

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam penelitian yang dilakukan menjadi pendukung skripsi ini dapat dilihat pada table 2.1 yang terdiri dari 4 literatur antara tahun 2015 sampai 2017 berikut ini merupakan tinjauan studi yang di ambil yaitu:

Tabel 2. 1 Daftar Literatur

No Literatur	Penulis	Tahun	Judul
Literatur 01	Koyuko, Harnolus Sinsuw, Alicia A.E. Najoan, Xaverius B.N.	2016	Perancangan Aplikasi <i>Monitoring</i> Pemadaman Listrik Berbasis Android Studi Kasus PT.PLN Area Manado
Literatur 02	Faisal Pangestu, Prasetya Mintarsih, Fitri Angraini, Nenny	2016	Sistem Informasi Eksekutif (SIE) Untuk Memantau Gangguan Gardu Listrik Berbasis Lokasi Pada Platform Android(Studi Kasus: PT. PLN Area Bandung)
Literatur 03	C. Bambang Dwi Kuncoro, M Anda Falahuddin, Tandi Sutandi, Raden M.Hadi, Terapan.	2017	Pengembangan Sistem Monitoring Mini <i>Showcase</i> Bertenaga Listrik <i>Hybrid</i> Berbasis Aplikasi Android
Literatur 04	Anjani	2015	Rancang Bangun Sistem Informasi Monitoring Kondisi Mesin genset Terpusat Berbasis Web Pada PT. Jichai Power Indonesia.

2.1.1 Tinjauan Literatur 01

Jurnal ini meneliti tentang aplikasi monitoring pemadaman listrik berbasis android. Penelitian ini bertujuan untuk memberikan informasi terlebih dahulu kepada pelanggan untuk mengamankan barang-barang elektronik atau mempersiapkan sumber listrik lainnya. Selanjutnya aplikasi ini berguna untuk membantu masyarakat dalam hal memberikan informasi ketika terjadi gangguan dalam hal ini pemadaman yang terjadi yang disebabkan oleh kondisi alam, atau kondisi teknis yang mendadak di lapangan.

Dalam penelitian ini terdapat dua buah aplikasi , aplikasi yang pertama adalah aplikasi berbasis Android yang digunakan oleh masyarakat umum (*user*) untuk melihat Informasi Pemadaman dan Melaporkan gangguan yang terjadi. Aplikasi yang kedua adalah aplikasi berbasis web yang digunakan oleh *admin* (operator) untuk melakukan manajemen data dan memproses laporan gangguan dari user. Pada aplikasi android terdapat fitur untuk melihat info pemadaman untuk melihat info pemadaman dan laporan gangguan yang bisa digunakan oleh user untuk mem-posting info gangguan yang ada. Seluruh fitur ini memerlukan sebuah sistem yang memiliki komponen aplikasi android dan *database* Server. Aplikasi android pada sistem ini dibuat dengan menggunakan cordova. Untuk melakukan fungsi-fungsinya aplikasi ini akan berhubungan dengan *database* server yang menggunakan *mysql*.

MySQL adalah sebuah perangkat lunak *system* manajemen SQL atau lebih dikenal dengan *database management system* (DBMS) yang *multithread*, multi user karena karakteristik mysql adalah multi user maka penulis menggunakan mysql sebagai software management database pada perancangan aplikasi monitoring pemadaman listrik berbasis android Kota Manado.

SRS bertujuan untuk menggambarkan kebutuhan sistem aplikasi monitoring pemadaman listrik Manado secara spesifik dan lengkap. Selain itu akan menjelaskan tujuan dan fitur dari sistem, antar muka sistem, apa yang akan dilakukan sistem, dan bagaimana sistem akan beroperasi.

2.1.2 Tinjauan Literatur 02

Tujuan dari penelitian ini adalah untuk merancang aplikasi yang dapat menginput data asset, melihat dan mengirim pesan permintaan asset apabila dibutuhkan. Menurut

Laudon (2002) sistem informasi eksekutif merupakan sistem informasi pada level strategis dari organisasi yang ditujukan bagi pengambilan keputusan tidak terstruktur melalui grafik dan komunikasi. Sehingga Sistem Informasi Eksekutif adalah sebuah sistem yang memberikan informasi-informasi sangat penting kepada manajer untuk memonitori kinerja sebuah perusahaan sehingga dapat mengambil sebuah keputusan yang dapat meningkatkan kinerja perusahaan tersebut.

