

BAB II

TINJAUAN PUSTAKA

Pada bab ini berisikan teori-teori pendukung penelitian yang didapatkan dari hasil studi literatur dan jurnal-jurnal yang penulis dapatkan. Setiap teori dalam bab ini akan menjadi landasan dalam perancangan dan implementasi sistem yang akan penulis buat.

2.1 Pengertian Sistem

Sistem adalah kumpulan atau himpunan dari unsur atau variable-variabel yang saling terkait, saling berinteraksi, dan saling tergantung satu sama lain untuk mencapai tujuan.[1]

Sistem dapat didefinisikan sebagai kumpulan dari elemen-elemen berupa data, jaringan kerja dari prosedur-prosedur yang saling berhubungan, sumber daya manusia, teknologi baik hardware dan software yang saling berinteraksi sebagai kesatuan untuk mencapai tujuan atau sasaran tertentu yang sama.[2]

Dari beberapa kutipan di atas maka penulis dapat menyimpulkan bahwa sistem informasi adalah sistem di dalam suatu instansi atau organisasi perusahaan yang mempertemukan kebutuhan pengolahan transaksi harian dan memberikan laporan-laporan atau informasi yang dibutuhkan.

2.2 Pengertian Informasi

Informasi merupakan kumpulan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima. Tanpa suatu informasi, suatu sistem tidak akan berjalan dengan lancar dan akhirnya bisa mati. Suatu organisasi tanpa adanya suatu informasi maka organisasi tersebut tidak bisa berjalan.[3]

2.3 Pengertian Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi organisasi yang bersifat manajerial dalam kegiatan strategi dari suatu organisasi

untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan.[3]

2.4 Pengertian E-Dokumen

E-Document adalah segala bentuk file digital yang hanya terdapat dalam bentuk/format elektronik seperti data yang disimpan dikomputer, jaringan (*network*)*back-up* ke dalam *Compact Disk* (CD) ataupun DVD, atau media penyimpanan lainnya. Contoh bentuk dokumennya bisa berupa : *email*, *voice mail*, *instans messages (IM)*, *e-czalender*, *audio-file*, *video*, data-data hasil transfer *handphone*, animasi, grafik, foto, presentasi, web pages, dan data digital lainnya.[4]

2.5 Pengertian Monitoring

monitoring adalah untuk mengenali dan mengevaluasi perkembangan yang terjadi akibat tindakan yaitu mengenali apakah pelaksanaan tindakan sesuai dengan rencana tindakan dan apakah telah terjadi peningkatan dengan adanya tindakan.[5] *Monitoring* dapat diartikan yaitu mengumpulkan secara teratur, mengkaji dan bertindak atas informasi tentang pelaksanaan suatu proyek atau kegiatan, yang pada umumnya digunakan untuk memeriksa kinerja terhadap target serta memastikan kepatuhan terhadap peraturan,[6]

Berdasarkan definisi diatas disimpulkan bahwa *monitoring* adalah kegiatan yang mengkaji dan mengevaluasi atas informasi tentang kinerja pelaksanaan suatu proyek atau kegiatan dengan melihat apakah telah terjadi peningkatan dengan adanya tindakan serta memastikan kepatuhan terhadap peraturan.

A. Fungsi Monitoring

1. *Compliance* (kesesuaian/kepatuhan)

Menentukan kesesuaian implementasi kebijakan dengan standard dan prosedur yang telah ditentukan.

2. *Auditing* (pemeriksaan)

Menentukan ketercapaian sumber-sumber/pelayanan kepada kelompok sasaran (*target groups*).

3. *Accounting* (Akuntansi)

Menentukan perubahan sosial dan ekonomi apa saja yang terjadi setelah implementasi sejumlah program (kebijakan) dari waktu ke waktu.

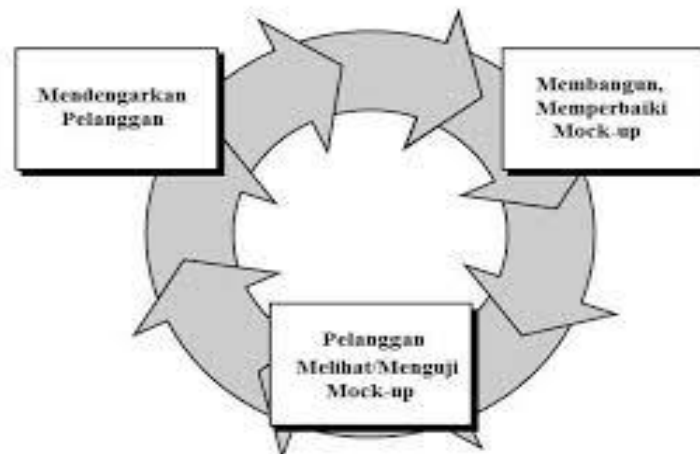
4. *Explanation* (Penjelasan)

Menjelaskan tingkat ketercapaian (hasil-hasil) program (kebijakan) relatif terhadap dengan tujuan yang ditetapkan.

2.6 Metode Pengembangan Sistem *Prototype*

Prototype adalah : Model prototipe dapat digunakan untuk menyambung ketidakpahaman pelanggan mengenai hal teknis dan memperjelas spesifikasi kebutuhan yang diinginkan pelanggan kepada pengembang perangkat lunak.[7]

Adapun tahapan – tahapan dalam metode *Prototype* dapat dilihat pada gambar 2.1:



Gambar 2.1 Tahapan - Tahapan Metode Prototype

Pada metode *prototypeter* dapat tiga tahap yaitu:[10]

1. Mendengarkan Pramuniaga dan kasir

Pada tahap ini, dilakukan pengumpulan kebutuhan dari sistem dengan cara mendengar kebutuhan admin dan Owner Pilar Properti 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.


2.7 Pengertian Bahasa Pemodelan Pengembangan Sistem (*UML*)

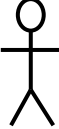




Bahasa Pemodelan Pengembangan Sistem (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. Beberapa jenis diagram *UML* antara lain sebagai berikut: [8]

2.7.1 Use Case Diagram

Use case diagram atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *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 gambar 2.1 di bawah ini:[8]

Tabel 2. 1 Simbol Diagram Use Case

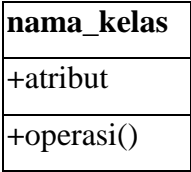



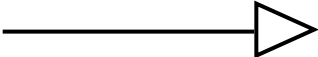

Simbol	Deskripsi
<p><i>Use Case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i></p>

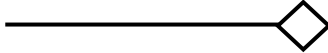
<p>Aktor/<i>actor</i></p> 	<p>Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i></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>

2.7.2 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,[8] simbol-simbol yang ada pada diagram kelas pada tabel *class diagram 2.2* di bawah ini:

Tabel 2. 2 Simbol Class Diagram

Simbol	Deskripsi
Kelas 	Kelas pada struktur system
Antarmuka/ <i>Interface</i>  nama_interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
Asosiasi/ <i>association</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





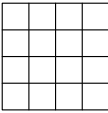

Simbol	Deskripsi
Agregasi/agregation 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)


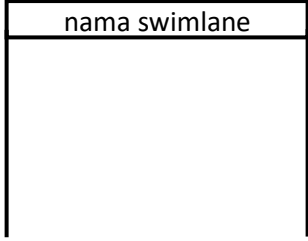
Sumber [8].

2.7.3 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,[8]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

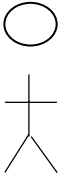
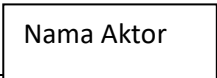
Simbol	Deskripsi
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi



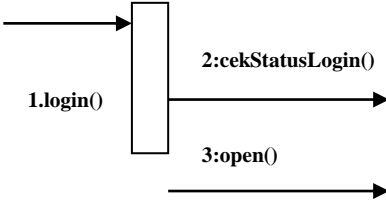
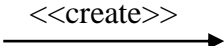
Sumber [8]

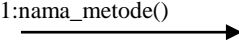
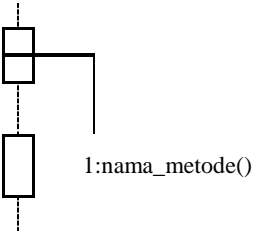
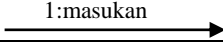
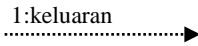
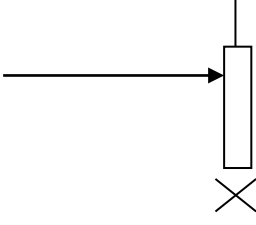
2.7.4 Sequence Diagram

Sequence Diagram adalah diagram interaksi yang merinci bagaimana operasi dilakukan. Mereka menangkap interaksi antar objek dalam konteks kolaborasi. Sequence Diagram adalah fokus waktu dan mereka menunjukkan urutan interaksi secara visual dengan menggunakan sumbu vertikal diagram untuk mewakili waktu pesan apa yang dikirim dan kapan.[8]

Tabel 2. 4 Simbol Sequence Diagram

Simbol	Deskripsi
Aktor  Namaaktor Atau 	Orang, proses, atau sistem lain yang berinteraksi dengan system informasi yang akan dibuat diluar system informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.

Tanpa waktu aktif	
Garishidup/ <i>lifeline</i> 	Menyatakan kehidupan suatu objek.
Objek <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;"> <u>Namaobjek: nama kelas</u> </div>	Menyatakan objek yang berinteraksi pesan.
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya, misalnya  <pre> sequenceDiagram actor Actor participant O1 participant O2 Actor->>O1: 1.login() activate O1 O1->>O2: 2:cekStatusLogin() O1->>O2: 3:open() deactivate O1 </pre> Maka cek Status Login() dan open() dilakukan di dalam metode login() Aktor tidak memiliki waktu aktif
Pesantipecreate 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat

<p>Pesantipecall</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objeklain atau dirinya sendiri,</p>  <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan objek yang berin teraksi.</p>
<p>Pesantipesend</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi keobjek lainnya, arah panah mengarah pada objek yang dikirim.</p>
<p>Pesantipereturn</p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian keobjek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>
<p>Pesan tipedestroy</p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy.</p>

2.8 Pengertian MySQL

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 managemen system (DBMS).[9]

2.9 PHP (Hypertext Preprocessor)

PHP (PHP: *Hypertext apareprocessor*) adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Karena PHP merupakan *server-side scripting* maka sintaks dan perintah-perintah PHP akan dieksekusi di server kemudian hasilnya dikirimkan ke browser dalam format HTML. Dengan demikian kode program yang ditulis dalam PHP tidak akan terlihat oleh user sehingga keamanan halaman web lebih terjamin. PHP dirancang untuk membuat halaman web yang dinamis, yaitu halaman web yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman web. PHP termasuk dalam *Open Source Product*, sehingga source code PHP dapat diubah dan didistribusikan secara bebas. Versi terbaru PHP dapat diunduh secara gratis di situs resmi PHP: <http://www.php.net>. PHP juga dapat berjalan pada berbagai web server seperti IIS (*Internet Information Server*), PWS (*Personal Web Server*), Apache, Xitami. PHP juga mampu lintas platform. Artinya PHP dapat berjalan dibanyak sistem operasi yang beredar saat ini, di antaranya : Sistem Operasi Microsoft Windows (semua versi), Linux, Mac OS, Solaris. PHP dapat dibangun sebagai modul pada *web server Apache* dan sebagai *binary* yang dapat berjalan sebagai CGI (*Common Gateway Interface*). PHP dapat mengirim HTTP header, dapat mengatur cookies, mengatur *authentication* dan *redirect user*.[9]

2.10 XAMPP

XAMPP merupakan perangkat lunak bebas (*open source*) yang mendukung banyak sistem operasi, merupakan campuran dari beberapa *program*. Yang mempunyai fungsi sebagai server yang berdiri sendiri (*localhost*), yang terdiri dari program MySQL *database*, Apache *HTTP Server*, dan penerjemah ditulis dalam bahasa pemrograman PHP dan Perlu. Nama XAMPP merupakan singkatan dari X (empat sistem operasi), Apache, MySQL, PHP dan Perlu. