

BAB II LANDASAN TEORI

2.1. Tinjauan Pustaka

PT Bukit Asam, Tbk. Unit Pelabuhan Tarahan merupakan Badan Usaha Milik Negara (BUMN) yang didirikan pada tanggal 2 Mei 1981 yang beralamatkan di Jl. Soekarno Hatta Km. 15. Tarahan, Bandar Lampung. Berdasarkan Peraturan Pemerintah No. 42 tahun 1980 tanggal 15 Desember 1980, dengan kantor pusat di TanjungEnim, Sumatera Selatan (PT.Bukit Asam, 2018). Dalam penelitian ini akan digunakan lima tinjauan pustaka yang nantinya dapat mendukung penelitian, berikut ini merupakan tinjauan pustaka yang diambil yaitu pada tabel 2.1:

Tabel 2.1 Literatur Review

No Literatur	Penulis	Tahun	Judul
Literatur 1	Wawan Kurniawan, Kriswanto, M. Fatchurrochman	2015	Perancangan Dan Pembuatan Sistem Informasi Pemeliharaan Dan Perawatan Di Jurusan Teknik Mesin UNNES.
Literatur 2	Junaidi Novi Cholisoh, Nur Hasanah	2018	Rancang Bangun Sistem Manajemen IT Untuk Pencatatan Story Pemeliharaan Sebagai Pendukung Ke-putusan

Tabel 2.1 Literatur Review (Lanjutan)

Literatur 3	Aufal Marom, Nur Alamsyah, dan Haji Sirajuddin	2017	Sistem Informasi Manajemen Perawatan Mesin Produksi Berbasis Web Pada PT.Budi Jaya Benjarindo
Literatur 4	Tries Augustrianto	2017	Rancang Bangun Sistem Informasi Pengelolaan dan Perawatan Peralatan Laboratorium Komputer Berbasis Web.
Literatur 5	Mulyono, Prima Widyawati Wardaningsih , Agung Satrio Nugroho	2016	Sistem Informasi Penjadwalan Dan Pemeliharaan Kalibrasi Alat Kesehatan

2.1.1 Tinjauan Terhadap Literatur 1

Penelitian (Wawan Kurniawan, Kriswanto, M. Fatchurrochman., 2015) dari jurusan Teknik Mesin Di Universitas Negeri Semarang yang berjudul **“Perancangan Dan Pembuatan Sistem Informasi Pemeliharaan Dan Perawatan Di Jurusan Teknik Mesin UNNES”**. Masalah yang terjadi pada penelitian tersebut ialah tidak adanya sistem dalam melakukan manajemen pemeliharaan pada universitas negeri semarang sehingga tujuan dari penelitian tersebut ialah : membangun sistem informasi manajemen pemeliharaan dan perawatan menggunakan basis data; dengan menerapkan sistem schedule dalam sistem informasi manajemen pemeliharaan dan perawatan basis data. Metode dalam penelitian tersebut menggunakan prosedur penelitian terapan, sedangkan tahapan perancangan SIM basis data adalah: merumuskan spesifikasi SIM dengan pemeliharaan dan perawatan basis data yang akan disusun: penulisan program; Pengujian dan debugging dokumentasi. Penelitian tersebut sudah menghasilkan

desain Sistem Informasi Manajemen Pemeliharaan dan Perawatan (SIMPP) yang memiliki kemampuan dalam mengelola kegiatan pemeliharaan dan perawatan inventaris pemesanan yang meliputi pendataan inventaris dan pemeliharannya. SIMPP sendiri dihasilkan guna menerapkan sistem penjadwalan karena dalam SIMPP terdiri dari tabel satuan kerja/teknisi/petugas, tabel inventaris, tabel, jadwal pemeliharaan dan perawatan, serta jenis kegiatan berdasar dengan tipe perawatannya..

2.1.2 Tinjauan Terhadap Literatur 2

Penelitian (Junaidi Novi Choliso, Nur Hasanah., 2018) Dari jurusan Sistem informasi STMIK Raharjama dengan judul **“Rancang Bangun Sistem Manajemen IT Untuk Pencatatan Story Pemeliharaan Sebagai Pendukung Keputusan”** Dalam upaya untuk mengatasi permasalahan- permasalahan yang ada perlu dikembangkan sistem yang lebih baik untuk pengelolaan aset IT khususnya history Pemeliharaan yang nanti nya dapat digunakan sebagai pendukung keputusan. Manajemen aset IT khususnya pencatatan history Pemeliharaan adalah suatu praktek bisnis yang sangat penting pada pada suatu perusahaan. Aset IT yang dimaksud mencakup pemeliharaan aset perangkat keras (hardware) serta sistem (perangkat lunak). Tujuan dari penelitian tersebut untuk menganalisa serta merancang sistem info manajemen aset IT buat pencatatan history Pemeliharaan. karena sistem yang ada saat ini buat history Pemeliharaan masih berupa pencatatan manual dengan menggunakan Ms. Excel sehingga dalam pencarian data masih beredar pada masing-masing petugas IT serta kurangnya keakuratan data. Metode penelitian yang digunakan dalam penelitian ini merupakan metode analisis memakai analisa PIECES, dan menggunakan melakukan observasi secara langsung pada perusahaan, kepustakaan menggunakan mempelajari dokumen- dokumen yang sudah ada, dan wawancara pada pihak yang terkait pada perusahaan. Perancangannya contoh sistem menggunakan UML (*Unified Modelling Language*) serta diimplementasikan menggunakan bahasa pemrograman PHP serta database MySQL. Manfaat asal penelitian ini merupakan buat membentuk sistem yang memudahkan dalam pencatatan history Pemeliharaan sebagai akibatnya data yang ada dapat digunakan

dalam pengambilan keputusan pembelian aset IT serta menghindari pembelian aset yang tidak perlu.

