BAB II
LANDASAN TEORI

2.1 Tinjauan Pustaka

Berikut ini adalah beberapa tinjauan pustaka yang dilakukan oleh penulis pada penelitian sebelumnya untuk menjadi pendukung penelitian yang sedang dilakukan dapat dilihat pada Tabel 2.1.

Tabel 2. 1 Daftar Literatur

No	Nama Peneliti	Tahun	Judul
1	(Fitri Ayu, Ari Mustofa)	2020	Sistem Aplikasi Absensi Menggunakan Teknologi Barcode Scanner Berbasis Android
2	(Uci Rahmalisa, Yuda Irawan, Refni Wahyuni)	2020	Aplikasi Absensi Guru Pada Sekolah Berbasis Android dengan Keamanan *Qr Code*
3	(Nur Rubiati, Sahara Widya Harahap)	2019	Aplikasi absensi Siswa Menggunakan Qr Code dengan Bahasa Pemrograman Php Di Smkit Zunurain Aqila Zahra Di Pelintung
4	(Sherly Christina, Enny Dwi Oktaviyani, Deddy Ronaldo, Rosya M. Zaini)	2019	Aplikasi Absensi Siswa Berbasiskan *Android*
5	(Fachrival Mustari)	2018	Aplikasi Absensi Guru Pada Sekolah Berbasis *Android*
6	(Sukatmi, Endah Septia Pitri)	2018	Aplikasi Absensi Siswa Berbasis *Web* Dengan Dukungan *Sms Gateway* pada Smk Kridawisata Bandar Lampung
7	(Billy B. Sumolang, Steven R. Sentinuwo, Xaverius B. N. Najoan)	2018	Aplikasi Absensi Jemaat Berbasis *Android*

2.1.1 Literatur 1

Beberapa kesimpulan dapat diambil dari penelitian ini.

1. Saat melakukan *verifikasi* terhadap absensi dengan menggunakan kode *barcode* melalui aplikasi *android*, kelemahan aplikasi absensi manual ini dapat diatasi dengan menggunakan formulir absensi yang ada di kertas sehingga data absensi dapat terdata dengan baik dan akurat.

2. Data kehadiran yang dimasukkan dalam data berupa data siswa, jadwal atau presensi, dan waktu selama prosedur kehadiran sehingga petugas BAAK dan pengguna terkait dapat bergerak dengan cepat diberitahukan dalam prosedur kehadiran.
3. Data absensi disimpan dalam *file database* yang dapat diakses langsung dari BAAK, sehingga proses rekapitulasi kehadiran menjadi lebih mudah, cepat dan akurat, sehingga tidak perlu lagi perhitungan manual dalam *proses* rekapitulasi kehadiran.
4. Aplikasi tidak dapat merekam lokasi titik pengguna yang ada (titik peta) dan tidak terkait dengan masalah keuangan terkait masalah ringkasan kehadiran yang dapat dilihat oleh instruktur (Fitri Ayu, 2020).

2.1.2 Literatur 2

Berdasarkan hasil penelitian pembuatan Aplikasi Absensi Guru Pada Sekolah Berbasis *Android* Dengan Keamanan *Qr Code* ini maka dapat diambil kesimpulan yaitu :

1. Penelitian ini menghasilkan Aplikasi Absensi Guru Pada Sekolah Berbasis Android Dengan Keamanan *Qr Code* yang mana dapat dapat memproses absensi. Dalam hasil uji coba sistem yang telah selesai dan berhasil dibuat, sistem akan segera diimplementasikan pada smartphone *Android*.
2. Aplikasi ini dapat memberikan informasi tentang absensi dalam bentuk grafik.

Dari kesimpulan yang dijelaskan, maka dapat dikemukakan beberapa saran untuk pengembangan Aplikasi Absensi Guru Pada Sekolah Berbasis Android Dengan Keamanan *Qr Code* ini yang lebih baik, diantaranya yaitu:

1. Untuk pengembangan selanjutnya aplikasi ini dapat dibuat untuk menentukan posisi guru yang sedang melakukan pengambilan absensi.
2. Penambahan fitur menarik yang tentunya mengikuti perkembangan teknologi kedepannya.
3. Menambahkan fitur chat pada aplikasi agar admin dapat berkomunikasi dengan guru (Uci Rahmalisa, 2020).

2.1.3 Literatur 3

Berdasarkan hasil penelitian maka dapatdiambil kesimpulan antara lain:

1. Data absensi siswa sudah dapat tersimpan dalam *database* sehingga mempermudah dalam membuat laporan Daftar kehadiran siswa.
2. Mengintegrasikan informasi dari proses absensi kedalam sistem yang akan dibangundan merancang suatu sistem agar lebih cepat, mudah dan tepat dalam proses absen sisiswa.
3. Kecurangan dalam proses absensi bisa diminimalisir karena tidak menggunakana bsensi secara tertulis (Nur Rubiati, 2019).

2.1.4 Literatur 4

Aplikasi Absensi ini dibangun dengan metode pengembangan perangkat lunak *Waterfall*. Data absensi yang diinputkan pada aplikasi disimpan oleh sistem pada *Google Sheet*. Hasil pengujian terhadap fitur-fitur di dalam aplikasi menunjukkan Aplikasi Absensi dapat berfungsi sesuai tujuannya. Saran bagi penelitian selanjutnya adalah meningkatkan teknik pengelolaan rekamandata absensi. Pengelolaan data absensi dapat dibuat pada suatu sistem basis data, agar data absensi dapat diolah lebih *optimal* lagi untuk memenuhi kebutuhan pihak sekolah (Sherly Christina, 2019).

2.1.5 Literatur 5

Adapun kesimpulan yang dapat diambil dari penelitian sistem absensi ini mampu memperbaharui metode absensi guru yang telah berjalan pada Sekolah SMP Negeri 1 bulu kumba. Perancangan yang dilakukan telah menghasilkan sebuah Sistem Absensi Guru Menggunakan *Qr Code Scanner* Berbasis *Android* yang dapat memproses absensi, selain itu aplikasi ini dapat memproses absensi dengan tepat dan cepat. Dalam hasil uji coba *system* yang telah selesai dan berhasil dibuat, *system* akan segera di implementasikan pada *smartphone Android*.

Saran Aplikasi sistem absensi guru pada sekolah dengan *Qr Code* menggunakan berbasis *android* adalah aplikasi absen berbasis android pertama yang dirancang di SMP Negeri 1 Bulu kumba. Oleh karena itu diperlukan pengembangan kearah yang lebih baik untuk meningkatkan keefektifan penggunaan aplikasi. Adapun saran saran terhadap pengembangan aplikasi:

1. Untuk pengembangan selanjutnya aplikasi ini dapat dibuat untuk menentukan posisi guru yang sedang melakukan pengambilanabsen.
2. Penambahan fitur menarik yang tentunya mengikuti perkembangan teknologi kedepannya.
3. Menambahkan fitur chat pada aplikasi agar admindapat dapat berkomunikasi dengan guru (Fachrival Mustari, 2018).

2.1.6 Literatur 6

Database yang dirancang di gunakan untuk menyimpan data presensi siswa sesuai dengan kebutuhan SMK Kridawisata. Aplikasi absensi siswa yang di rancangan merupakan aplikasi berbasis web dengan bahasa pemrograman *php, databae MySQL*, dan dilengkapi dengan dukungan *SMS Gateway* untuk

memberikan informasi kepada wali murid, bagian kesiswaan ataupun pihak pengguna lainnya. Keluaran aplikasi adalah rekapitulasi absensi siswa per hari, per bulan, dan per semester yang digunakan untuk pelaporan kehadiran siswa (Sukatmi, 2018).

2.1.7 Literatur 7

Berdasarkan hasil pengujian sistem yang telah dilakukan untuk aplikasi absensi jemaat berbasis *android* maka dapat disimpulkan sebagai berikut:

1. Aplikasi Absensi Jemaat Berbasis *Android* dengan menggunakan metode *Rapid Application Development (RAD)* berhasil dibangun. Presensi anggota jemaat dapat di aplikasikan dengan menggunakan android yang dijalankan pada ponsel pintar pada setiap peribadatan di tingkat kolom.
2. Pelaporan persembahan dapat di implementasikan dengan menggunakan android yang dijalankan pada ponsel pintar pada setiap peribadatan di tingkat kolom.
3. Aplikasi dapat berjalan dengan menggunakan *koneksi internet* (Billy B. Sumolang, 2018).

2.2 Absensi

Absensi adalah suatu pendataan kehadiran, bagian dari pelaporan aktivitas suatu institusi, suatu komponen institute itu sendiri yang berisi data-data kehadiran yang disusun dan diatur sedemikian rupa sehingga mudah untuk dicari dan dipergunakan apabila sewaktu-waktu di perlukan oleh pihak yang berkepentingan (Wardoyo, dkk 2016). Sedangkan menurut (Arya, dkk 2020) Absensi dapat dikatakan suatu pendataan kehadiran yang merupakan bagian dari aktifitas pelaporan yang ada dalam sebuah institusi. Dalam sebuah perusahaan memerlukan kebijakan terutama tingkat kedisplinan pegawai. Kedisiplinan dari pegawai

merupakan tolak ukur utama dalam melihat kinerja pegawai berdasarkan kehadirannya di perusahaan. Sebuah perusahaan harus memiliki sistem absensi kehadiran pegawai yang dapat mengatur kehadiran pegawai berdasarkan kewajiban, larangan, dan sanksi apabila kewajiban seorang pegawai tidak ditaati atau dilanggar.

2.3 Location Based Service

Layanan Berbasis Lokasi atau lebih dikenal dengan *Location Based Service (LBS)* istilah umum yang digunakan untuk menggambarkan teknologi yang digunakan untuk menemukan lokasi perangkat yang kita gunakan. *LBS* adalah layanan informasi yang dapat diakses melalui *mobile device* dengan menggunakan *mobile network*, yang dilengkapi kemampuan untuk memanfaatkan lokasi dari *mobile device* tersebut. Terdapat dua unsur utama pada *LBS* yaitu:

a. *Location Manager (API Maps)* Menyediakan *tools/source* untuk *LBS*, *Application Programming Interface (API) Maps* menyediakan fasilitas untuk menampilkan, memanipulasi maps/peta beserta *feature – feature* lainnya seperti tampilan satelit, *street* (jalan), maupun gabungannya. Paket ini berada pada *com.google.adroid.maps*.
b. *Location Provider (API Location)* Menyediakkan teknologi pencarian lokasi yang digunakan oleh *device*/perangkat. *API Location* berhubungan dengan data *GPS (Global Positioning System)* dan data lokasi *real-time*. *API Location* berada pada paket *android* yaitu dalam paket *android.location*. Dengan *Location Manager*, kita dapat menentukan lokasi kita saat ini, *track* gerakan/perpindahan, serta kedekatan dengan lokasi tertentu dengan mendeteksi perpindahan.

2.4 Global Position System

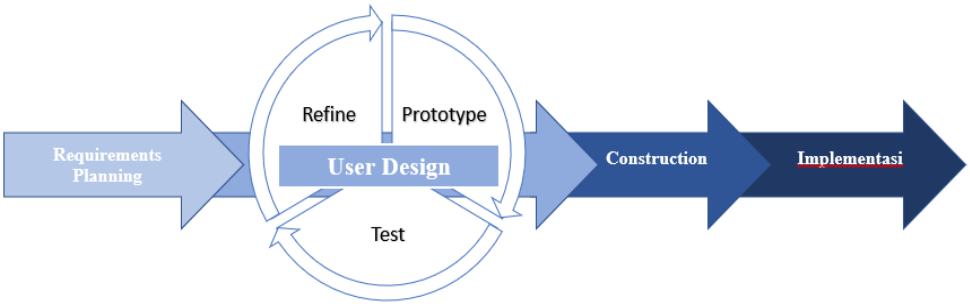
GPS adalah singkatan dari *Global Positioning System* merupakan sistem untuk menentukan posisi dan navigasi secara *global* dengan menggunakan satelit. Sistem yang pertama kali dikembangkan oleh Departemen Pertahanan Amerika Serikat ini digunakan untuk kepentingan militer maupun sipil (Winardi, 2006). *GPS* adalah satu-satunya sistem satelit *navigasi* global untuk penentuan lokasi, kecepatan, arah, dan waktu yang telah beroprasi secara penuh didunia saat ini. *GPS* menggunakan konstelasi 27 buah satelit yang mengorbit bumi, dimana sebuah *GPS receiver* menerima informasi dari tiga atau lebih satelit tersebut untuk menentukan posisi. *GPS receiver* harus berada *dalam line-of sight (LoS)* terhadap ketiga satelit tersebut untuk menentukan posisi, sehingga *GPS* hanya ideal untuk diguakan dalam *outdoor positioning.*