Penelitian ini menitik beratkan bagaimana user area dan user unit dapat memasukan laporan-laporan yang dibutuhkan pihak menejerial tepat waktu. Proses awal adalah administrator menentukan *user* unit dan user area maupun pihak menejerial yang dapat mengakses aplikasi ini. Mengingat laporan-laporan perusahaan bersifat rahasia dan penting maka semua informasi data siapa yang membuat, siapa yang mengakses harus jelas dan akurat, karena nantinya akan dipertanggungjawabkan semua kebenaran data laporan yang diinput ke dalam aplikasi. Setelah itu, *user* unit dan user area yang sudah ditentukan harus login terlebih dahulu, ketika user akan menginput laporan-laporan perusahaan maka *user* dapat memilih *page* gangguan listrik, dan dapat menginput data aset jaringan listrik. Sedangkan untuk manajerial hanya bisa melihat dan men- download laporan-laporan perusahaan dan menggunakan fitur navigasi.

Aplikasi ini dibuat dengan fitur *notification* yang dapat memunculkan sebuah allert berupa suara dan perubahan lampu LED HP apabila terdapat laporan yang masuk sehingga laporan tersebut sesuai dengan waktu kejadian. Dengan aplikasi ini admin, *user* area dan unit dapat menginput persediaan aset di setiap unit atau area, sehingga pegawai lapangan mengetahui persediaan aset dan mengirim pesan meminta aset tersebut apabila terjadi kerusakan di suatu wilayah menggunakan aplikasi ini.

2.1.3 Tinjauan Literatur 03

Penelitian ini membahas tentang sistem monitoring mini showcase bertenaga listrik *hybrid* yang merupakan sistem monitoring pada *smartphone* Android yang diaplikasikan pada mini showcase berdaya listrik sistem hybrid dengan menggunakan sumber energi matahari dan sumber listrik PLN. Sistem monitoring yang dibangun pada dasarnya bekerja untuk mendukung sistem kontrol yang mengatur kerja sistem *refrigerasi mini showcase*, dan juga mengatur pasokan energi listriknya. Dengan penggunaan sistem

monitoring yang dikembangkan maka akan memudahkan *user/operator* sistem mendapat informasi terkait dengan sistem yang dipantau secara *remote* dan *real time*. Perangkat monitoring yang dikembangkan berbasis aplikasi Android dan *platform* Arduino. Fungsi sistem monitoring mampu memberikan informasi data *temperature*, tegangan, dan intensitas cahaya untuk mengatur kerja sistem pembangkit listrik *hybrid* dan kerja sistem *refrigerasi mini showcase*. Sistem monitoring yang dirancang juga mampu menampilkan data-data terukur pada LCD, dan mengirim data tersebut ke *smartphone* Android selama periode monitoring.

2.1.4 Tinjauan Literatur 04

Penelitian ini membahas tentang system informasi monitoring kondisi mesin genset terpusat pada PT. Jichai Power Indonesia. Penelitian ini bertujuan untuk membuat suatu sistem yang akan memudahkan staff yang berada dikantor pusat untuk mengetahui kondisi mesin di lapangan serta sebagai acuan dalam perhitungan denda yang harus dibayarkan oleh PT. Jichai Power Indonesia. Sistem ini akan memberikan informasi yang akurat dan mudah digunakan oleh user/admin dalam pengelolaannya, juga dapat meningkatkan mutu kinerja pelayanan sehingga proses demi proses dilakukan seara cepat dan berjalan lancar.

Metode pengembangan sistem yang digunakan pada penelitian ini adalah SDLC (*System Development Life Cycle*). Sedangkan untuk perancangan proses pada sistem yang akan dibuat menggunakan *Flowchart*, *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram*. Dan pada perancangan program yang akan dibuat penulis menggunakan bahasa pemrograman PHP dan MySQL. Perancangan sistem ini diharapkan dapat meningkatkan kinerja Operator *site* dalam mengirimkan data monitoring dan kerusakan setiap harinya serta keakuratan dari informasi tersebut. Juga dapat mempermudah *Admin site* dalam pengelolaan laporan serta mempermudah Pimpinan untuk melakukan pengawasan.

2.2 Monitoring

Monitoring, dalam bahasa Indonesia dikenal dengan istilah pemantauan. Monitoring merupakan sebuah kegiatan untuk menjamin akan tercapainya semua tujuan organisasi dan manajemen (Handoko, 1995). Dalam kesempatan lain, monitoring juga

didefinisikan sebagai langkah untuk mengkaji apakah kegiatan yang dilaksanakan telah sesuai dengan rencana, mengidentifikasi masalah yang timbul agar langsung dapat diatasi, melakukan penilaian apakah pola kerja dan manajemen yang digunakan sudah tepat untuk mencapai tujuan, mengetahui kaitan antara kegiatan dengan tujuan untuk memperoleh ukuran kemajuan (Sutabri, 2012). Dengan kata lain, monitoring merupakan salah satu proses didalam kegiatan organisasi yang sangat penting yang dapat menentukan terlaksana atau tidaknya sebuah tujuan organisasi. Tujuan dilakukannya monitoring adalah untuk memastikan agar tugas pokok organisasi dapat berjalan sesuai dengan rencana yang telah ditentukan (Aviana, 2012)..