2.1.3 Tinjauan Terhadap Literatur 3

Penelitian (Aufal Marom, Nur Alamsyah, dan Haji Sirajuddin., 2018) dari Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Islam Kalimantan Muhammad Arsyad Al Banjari Banjarmasin, dengan judul **“Sistem Informasi Manajemen Pemeliharaan Mesin Produksi Berbasis Web Pada PT.Budi Jaya Banjarindo”** dari masalah pada penelitian tersebut sistem Pemeliharaan Mesin PT. Budi Jaya Banjarindo pada sistem penjadwalan yang masih manual, pendataan mesin produksi, suku cadang mesin masih menggunakan data dari teknisi yang di catat manual di buku catatan, informasi mesin produksi terbilang masih terbatas sehingga menjadi tidak efektif. Pada pernyataan tersebut diperlukannya sistem pemeliharaan mesin produksi yang akurat dan tepat. Sistem pemeliharaan mesin pada PT Budi Jaya banjarindo masih manual dengan catatan buku yang di tulis kedalam pendataan penjadwalan servis, pendataan servis, dan pendataan mesin rusak. Sehingga sangat menyulitkan saat perekapan data mesin perbulan dan pertahunnya. Dengan metode tersebut maka dihasilkan sistem berbasis web yang dapat membantu perusahaan dalam memproses pengolahan sistem pengarsipan data penjadwalan servis, data pengerjan servis, data kerusakan komponen serta perbaikannya, data keluhan operator mesin, data stok barang, data produksi, data harga barang, data biaya pemeliharaan mesin dan memberikan kemudahan dalam pembuatan laporan yang tepat dan akurat. Penelitian tersebut telah menghasilkan sebuah rancangan sistem yang memudahkan perusahaan dalam melakukan proses pengolahan sistem dengan pengarsipan data penjadwalan servis, data pengerjan servis, data kerusakan komponen serta perbaikan, data keluhan operator mesin, data stok barang, data produksi, data harga barang, data biaya pemeliharaan mesin sehingga menghindari terjadinya kesalahan pada saat melakukan pendataan dan pembuatan laporan

2.1.4 Tinjauan Terhadap Literatur 4

Penelitian (Tries Augustrianto, 2017) Dari Jurusan Fakultas Teknologi Informasi & Elektro Universitas Teknologi Yogyakarta dengan judul “**Rancang Bangun Sistem Informasi Pengelolaan dan Perawatan Peralatan Laboratorium Komputer Berbasis Web..**” Masalah yang dihadapi dalam penelitian tersebut ialah belum adanya system untuk menentukan jadwal pemeliharaan terhadap peralatan pada laboratorium komputer sehingga menyulitkan bagi petugas baik itu siswa ataupun guru dalam melakukan pengelolaan lab. Tujuan dari penelitian tersebut ialah membuat sistem informasi pengelolaan dan perawatan laboratorium komputer berbasis web ini adalah untuk mempermudah baik siswa maupun para guru dalam meminjam, merawat dan mengelolala segala sesuatu yang berada didalam laboratorium tersebut. Dalam pembuatan sistem ini perangkat lunak yang digunakan adalah xampp untuk menjalankan web servernya, MySql untuk pengelolaan database, Adobe Dreamweaver CS5, tools yang digunakan script menggunakan bahasa pemrograman html, php, SDLC sebagai metodenya dan javascrpt. Sistem tersebut sudah dapat membantu atau mempermudah segala kegiatan yang berkaitan dengan laboratorium baik untuk siswa maupun guru.

2.1.5 Tinjauan Terhadap Literatur 5

Penelitian (Tries Augustrianto, 2016) Dari Jurusan Fakultas Teknologi Informasi & Elektro Universitas Teknologi Yogyakarta dengan judul “**Sistem Informasi Penjadwalan Dan Pemeliharaan Kalibrasi Alat Kesehatan**” Masalah pada penelitian tersebut ialah jumlah teknisi elektromedis sangat terbatas. sehingga perlu adanya suatu sistem yang bisa mengetahui prioritas alat mana saja yang perlu segera dilakukan inspeksi dan pemeliharaan preventif maupun kalibrasi. Untuk penentuan jumlah inspeksi yang perlu dilakukan didasarkan dari nilai Equipment Management. Pada penelitian tersebut dibuatlah sebuah sistem informasi penjadwalan pemeliharaan dan kalibrasi berbasis web dengan bahasa pemrograman PHP serta dalam pengolahan basis datanya menggunakan *firebase database platform*. Pada gambaran interaksi petugas dengan sistem dibuat menggunakan use case sedangkan pemodelan alir aktifitas petugas dalam sistem

dibuat dalam bentuk activity diagram, interaksi petugas dengan sistem berdasarkan urutan waktu yang digambarkan menggunakan *Sequence Diagram*. Sistem informasi yang dibuat untuk mengetahui jadwal pemeliharaan dan kalibrasi yang disajikan berdasarkan informasi secara mudah dan cepat yang bisa disajikan secara *real time*.

Berdasarkan penelitian terdahulu seperti diatas dapat dilihat perbedaannya yaitu :

1. Metode pengembangan sistem dan alat pengembangan sistem yang dipakai yaitu *extreme programming* dan UML (*Unit Modelling Language*) sebagai alat pengembangan sistem serta metode pengujian sistem menggunakan *black box testing*.
2. Informasi yang ditampilkan berupa aset peralatan, penjadwalan, Pemeliharaan dan pelaporan.

2.2. Pengembangan

Menurut Jogiyanto (2016: 35), pengembangan (development) yakni dapat diartikan menyusun sistem yang baru untuk menggantikan sistem yang lama yang tidak efektif dan secara keseluruhan memperbaiki sistem yang telah ada. Sistem lama perlu dikembangkan atau diganti disebabkan karena beberapa hal.

2.3. Sistem Informasi

Menurut Krismiaji (2015) Sistem informasi adalah tata cara yang didalam organisasi untuk mengumpulkan, menginputkan, mengolah serta menyimpan data, dan cara-cara yang diorganisasi untuk menyimpan, mengelola, mengendalikan, dan melaporkan informasi dengan sedemikian rupa sehingga sebuah organisasi dapat mencapai tujuan yang telah ditetapkan.

Jadi berdasarkan pengertian diatas dapat disimpulkan bahwa sistem informasi adalah kumpulan-kumpulan data yang terintegritasi dan saling melengkapi dengan menghasilkan *output* yang baik guna untuk memecahkan masalah dan pengambilan keputusan.