2.5 Metode Pengembangan Sistem

2.5.1 Metode *Rapid Application Development*

Menurut (Academy, 2020) metode *Rapid Application Development* salah satu metode pengembangan aplikasi yang kerap dipakai saat ini. Metode ini menekankan dalam proses pembuatan aplikasi berdasarkan pembuatan *prototype, iterasi,* dan *feedback* yang berulang-ulang. Dengan begitu, aplikasi yang dibuat bisa dikembangkan dan diperbaiki dengan cepat. Sangat cocok dengan kebutuhan dan perkembangan dunia digital yang super cepat. Seperti yang dikatakan sebelumnya ,metode ini lebih fokus pada pembuatan *prototype* secara cepat dan mengandalkan *feedback* dari *user,* dan keuntungan utama menjalankan metode *rapid application development* adalah jangka waktu pengembangan lebih cepat dikarenakan *feedback* dari *user* cepat didapatkan dan semua perubahan yang dilakukan sesuai hasil

tersebut. Adapun tahapan Metode *Rapid Application Development* dapat dilihat pada gambar 2.1.



Gambar 2. 1 Tahapan Metode *Rapid Application Development*

Berdasarkan Gambar 2.1 ada empat tahapan metode *Rapid Application Development* yang perlu dilalui Ketika mengembangkan aplikasi. Keempat tahapan itu adalah:

1. Menentukan *Project Requirements*

Menentukan *project requirement* merupakan proses awal yang dilakukan untuk menentukan kebutuhan apa saja yang diperlukan untuk membangun sebuah *project*, setelah mendapatkan kebutuhan yang jelas, barulah menetukan hal-hal yang mendetail. Contoh seperti tujuan, *timeline*, dan *budget* yang diperlukan. Pada intinya, tahapan awal ini berguna untuk memberikan gambaran pada *project* yang akan di kerjakan nanti.

Pada tahap ini, akan dilakukan pengumpulan informasi dengan melakukan *observasi* dan wawancara pada pihak SMA Negeri 2 Oku Tanzania untuk mengetahui apa saja kebutuhan dan bagaimana cara mengatasi kebutuhan *user* tersebut.

2. Membuat *Prototype*

Pada tahapan selanjutnya adalah membuat *prototype*, dimana *developer* secepat mungkin akan membuat *prototype* dari aplikasi yang diinginkan. Lengkap dengan fitur dan fungsi yang berbeda-beda. Tujuannya, sekedar untuk memberitahu apakah *prototype* yang dibuat sudah sesuai dengan keinginan dan kebutuhan *user*. Dan juga nanti akan melibatkan *user* untuk *testing* dan memberikan *feedback*, proses ini memungkinkan untuk mempelajari *error* yang mungkin akan muncul kedepannya, dan juga tahap ini berguna untuk mengurangi *error* dan *debugging*.

3. *Rapid Construction* dan Pengumpulan *Feedback*

Pada tahapan ketiga ini pengumpulan *feedback* yang diberikan oleh *user*. *Feedback* yang dimaksud disini mencakup fitur, fungsi, *visual*, dan juga *interface* dari program yang sedang dikembangkan. Setelah itu, *Prototype* akan dikembangkan lagi sampai klien memberikan persetujuan untuk finalisasi produk. Dan seperti bahasan sebelumnya, kedua tahapan ini akan diulang terus-menerus, sampai hasilnya maksimal dengan keinginan klien.

