabel. Operasional juga dapat diartikan sebagai suatu pedoman dalam melakukan kegiatan atau pekerjaan penelitian. Definisi operasional yaitu juga didasarkan pada karakteristik yang dapat diobservasi dari apa yang sedang didefinisikan atau mengubah konsep yang berupa konstruk dengan kata yang menggambarkan perilaku atau gejala yang dapat diamati dan diuji serta ditentukan kebenarannya oleh orang lain.

Pengertian definisi operasional dalam variabel penelitian adalah suatu atribut atau sifat atau nilai dari objek atau kegiatan yang mempunyai variasi tertentu yang telah ditetapkan oleh peneliti untuk dipelajari dan kemudian ditarik kesimpulannya

Sehingga operasional merupakan suatu variabel atau kegiatan suatu atribut atau sifat atau nilai dari objek atau kegiatan tersebut untuk menghasilkan suatu informasi kegiatan yang telah dilakukan.

2.4 Kontrak Kerja

Menurut Prihadi *et al.*, (2020) kontrak kerja adalah suatu perjanjian antara pekerja dan pengusaha secara lisan dan atau tulisan, baik untuk waktu tertentu maupun untuk waktu tidak tertentu yang memuat syarat-syarat kerja, hak dan

kewajiban. Setiap perusahaan wajib memberikan kontrak kerja di hari pertama anda bekerja.

Kontrak Kerja/Perjanjian Kerja menurut Undang-Undang No.13/2003 tentang Ketenagakerjaan adalah perjanjian antara pekerja/buruh dengan pengusaha atau pemberi kerja yang memuat syarat-syarat kerja, hak, dan kewajiban para pihak.

Sehingga kontrak kerja merupakan perjanjian antara dua pihak sesuai syarat dan kewajiban yang telah disepakati bersama dalam hal pekerjaan.

2.5 Web Based

Menurut Urbietta *et al.* (2019), *Web Based* adalah aplikasi yang dibuat berbasis *web* yang membutuhkan *web server* dan *browser* untuk menjalankannya.

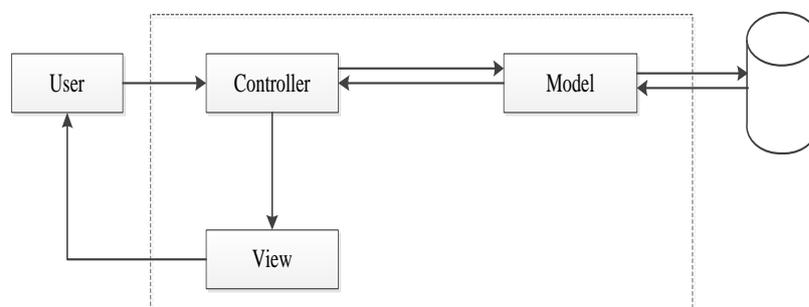
Dengan membuat sistem berbasis *web based* ada beberapa hal yang penting dan harus kita pikirkan sebelum membangun sistem tersebut, diantaranya:

1. Tidak membutuhkan *hardware* dengan spesifikasi yang tangguh untuk menjalankan aplikasinya.
2. Server yang dibutuhkan cukup diinstallkan *tools* pendukung saja agar klien mudah menjalankan aplikasi
3. Infrastruktur jaringan yang dibutuhkan juga cukup besar karena aplikasi yang dibuat dapat diakses dari jaringan luar (internet).
4. Aplikasi berbasis *web based* dapat diakses dari berbagai perangkat dengan syarat menggunakan *web browser* saja sudah dapat mengaksesnya.
5. Jika aplikasi yang sudah jadi ingin di *update*, sangat mudah untuk melakukannya karena tidak membutuhkan membuka keseluruhan aplikasi.

2.5.1 CodeIgniter

Menurut Raharjo (2018), CodeIgniter adalah Framework untuk bahasa pemrograman PHP, yang dibuat Rick Ellis pada tahun 2006. CodeIgniter memiliki banyak fitur yang membantu para pengembang PHP untuk dapat membuat aplikasi secara mudah dan cepat serta memiliki sifat yang fleksibel dapat dikembangkan dalam perangkat web, dekstop maupun mobile.

CodeIgniter memiliki konsep atau pola Model-View-Controller (MVC) sehingga kode-kode dapat di sederhanakan.



Gambar 2.1 Arsitektur MVC

2.5.2 PHP

Menurut Subagja (2018), PHP adalah bahasa *server-side-scripting* yang menyatudengan HTML untuk membuat halaman *web* yang dinamis.