2.4. Manajemen

Manajemen sendiri ialah proses pengambilan keputusan dan implementasi sesuai dengan perolehan, penggunaan, dan distribusi aset selain itu, proses sistematis yang bertujuan untuk memelihara, memperbaharui, menghapus serta melakukan laporan pada PT. Bukit Asam, Tbk sendiri sudah memiliki sistem namun, pada sistem sebelumnya menggunakan sistem web yang dihubungkan ke jaringan *lokalhost* sehingga tidak bisa menangani akses secara *real-time*, akurat, terintegrasi, dan petugas *friendly*. akses yang diberikan tidak petugas *friendly* , Selain itu selain itu membutuhkan *Knowledge* dalam menentukan perencanaan Pemeliharaan dan mencari solusi baru. Menurut (Sucahyowati 2017, 5) mengatakan bahwa Manajemen adalah suatu proses yang terdiri dari rangkaian kegiatan, seperti *Planning, Organizing, Staffing, Directing* dan *Controlling* yang dilakukan oleh para anggota organisasi dengan menggunakan seluruh sumber daya organisasi untuk menentukan dan mencapai tujuan yang telah ditetapkan.

Menurut Hasibuan dalam (Anwar 2015, 1) mengatakan bahwa : Manajemen adalah ilmu dan seni mengatur proses pemanfaatan sumber daya manusia dan sumber-sumber lainnya secara efektif dan efisien untuk mencapai suatu tujuan tertentu .Dari definisi mengenai manajemen di atas, maka dapat disimpulkan bahwa manajemen adalah sebuah ilmu dan seni dalam proses menggunakan seluruh sumber daya dengan cara *planning, organizing, staffing, directing* dan *controlling* yang dilakukan oleh anggota organisasi untuk mencapai tujuan yang sudah ditentukan.

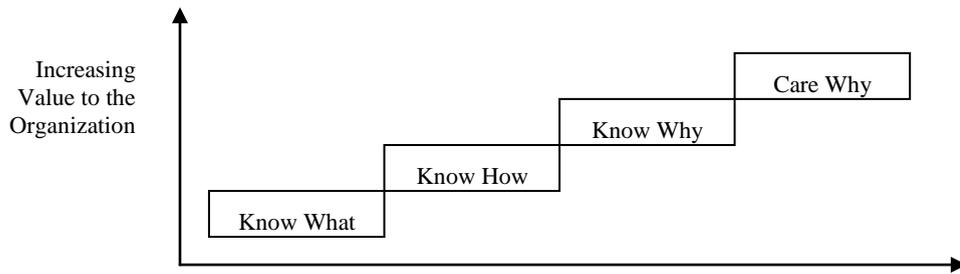
2.5. Knowledge management System (KMS)

Menurut Thomas Davenport dan Laurence *Knowledge management system* merupakan perpaduan dari pengalaman, nilai, informasi, ppenggunaan pakar dan intuisi mendasar yang memberikan suatu lingkungan dan kerangka untuk mengevaluasi dan menyatukan solusi baru dengan informasi. Di perusahaan *Knowledge* sering terkait tidak hanya pada dokumen atau tempat penyimpanan peralatan berharga, tetapi juga pada rutinitas, proses, praktek dan norma perusahaan. Berdasarkan definisi tersebut diatas, *Knowledge* menjadi sangat penting dengan alasan sebagai berikut:

1. *Knowledge* adalah aset perusahaan, yang menentukan jenis barang/alat, tenaga kerja, Informasi, ketrampilan dan struktur yang diperlukan.
2. Pengetahuan dan pengalaman perusahaan merupakan sumber daya yang berkelanjutan (*sustainable resources*) dari keuntungan daya saing kompetitif (*competitive advantages*) dibandingkan dengan produk penggunaan dan teknologi terancang yang dimiliki.
3. Pengetahuan dan pengalaman mampu menciptakan, mengkomunikasikan dan mengsystemkan pengetahuan mengenai semua hal terkait untuk mencapai tujuan bisnis.

Pendekatan lainnya mendefinisikan *Knowledge* dalam 4 level operasional sebagai berikut:

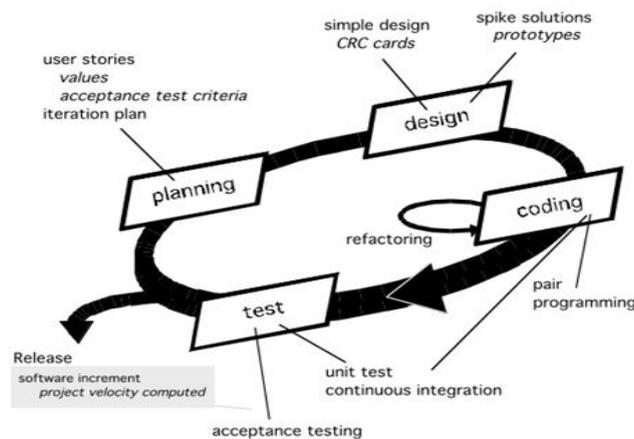
- a. *Know what* atau *cognitive Knowledge* merupakan *Knowledge* yang diperoleh melalui pelatihan, pembelajaran dan kualifikasi formal. Level ini sangat penting bagi perusahaan namun umumnya masih kurang mencukupi bagi keberhasilan komersial.
- b. *Know how* - merupakan level sistem praktis
 Pada level ini apa yang telah didapat pada level I diterjemahkan dalam pelaksanaan. Pada tahap ini merupakan area dimana *Knowledge* menambahkan nilai dalam suatu organisasi melalui kemampuan untuk menterjemahkan *Knowledge* yang bersifat teoritis menjadi eksekusi yang efektif.
- c. *know why* disebut juga *system understanding*
 Merupakan *Knowledge* terdalam dari jaringan hubungan sebab akibat yang ada pada suatu disiplin ilmu. Level ini memungkinkan profesional untuk berpindah dari pelaksanaan kerja ke pemecahan masalah yang lebih besar dan kompleks dan menciptakan solusi baru bagi permasalahan yang baru.
- d. *Care why* - tahap lanjutan dari kreativitas diri (*self-motivated creativity*)
 Merupakan level dimana inovasi radikal dapat terjadi melalui lompatan imajinatif dan pemikiran lateral.



