

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Dibawah ini beberapa penelitian yang pernah dilakukan, dapat dilihat pada dibawah ini :

I. Pratama (2016), meneliti tentang Penerapan Sistem Informasi Posyandu Mawar Kelurahan Simpang Empat. Pencatatan yang masih dilakukan secara manual mengakibatkan kader merasa kesulitan dalam memonitoring seluruh peserta. Penulisan ini memaparkan sebuah aplikasi yang dapat membantu dan mempermudah kader dalam pencatatan kegiatan Posyandu khususnya kesehatan ibu dan anak. Alur dari penggunaan aplikasi ini diadopsi dari alur kegiatan yang ada di Posyandu, dimulai pada saat ibu datang ke Posyandu dan mendaftar sebagai ibu hamil, kemudian anak lahir dan didata oleh kader yang selanjutnya anak melakukan penimbangan setiap bulan sampai usia lima tahun. Data penimbangan anak yang dimasukkan ke dalam sistem meliputi data penimbangan berat dan tinggi badan, vitamin A, imunisasi, dan keluhan kesehatan. Selanjutnya adalah pendataan ibu yang akan mengikuti program Keluarga Berencana (KB). Berkaitan dengan hal tersebut, aplikasi ini menyediakan form masukan yaitu data identitas ibu dan metode KB yang akan digunakan. Hasil dari pengujian sistem didapatkan bahwa aplikasi dapat membantu kader dalam melakukan kegiatan di Posyandu.

2. Savitri and Nurwasito (2018) Pengembangan Aplikasi Mobile untuk Pelayanan Administrasi Posyandu dengan menggunakan Google Maps Api Geolocation Tagging. Aplikasi yang dikembangkan dapat digunakan untuk menggantikan sistem administrasi Posyandu yang masih bersifat manual dan konvensional. Proses pencatatan data balita, jadwal imunisasi, dan perkembangan balita dapat dilakukan dengan mudah dan cepat menggunakan aplikasi Android pada smartphone. Dengan aplikasi ini, kegiatan administrasi di Posyandu menjadi lebih efisien dan sekaligus mendorong ibu-ibu untuk memanfaatkan smartphone dengan aplikasi-aplikasi yang lebih produktif.
3. Sundari (2016), meneliti tentang Sistem Informasi Pelayanan Puskesmas Berbasis Web. Sistem informasi pasien pada puskesmas adalah sistem informasi yang mempunyai kegiatan antrian, registrasi, dan rekam medis pasien. Adapun kinerja sistem dalam pelayanan pasien yang berjalan pada puskesmas secara umum belum optimal karena masih pada pengolahan data pasien dan data rekam medis masih menggunakan media pembukuan atau manual. Maka pelayanan pasien pada puskesmas menjadi tidak efektif dan efisien, karena media pembukuan memperlambat pembuatan laporan. Metode pengembangan sistem pelayanan pasien pada puskesmas menggunakan metode waterfall dengan alat perancangan ERD (Entity Relationship Diagram) dan LRS (Logical Record Structure). Implementasi program menggunakan bahasa PHP dengan database menggunakan MySQL
4. Nabila and Kusumadewi (2015) meneliti tentang Sistem Informasi Posyandu Kesehatan Ibu Dan Anak, Penulisan ini memaparkan sebuah aplikasi yang

dapat membantu dan mempermudah kader dalam pencatatan kegiatan Posyandu khususnya kesehatan ibu dan anak. Alur dari penggunaan aplikasi ini diadopsi dari alur kegiatan yang ada di Posyandu, dimulai pada saat ibu datang ke Posyandu dan mendaftar sebagai ibu hamil, kemudian anak lahir dan didata oleh kader yang selanjutnya anak melakukan penimbangan setiap bulan sampai usia lima tahun. Data penimbangan anak yang dimasukkan ke dalam sistem meliputi data penimbangan berat dan tinggi badan, vitamin A, imunisasi, dan keluhan kesehatan. Selanjutnya adalah pendataan ibu yang akan mengikuti program Keluarga Berencana (KB). Berkaitan dengan hal tersebut, aplikasi ini menyediakan form masukan yaitu data identitas ibu dan metode KB yang akan digunakan. Hasil dari pengujian sistem didapatkan bahwa aplikasi dapat membantu kader dalam melakukan kegiatan di Posyandu.