Menurut Oetomo and Maharginono (2020), berpendapat bahwa *PHP Hypertext Preprocessor* adalah bahasa pemograman *web server-side* yang bersifat *open source*. PHP merupakan *script* yang terintegrasi dengan HTML dan berada pada *server (server side HTML embedded scripting)*. PHP adalah *script* yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima *client* selalu yang terbaru/*up to date*. Semua *script* PHP dieksekusi pada *server* dimana *script*

tersebut dijalankan. Dengan menggunakan program PHP, sebuah *website* akan lebih interaktif dan dinamis.

Sehingga PHP merupakan bahasa pemrograman yang digunakan oleh pengembang untuk membuat sistem *website* dengan kumpulan bahasa HTML dan *script* lainnya.

2.5.3 MySQL

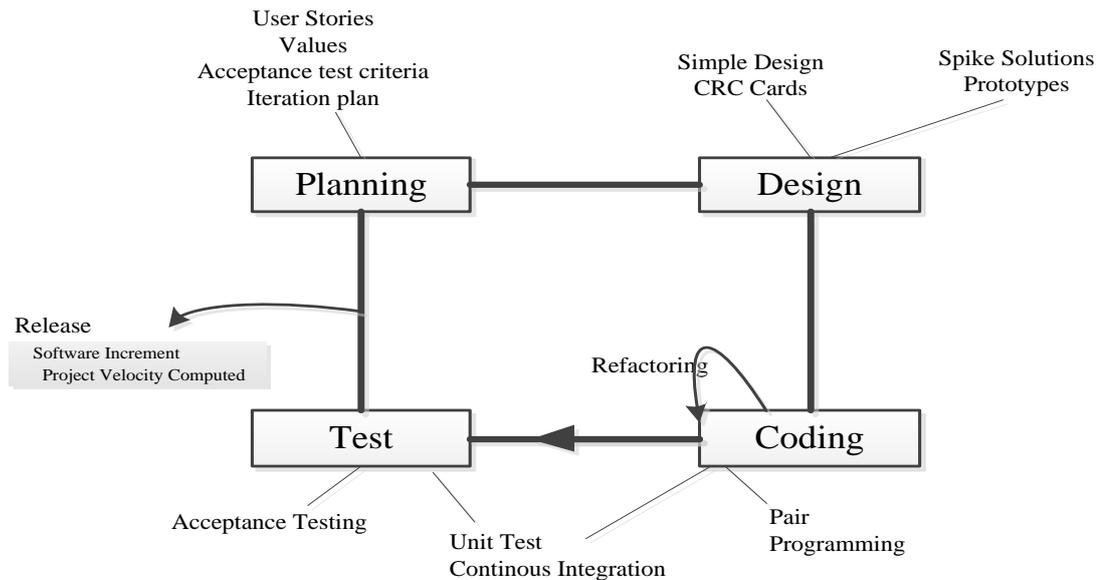
Menurut MySQL (2018), *MySQL* adalah singkatan dari *Structure Query Language* yang digunakan untuk mendefinisikan structure data, memodifikasi data pada basis data, menspesifikasi batasan keamanan (*security*), hingga pemeliharaan data.

Menurut Amin (2018) mendefinisikan *mysql* adalah RDBMS yang cepat dan mudah digunakan, serta sudah banyak digunakan untuk berbagai kebutuhan.

MySQL merupakan bahasa standar yang paling banyak digunakan untuk mengakses *database* relasional dan merupakan aplikasi yang dapat dipergunakan secara bebas.

2.6 Metode Extreme Programming

Menurut Suryantara (2017) *extreme programming* berdasarkan sejarah singkat bahwa pengembangan perangkat lunak banyak digunakan untuk pengembangan yang lebih cepat dengan meliputi tahapan *planning*, *design*, *coding* dan *testing*. Berikut merupakan konsep *Extreme programming*.



Gambar 2.2 *Extreme Programming*

Sumber: (Suryantara, 2017)

2.6.1 Kerangka Kerja Extreme Programming

Pengembangan yang dilakukan menggunakan XP dengan proses yang lebih cepat dengan tahapan seperti *planning*, *design*, *coding* dan *testing*.

1. *Planning*/Perencanaan

Tahap ini dimulain dengan pemahaman konteks bisnis dari aplikasi dengan mendefinisikan keluaran seperti fitur, fungsi, penentuan waktu dan biaya serta alur pengembangan.

2. *Design*/Perancangan

Tahap perencanaan secara sederhana dengan alat mendesain kartu CRC (*Class Responsibility Collaborator*) yang digunakan untuk pemetaan kelas-kelas yang akan diguanakna pada diagram UML.