Gambar 2.1 Level Operasional dari Definisi *Knowledge*

2.6. Metode Extreme Programming (XP)

Menurut Pressman (2012) *extreme programming* adalah pendekatan *extreme programming* merupakan suatu pendekatan berorientasi objek dan sebagai pengembangan perangkat lunak cepat sedikit lebih rinci dengan tujuan memberikan ulasan secara ringkas. Pengembangan *extreme programming* dapat dilihat menggunakan suatu alur tahapan pengembangan yang dapat dilihat pada gambar 2.2 :



Gambar 2.2 *Extreme Programming*

Sumber: (Pressman, 2012)

Berdasarkan tahapan tersebut merupakan suatu paradigma yang diinginkan mencakup didalam seperangkat aturan dan praktik-praktik dalam empat konteks kegiatan kerangka kerja yaitu :

1. *Planning* (Perencanaan)

Perencanaan yang bertujuan awal untuk mendengarkan kebutuhan secara fungsional dan non fungsional serta membuat penggambaran sistem secara *interface* sebagai keperluan *outputnya*. Pada ahapan ini dimulai

dengan mendengarkan kumpulan kebutuhan aktifitas suatu sistem yang memungkinkan pengguna memahami proses untuk sistem dan mendapatkan gambaran yang jelas mengenai fitur utama, fungsionalitas dan keluaran yang diinginkan. Dalam pengembangan sistem informasi manajemen aset berbasis android, pada tahapan ini dimulai dari mengidentifikasi permasalahan yang timbul pada sistem sebelumnya, kemudian dilakukan analisa kebutuhan pengguna terhadap sistem yang akan dikembangkan.

2. *Design* (Perancangan)

Penggunaan rancangan bertujuan untuk membangun dan menggambarkan sistem yang akan dibuat guna mempermudah proses pengkodean. Pada tahapan ini perancangan dilakukan pembuatan pemodelan sistem berdasarkan hasil analisa kebutuhan yang didapatkan. Selain itu dibuatkan juga pemodelan basis data untuk menggambarkan hubungan antar data dan *adobe xd* sebagai pendukung dalam membuat desain sistem.

3. *Coding* (Pengkodean)

Pengkodean bertujuan untuk membuat hasil perancangan dengan melakukan konstruksi atau pembuatan sistem secara seluruh menggunakan kode. Tahapan ini merupakan implementasi dari perancangan model sistem yang telah dibuat kedalam kode program yang menghasilkan prototipe dari perangkat lunak. Dalam pengembangan sistem informasi manajemen aset berbasis android menggunakan bahasa pemrograman mobile. Untuk implementasi basis data, *Database Management System* yang digunakan adalah *Firestore Database Platform*.

4. *Testing* (Pengujian)

Pengujian dilakukan untuk menguji hasil pengkodean dan selanjutnya dilakukan evaluasi dan implementasi. Pada tahapan ini merupakan tahapan pengujian terhadap sistem yang sudah dibangun, pada tahapan ini ditentukan oleh pengguna sistem dan berfokus pada fitur dan fungsionalitas dari keseluruhan sistem kemudian ditinjau oleh pengguna sistem. Metode yang digunakan dalam melakukan pengujian terhadap sistem web seleksi peserta pelatihan kerja adalah *Black-Box Testing*

dengan melakukan pengujian terhadap masukan dan keluaran yang dihasilkan sistem.

5. *Software Increment* (Peningkatan Perangkat Lunak) .

Tahapan ini merupakan tahap pengembangan sistem yang sudah dibuat secara bertahap yang dilakukan setelah sistem diterapkan dalam organisasi dengan menambahkan layanan atau konten yang mengakibatkan bertambahnya kemampuan fungsionalitas dari sistem.

2.6.1. Kelebihan Model *Extreme Programming*

Komunikasi dalam XP dibangun dengan melakukan pemrograman berpasangan (*pair programming*). *Developer* didampingi oleh pihak klien dalam melakukan *coding* dan *unit testing* sehingga klien bisa terlibat langsung dalam pemrograman sambil berkomunikasi dengan *developer*. Selain itu perkiraan beban tugas juga diperhitungkan.

1. Menekankan pada kesederhanaan dalam pengkodean: “*What is the simplest thing that could possibly work?*” Lebih baik melakukan hal yang sederhana dan mengembangkannya besok jika diperlukan. Komunikasi yang lebih banyak mempermudah, dan rancangan yang sederhana mengurangi penjelasan.
2. Setiap *feed back* ditanggapi dengan melakukan tes, *unit test* atau *system integration* dan jangan menunda karena biaya akan membengkak (uang, tenaga, waktu).
3. Banyak ide baru dan berani mencobanya, berani mengerjakan kembali dan setiap kali kesalahan ditemukan, langsung diperbaiki.

2.6.2. Kelemahan Model *Extreme Programming*

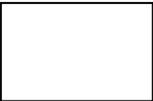
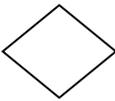
Pengembangan *extreme programming* yang digunakan tentunya masih memiliki beberapa kelemahan atau kekurangan seperti berikut.

1. *Developer* harus selalu siap dengan perubahan karena perubahan akan selalu diterima.
2. Tidak bisa membuat kode yang detail di awal (*prinsip simplicity* dan juga anjuran untuk melakukan apa yang diperlukan hari itu juga).

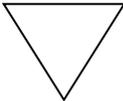
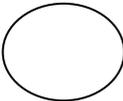
2.7. Bagan Alir Dokumen

Bagan Alir Dokumen (*Dokument Flowchart*) Rosa & Salahuddin (2013) adalah Salah satu alat dan teknik yang digunakan dalam pengembangan sistem informasi, dokument *flowchart* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusannya. Simbol-simbol yang dipergunakan dalam pembuatan bagan alir dokumen dapat dilihat pada tabel 2.2 :

Tabel 2.2 Simbol-simbol dalam Bagan Alir Dokumen

No.	Simbol	Keterangan
1.		Simbol <i>Offline Connector</i> Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang berbeda.
2.		Simbol <i>Manual</i> Untuk menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer.
3.		Simbol <i>Decision/Logika</i> Untuk menunjukkan suatu kondisi tertentu yang menghasilkan dua kemungkinan jawaban, ya/tidak.
4.		Simbol <i>Predefined Proses</i> Untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
5.		Simbol <i>Terminal</i> Untuk menyatakan permulaan atau akhir suatu program.

Tabel 2.2 Simbol-simbol dalam Bagan Alir Dokumen (Lanjutan)

6.		Simbol Keying Operation Untuk menyatakan segala jenis operasi yang diproses.
		Simbol Off-Line Storage Untuk menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu.
8.		Simbol Manual Input Untuk memasukkan data secara manual dengan menggunakan <i>online keyboard</i> .
9.		Simbol Input-Output Untuk menyatakan proses input dan <i>output</i> tanpa tergantung dengan jenis peralatanya.
10.		Simbol Punched Card Untuk menyatakan input berasal dari kartu atau <i>output</i> ditulis ke kartu
11.		Simbol Magnetic-tape Unit Untuk menyatakan input berasal dari pita magnetic atau <i>output</i> disimpan ke pita magnetic.
12.		Simbol Disk storage Untuk menyatakan input berasal dari disk atau <i>output</i> disimpan ke <i>disk</i> .
13.		Simbol Document Untuk mencetak laporan ke printer.
14.		Simbol Display Untuk menyatakan peralatan <i>output</i> yang digunakan berupa layar (video, komputer).

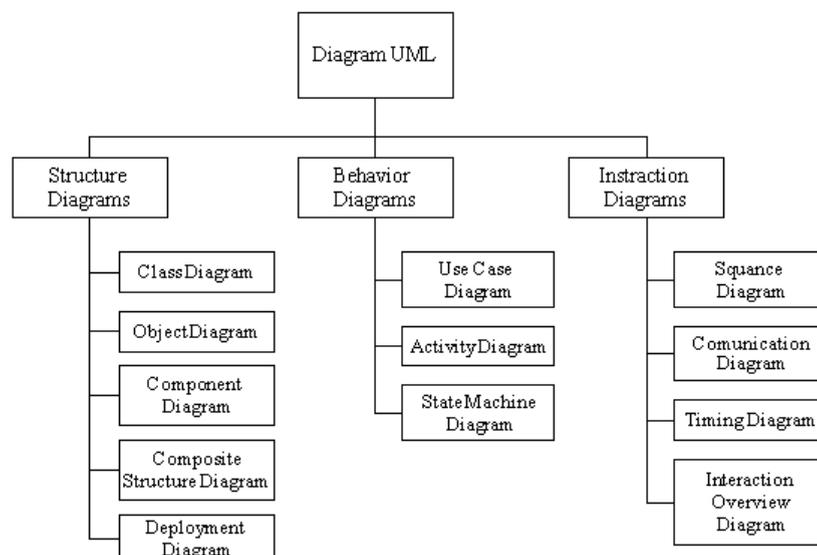
Tabel 2.2 Simbol-simbol dalam Bagan Alir Dokumen (Lanjutan)

15.		<p>Simbol <i>Connector</i></p> <p>Untuk menyatakan sambungan dari satu proses ke proses dalam halaman sama.</p>
-----	---	--

2.8.UML (*Unified Modelling Language*)

UML adalah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan *artifacts* (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, *artifact* tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya .

Menurut Rosa & Salahuddin (2013), UML (*Unified Modelling Language*) adalah bahasa visual untuk pemodelan serta komunikasi dalam sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. Berikut adalah bagan UML pada Gambar 2.3.

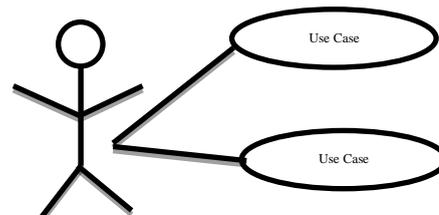


Gambar 2.3 Bagan UML
(Sumber : Rosa & Salahuddin, 2013)

Berikut ini merupakan penjelasan tentang masing-masing diagram yang ada pada UML (*Unified Modelling Language*).

2.8.1. Use Case Diagram

Use Case adalah *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 (Salahuddin, 2013). Berikut adalah konsep *Use Case Diagram* pada Gambar 2.4



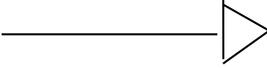
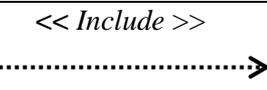
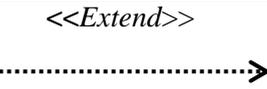
Gambar 2.4 *Use Case Diagram*

Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Use Case Diagram* dapat dilihat pada tabel 2.3 berikut ini:

Tabel 2.3 Simbol *Use Case Diagram*

No	Simbol	Deskripsi
1.		<i>Use case</i> : Fungsionalitas yang sudah disediakan pada sistem sebagai unit yang saling bertukar atau memberikan pesan antar unit atau aktor, biasanya dinyatakan menggunakan kata kerja di awal <i>frase</i> nama <i>use case</i> .
2.		Aktor: seseorang/karakter yang berinteraksi dengan yang akan dibuat. diluar sistem informasi. Biasanya dinyatakan menggunakan kata benda
3.		Asosiasi (<i>association</i>): merupakan komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.

Tabel 2.3 Simbol *Use Case Diagram* (Lanjutan)

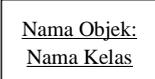
4.		Generalisasi (<i>generalization</i>): merupakan hubungan (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum
5.		<i>Include</i> berarti <i>use case</i> yang ditambahkan akan dipanggil saat <i>use case</i> tambahan dijalankan.
6.		Ekstensi (<i>extend</i>) merupakan <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.

