BAB II
LANDASAN TEORI

2.1. Tinjauan Pustaka

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

Tabel 2. 1 Tinjauan Pustaka

No	Nama (Tahun)	Judul	Metode	Hasil
1	Lestanti and Susana (2016),	Sistem Pengarsipan Dokumen Guru Dan Pegawai Menggunakan Metode Mi*xture Modelling* Berbasis Web	Metode *Mixture Modelling*	Hasil yang diharapkan dari penelitian adalah aplikasi pengelolaan data kearsipan yang dapat menghasilkan output seperti yang diharapkan dan proses pengarsipan dokumen dapat dilakukan dengan lebih baik, cepat, dan mudah.
2	Prastika, Supriady and Setiawan (2019)	Sistem Pengelolaan Berkas Permohonan Dana Kegiatan Menggunakan Metode *Mixture Modelling* Berbasis Web	Metode *Mixture Modelling*	Hasil penelitian ini adalah sistem pengelolaan berkas permohonan dana kegiatan yang dikelola secara cepat
3	Kirana and Etisa (2017)	Penerapan Mixture Model Kepada Aplikasi Helpdesk Berbasis Web	Metode *Mixture Modelling*	Penggunaan aplikasi helpdesk adalah untuk mengelola data keluhan atau pun permasalahan

				yang dimiliki oleh pelanggan, sehingga membutuhkan penyedia jasa. Karena
4	Agusta (2005)	Mixture Modelling Menggunakan Prinsip Minimum Message Length	Metode *Mixture Modelling*	Di dalam pengembangan metode penganalisaan data berbasis mixture modelling ini, prinsip Minimum Message Length (MML) dapat diaplikasikan secara bersamaan baik dalam pengestimasian parameter dan pencarian model. Di dalam tulisan ini, metode mixture modelling untuk penganalisaan data kontinyu univariate dan multivariate tidak berkorelasi akan dikembangkan. Metode ini juga dilengkapi dengan fasilitas untuk mengamati keberadaan outliers di dalam kelompok-kelompok yang dihasilkan

5	Pratama (2018)	Dari Dutch Manual Hingga Records In Contexts: Perubahan Dan Kesinambungan Prinsip-Prinsip Kearsipan	Metode *Mixture Modelling*	Hasil penelitian ini menguraikan perubahan dan kesinambungan prinsip tersebut sejak Dutch Manual, beragam standar deskripsi nasional, dan Records in Contexts, pedoman tunggal dan terbaru itu. Kata

Berdasarkan penelitian diatas maka dapat dibedakan dengan peneliti sebelumnya sebagai berikut :

1. Metode pengembangan sistem menggunakan *prototype*
2. metode perancangan sistem menggunakan UML
3. sistem yang dibangun dapat melakukan pencarian berdasarkan waktu pengarsipan dokumen
4. sistem yang dibangun dapat mencetak laporan data dokumen yang telah diarsipkan secara priode.
5. Implementasi sistem menggunakan bahasa pemprograman PHP dan *MySQL* sebagai *database.*
6. Pengujian menggunakan menggunakan iso 9126 dan dengan menggunakan karakteristik pengujian aspek *functionality, usability, realibility,* dan *efficiency.*

2.2. Arsip

Berdasarkan Undang-Undang nomor 43 tahun 2009, arsip adalah rekam kegiatan atau peristiwa dalam berbagai bentuk dan media sesuai dengan perkembangan teknologi informasi dan yang dibuta dan diterima oleh lembaga Negara, pemerintahan daerah, lembaga pendidikan, perusahaan, organisasi politik, organisasi kemasyarakatan, dan perseorangan dalam pelaksanaan kehidupan bermasyarakat, bangsa, dan Negara Indonesia.

2.3. Dokumen

Menurut Gottschalk (2014) bahwa dokumen (dokumentasi) dalam pengertiannya yang lebih luas berupa setiap proses pembuktian yang didasarkan atas jenis sumber apapun, baik itu yang bersifat tulisan, lisan, gambaran, atau arkeologis.

Menurut Kosim.E (2009) dokumen merupakan sumber data tertulis, maka terbagi dalam dua kategori yaitu sumber resmi dan tidak resmi.

2.4. Pengertian *Mixture Modeling*

Mixture modelling (*Mixture Modeling* atau *Mixture Model*) adalah suatu metode yang memodel atau mengelompokkan data-data di dalam suatu dataset menjadi kelompok-kelompok data yang sebelumnya tidak terdifinisikan. Metode yang diulas adalah pengelompokan data yang memodel suatu distribusi statistik bercampur dengan distribusi statistik yang lain dalam bentuk mixture (penjumlahan berproporsi) (Lestanti and Susana, 2016).

2.4.1. Jenis Mixture Model

Terdapat dua jenis mixture model, yaitu *direct* dan *indirect*. Keadaan finansial yang berubah-ubah pada contoh diatas adalah contoh aplikasi direct dari mixture model, sebuah keadaan dimana setiap observasi memberikan suatu komponen atau kategori yang berbeda. Dalam bentuk *mixture*, setiap komponen digambarkan oleh sebuah *probability density function* dan nilai *mixture* yang adalah probabilitas dari setiap komponen (Lestanti and Susana, 2016).

2.4.2. Kelebihan dan Kekurangan *Mixture modelling*

Kelebihan *mixture modelling* yaitu :

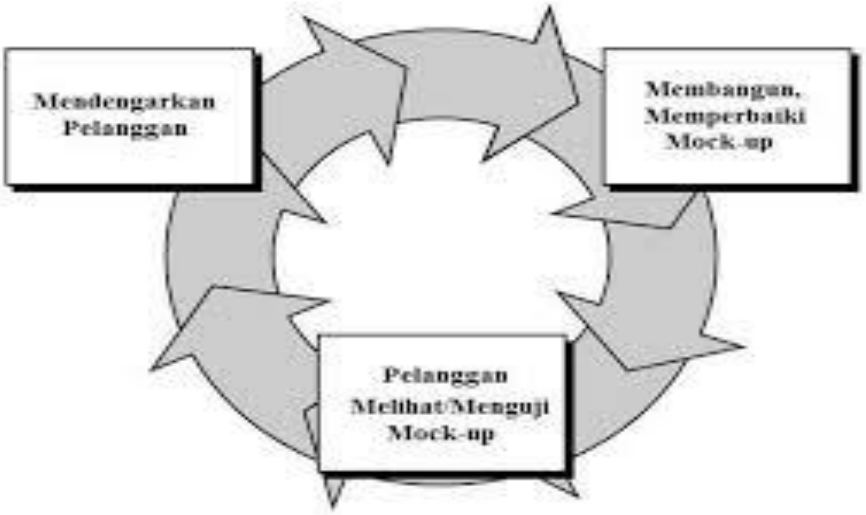
a. Dapat menganalisa data kontinyu dan tidak berkolerasi untuk pengembangan
b. Dilengkapi oleh fasilitas untuk pengamatan keberadaan didalam kelompok
c. Dapat diaplikasikan secara bersama

Kekurangan *mixture modelling* yaitu :

a. Jumlah relative sesuai kebutuhan data
b. Hanya menghasilkan pada kompok yang dilakukan
c. Penentuan kelompok menjadi masalah yang cukup kompleks.

2.5. Pengembangan Sistem *Prototype*

Menurut Rosa dan Shalahuddin (2018) menyatakan bahwa : Model prototipe dapat digunakan untuk menyambung ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak. Adapun tahapan – tahapan dalam metode *Prototype* dapat dilihat pada gambar 2.1:



Gambar 2. 1 Tahapan – Tahapan Metode *Prototype*

Sumber : Rosa dan Shalahuddin (2018)

Menurut Rosa dan Shalahuddin (2018) menyatakan bahwa : Pada metode *prototype* terdapat tiga tahap yaitu :

1. Mendengarkan Pelanggan

   Pada tahap ini, dilakukan pengumpulan kebutuhan dari sistem dengan cara mendengar kebutuhan pelanggan sebagai pengguna sistem perangkat lunak untuk menganalisis serta mengembangkan kebutuhan pengguna.