3. *Coding*/Pengkodean

Hal utama dalam pengembangan menggunakan XP yaitu *pari programming* (Proses pembuatan program melibatkan 2 atau lebih programmer).

4. *Testing/Pengujian*

Tahap ini fokus pada pengujian fitur pada aplikasi sehingga tidak ada kesalahan dan sesuai dengan proses bisnisnya.

2.6.2 Keuntungan dan Kerugian *Extreme Programming*

Keuntungan pada penerapan metode XP yaitu:

- a. Dalam hal XP menjalin komunikasi yang baik dengan klien pada pengembangan aplikasi
- b. Saling menghargai antar developer dan meningkatkan komunikasi
- c. Dapat menjadi pembelajaran bagi orang lain
- d. Klien mendapatkan umpan balik yang akurat mengenai aplikasi yang dibuat
- e. Dengan XP dapat mengubah pemikiran pelanggan terhadap aplikasi yang dibuat
- f. Developer tidak berkerja secara berlebihan
- g. Dengan XP dapat membuat keputusan yang bersifat teknikal

2.6.3 *Class Responsibility Collaboration (CRC) Card*

Menurut Rosa and Shalahuddin (2019) *Class Responsibility Collaboration (CRC) Card* merupakan tools yang digunakan untuk mendefinisikan behavior dan responsibility dari masingmasing class dan hubungan kolaborasi antara *class-class* tersebut. *CRC Card* dikembangkan dengan interaksi antara analis dan pengguna. Masing-masing orang diminta untuk mendeskripsikan logika yang diperlukan untuk memenuhi suatu *responsibility*, dan informasi apa saja yang dibutuhkan

tetapi tidak dimilikinya. *Class* lain yang memiliki informasi yang dibutuhkan akan menjadi *collaborator* untuk *behavior* tersebut.

2.7 UML (*Unified Modelling Language*)

Menurut Rosa and Shalahuddin (2019) UML (*unified Modelling Language*) adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada UML (*Unified Modelling Language*).

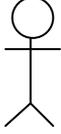
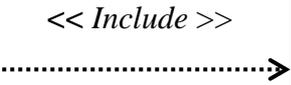
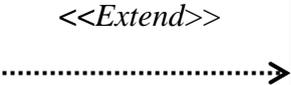
2.7.1 *Use Case Diagram*

Menurut Rosa and Shalahuddin (2019) *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi 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 tersebut. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Use Case Diagram* dapat dilihat pada Tabel 2.3.

Tabel 2.2 Simbol *Use Case Diagram*

No	Simbol	Deskripsi
1.		<i>Usecase</i> Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal <i>frase</i> nama <i>use case</i> .

Tabel 2.3 Simbol *Use Case Diagram* (Lanjutan)

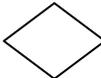
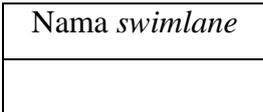
2.		Aktor Aktor seseorang/sesuatu yang berinteraksi dengan yang akan dibuat. diluar sistem informasi. Biasanya dinyatakan menggunakan kata benda
3.		Asosiasi/association merupakan komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
4.		Generalisasi (<i>generalization</i>) merupakan hubungan (umum – khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum
5.		Include berarti use case yang ditambahkan akan dipanggil saat use case tambahan dijalankan.
6.		Ekstensi (<i>extend</i>) merupakan use case tambahan ke sebuah use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu.

Sumber : (Rosa and Shalahuddin, 2019)

2.7.2 Activity Diagram

Menurut Rosa and Shalahuddin (2019) *activity* diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *activity diagram* dapat dilihat pada Tabel 2.3:

Tabel 2.4 Simbol *Activity Diagram*

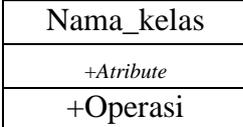
No.	Simbol	Keterangan
1.		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Percabangan (<i>Decision</i>) merupakan asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari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

Sumber : (Rosa and Shalahuddin, 2019)

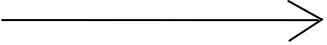
2.7.3 Class Diagram

Menurut Rosa and Shalahuddin (2019) *Class diagram* mengembangkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Class Diagram* dapat dilihat pada Tabel 2.4.

Tabel 2.5 Simbol *Class Diagram*

No.	Simbol	Deskripsi
1.		Kelas pada struktur sistem.