Sumber: (Rosa & Salahuddin, 2013)

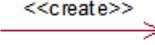
2.8.2. *Sequence Diagram*

Menurut Rosa dan M. Salahuddin (2013), diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *message* yang dikirim dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlihat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Berikut simbol-simbol yang akan digunakan dalam menggambarkan *Sequence Diagram* dapat dilihat pada tabel 2.4 berikut ini:

Tabel 2.4 Simbol *Sequence Diagram*

No	Simbol	Nama	Deskripsi
1		Aktor	Orang, proses, atau sistem yang berinteraksi dengan sistem informasi lain yang dibuat di luar dari sistem informasi itu sendiri.
2		Garis Hidup	Menyatakan kehidupan suatu objek.
3		Objek	Menyatakan objek yang berinteraksi dengan pesan.

Tabel 2.4 Simbol *Sequence Diagram* (Lanjutan)

4		Waktu Aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
5		Pesan Tipe Create	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.

Sumber: (Rosa & Salahuddin, 2013)

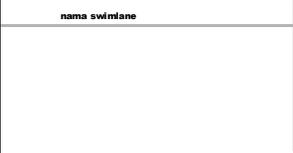
2.8.3. Activity Diagram

Activity Diagram menggambarkan rangkaian aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Aktivitas-aktivitas yang dikerjakan oleh sistem, bukan aktivitas yang dikerjakan oleh aktor (Rosa & Salahuddin, 2013). Simbol yang menggambarkan *activity diagram* dapat dilihat pada tabel 2.5 berikut ini :

Tabel 2.5 Simbol Activity Diagram

No	Simbol	Nama	Deskripsi
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 / Decision	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.

Tabel 2.5 Simbol *Activity Diagram* (Lanjutan)

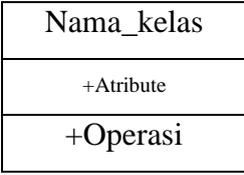
4		Penggabungan / Join	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5		Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6		Swimlane	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber: (Rosa & Salahuddin, 2013)

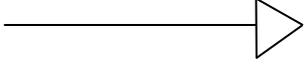
2.8.4. *Class Diagram*

Menurut Rosa & Salahuddin (2013) *Class Diagram* adalah *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 2.6 berikut ini:

Tabel 2.6 Simbol *Class Diagram*

No.	Simbol	Deskripsi
1.		Kelas pada struktur sistem.
2.	<p>Antar Muka/<i>Interface</i></p>  <p>Nama_<i>Interface</i></p>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.

Tabel 2.6 Simbol *Class Diagram* (Lanjutan)

3.	Asosiasi / <i>Association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan symbol
4.	Asosiasi Berarah / <i>Directed Association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan symbol.
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
7.	Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan maksna semua bagian (<i>whole-part</i>)

Sumber: (Rosa & Salahuddin, 2013)

2.9. *Adobe xd*

Adobe xd adalah alat desain berbasis vektor untuk membuat desain situs website dan sistem. Alat ini dikembangkan dan diterbitkan oleh Inc untuk pengguna Mac dan Windows. Dengan *Adobe xd*, pengguna juga dapat membuat prototype, mockup, atau full design untuk situs website serta sistem seluler.

Setelah mengetahui apa itu *Adobe xd* berikut beberapa fitur di dalamnya. Perlu diketahui, bahwa *Adobe xd* secara rutin mengupdate dan peningkatan pada fitur-fitur yang disediakan. Berikut beberapa pembaruan fitur terbaru yang telah disediakan :

1. *Offline support* untuk *coediting*

Saat ini sudah banyak perusahaan yang menerapkan sistem *remote*, sehingga kemampuan untuk berkolaborasi secara real time sangat diperlukan. Oleh karena itu, *Adobe xd* mempunyai dukungan offline untuk *co-publishing*. ini memudahkan pengguna untuk mengatur desain dengan tim secara *online* atau *offline*.

2. Memperbesar dan memperkecil link *prototype*

Jadi pengguna harus memiliki pemahaman yang baik tentang prototipe desain. memungkinkan pengguna untuk memperbesar atau memperkecil

bagian tertentu dari sebuah desain pada pengembangan sistem sehingga pengguna dapat melihat desain yang telah dibuat. Fitur ini tersedia untuk browser desktop atau seluler.

Dengan menggunakan *Adobe xd* kita dimudahkan dalam pembuatan desain *user interface* karena *Adobe xd* sudah menyediakan *tools* yang dapat memudahkan dalam membuat desain *prototyping* sistem yang akan kita buat. *Software* ini berfokus pada konten yang ingin digambar dan fungsionalitas yang dibutuhkan oleh pengguna.

2.9.1. Keunggulan & kekurangan *Adobe xd*

1.Keunggulan *Adobe xd*

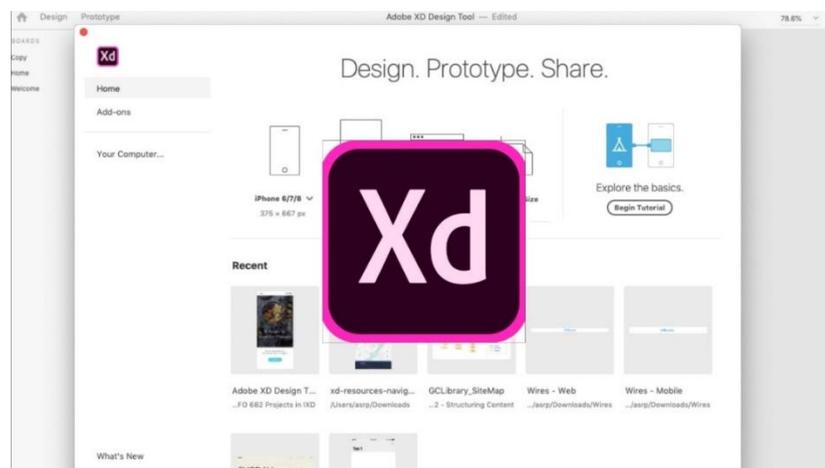
Adobe xd tentunya memiliki beberapa keunggulan yang sangat berguna bagi para pengembangan sistem berbasis android, karena dapat memberikan alur kerja dan kenyamanan yang lebih baik lagi. Keuntungan berikut dapat dirasakan:

- a. *Prototyping tools* : Dengan *tools* ini, desainer sistem mobile dapat membuat alur kerja dan ketertiban yang lebih baik dan lebih detail sehingga tidak saling mengganggu dalam proyek desain lain dan dipertahankan dengan cara tersendiri.
- b. *Focused Design Tools* : Alat ini digunakan oleh desainer sistem seluler untuk membuat bingkai, maket tata letak layar, dan produksi untuk desain lainnya.
- c. *Photoshop, Sketch, Illustrator aset desain* : Adobe diketahui memiliki beberapa program lain yang juga digunakan untuk desain, seperti *Photoshop, Sketch, dan Illustrator*. Untungnya, platform ini dapat menggabungkan aset desain perangkat lunak ke dalam proyek dan membuat segalanya lebih mudah.
- d. *Berbagi Terintegrasi* : Alat ini memungkinkan pengguna dalam merekam video selama proses desain untuk ditinjau oleh desainer lain menggunakan fitur *Creative Cloud*.
- e. *Performa cepat* : Tentu saja, performa cepat disediakan agar pengguna dapat menyelesaikan pekerjaan tanpa penundaan.

2. Kekurangan Adobe xd

Sedangkan kelebihan Adobe xd sangat menggiurkan, ada kekurangan yang mungkin Pengguna rasakan saat menggunakannya, seperti:

- a. Animasi antarmuka pengguna yang sulit : Prototype menunjukkan apa yang ditampilkan di sistem seluler, tetapi Adobe SD tidak memiliki alat animasi yang kompatibel dengan Adobe xd sehingga ini bisa sulit.
- b. Tidak ada ekspor CSS : Ekspor CSS dapat membuat pekerjaan desainer menjadi lebih mudah dan lebih cepat. Namun, fungsi ini tidak disertakan. Meskipun platform *Photoshop* sendiri sudah menggunakan plugin Zeplin. Dibawah ini merupakan tampilan *adobe xd* pada gambar 2.5 berikut ini :



Gambar 2.5 Adobe xd

2.10 Java

Kurniawan (2016:3) "Java adalah bahasa pemrograman yang dapat digunakan di banyak komputer, termasuk ponsel. Dikembangkan oleh Sun Microsystems dan dirilis pada tahun 1995.

Sedangkan menurut Sukanto dan Salahuddin (2013:103) "Java adalah nama yang diberikan untuk teknologi yang digunakan untuk membuat dan menjalankan perangkat lunak pada komputer mandiri atau dalam lingkungan jaringan".

Java adalah mesin juru bahasa yang disebut Java Virtual Machine (JVM). JVM membaca bytecode dari file .class program sebagai representasi langsung dari program yang berisi bahasa mesin. Inilah mengapa Java disebut sebagai bahasa pemrograman portable karena dapat berjalan di beberapa sistem operasi asalkan sistem operasi tersebut memiliki JVM.

Java adalah bahasa pemrograman yang biasa digunakan untuk mengembangkan perangkat lunak, sistem Android, dan situs web back-end.

Java juga dikenal dengan motonya "Tulis sekali, jalankan di mana saja". Ini berarti bahwa Java dapat berjalan di banyak platform tanpa harus menyesuaikannya dengan platform tersebut. Misalnya di Android, Linux, Windows dan sistem operasi lainnya.

2.11 Android Studio

Android Studio merupakan perangkat lunak IDE (*Integrated Development Environment*) atau sebuah Lingkungan dalam Pengembangan Terpadu yang digunakan untuk mengembangkan sistem menjadi sistem *Android*. *Android Studio* menggunakan bahasa pemrograman Java dan Kotlin sebagaimana yang digunakan pada sistem operasi *Android*. Sebagai editor kode IntelliJ dan alat pengembang yang serba guna, *Android Studio* menggantikan IDE pendahulunya yaitu Eclipse ADT (*Android Development Tools*). berikut adalah tampilan awal *Android Studio* pada Gambar 2.6.



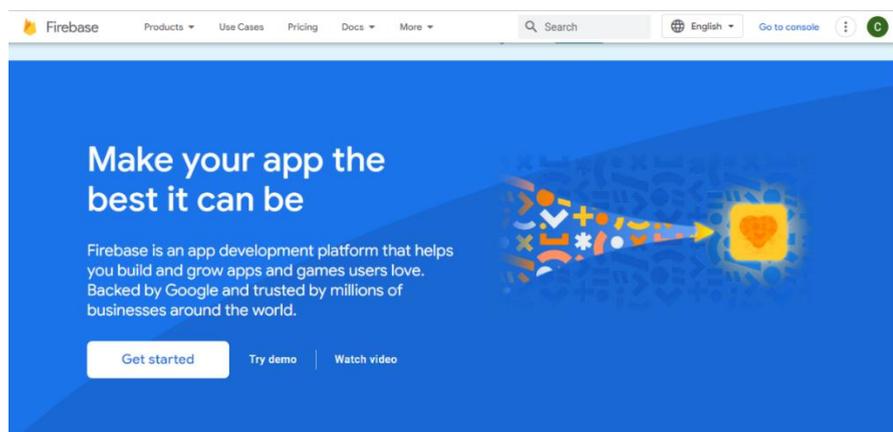
Gambar 2.6 *Android Studio*

2.12 Firebase Database Platform

Pengertian *Firebase* adalah suatu stage dari Google berupa database berbasis NoSQL yang kini mulai banyak dikenal dan digunakan oleh para designer. Hal tersebut tak mengherankan sebab equipment layanan *Firebase* yang cukup sederhana terbukti mampu mempermudah sekaligus mempercepat pekerjaan mereka.

Fungsi *firebase* disini memberi para designer dan developer alat yang ideal dengan layanan untuk membantu mengembangkan sistem, menumbuhkan source pengguna, dan mendapatkan keuntungan. Definisi lain dari *Firebase* adalah platform atau stage dari Google yang menyediakan berbagai layanan untuk memudahkan pengembangan sistem web, Android, iOS, maupun Unity dalam membangun sistem game. Platform ini jugamampu mempermudah developer dalam pengembangan sistem bersifat Realtime Database dan petugas friendly. Untuk menjalankan platform tersebut, fungsi *firebase* juga memerlukan akses internet karena tempat penyimpanannya berbasis online.

pengertian *Firebase* sendiri ialah *Backend-as-a-Service* (Baas) yang berguna mempercepat pekerjaan para designer ataupun developer. Dengan *Firebase*, designer bisa fokus mengembangkan sistem tanpa harus memberikan usaha yang besar. Mulanya *Firebase* berbentuk database yang tidak terbatas pada sebuah komputer fisik.berikut adalah tampilan pertama dari *Firebase* pada Gambar 2.7.