2. Merancang dan Membuat *Prototype*

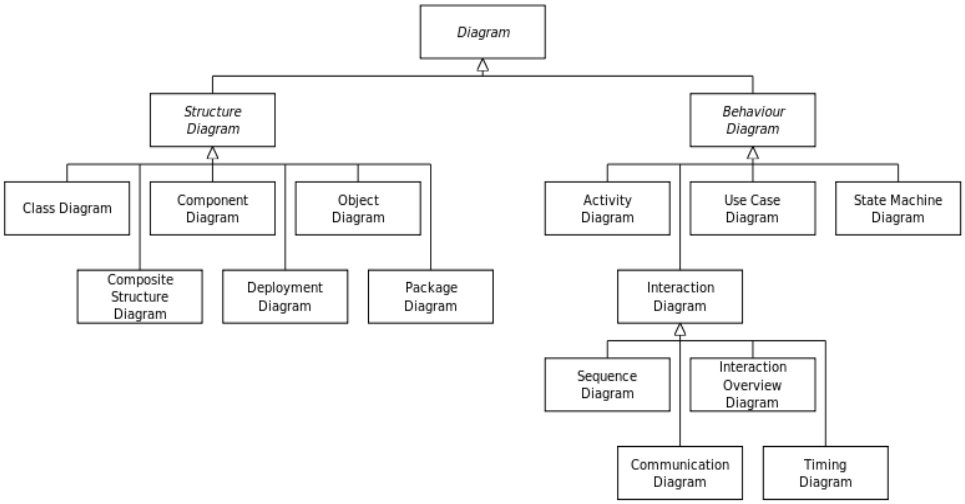
   Pada tahap ini, dilakukan perancangan dan pembuatan *prototype* sistem yang disesuaikan dengan kebutuhan pengguna.

3. Uji Coba

Pada tahap ini, dilakukan pengujian *prototype* sistem oleh pengguna kemudian dilakukan evaluasi sesuai dengan kekurangan-kekurangan dari kebutuhan pelanggan. Jika sistem sudah sesuai dengan *prototype*, maka sistem akan diselesaikan sepenuhnya.

2.6. Pengertian UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (A.S Rosa & Shalahuddin, 2018). Struktur UML dapat dilihat pada gambar berikut.



Gambar 2.6.2 UML (*Unified Modeling Langue*)

1. *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. simbol-simbol yang ada pada diagram kelas pada Tabel 2.2 di bawah ini:

Tabel 2. 2 Simbol *Class Diagram*

Simbol	Deskripsi
Kelas **nama_kelas** +atribut +operasi()	Kelas pada struktur sistem
Antarmuka/*Interface* **nama_interface**	Sama dengan konsep *interface* dalam pemrograman berorientasi objek
Asosiasi/*asociation*	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*
Asosiasi berarah/*directed association*	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan *multiplicity*
Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi(umum khusus)
Kebergantungan/*dependecy*	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi/*agregation*	Relasi antar kelas dengan makna semua bagian (*whole-part*)

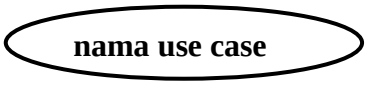
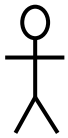
Sumber : (A.S Rosa & Shalahuddin, 2018)

2. *Use Case Diagram*

Use case diagram atau diagram *use case* merupakan pemodelan untuk kelakuan *(behavior)* sistem informasi yang akan dibuat (A.S Rosa & Shalahuddin, 2018).

Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat simbol-simbol yang ada pada diagram *use case* dapat dilihat pada Tabel 2.3 di bawah ini:

Tabel 2. 3 Simbol diagram *use case*

Simbol	**Deskripsi**
Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama *use case*
Aktor/*actor* 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama actor
Asosiasi/*association* 	Komunikasi antara aktor dan *use case* yang berpatisipasi pada *use case* atau *use case* memiliki interaksi dengan actor

Ekstensi/*extend* <<*extend*>> 	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu mirip dengan prinsip *inheritance* pada pemrograman berorientasi objek biasanya *use case* tambahan memiliki nama depan
Generalisasi/***generalization*** 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan/*Include/uses* <<*include*>> 	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini

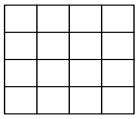
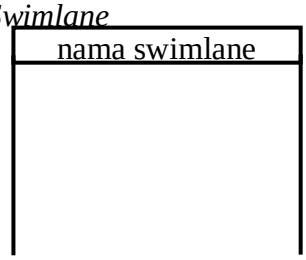
Sumber: (A.S Rosa & Shalahuddin, 2018)

3. *Activity Diagram*

Activity diagram atau Diagram aktivitas menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Simbol-simbol yang ada pada *activity diagram* dapat dilihat pada Tabel 2.4 berikut :

Tabel 2. 4 Simbol *Activity Diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem,sebuah diagram aktivitas memiliki sebuah status awal.

Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/*decision* 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan/*join* 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Tabel 	Suatu file komputer dari mana data bisa dibaca atau direkam selama kejadian bisnis
Dokumen 	Menunjukan dokumen sumber atau laporan
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber: (Rosa & Shalahuddin, 2018)

2.7. Pengertian *MySQL*

MySQL adalah RDBMS (*Relational Database Manajement System*) yang dapat menangani data yang bervolume besar (Sadeli, 2014)

MySQL mampu mengirim dan menerima data dengan sangat cepat memiliki tingkat kestabilan yang baik karena telah digunakan dalam proyek skala besar seperti *facebook, google, youtube* dan lain-lain. Beberapa kelebihan yang dimiliki oleh *MySQL* diantaranya :

1. *MySQL* merupakan RDBMS *OpenSource* yaitu *software* yang bersifat *free* atau bebas digunakan oleh perseorangan atau instansi tanpa harus membeli atau membayar kepada pembuatnya.
2. *MySQL* dapat diakses melalui protokol ODBC (*Open Database Connectivity*) sehingga dapat diakses oleh banyak *software*.
3. Semua *user* dapat mengakses *server* dalam satu waktu, tanpa harus menunggu yang lain untuk mengakses *database*.
4. *MySQL* dapat berjalan di berbagai jenis *operating system* seperti *Windows, Linux, Solaris* dan lain-lain.

2.8. PHP *(Personal Home Page)*

PHP (*Personal Home Page*) adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah web dan bisa digunakan pada *HTML*. PHP merupakan singkatan dari "PHP: *Hypertext Prepocessor*", dan merupakan bahasa yang disertakan dalam dokumen HTML sekaligus bekerja di sisi server (server-side HTML-embedded scripting). Artinya sintaks dan perintah yang diberikan akan sepenuhnya dijalankan di server tetapi disertakan pada halaman HTML biasa, sehingga script-nya tak tampak di sisi client (Betha and Husni, 2014)

PHP dirancang untuk dapat bekerja sama dengan database server dan dibuat sedimikian rupa sehingga pembuatan dokumen *HTML* yang dapat mengakses

database menjadi begitu mudah. Tujuan dari bahasa scripting ini adalah untuk membuat aplikasi dimana aplikasi tersebut yang dibangun oleh PHP pada umumnya akan memberikan hasil pada *web browser*, tetapi prosesnya secara keseluruhan dijalankan di *server*.

2.9. XAMPP

XAMPP adalah paket program web lengkap yang dapat Anda pakai untuk belajar pemrograman web, khususnya PHP dan MySQL (Nugroho, 2015).

XAMPP adalah perangkat lunak open source yang diunggah secara geratis dan bisa dijalankan di semua semua operasi seperti *windows, linux, solaris, dan mac* (Buana, 2014)

2.10. *Cloud*

Cloud adalah suatu model komputasi di mana kapabilitas terkait teknologi informasi disajikan sebagai suatu layanan, sehingga pengguna dapat mengaksesnya lewat internet (Biem Prima, 2018).

2.10.1. *OwnCloud*

OwnCloud merupakan salah satu berbagi perangkat lunak berkas (file hosting) gratis dan bebas layaknya Dropbox atau Google Drive. OwnCloud menyediakan pengamanan yang baik, memiliki tata cara yang baik bagi pengguna aplikasi untuk membagi dan mengakses data yang secara lancar terintegrasi dengan perangkat teknologi informasi yang tujuannya mengamankan, melacak, dan melaporkan penggunaan data. OwnCloud merupakan salah satu software open source yang mengizikan siapa saja untuk menginstall dan mengoperasikannya tanpa dikenanakan biaya. Software ini tidak melimitasi kapastias penyimpanan

(kecuali space yang ada di hardisk server) dan jumlah client yang terkoneksi dengan software ini. Dikembangkan awal tahun 2010 oleh Frank Karlitschek seorang develover software KDE untuk menyediakan alternatif software penyedia layanan penyimpanan berbayar. Pembuatannya menggunakan scripting PHP dan Java Script dan dirancang agar bisa bekolaborasi dengan beberapa sistem database diantaranya SQLite, MariaDB, MySQL Oracle dan PostgreSQL. OwnCloud server dapat melakukan sinkronisasi file dengan operating sistem lain yaitu Windows, OS X atau Linux. Versi mobile client juga tersedia untuk Android dan iOS. File atau data lain berupa kalender, kontak atau bookmark dapat diakses menggunakan web browser tanpa software tambahan.