4. *Implementasi* dan finalisasi

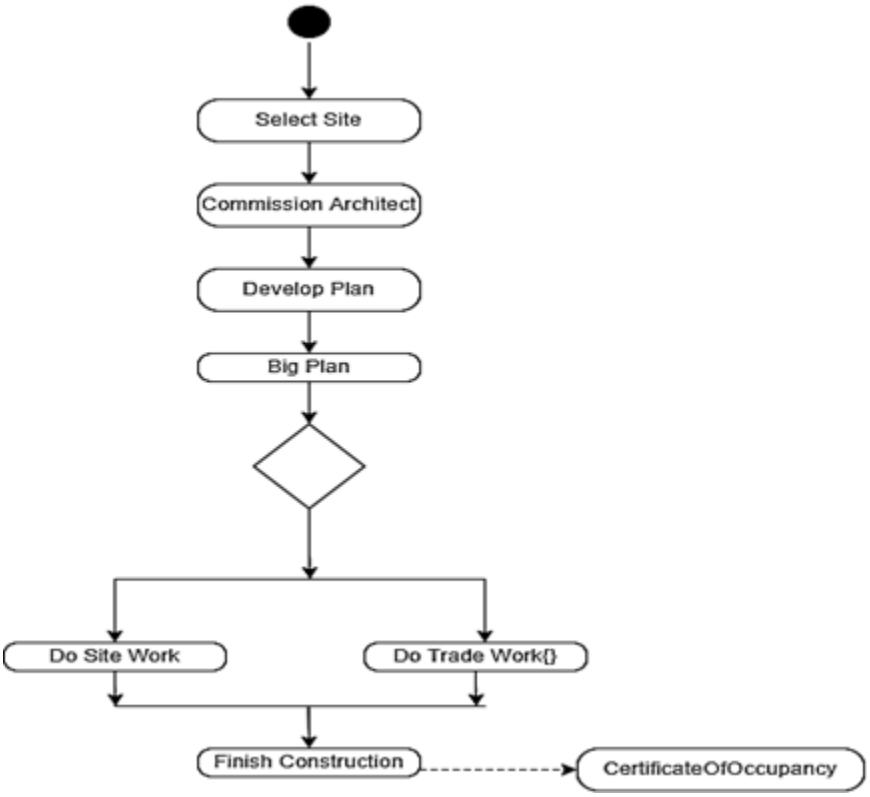
Langkah terakhir adalah *implementasi* hasil *feedback* yang diberikan *user* dan membuat *project* akhir. Dimana tugas utama develover adalah menutupi kekurangan yang mungkin terjadi Ketika proses pengembangan aplikasi nanti, tugas ini penting dikarenakan melakukan *optimasi* untuk *stabilitas* aplikasinya, memperbaiki *interface*, hingga melakukan *maintenance* dan menyusun dekumentasi.

2.6 Unified Modelling Language (UML)

Menurut (Booch et al., 1998), *Unified Modeling Language* (UML) adalah bahasa standar untuk menulis cetak biru perangkat lunak. *UML* dapat digunakan untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan *artefak* dari perangkat lunak yang intensif sistem.

1. *Activity Diagram*

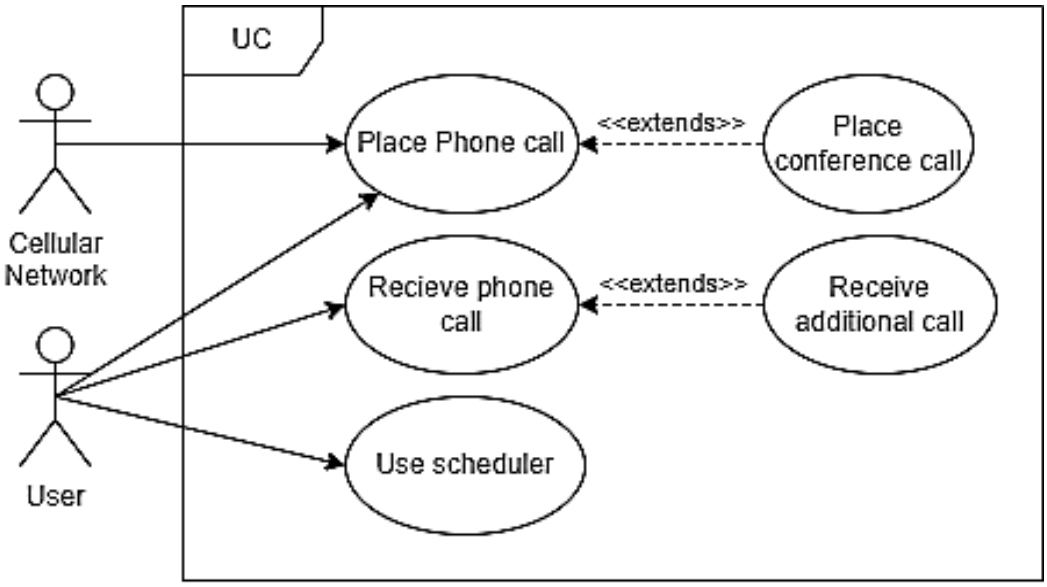
Diagram aktivitas menurut (Booch, 2005), adalah jenis diagram *statechart* khusus yang menunjukkan aliran dari aktivitas ke aktivitas dalam suatu sistem. Diagram aktivitas membahas tampilan dinamis dari suatu sistem. Diagram ini sangat penting dalam memodelkan fungsi sistem dan menekankan aliran control di antara objek, (Booch, 2005). Adapun contoh *Activity Diagram* dapat dilihat pada Gambar 2.2.