Monitoring akan memberikan informasi tentang status dan kecenderungan bahwa pengukuran serta evaluasi yang diselesaikan berulang dari waktu ke waktu. Pemantauan umumnya dilakukan untuk tujuan tertentu, seperti untuk memeriksa terhadap proses berikut objek atau untuk mengevaluasi kondisi atau kemajuan menuju tujuan hasil manajemen atas efek tindakan dari beberapa jenis antara lain tindakan untuk mempertahankan manajemen yang sedang berjalan.

2.3 Android

Android merupakan subset perangkat lunak untuk perangkat mobile yang meliputi sistem operasi, *middleware*, dan aplikasi inti yang dirilis oleh Android adalah sistem operasi bergerak (*mobile operating system*) yang mengadopsi sistem operasi linux, namun telah dimodifikasi. Android diambil alih oleh Google pada tahun 2005 dari Android, Inc sebagai bagian strategi untuk mengisi pasar sistem operasi bergerak. Google mengambil alih seluruh hasil kerja Android termasuk tim yang mengembangkan Android.(Lengkong et al., 2015)

Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang digunakan oleh bermacam piranti bergerak. Awalnya android dikembangkan oleh Android, Inc. dengan dukungan finansial dari Google yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya *Open Handset Alliance*, konsorium dari perusahaan-perusahaan perangkat keras, perangkat lunak dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Sudah banyak *platform*

untuk perangkat selular saat ini, termasuk didalamnya *Symbian*, *iPhone*, *Windows Mobile*, *BlackBerry*, *Java Mobile Edition*, *Linux Mobile (LiM0)*, dan banyak lagi. Namun ada beberapa hal yang 12 menjadi kelebihan Android. Beberapa keunggulan android dengan *platform* lain yaitu:

1. Keterbukaan, Android menyediakan akses ke fungsi dasar perangkat mobile menggunakan standar panggilan ke API.
2. Penghancuran perbatasan, anda dapat menggabungkan informasi dari internet ke dalam telepon, seperti informasi kontak, atau data pada lokasi geografis.
3. Cepat dan mudah perkembangannya, dalam SDK memiliki semua yang anda butuhkan untuk membuat dan menjalankan aplikasi Android, termasuk simulator dan alat *debugging*.

2.4 Android Studio

Android Studio adalah *Integrated Development Environment (IDE)* untuk pengembangan aplikasi Android. Selain merupakan kode editor dan alat pengembang aplikasi Android, Android Studio menawarkan fitur lebih banyak untuk membuat aplikasi Android, misalnya:

1. *Gradle* yang fleksibel
2. Emulator yang cepat dan memiliki banyak fitur
3. Environment untuk pengembangan semua perangkat Android
4. instant Run untuk menjalankan aplikasi tanpa perlu membuat apk baru
5. terdapat beberapa template kode dan integrasi dengan GitHub
6. memiliki alat pengujian dan kerangka kerja yang ekstensif
7. Terdapat beberapa bantuan untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain
8. dukungan C++ dan NDK
9. terdapat *Google Cloud Platform*, untuk mempermudah integrasi dengan *Google Cloud Messaging* dan *App Engine*.(Tanuwijaya, 2018).

2.5 Java

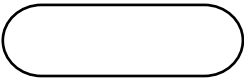

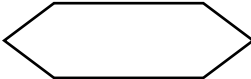

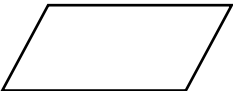

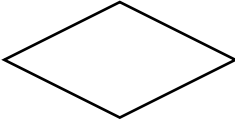
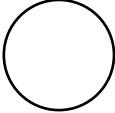
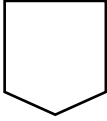
Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih

bergabung di *Sun Microsystems* saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksi model objek yang lebih sederhana serta dukungan rutin-rutin atas bawah yang minimal. Aplikasi-aplikasi berbasis JAVA umumnya dikompilasi ke dalam p-code (bytecode) dan dapat dijalankan pada berbagai mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus di disain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitas-nya yang memungkinkan aplikasi java mampu berjalan di beberapa platform system operasi yang berbeda, java dikenal pula dengan slogannya, “Tulis sekali, jalankan di mana pun”. Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web. Java adalah bahasa pemrograman yang multi *platform* dan *multi device*. Sekali anda menuliskan sebuah program dengan menggunakan java, anda dapat menjalankan hampir di semua computer dan perangkat lain yang mensupport java, dengan sedikit perubahan atau tanpa perubahan sama sekali dalam kodenya. Aplikasi dengan berbasis Java ini dikompilasikan ke dalam *p-code* dan bisa dijalankan dengan java virtual machine. Fungsionalitas dari java ini dapat berjalan dengan *platform system* operasi yang berbeda karena sifatnya yang umum dan non-spesifik.(Barri et al., 2015)