Gambar 2.7 *Firebase Database Platform*

2.13 Pengujian (*Testing*)

Menurut (Hetzel) testing merupakan aktivitas yang digunakan untuk dapat melakukan evaluasi suatu parameter ataupun kemampuan dari program atau sistem serta memastikan apakah sudah memenuhi kebutuhan ataupun hasil yang diharapkan. Untuk memastikan kualitas mutu sistem testing harus dilakukan yaitu dengan menguji apakah sistem data yang dihasilkan sesuai dengan testing yang telah dilakukan untuk memastikan kualitas (*quality assurance*) Proses menganalisa sesuatu entitas sistem untuk mendeteksi perbandingan antara keadaan yang terdapat dengan keadaan yang diinginkan (*defect*) serta mengevaluasi fitur-fitur dari entitas *software*.

Menurut Yakub (2012), pengujian sistem adalah proses untuk memeriksa apakah perangkat lunak yang dihasilkan dapat berjalan sesuai dengan pengguna atau tidak. Pengujian sistem dapat menggunakan metode pengujian *black box*, yang merupakan pendekatan pelengkap untuk teknik pengujian kotak putih, karena pengujian *black box* dapat mengungkapkan kesalahan yang lebih besar. Pengujian *black box* berfokus pada pengujian persyaratan fungsional perangkat lunak untuk mendapatkan serangkaian kondisi input yang sesuai dengan persyaratan fungsional suatu program. Jadi, pada pengujian sistem dapat dilakukan dengan cara pengecekan input, *output* dan pengecekan proses lainnya sebagai berikut:

- a. Pengecekan input, yang meliputi perlengkapan item input, kemudahan dalam pengoperasian, kemudahan dalam memanipulasi data, serta pengendalian kesalahan.
- b. Pengecekan pemroses, yang dilakukan dengan cara pengecekan *output* program.
- c. Pengecekan *output*, yang meliputi pengecekan terhadap format serta bentuk laporan.

Kelebihan pada *Black box Testing* sebagai berikut:

- a. Pengguna tidak perlu melihat *source code* secara detail.
- b. Dapat mendeteksi kesalahan dalam pengetikan (*typo*).
- c. Dapat mendeteksi kesalahan *Design* atau *User interface* dari sebuah *software* ataupun dari *website*.

- d. Dapat menampilkan asumsi yang tidak sesuai nantinya akan di analisa dan diperbaiki.
- e. Seorang *Tester* tidak harus *Programmer*.

Kekurangan *Black box Testing* adalah:

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

2.13.1 Skala Likert

Menurut Sugiyono (2018), skala tes *Likert* adalah skala *Likert* yang mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Ketika menyelidiki fenomena sosial ini, secara khusus ditentukan oleh peneliti, yang selanjutnya disebut variabel penelitian. Dengan skala *Likert*, variabel yang akan diukur dijabarkan menjadi variabel indikator.

Variabel indikator kemudian menjadi titik tolak penyusunan unsur-unsur instrumen, yang dapat berupa pernyataan atau pertanyaan. Pengujian menggunakan 5 kategori respon dengan bobot yang berbeda untuk setiap respon, seperti terlihat pada Tabel 2.7 dan keterangan pada Tabel 2.8.

Tabel 2.7 Bobot Jawaban *Usability*

No.	Pertanyaan	Jawaban				
		SS	S	TP	TS	STS
1.	apakah sistem ini mudah digunakan?	5	4	3	2	1
2.	Bagaimana untuk tampilannya ?	5	4	3	2	1

Tabel 2.8 Keterangan Skala *Likert*

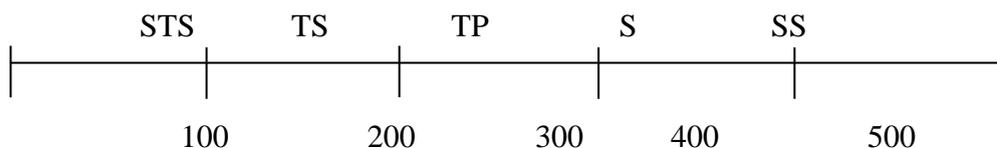
Keterangan Skala	Skor
Sangat Tidak Setuju (STS)	1
Tidak Setuju (TS)	2
Tidak Pasti(TP)	3
Setuju (S)	4
Sangat Setuju (SS)	5

Dengan menggunakan skala likert 5 (lima) data yang dinilai, yaitu data yang dikumpulkan, dianalisis dan dilaporkan menurut frekuensinya. Dalam merangkum keputusan pengambilan keputusan dari analisis data bersama dari tiga kategori yang ditunjukkan pada tabel 2.9.

Tabel 2.9 Pemingkatan Skala *Likert*

Keterangan Skala 5 Peringkat	Skala 3 Peringkat
1 (STS)	Tidak Setuju
2 (TS)	
3 (TP)	Tidak Pasti
4 (S)	Setuju
5 (SS)	

Secara hasil kesimpulan dari hasil kuisisioner yang didapat menggunakan pengukuran dari hasil maksimal nilai tertinggi, berikut merupakan gambar aspek *usability* pada Gambar 2.8.



Gambar 2.8 Pengukuran Aspek *Usability*

Rumus skala *likert* untuk menghitung presentasi cukup sederhana, yakni dengan menjumlahkan seluruh poin yang diterima dari pengujian sistem, kemudian dibagi dengan poin maksimum. Nantinya hasil dari perhitungan yang didapatkan dari angket, berikutnya melakukan perbandingan dengan rentang kriteria interpretasi skor untuk menyatakan hasil yang didapatkan dengan rentang pada tabel 2.10.

Tabel 2.10 Rentang *Criteria Interpretasi*

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

(Sumber: Sugiyono, 2013)