Tabel 2.4 Simbol *Class Diagram* (Lanjutan)

2.	<p>Antar Muka/<i>Interface</i></p>  <p>Nama_<i>Interface</i></p>	Sama dengan konsep interface dalam pemrograman berorientasi objek.
3.	<p>Asosiasi / <i>Association</i></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan symbol
4.	<p>Asosiasi Berarah / <i>Directed Association</i></p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan symbol.
5.	<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	<p>Agregasi / <i>aggregation</i></p> 	Relasi antar kelas dengan maksna semua bagian (<i>whole-part</i>)

Sumber: (Rosa and Shalahuddin, 2019)

2.8 Pengujian *Black Box*

Menurut Rosa and Shalahuddin, (2019) Pengujian sistem adalah proses untuk mengecek apakah suatu perangkat lunak yang dihasilkan sudah dapat dijalankan sesuai standar atau belum. Pengujian sistem dapat menggunakan metode *black box testing* yaitu merupakan pendekatan komplementer dari teknik *white box testing*, karena pengujian *black box testing* mampu mengungkap kesalahan yang lebih luas. *Black box testing* berfokus pada pengujian persyaratan fungsional perangkat lunak, karena untuk mendapatkan serangkaian kondisi input yang sesuai dengan persyaratan fungsional suatu program.

Jadi, pengujian sistem dapat dilakukan dengan pengecekan *input*, pengecekan *output* dan pengecekan proses sebagai berikut:

- a. Pengecekan *input*, meliputi kelengkapan item-item input, kemudahan pengoperasian, kemudahan manipulasi data, dan pengendalian kesalahan.
- b. Pengecekan proses, dilakukan dengan pengecekan output program.
- c. Pengecekan *output*, meliputi pengecekan terhadap format dan bentuk-bentuk laporan

2.8.1 Kelebihan *Black Box Testing*

- a. Tidak perlu melihat *source code* secara detail.
- b. Mendeteksi kesalahan pengetikan (*typo*).
- c. Mendeteksi kesalahan *Design* atau *User Interface* dari sebuah *software* atau *website*.
- d. Menampilkan asumsi yang tidak sesuai dengan kenyataan, untuk di analisa dan diperbaiki.
- e. Seorang *Tester* tidak harus *Programmer*.

2.8.2 Kekurangan *Black Box Testing*

- a. Ketergantungan dengan dokumen dan design *software* tersebut.
- b. Tidak sampai level *code*, sehingga tester tidak mengetahui *level security* dari *software* tersebut.

2.8.3 Pengukuran Skala *Likert*

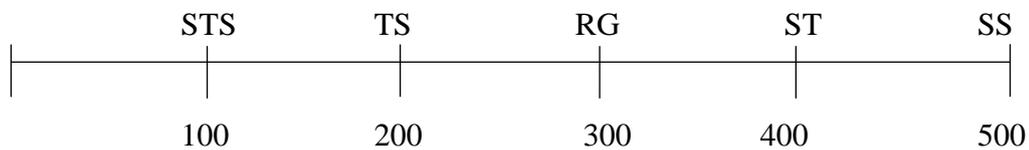
Menurut Sugiyono (2018), *skala likert testing* adalah Skala *Likert* digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Dalam penelitian fenomena sosial ini telah ditetapkan secara spesifik oleh peneliti, yang selanjutnya disebut sebagai variabel penelitian.

Dengan *skala likert* maka variabel yang akan diukur dijabarkan menjadi indikator variabel. Kemudian indikator variabel tersebut dijadikan sebagai titik tolak untuk menyusun item-item instrumen yang dapat berupa pernyataan atau pertanyaan. Pengujian menggunakan 5 kategori jawaban dengan bobot yang berbeda untuk setiap jawabannya seperti pada Tabel 2.5.

Tabel 2.6 Bobot Jawaban *Usability*

No.	Pertanyaan	Jawaban				
		SS	ST	RG	TS	STS
1.	Tampilan pada sistem mudah digunakan	5	4	3	2	1

Secara hasil kesimpulan dari hasil kuisioner yang diperoleh dapat menggunakan pengukuran dari hasil maksimal nilai tertinggi, berikut merupakan gambar aspek *usability* pada Gambar 2.3.



Gambar 2.3 Hasil Pengukuran Aspek *Usability*

Kemudian hasil perhitungan yang didapatkan dari angket, selanjutnya dibandingkan dengan rentang kriteria interpretasi skor untuk menyatakan hasil yang didapatkan dengan rentang pada Tabel 2.6.

Tabel 2.7 Rentang *Criteria Interpretasi*

No	Rentang Kriteria	Kriteria
1	0% - 20%	Sangat Tidak Baik
2	21% - 40%	Tidak Baik
3	41% - 60%	Kurang Baik
4	61% - 80%	Baik
5	81% - 100%	Sangat Baik

(Sumber: Sugiyono, 2018)