2.6 Flowchart Diagram

Flowchart adalah representasi secara simbolik dari suatu algoritma atau prosedur untuk menyelesaikan suatu masalah, dengan menggunakan flowchart akan memudahkan pengguna melakukan pengecekan bagian-bagian yang terlupakan dalam analisis masalah, disamping itu *flowchart* juga berguna sebagai fasilitas untuk berkomunikasi antara pemrogram yang bekerja dalam tim suatu proyek. *Flowchart* membantu memahami urutan-urutan logika yang rumit dan panjang. *Flowchart* membantu mengkomunikasikan jalannya program ke orang lain (bukan pemrogram) akan lebih mudah. (Santoso & Nurmalina, 2017), berikut table flowchart diagram tersebut :

Table 2.2 Daftar Flowchart Diagram

Simbol	Nama	Fungsi
	Terminator	Permulaan / akhir program
	Garis Alir	Arah aliran program
	Preparation	Proses inisialisasi / pemberian harga awal
	Proses	Proses perhitungan / proses pengolahan data
	Input/output data	Proses input / output data, parameter, informasi
	Predefined Proses	Permulaan sub prigram / proses menjalankan program
	Decision	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	On Page Connector	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	Off Page Connector	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda

Flowchart terbagi atas lima jenis, yaitu:

- a. Flowchart Sistem (*System Flowchart*)
- b. Flowchart Dokumen (*Document Flowchart*)
- c. Flowchart Skematik (*Schematic Flowchart*)
- d. Flowchart Program (*Program Flowchart*)

e. *Flowchart Proses (Process Flowchart)*

a. ***Flowchart Sistem (System Flowchart)***

Flowchart Sistem merupakan bagan yang menunjukkan alur kerja atau apa yang sedang dikerjakan di dalam sistem secara keseluruhan dan menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem. *Flowchart* ini merupakan dekripsi secara grafik dari urutan prosedur-prosedur yang terkombinasi yang membentuk suatu sistem. *Flowchart* ini terdiri dari data yang mengalir melalui sistem dan proses yang mentransformasikan data itu. Data dan proses dalam *Flowchart* ini dapat digambarkan secara online (dihubungkan langsung dengan komputer) atau *offline* (tidak dihubungkan langsung dengan komputer, misalnya mesin tik, cash register atau kalkulator).

b. ***Flowchart Dokumen (Document Flowchart)***

Bagan alir dokumen (*document Flowchart*) atau disebut juga bagan alir formulir (*form Flowchart*) atau *paperwork Flowchart* merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya. Bagan alir dokumen ini menggunakan simbol-simbol yang sama dengan yang digunakan di dalam bagan alir sistem.

c. ***Flowchart Skematik (Schematic Flowchart)***

Bagan alir skematik (*schematic Flowchart*) merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem. Perbedaannya adalah bagan alir skematik selain menggunakan simbol-simbol bagan alir sistem, juga menggunakan gambar-gambar komputer dan peralatan lainnya yang digunakan. Maksud penggunaan gambar-gambar ini adalah untuk memudahkan komunikasi kepada orang yang kurang paham dengan simbol-simbol bagan alir. Penggunaan gambar-gambar ini memudahkan untuk dipahami, tetapi sulit dan lama menggambarinya.

d. ***Flowchart Program (Program Flowchart)***

Bagan alir program (*program Flowchart*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Bagan alir program dibuat dari derivikasi bagan alir sistem. Bagan alir program dapat terdiri dari dua macam, yaitu bagan alir logika program (*program logic Flowchart*) dan bagan alir

program komputer terinci (*detailed computer program Flowchart*). Bagan alir logika program digunakan untuk menggambarkan tiap-tiap langkah di dalam program komputer secara logika. Bagan alir logika program ini dipersiapkan oleh analis sistem.

e. *Flowchart Proses (Process Flowchart)*

Flowchart Proses merupakan teknik penggambaran rekayasa industrial yang memecah dan menganalisis langkah-langkah selanjutnya dalam suatu prosedur atau sistem. Bagan alir proses menggunakan lima buah simbol tersendiri, *Flowchart* Proses digunakan oleh perekayasa industrial dalam mempelajari dan mengembangkan proses-proses *manufacturing*. Dalam analisis sistem, *Flowchart* ini digunakan secara efektif untuk menelusuri alur suatu laporan atau *form*.

2.7 Unified Modeling Language (UML)

UML merupakan pengganti dalam metode analisa dan menjadi standarisasi dalam perancangan system piranti lunak, merupakan bahasa dalam metode pemodelan berorientasi *object*. (Darmanto et al., n.d.)

Tujuan UML antara lain adalah:

1. Memberikan model yang siap digunakan, bahasa pemodelan visual yang ekspresif untuk pengembangan dan saling beretukar model dengan mudah dan dimengerti secara umum.
2. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
3. Menadukan teknik terbaik yang terdapat dalam pemodelan.