5. Muflikhah, Jauhari, and Indriati (2018) meneliti tentang Peningkatan Manajemen Data Melalui Sistem Aplikasi Posyandu Di Kecamatan Lowokwaru Malang. Sistem ini ditujukan untuk pencatatan selama kegiatan posyandu dan pemberian informasi seputar kegiatan informasi serta perkembangan balita. Adapun tahapan kegiatan dimulai dengan surveykegiatan terkait dengan analisis kebutuhan sistem, perancangan dan implmentasi sistem, tahapan selanjutnya adalah pelatihan terhadap pemakai aplikasi, baik bagi kader posyandu maupun ibu balita. Tahap akhir merupakan evaluasi terhadap tingkat kegunaan sistem bagi petugas dan peserta posyandu. Berdasarkan hasil kuisisioner diperoleh bahwasannnya

sistem aplikasi tersebut telah memenuhi kebutuhan untuk dioperasikan secara efektif (83.67% responden), efisien (65% responden) dengan kemudahan (89.3% responden) serta 79% menyatakan tingkat kepuasan yang relatif tinggi.

2.2. Sistem Informasi Manajemen

Sistem Informasi Manajemen (SIM) merupakan sistem yang mengolah serta mengorganisasikan data dan informasi yang berguna untuk mendukung pelaksanaan tugas dalam suatu organisasi. Sistem informasi manajemen atau SIM (bahasa Inggris: management information system, MIS) adalah sistem perencanaan bagian dari pengendalian internal suatu bisnis yang meliputi pemanfaatan manusia, dokumen, teknologi, dan prosedur oleh akuntansi manajemen untuk memecahkan masalah bisnis seperti biaya produk, layanan, atau suatu strategi bisnis. Sistem informasi manajemen dibedakan dengan sistem informasi biasa karena SIM digunakan untuk menganalisis sistem informasi lain yang diterapkan pada aktivitas operasional organisasi (Hariyanto 2020).

2.2.1. Fungsi Sistem Informasi Manajemen

Beberapa kegunaan/fungsi sistem informasi manajemen antara lain adalah sebagai berikut (Setiawan 2019):

1. Meningkatkan aksesibilitas data yang tersaji secara tepat waktu dan akurat bagi para pemakai, tanpa mengharuskan adanya prantara sistem informasi.
2. Menjamin tersedianya kualitas dan keterampilan dalam memanfaatkan sistem informasi secara kritis.
3. Mengembangkan proses perencanaan yang efektif.

4. Mengidentifikasi kebutuhan-kebutuhan akan keterampilan pendukung sistem informasi.
5. Menetapkan investasi yang akan diarahkan pada sistem informasi.
6. Mengantisipasi dan memahami konsekuensi-konsekuensi ekonomis dari sistem informasi dan teknologi baru.
7. Memperbaiki produktivitas dalam aplikasi pengembangan dan pemeliharaan sistem.
8. Organisasi menggunakan sistem informasi untuk mengolah transaksi-transaksi, mengurangi biaya dan menghasilkan pendapatan sebagai salah satu produk atau pelayanan mereka.
9. Bank menggunakan sistem informasi untuk mengolah cek-cek nasabah dan membuat berbagai laporan rekening koran dan transaksi yang terjadi.
10. Perusahaan menggunakan sistem informasi untuk mempertahankan persediaan pada tingkat paling rendah agar konsisten dengan jenis barang yang tersedia.
11. SIM untuk Pendukung Pengambilan Keputusan Sebuah sistem keputusan, yaitu model dari sistem dengan mana keputusan diambil, dapat tertutup atau terbuka. Sebuah sistem keputusan tertutup menganggap bahwa keputusan dipisah dari masukan yang tidak diketahui dari lingkungan.
12. SIM Berdasarkan Aktivitas/Kegiatan Manajemen Kegiatan dan proses informasi untuk tiga tingkat adalah saling berhubungan. Contohnya pengendalian inventaris pada tingkatan operasional bergantung pada proses yang tepat dari transaksi; pada tingkat dari pengendalian manajemen,