2.11. ISO 9126

Salah satu tolak ukur kualitas perangkat lunak adalah ISO 9126, yang dibuat oleh *International Organization for Standardization (ISO)* dan *International Electrotechnical Commission (IEC)*. ISO 9126 mendefinisikan kualitas produk perangkat lunak, model, karakteristik mutu, dan metrik terkait yang digunakan untuk mengevaluasi dan menetapkan kualitas sebuah produk software. Standar ISO 9126 telah dikembangkan dalam usaha untuk mengidentifikasi atribut-atribut kunci kualitas untuk perangkat lunak komputer. ISO 9126 adalah standar internasional yang diterbitkan oleh ISO untuk evaluasi kualitas perangkat lunak dan merupakan pengembangan dari ISO 9001. Faktor kualitas menurut ISO 9126 meliputi enam karakteristik kualitas sebagai berikut (Abran *et al.*, 2008)

1) *Functionality* (Fungsionalitas). Kemampuan perangkat lunak untuk menyediakan fungsi sesuai kebutuhan pengguna, ketika digunakan dalam kondisi tertentu.

2) *Reliability* (Kehandalan). Kemampuan perangkat lunak untuk mempertahankan tingkat kinerja tertentu, ketika digunakan dalam kondisi tertentu.

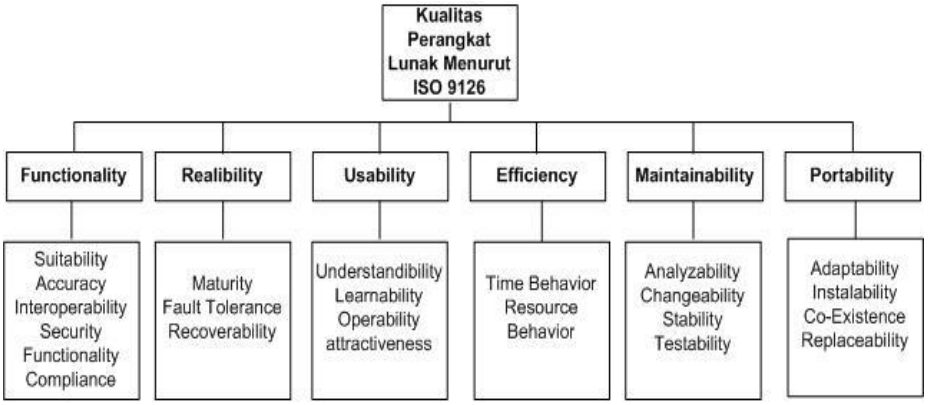
3) *Usability* (Kebergunaan). Kemampuan perangkat lunak untuk dipahami, dipelajari, digunakan, dan menarik bagi pengguna, ketika digunakan dalam kondisi tertentu.

4) *Efficiency* (Efisiensi). Kemampuan perangkat lunak untuk memberikan kinerja yang sesuai dan relatif terhadap jumlah sumber daya yang digunakan pada saat keadaan tersebut.

5) *Maintainability* (Pemeliharaan). Kemampuan perangkat lunak untuk dimodifikasi.Modifikasi meliputi koreksi, perbaikan atau adaptasi terhadap perubahan lingkungan, persyaratan, dan spesifikasi fungsional.

6) *Portability* (Portabilitas). Kemampuan perangkat lunak untuk ditransfer dari satu lingkungan ke lingkungan lain.

Karakteristik Kualitas Perangkat Lunak Model ISO 9126



Gambar 2. 2 Model Kualitas Perangkat Lunak Model ISO 9126 (Al-Qutaish 2010)

Masing-masing karakteristik kualitas perangkat lunak model ISO 9126 dibagi menjadi beberapa sub-karakteristik kualitas. Berikut adalah tabel 2.5 karakteristik Kualitas Perangkat Lunak Model ISO 9126 :

Tabel 2. 5 Karakteristik ISO 9126

Karakteristik	**Sub Karakteristik**	**Deskripsi**
Functionality	*Suitability*	Kemampuan perangkat lunak untuk menyediakan serangkaian fungsi yang sesuai untuk tugas-tugas tertentu dan tujuan pengguna.
	Accuracy	Kemampuan perangkat lunak dalam memberikan hasil yang presisi dan benar sesuai dengan kebutuhan.
	Security	Kemampuan perangkat lunak untuk mencegah akses yang tidak diinginkan, menghadapi penyusup (*hacker*) maupun otorisasi dalam modifikasi data.
	Interoperabiy	Kemampuan perangkat lunak untuk berinteraksi dengan satu atau lebih sistem tertentu.
	Compliance	Kemampuan perangkat lunak dalam memenuhi standar dan kebutuhan sesuai peraturan yang berlaku.
Reliability	*Maturity*	Kemampuan perangkat lunak untuk menghindari kegagalan sebagai akibat dari kesalahan dalam perangkat lunak.
	Fault tolerance	Kemampuan perangkat lunak untuk mempertahankan kinerjanya jika terjadi

		kesalahan perangkat lunak
	Recoverabiliy	Kemampuan perangkat lunak untuk membangun kembali tingkat kinerja ketika terjadi kegagalan sistem, termasuk data dan koneksi jaringan.
Usability	*Understandibity*	Kemampuan perangkat lunak dalam kemudahan untuk dipahami.
	Learnability	Kemampuan perangkat lunak dalam kemudahan untuk dipelajari.
	Operability	Kemampuan perangkat lunak dalam kemudahan untuk dioperasikan.
	Attractiveness	Kemampuan perangkat lunak dalam menarik pengguna.
Efficiency	*Time behavior*	Kemampuan perangkat lunak dalam Memberikan respon dan waktu pengolahan yang sesuai saat melakukan fungsinya
	Resource behavior	Kemampuan perangkat lunak dalam menggunakan sumber daya yang dimilikinya ketika melakukan fungsi yang ditentukan.
Maintainabiliy	*Analyzability*	Kemampuan perangkat lunak dalam mendiagnosis kekurangan atau penyebab kegagalan.
	Changeabiliy	Kemampuan perangkat lunak untuk dimodifikasi tertentu.
	Stability	Kemampuan perangkat lunak untuk meminimalkan efek tak terduga dari modifikasi perangkat lunak.
	Testability	Kemampuan perangkat lunak untuk

		dimodifikasi dan divalidasi perangkat lunak lain.
Portability	*Adaptability*	Kemampuan perangkat lunak untuk diadaptasikan pada lingkungan yang berbeda-beda.
	Instalability	Kemampuan perangkat lunak untuk diinstal dalam lingkungan yang berbeda-beda.
	Coexistence	Kemampuan perangkat lunak untuk berdam[pingan dengan pernagkat lunak lainnya dalam satu lingkungan dengan berbagai sumber daya
	Replaceability	Kemampuan perangkat lunak untuk digunakan sebagai pengganti perangkat lunak lainnya

Sumber: (Al-Qutaish 2010)

Adapun alasan penggunaan ISO 9126 karena ISO sudah berstandar *International Organization for Standardization (ISO)* dan *International Electrotechnical Commission (IEC)*. Kualitas produk perangkat lunak ISO 9126 memiliki enam karakteristik pendukung yang dapat digunakan sebagai acuan dalam menilai maupun memberikan masukan terhadap kualitas perangkat lunak yang akan dibangun yang akan menghasilkan nilai uji yang terukur. Indikator yang digunakan dalam pengujian ISO 9126 dilihat dari sisi *Functionality, Usability*, dan *Efficiency.*

Adapun kriteria hasil perhitungan kelayakan sistem dari pengujian ini adalah sebagai berikut:

Tabel 2. 6 Kriteria Persentase Skor Tanggapan Responden

Jumlah Skor	Kriteria
0.00 – 36.00	Tidak Baik / Tidak Layak
36.01 – 52.00	Kurang Baik / Kurang Layak
52.01 – 68.00	Cukup Baik / Cukup Layak
68.01 – 84.00	Baik / Layak
84.01 – 100	Sangat Baik / Sangat Layak

Sumber: Narimawati (2007)

Skala pengukuran yang digunakan adalah skala Likert, skala yang didasarkan pada penjumlahan sikap responden dalam merespon pernyataan berkaitan indikator-indikator suatu konsep atau variable yang sedang diukur (Sanusi, 2012). Skala Likert umumnya menggunakan lima titik dengan label netral pada posisi tengah (ketiga).Skala Likert apat dilihat pada Tabel 2.7.

Tabel 2. 7 Skala Likert

Jawaban	Skor
Sangat Setuju	5
Setuju	4
Netral	3
Tidak Setuju	2
Sangat Tidak Setuju	1