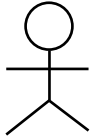


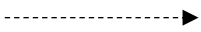

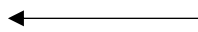
Berikut klasifikasi model diagram pada UML:

A. *Use Case Diagram*

Use case diagram adalah rangkaian maupun uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi. “*Use Case Diagram* merupakan rangkaian tindakan yang dilakukan oleh sistem, aktor mewakili user atau sistem lain yang berinteraksi dengan sistem yang dimodelkan”.

Berikut ini adalah table penjelasan tentang fungsi *symbol* yang di gunakan pada *use case*:

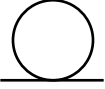
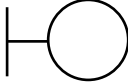

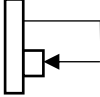


Table 2.3 Daftar Use Case Diagram

Simbol	Penjelasan
	Actor mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan use case
	Use Case: Abstraksi dan interaksi antara sistem dalam aktor
	Association: Abstraksi dari penghubung antara aktor dan use case
	Generalisasi menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan use case
	Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya
	Menunjukkan bahwa suatu use case merupakan tambahan fungsional dari use case lainnya jika suatu kondisi terpenuhi

B. Sequence Diagram

Sequence Diagram (diagram urutan) adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam system yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, *display*, dan sebagainya berupa pesan/*message*. Berikut ini merupakan table dalam *sequence diagram*.

Table 2.4 Daftar Sequence Diagram

Gambar	Nama	Keterangan
	Entitiy Class	Gambaran sistem sebagai landasan dalam menyusun basis data
	Boundary Class	Menangani komunikasi antar lingkungan sistem
	Control Class	Bertanggung jawab terhadap kelas-kelas terhadap objek yang berisi logika
	Recursive	Pesan untuk dirinya
	Activation	Mewakili proses durasi aktivasi sebuah operasi
	Line Life	Komponen yang digambarkan garis putus terhubung dengan objek

2.8 Database

Database adalah susunan *record* data operasional lengkap dari suatu organisasi atau perusahaan, yang diorganisir dan disimpan secara terintegrasi dengan menggunakan metode tertentu sehingga mampu memenuhi informasi yang optimal yang dibutuhkan oleh para pengguna. Pengguna *Database* Menurut Fathansyah dalam Pengguna basis data yang dibedakan berdasarkan cara mereka berinteraksi terhadap sistem, ada beberapa tipe yaitu:

1. Programmer Aplikasi Pemakai berinteraksi dengan basis data melalui *Data Manipulation Language*, yang disertakan dalam program yang ditulis dalam bahasa pemrograman induk
2. *User Mahir (Casual User)* Pemakai yang berinteraksi dengan system tanpa menulis modul program.

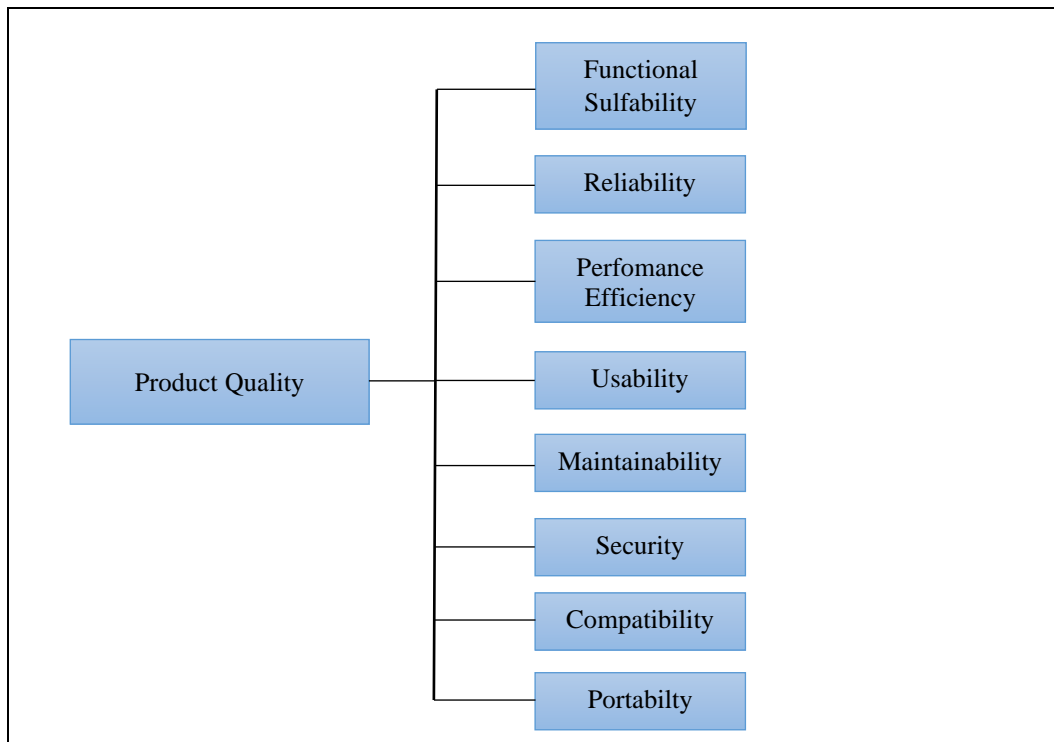
3. *User Umum (End User Naïve User)* Pemakai yang berinteraksi dengan sistem basis data melalui pemanggilan satu program aplikasi permanent yang telah ditulis atau disediakan sebelumnya.
4. *User Khusus (Specialized User)* Pemakai yang menulis aplikasi basis data non konvensional, tetapi untuk keperluan khusus. (Santoso & Nurmalina, 2017)

2.9 Mysql

MySQL adalah sebuah *Database Manajemen System* (DBMS) populer yang memiliki fungsi sebagai *relational database manajemen system* (RDBMS). Selain itu MySQL *software* merupakan suatu aplikasi yang sifatnya *open source* serta server basis data MySQL memiliki kinerja sangat cepat, *reliable*, dan mudah untuk digunakan serta bekerja dengan arsitektur *client server* atau *embedded systems*. Dikarenakan faktor *open source* dan populer tersebut maka cocok untuk mendemonstrasikan proses replikasi basis data. (Yuliansyah et al., 2014)

2.10 Pengujian ISO 25010

Model ISO-25010 merupakan bagian dari *Software product Quality Requirements and Evaluation (SQuaRE)*, yang merupakan pengembangan dari model kualitas perangkat lunak sebelumnya yaitu ISO-9126. Dalam model ISO-25010 ini digunakan untuk melihat kualitas suatu perangkat lunak yang digunakan oleh perusahaan, instansi ataupun organisasi. Metode ISO 25010 ini dapat digunakan untuk mengevaluasi kualitas sistem perangkat lunak secara spesifik berdasarkan dua dimensi umum, yaitu dimensi *product quality*, dimana prosesnya mengacu pada karakteristik intrinsik dari sebuah produk perangkat lunak, memiliki beberapa elemen antara lain meliputi *functional suitability*, *reliability*, *operability*, *performance efficiency*, *security*, *compatibility*, *maintainability* dan *transferability*. *Quality in use* dan *product quality*. Sedangkan pada *dimensi quality in use*, terdapat beberapa karakteristik relatif yang ditinjau dari perspektif *user* antara lain *Usability in use*, *Flexibility in use*, dan *Safety* (Alfian, 2010)



Gambar 2. 1 Model kualitas produk ISO/IEC 25010

Adapun dimensi yang pertama terdapat beberapa faktor elemen diantaranya:

1) *Functionality* (Fungsionalitas).

Kemampuan perangkat lunak untuk Merupakan tingkatan dimana perangkat lunak dapat menyediakan fungsionalitas yang dibutuhkan ketika perangkat lunak digunakan pada kondisi spesifik tertentu dalam hal ini perangkat lunak dapat memenuhi kelayakan dari sebuah fungsi untuk melakukan pekerjaan yang spesifik bagi pengguna dan dapat memberikan hasil yang tepat dan ketelitian terhadap tingkat kebutuhan pengguna. Karakteristik ini dibagi menjadi beberapa karakteristik yaitu.

- a. *Functional completeness*, sejauh mana fungsi yang disediakan mencakup semua tugas dan tujuan pengguna secara spesifik.
- b. *Functional correctness*, sejauh mana produk atau sistem menyediakan hasil yang benar sesuai kebutuhan.
- c. *Functional appropriateness*, sejauh mana fungsi yang disediakan mampu memfasilitasi penyelesaian tugas dan tujuan tertentu.

2) *Compatibility*

Sejauh mana sebuah produk, sistem atau komponen dapat bertukar informasi dengan produk, sistem atau komponen dan/atau menjalankan fungsi lain yang diperlukan secara bersamaan ketika berbagi perangkat keras dan environment perangkat lunak yang sama. Karakteristik ini dibagi menjadi 2 karakteristik yaitu.

- a. *Co-existence*, sejauh mana produk atau sistem dapat menjalankan fungsi yang dibutuhkan secara efisien sementara berbagi sumber daya dengan produk atau sistem yang lain tanpa merugikan produk atau sistem tersebut.

3) *Interoperability*, sejauh mana dua atau lebih produk, sistem atau komponen dapat bertukar informasi dan menggunakan informasi tersebut. *Usability*

Sejauh mana sebuah produk atau sistem dapat digunakan oleh user tertentu untuk mencapai tujuan dengan efektif, efisiensi, dan kepuasan tertentu dalam konteks penggunaan. Karakteristik ini terbagi menjadi beberapa karakteristik yaitu.

- a. *Appropriateness recognizability*, sejauh mana pengguna dapat mengetahui apakah sistem atau produk sesuai kebutuhan mereka.
- b. *Learnability*, sejauh mana produk atau sistem dapat digunakan oleh pengguna untuk mencapai tujuan tertentu yang belajar menggunakan sistem atau produk dengan efisien, efektif, kebebasan dari resiko dan kepuasan dalam konteks tertentu.
- c. *Operability*, sejauh mana produk atau sistem mudah dioperasikan dan dikontrol.
- d. *User error protection*, sejauh mana produk atau sistem melindungi pengguna terhadap membuat kesalahan.
- e. *User interface aesthetics*, sejauh mana antarmuka pengguna dari produk atau sistem memungkinkan interaksi yang menyenangkan dan memuaskan pengguna.

- 4) *Accessibility*, sejauh mana produk atau sistem dapat digunakan oleh semua kalangan untuk mencapai tujuan tertentu sesuai konteks penggunaan.

Reliability

Merupakan tingkatan dimana perangkat lunak dapat bertahan pada tingkatan tertentu ketika digunakan oleh pengguna pada kondisi yang spesifik dalam hal ini perangkat lunak dapat beroperasi dan siap ketika dibutuhkan untuk digunakan dan juga dapat bertahan pada tingkat kemampuan tertentu terhadap kegagalan, kesalahan serta perangkat lunak kembali pada tingkat tertentu dalam mengembalikan pengembalian data yang disebabkan kegagalan atau kesalahan pada perangkat lunak. Karakteristik ini terbagi menjadi beberapa subkarakteristik yaitu.

- a. *Maturity*, sejauh mana produk atau sistem mampu memenuhi kebutuhan secara handal di bawah keadaan normal.
- b. *Availability*, sejauh mana produk atau sistem siap beroperasi dan dapat diakses saat perlu digunakan.
- c. *Fault tolerance*, sejauh mana produk atau sistem tetap berjalan sebagaimana yang dimaksud meskipun terjadi kesalahan pada perangkat keras atau perangkat lunak.
- d. *Recoverability*, sejauh mana produk atau sistem mampu dapat memulihkan data yang terkena dampak secara langsung dan menata ulang kondisi sistem seperti yang diinginkan ketika terjadi gangguan.

- 5) *Security*

Sejauh mana sebuah produk atau sistem melindungi informasi dan data sehingga seseorang atau sistem lain dapat mengakses data sesuai dengan jenis dan level otorisasi yang dimiliki. Karakteristik ini terbagi menjadi beberapa karakteristik yaitu.

- a. *Confidentiality*, sejauh mana produk atau perangkat lunak memastikan data hanya bisa diakses oleh mereka yang berwenang untuk memiliki akses.

- b. *Integrity*, sejauh mana produk atau perangkat lunak mampu mencegah akses yang tidak sah untuk memodifikasi data.
- c. *Non-repudiation*, sejauh mana peristiwa atau tindakan dapat dibuktikan telah terjadi, sehingga tidak ada penolakan terhadap peristiwa atau tindakan tersebut.
- d. *Accountability*, sejauh mana tindakan dari suatu entitas dapat ditelusuri secara unik untuk entitas.

6) *Authenticity*,

Sejauh mana identitas subjek atau sumber daya dapat terbukti menjadi salah satu yang diklaim. *Portability*

Sejauh mana keefektifan dan efisiensi sebuah sistem, produk atau komponen dapat dipindahkan dari satu perangkat keras, perangkat lunak atau digunakan pada lingkungan yang berbeda. Karakteristik ini dibagi menjadi beberapa karakteristik yaitu.

- a. *Adaptability*, sejauh mana produk atau sistem dapat secara efektif dan efisien disesuaikan pada perangkat lunak, perangkat keras dan lingkungan yang berbeda.
- b. *Installability*, sejauh mana produk atau sistem dapat berhasil dipasang atau dihapus dalam lingkungan tertentu.
- c. *Replaceability*, sejauh mana produk atau sistem dapat menggantikan produk atau sistem lain yang ditentukan untuk tujuan yang sama pada lingkungan yang sama.

7) *Performance efficiency*

Merupakan tingkatan dimana perangkat lunak dapat memberikan kinerja terhadap sejumlah sumber daya yang digunakan pada kondisi tertentu dalam hal ini *performance efficiency* dapat memberikan reaksi dan waktu yang dibutuhkan ketika melakukan aksi dari sebuah fungsi dan perangkat lunak dapat menggunakan sejumlah sumber daya ketika melakukan aksi dari sebuah fungsi. Kinerja relatif terhadap sumber daya yang digunakan dalam kondisi tertentu. Karakteristik ini terbagi menjadi beberapa subkarakteristik yaitu.

- a. *Time behaviour*, sejauh mana respon dan pengolahan waktu produk atau sistem dapat memenuhi persyaratan ketika menjalankan fungsi.
- b. *Resource utilization*, sejauh mana jumlah dan jenis sumber daya yang digunakan oleh produk atau sistem dapat memenuhi persyaratan ketika menjalankan fungsi.
- c. *Capacity*, sejauh mana batas maksimum parameter produk atau sistem dapat memenuhi persyaratan.

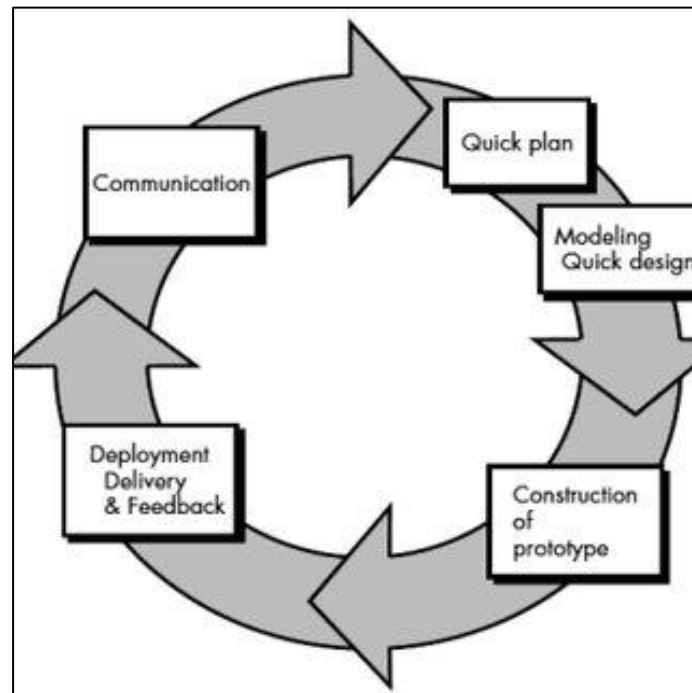
8) *Maintainability*

Merupakan tingkat dimana sebuah perangkat lunak dapat dimodifikasi. Dalam hal ini modifikasi adalah perbaikan, perubahan atau penyesuaian perangkat lunak untuk dapat berubah pada lingkungan, kebutuhan dan fungsionalitas yang spesifik. Selain itu perangkat lunak dapat dianalisis untuk mengetahui apa yang menyebabkan kegagalan pada perangkat lunak untuk mengidentifikasi bagian yang dapat dimodifikasi. Sejauh mana keefektifan dan efisiensi dari sebuah produk atau sistem dapat dirawat. Karakteristik ini terbagi menjadi beberapa subkarakteristik yaitu.

- a. *Modularity*, sejauh mana sistem terdiri dari komponen terpisah sehingga perubahan atau modifikasi pada salah satu komponen tersebut memiliki dampak yang kecil terhadap komponen yang lain.
- b. *Reusability*, sejauh mana aset dapat digunakan lebih oleh satu sistem atau digunakan untuk membangun aset lain.
- c. *Analyzability*, tingkat efektivitas dan efisiensi untuk mengkaji dampak perubahan pada satu atau lebih bagian-bagian produk atau sistem, untuk mendiagnosis kekurangan atau penyebab kegagalan produk, untuk mengidentifikasi bagian yang akan diubah.
- d. *Modifiability*, sejauh mana produk atau sistem dapat dimodifikasi secara efektif dan efisien tanpa menurunkan kualitas produk yang ada.
- e. *Testability*, tingkat efektivitas dan efisiensi untuk membentuk kriteria uji dari produk, sistem atau komponen dan uji dapat dilakukan untuk menentukan apakah kriteria tersebut telah terpenuhi.
- f.

1.11 Metode Pengembangan Sistem

Metode Prototype merupakan metode pengembangan perangkat lunak yang memungkinkan adanya interaksi antara pengembang sistem dengan pengguna sistem, sehingga dapat mengatasi ketidakserasian antara pengembang dan pengguna (Pressman, 2012: 50). Adapun model pengembangan Prototype digambarkan pada gambar 2.2.



Gambar 2.2 Prototype Model (Sumber: Roger S. Pressman, 2012:50)

Seringkali pelanggan mendefinisikan sejumlah sasaran perangkat lunak secara umum, tetapi tidak bisa mengidentifikasi spesifikasi kebutuhan yang rinci untuk fungsi-fungsi dan fitur-fitur yang nantinya akan dimiliki perangkat lunak yang akan dikembangkan. Dalam kasus yang lain, pengembangan perangkat lunak mungkin merasa tidak pasti tentang efisiensi suatu algoritma yang akan digunakan dalam pengembangan perangkat lunak, atau juga merasa tidak pasti akan kemampuan perangkat lunak untuk beradaptasi dengan manusia-komputer yang digunakan. Dalam kasus-kasus seperti ini dan dalam banyak situasi yang lain, paradigma pembuatan prototype (prototyping) mungkin menawarkan pendekatan yang paling baik (Pressman, 2012:50).