pembuatan keputusan tentang keamanan persediaan dan frekuensi memesan lagi bergantung pada pembetulan ringkasan dari hasil operasi- operasi; pada tingkat strategi, hasil dalam operasi-operasi dan pengendalian manajemen yang dihubungkan pada tujuan-tujuan strategi, saingan tindak tanduk dan sebagainya untuk mencapai strategi inventaris. Tampaknya terdapat kontras tajam antara ciri-ciri informasi untuk perencanaan pengendalian dan taktis berada di tengahnya.

13. Sistem Informasi Untuk Pengendalian Operasional. Pengendalian operasional adalah proses pemantapan agar kegiatan operasional dilaksanakan secara efektif dan efisien. Pengendalian operasional menggunakan prosedur dan aturan keputusan yang sudah ditentukan lebih dahulu. Sebagian besar keputusan bisa diprogramkan.
14. Sistem Informasi Untuk Pengendalian Manajemen. Informasi pengendalian manajemen diperlukan oleh manajer departemen untuk mengukur pekerjaan, memutuskan tindakan pengendalian, merumuskan aturan keputusan baru untuk diterapkan personalia operasional, dan mengalokasikan sumber daya.

2.2.2. Tujuan Sistem Informasi Manajemen

Tujuan sistem informasi manajemen antara lain adalah sebagai berikut (Setiawan 2019):

1. Menyediakan informasi yang dipergunakan di dalam perhitungan harga pokok jasa, produk, dan tujuan lain yang diinginkan manajemen.
2. Menyediakan informasi yang dipergunakan dalam perencanaan, pengendalian, pengevaluasian, dan perbaikan berkelanjutan.

3. Menyediakan informasi untuk pengambilan keputusan.

2.2.3. Manfaat Sistem Informasi Manajemen

Sistem Informasi Manajemen memiliki sejumlah manfaat yang bisa diambil oleh para penggunanya, berikut ini beberapa manfaat dari sistem informasi manajemen (Setiawan 2019):

1. Dapat meningkatkan aksestabilitas data yang disajikan dengan akurat dan tepat waktu untuk para pengguna informasi tanpa melalui perantara sistem informasi
2. Sistem Informasi Manajemen bisa menjamin kualitas dan keterampilan didalam memanfaatkan sistem informasi manajemen secara kritis
3. Sistem informasi manajemen dapat mengantisipasi serta memahami konsekuensi ekonomis dari sebuah sistem informasi dan tekhnologi baru
4. Mengembangkan sebuah perencanaan yang lebih efektif
5. Mengidentifikasi kebutuhan pendukung sistem informasi
6. Menetapkan investasi yang nantinya akan diarahkan kepada sistem informasi
7. Memperbaiki produktifitas didalam aplikasi pengembangan dan maintenance sistem
8. Mengolah seluruh transaksi, mengurangi biaya serta menciptakan pendapatan

2.2.4. POAC

Fungsi POAC sendiri dalam suatu organisasi adalah untuk meningkatkan efektifitas dan efisiensi suatu organisasi dalam pencapaian tujuannya. Berikut

adalah pemaparan singkat tentang tiap bagian dari POAC, yang mana akan dibahas lebih dalam di bab lain (Dakhi, 2016):

1. **Planning** meliputi pengaturan tujuan dan mencari cara bagaimana untuk mencapai tujuan tersebut. Planning telah dipertimbangkan sebagai fungsi utama manajemen dan meliputi segala sesuatu yang manajer kerjakan. Di dalam planning, manajer memperhatikan masa depan, mengatakan “Ini adalah apa yang ingin kita capai dan bagaimana kita akan melakukannya”. Membuat keputusan biasanya menjadi bagian dari perencanaan karena setiap pilihan dibuat berdasarkan proses penyelesaian setiap rencana. Planning penting karena banyak berperan dalam menggerakkan fungsi manajemen yang lain. Contohnya, setiap manajer harus membuat rencana pekerjaan yang efektif di dalam kepegawaian organisasi.
2. **Organizing** adalah proses dalam memastikan kebutuhan manusia dan fisik setiap sumber daya tersedia untuk menjalankan rencana dan mencapai tujuan yang berhubungan dengan organisasi. Organizing juga meliputi penugasan setiap aktifitas, membagi pekerjaan ke dalam setiap tugas yang spesifik, dan menentukan siapa yang memiliki hak untuk mengerjakan beberapa tugas. Aspek utama lain dari organizing adalah pengelompokan kegiatan ke departemen atau beberapa subdivisi lainnya. Misalnya kepegawaian, untuk memastikan bahwa sumber daya manusia diperlukan untuk mencapai tujuan organisasi. Memekerjakan orang untuk pekerjaan merupakan aktifitas kepegawaian yang khas. Kepegawaian adalah suatu aktifitas utama yang terkadang diklasifikasikan sebagai fungsi yang terpisah dari organizing.

3. **Actuating** adalah peran manajer untuk mengarahkan pekerja yang sesuai dengan tujuan organisasi. Actuating adalah implementasi rencana, berbeda dari planning dan organizing. Actuating membuat urutan rencana menjadi tindakan dalam dunia organisasi. Sehingga tanpa tindakan nyata, rencana akan menjadi imajinasi atau impian yang tidak pernah menjadi kenyataan.
4. **Controlling**, memastikan bahwa kinerja sesuai dengan rencana. Hal ini membandingkan antara kinerja aktual dengan standar yang telah ditentukan. Jika terjadi perbedaan yang signifikan antara kinerja aktual dan yang diharapkan, manajer harus mengambil tindakan yang sifatnya mengoreksi. Misalnya meningkatkan periklanan untuk meningkatkan penjualan. Fungsi dari controlling adalah menentukan apakah rencana awal perlu direvisi, melihat hasil dari kinerja selama ini. Jika dirasa butuh ada perubahan, maka seorang manajer akan kembali pada proses planning. Di mana ia akan merencanakan sesuatu yang baru, berdasarkan hasil dari *controlling*.

2.3. Posyandu

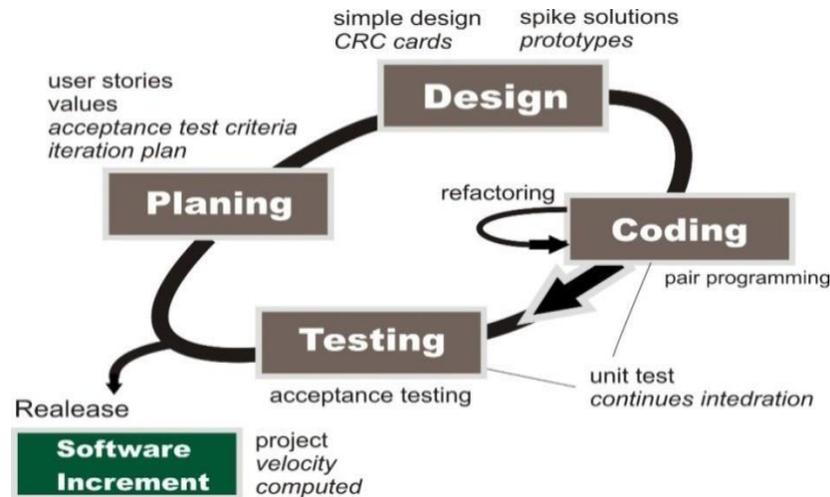
Pos Pelayanan Terpadu atau Posyandu adalah kegiatan yang berhubungan dengan Upaya Kesehatan Bersumber Daya Masyarakat (UKBM) yang diselenggarakan dan dikelola dari oleh, untuk dan bersama masyarakat dalam pembangunan dan penyelenggaraan kesehatan, agar dapat memberdayakan masyarakat dan memberikan kemudahan kepada masyarakat dalam memperoleh pelayanan kesehatan dasar sehingga membantu menurunkan angka kematian ibu dan bayi (Kemenkes 2012).

2.4. Pelayanan

Menurut Wasistiono (2015) pelayanan adalah pemberian jasa baik oleh pemerintah, pihak swasta atas nama pemerintah ataupun pihak swasta kepada masyarakat, dengan atau tanpa pembayaran guna memenuhi kebutuhan dan kepentingan masyarakat.

2.5. Metode Pengembangan *Extreme Programming*

Extreme Programming (XP) adalah metodologi pengembangan perangkat lunak yang ditujukan untuk meningkatkan kualitas perangkat lunak dan tanggap terhadap perubahan kebutuhan pelanggan. Jenis pengembangan perangkat lunak semacam ini dimaksudkan untuk meningkatkan produktivitas dan memperkenalkan pos pemeriksaan dimana persyaratan pelanggan baru dapat diadopsi. Tahapan-tahapan dari *Extreme Programming* terdiri dari *planning* seperti memahami kriteria pengguna dan perencanaan pengembangan, *designing* seperti perancangan *prototype* dan tampilan, *coding* termasuk pengintegrasian, dan yang terakhir adalah *testing* (Pressman 2012). Pressman (2012) menggambarkan tahapan *Extreme Programming* (XP) disajikan pada Gambar 2.1



Gambar 2. 1 Model *Extreme Programming (XP)*

Sumber : (Pressman 2012)

Dibawah ini adalah penjelasan tahapan *Extreme Programming* yaitu :

1. *Planning* (Perencanaan)

Kegiatan Perencanaan (disebut juga *planning game*) biasanya dimulai dengan mendengarkan suatu kegiatan yang bertujuan mengumpulkan kebutuhan-kebutuhan untuk memahami konteks bisnis dan perlunya keluaran-keluaran (*output*), fungsi utama, dan *fungsionalitas*.

Pada perencanaan terdapat *user stories values* yaitu story dengan value tertinggi akan dipindahkan dari jadwal dan diimplementasikan pertama, *acceptance test criteria iteration plan* melakukan perhitungan kecepatan project selama development, customer dapat menambah story, merubah value, membagi story atau menghapusnya.

2. *Design* (Perancangan)

Perancangan yang simple, menarik, dan sederhana selalu memberikan hasil yang lebih disukai daripada gambaran-gambaran yang lebih kompleks.

Perancangan XP memberikan panduan implementasi untuk suatu cerita ketika ditulis, tidak kurang, tidak lebih.

Terdapat *simple design CRC Cards* untuk mengenali dan mengatur *object oriented class* sesuai dengan *software increment* dan *spike solutions prototypes* melakukan spesifikasi solusi dari *object oriented class*.

3. Coding (Pengkodean)

Pengkodean ini dilanjutkan setelah cerita yang telah dikembangkan dan rancangan yang telah dilakukan oleh tim perangkat lunak. Pengkodean ini tidak langsung mengarah ke kode-kode program. Tim akan mengembangkan serangkaian unit pengujian lalu beralih ke pengkodean.

Pada tahapan *pair programming* melakukan kerja sama untuk membuat code dari satu story. Dan *refactoring* adalah proses restrukturisasi kode program computer yang ada tanpa mengubah perilaku eksternalnya.

4. Pengujian (Pengujian)

Unit pengujian yang harus dibuat dan kemudian dijalankan menggunakan kerangka kerja yang memungkinkan mereka untuk diotomatisasi sehingga dapat dijalankan dengan mudah dan dapat dijalankan berulang kali.

Pada tahapan pengujian yaitu *unit test continuous integration* yaitu tahapan pengujian code yang diintegrasikan dengan kerja lainnya dengan pengujian yang dilakukan oleh customer dan focus pada keseluruhan dan fungsional sistem, dan *acceptance testing* yaitu pengujian yang dilakukan *customer stories* yang akan diimplementasikan sebagai bagian dari *software release*.

Selanjutnya terdapat tahapan *software increment project velocity computed*

yaitu tahapan yang telah diimplementasikan dari *software realease* yang nantinya akan diterapkan dalam suatu sistem.

2.6. Pengertian Bahasa Pemodelan Pengembangan Sistem (UML)

Menurut Pressman (2012) *Unified Modeling Language (UML)* adalah bahasa standar untuk menulis perangkat lunak dalam bentuk gambar. *UML* dapat digunakan untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan sebuah sistem perangkat lunak. Beberapa jenis diagram *UML* antara lain sebagai berikut:

2.6.1. Use Case Diagram

Menurut Pressman (2012) *use case* diagram membantu anda menentukan fungsi dan fitur dari perangkat lunak. Dalam diagram ini, gambar yang menyerupai boneka kayu mewakili aktor yang berhubungan dengan kategori dari pengguna. Di dalam diagram *use case*. Para aktor terhubung oleh garis ke *use case* yang mereka kerjakan.

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 2.1 di bawah ini:

Tabel 2. 1 Simbol Diagram *Use Case*

Simbol	Deskripsi
<p><i>Use Case</i></p> 	<p>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 <i>use case</i></p>
Aktor/ <i>actor</i>	Orang, proses atau sistem lain yang

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

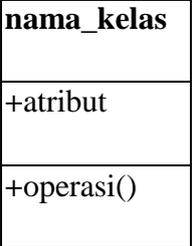
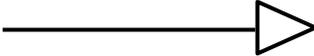
Sumber: Pressman(2012)

2.6.2. Class Diagram

Menurut Pressman (2012) Unsur-unsur utama dari diagram kelas adalah kotak, yang merupakan ikon yang digunakan untuk mewakili kelas dan *interface*. Setiap kotak dibagi menjadi bagian-bagian horisontal. Bagian atas berisi nama kelas. Bagian tengah berisi daftar atribut kelas, dan bagian bawah merupakan

operation dari kelas tersebut. simbol-simbol yang ada pada diagram kelas pada tabel *class diagram 2.2*.

Tabel 2. 2 Simbol *Class Diagram*

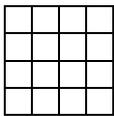
Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Antarmuka/ <i>Interface</i>  nama_interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi/ <i>asociation</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
Asosiasi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

Sumber: (Pressman, 2012)

2.6.3. Activity Diagram

Menurut (Pressman 2012) Sebuah diagram *activity* menggambarkan perilaku dinamis dari sistem atau bagian dari sistem melalui aliran kontrol antara tindakan yang sistem lakukan. Hal ini mirip dengan sebuah *flowchart* kecuali bahwa suatu diagram *activity* dapat menunjukkan arus bersamaan. Simbol-simbol yang ada pada *activity diagram* dapat dilihat pada tabel 2.3 di bawah ini :

Tabel 2. 3 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/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan/ <i>join</i> 	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 	Menunjukkan dokumen sumber atau laporan
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

<p><i>Swimlane</i></p> <table border="1" data-bbox="376 367 683 602"> <tr> <td data-bbox="376 367 683 405">nama swimlane</td> </tr> <tr> <td data-bbox="376 405 683 602"></td> </tr> </table>	nama swimlane		<p>Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi</p>
nama swimlane			

Sumber: (Pressman, 2012)

2.7. Pengertian *Website*

Menurut MADCOM (2016) Web adalah suatu program komputer yang mempunyai tanggung jawab atau tugas menerima permintaan HTTP dari komputer klien, yang dikenal dengan nama web browser dan melayani mereka dengan menyediakan repon HTTP berupa konten data.

Ada beberapa jenis software untuk membangun web server local atau localhost yang support sistem operasi windows diantaranya adalah Wampserver, Appserv, XAMPP, PHP Triad atau Vertigo (MADCOM, 2016)

2.8. *Dreamweaver*

Adobe Dreamweaver adalah :aplikasi desain dan pengembangan web yang menyediakan editor WYSIWYG visual (bahasa sehari-hari yang disebut sebagai Design view) dan kode editor dengan fitur standar seperti syntax highlighting, code completion, dan code collapsing serta fitur lebih canggih seperti real-time syntax checking dan code introspection untuk menghasilkan petunjuk kode untuk membantu pengguna dalam menulis kode (Destiningrum and Adrian 2017).

2.9. Pengertian MySQL

Menurut MADCOM (2016) MySQL adalah sistem manajemen Database SQL yang bersifat Open Source dan paling populer saat ini. Sistem Database MySQL mendukung beberapa fitur seperti multithreaded, multiuser dan SQL Database management system (DBMS).

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 *dimensionality in use*, terdapat beberapa karakteristik relatif yang ditinjau dari perspektif *user* antara lain *Usability in use*, *Flexibility in use*, dan *Safety*. (Alfian, 2010).

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.
- 2) *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.
- 3) *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.
- 4) *Operability* Perangkat lunak dapat dimengerti, dipelajari, digunakan, dan menarik pengguna ketika digunakan dalam hal ini perangkat lunak mudah

dipelajari oleh pengguna, perangkat lunak dapat digunakan dan dioperasikan oleh pengguna, perangkat lunak dapat memberikan bantuan ketika pengguna membutuhkan panduan, perangkat lunak dapat menarik perhatian pengguna, perangkat lunak memenuhi kebutuhan pengguna yang memiliki keterbatasan dan perangkat memungkinkan untuk dianalisis oleh pengguna apakah perangkat lunak sudah memenuhi kebutuhan mereka.

- 5) *Security* Merupakan perlindungan terhadap perangkat lunak dari berbagai ancaman atau keganjalan dalam hal ini perangkat lunak memiliki perlindungan terhadap data atau informasi dari pengguna dan merupakan dari kelengkapan, ketepatan dari sejumlah *asset* yang telah dijaga sehingga aksi atau tindakan yang dilakukan telah terbukti dan hal tersebut tidak dapat ditolak.
- 6) *Compability* Faktor ini merupakan kemampuan dari dua atau lebih komponen perangkat lunak dapat melakukan pertukaran informasi dan melakukan fungsi yang dibutuhkan ketika digunakan pada *hardware* atau lingkungan perangkat lunak yang sama.
- 7) *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.

- 8) *Transferability*. Merupakan kemudahan dimana sistem atau komponen dapat berpindah dari lingkungan satu ke lingkungan yang lain dalam hal ini perangkat lunak dapat beradaptasi dengan cepat pada spesifikasi lingkungan yang berbeda tanpa menerapkan aksi atau cara lain dari pada memberikan tujuan tertentu terhadap perangkat lunak yang telah ada.

2.11. Skala Pengukuran

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 dapat dilihat pada Tabel 2.4:

Tabel 2. 4 Skala Likert

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

Hasil penilaian responden akan dihitung *persentase* kelayakannya dengan menggunakan perhitungan, dapat dilihat dibawah ini:

$$\text{Persentase} = \frac{\text{Skor Aktual (f)}}{\text{Skor Ideal (n)}} \times 100\%$$

Persentase kelayakan yang diperoleh kemudian dibandingkan dengan Tabel

konversi yang berpedoman pada acuan konversi nilai, dapat dilihat pada Tabel

2.5.

Tabel 2. 5 Skala Konversi Nilai

Persentase Pencapaian (%)	Interpretasi
$90 \leq x$	Sangat Baik
$80 \leq x < 90$	Baik
$70 \leq x < 80$	Cukup
$60 \leq x < 70$	Kurang
$X < 60$	Sangat Kurang

Keterangan:

x = persentase hasil pengujian.