Gambar 2. 2 Contoh *Activity Diagram*
(Sumber: (Booch, 2005))

2. *Use Case Diagram*

Menurut (Booch, 2005), diagram *use case* menunjukkan satu *set use case* dan *actor* (jenis kelas khusus) dan *use case* saling hubungan. Diagram *use case* membahas tampilan *use case* statis dari suatu sistem. Diagram ini sangat penting dalam mengatur dan memodelkan perilaku sistem. Adapun contoh *Use Case Digram* dapat dilihat pada Gambar 2.3.



Gambar 2. 3 Contoh *Use Case Diagram*
(Sumber: (Booch, 2005))

Diagram *use case* hanyalah jenis diagram khusus dan memiliki properti umum yang sama seperti mengerjakan diagram lainnya,nama dan konten grafis yang merupakan proyeksi ke dalam model. Apa yang membedakan diagram *use case* dari semua jenis diagram lainnya adalah konten khususnya.Diagram *use case* biasanya berisi:

a. Kasus penggunaan

b. Aktor

c. Ketergantungan, generalisasi, dan hubungan asosiasi.

3. *Class Diagram*

Menurut (Booch, 2005), *Diagram* kelas adalah diagram yang paling umum ditemukan dalam pemodelan sistem berorientasi objek. Sebuah diagram kelas menunjukkan sekumpulan kelas, antarmuka, dan kolaborasi serta hubungannya. Adapun contoh *activity diagram* dapat dilihat pada Gambar 2.4.

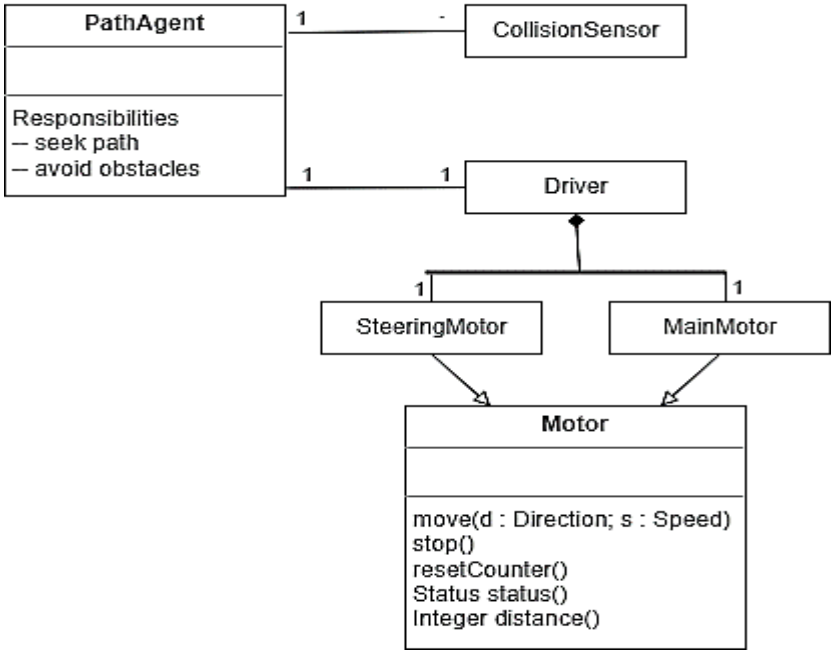
Diagram kelas biasanya berisi hal-hal berikut:

a. Kelas

b. Antarmuka

c. Kalaborasi

d. Ketergantungan, generalisasi, dan hubungan asosiasi.



Gambar 2. 4 Contoh *Class Diagram*
(Sumber: (Booch, 2005))

2.7 Analisis PIECES

Menurut (Suharto, 2018), "Metode PIECES Untuk mengidentifikasi masalah, maka harus dilakukan analisis terhadap kinerja, informasi, ekonomi, pengendalian, efisiensi, dan pelayanan. Panduan ini dikenal dengan analisis PIECES

(*Performance, Information, Economy, Control, Efficiency, Service*). Analisis dilakukan pada sistem informasi lama yang berupa hard copy seperti brosur apabila band tersebut akan mengadakan pentas. Dari analisis ini biasanya didapatkan beberapa masalah dan akhirnya dapat ditemukan masalah utamanya.

Menurut (Ita Dewi Sintawati, 2020),berikut ini merupakan komponen komponen dari analisis PIECES:

a. Analisis Kinerja Sistem (*Performance*) Merupakan suatu kemampuan sistem dalam menyelesaikan tugas dengan cepat sehingga sasaran dapat segera tercapai.
b. Analisa Informasi (*Information*) Hal penting karena dengan informasi tersebut pihak manajemen (*marketing*) dan *user* dapat melakukan langkah selanjutnya. Apabila kemampuan sistem informasi baik, maka *user* akan mendapatkan informasi yang akurat, tepat waktu dan relevan sesuai dengan yang diharapkan.
c. Analisis Ekonomi (*Economy*) Pemanfaatan biaya yang digunakan dari pemanfaatan informasi. Peningkatan terhadap kebutuhan ekonomis mempengaruhi pengendalian biaya dan peningkatan manfaat.
d. Analisis Pengendalian (*Control*) Analisis ini digunakan untuk membandingkan sistem yang dianalisa berdasarkan pada segi ketepatan waktu, kemudahan akses, dan ketelitian data yang diproses.
e. Analisis Efisiensi (*Efficiency*) Efisiensi berhubungan dengan bagaimana sumber tersebut dapat digunakan secara optimal. Operasi pada suatu perusahaan dika-takan efisien atau tidak biasanya didasarkan pada tugas dan tanggung jawabdalam melaksanakan kegiatan.

f. Analisis Pelayanan (*Service*) Peningkatan pelayanan memperlihatkan kategori yang beragam. Proyek yang dipilih merupakan peningkatan pelayanan yang lebih baik bagi manajemen (*marketing*), user dan bagian lain yang merupakan simbol kualitas darisuatu sistem informasi

2.8 *Android Studio*

Android Studio adalah *Integrated Development Environment (IDE)* yakni *software* yang bisa digunakan untuk mengembangkan aplikasi *android*. *Android Studio* awalnya muncul pada tahun 2013 dan diperkenalkan di acara *Google I/O Conference. Software* yang dikembangkan oleh *JetBrains* dan dirilis pertama kali ke publik pada tahun 2014. *Android Studio* menjadi *software* resmi yang didukung penuh oleh *Google* sebagai perusahaan induk *Sistem Operasi Android*.

2.9 *Visual Studio Code*

Visual Studio Code (VSCode) ini adalah sebuah teks editor ringan dan handal yang dibuat oleh *Microsoft* untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi *Linux, Mac, dan Windows*. *Teks editor* ini secara langsung mendukung bahasa pemrograman *JavaScript, Typescript, dan Node.js*, serta bahasa pemrograman lainnya dengan bantuan *plugin* yang dapat dipasang *via marketplace Visual Studio Code (seperti C++, C#, Python, Go, Java, dst).*

Banyak sekali fitur-fitur yang disediakan oleh *Visual Studio Code*, diantaranya *Intellisense, Git Integration, Debugging*, dan fitur ekstensi yang menambah kemampuan *text editor*. Fitur-fitur tersebut akan terus bertambah seiring dengan bertambahnya versi *Visual Studio Code*. Pembaruan versi *Visual Studio Code* ini juga dilakukan berkala setiap bulan, dan inilah yang membedakan *VS Code* dengan teks editor-teks editor yang lain.

2.10 *Object Oriented Programming*

Menurut (Techtarget, 2018) *Object Oriented Programming (OOP)* adalah suatu metode pemrograman yang berorientasi pada objek. Program-program yang telah ada merupakan gabungan dari beberapa komponen-komponen kecil yang sudah ada sebelumnya. Hal itu dapat mempermudah pekerjaan seorang programmer dalam melakukan pengembangan program.Objek-objek yang saling berkaitan dan disusun kedalam satu kelompok ini disebut dengan *class*.

Nantinya, objek-objek tersebut akan saling berinteraksi untuk menyelesaikan masalah program yang rumit.Jika sebelumnya *developer* harus berfokus pada *logic* yang akan dimanipulasi, dengan *OOP*, *developer* dapat lebih terfokus pada objeknya saja untuk dimanipulasi. Pendekatan ini menawarkan cara yang mudah untuk menangani kerumitan suatu pemrograman.Tujuan utama *OOP* adalah untuk mengatasi kelemahan pendekatan pemrograman konvensional.

2.11 My SQL

MySQL merupakan sebuah *database developer* yang juga bersifat *free*, *MySQL* banyak digunakan sebagai *database* karena mudah digunakan dan juga sangat banyak tersedia. *MySQL* sendiri menggunakan bahasa *SQL* yang saat ini sudah banyak digunakan.

MySQL merupakan *software database* yang termasuk paling popular di lingkungan *Linux atau Unix*, kepopuleran ini ditunjang karena *query* dari basis data yang saat itu bisa dikatakan paling cepat dan juga memiliki sedikit permasalahan (Wiguna et al., 2019)

2.12 Pengujian ISO 25010

Menurut (Wattiheluw, dkk., 2019), pengujian ISO 25010 merupakan bagian dari *Systems and Software Quality Requirements and Evaluation* (SQuaRE) yang merupakan versi lanjutan dari ISO 91261, yang telah direvisi secara teknis dengan menambahka beberapa struktur dan bagian dari standar model kualitas. Tujuan dari penggunaan kualitas ini adalah untuk mengukur sejauh mana produk atau sistem tersebut bisa digunakan oleh pengguna untuk memenuhi kebutuhan dalam mencapai tujuan yang diinginkan dengan efisiensi, efektivitas, kepuasan dalam konteks penggunaan yang spesifik, dan bebas dari resiko.

Menurut (Harun, 2018), ISO 25010 terdiri dari delapan karakteristik yang dibagi menjadi beberapa bagian yang berhubungan dengan sifat-sifat statis perangkat lunak dan sifat dinamis dari sistem komputer, yang dapat ditunjukkan pada gambar dibawah ini:

1. *Functional Suitability*, merupakan sistem atau produk yang memberikan fungsional untuk memenuhi kebutuhan saat sistem atau produk tersebut digunakan pada keadaan tertentu.
2. *Reliability*, merupakan tingkat dimana suatu sistem atau produk dapat mempertahankan kinerjanya pada level tertentu ketika digunakan pada keadaan tertentu.
3. *Performance Efficiency*, merupakan tingkat dimana sistem atau produk menyediakan performa yang baik dengan sejumlah *resource* yang akan digunakan pada sistem atau produk.
4. *Usability*, merupakan tingkat dimana pada suatu sistem atau produk mudah dimengerti, mudah dipakai, dan menarik untuk digunakan.

5. *Security*, merupakan tingkat dimana pada suatu sistem atau produk menyediakan layanan untuk melindungi akses, penggunaan, modifikasi, pengrusakan, ataupun pengungkapan yang berbahaya.
6. *Compatibality*, merupakan kemampuan pada suatu komponen atau sistem untuk bertukar informasi.
7. *Maintainability*, merupakan tingkat dimana pada suatu sistem atau produk dapat dimodifikasi, yang meliputi perbaikan, pengembangan untuk menyesuaikan dengan lingkungan, modifikasi pada kriteria, dan spesifikasi fungsi.
8. *Portability*, merupakan tingkat dimana pada suatu sistem atau produk dapat dipindahkan dari satu ruang ke ruang lainnya.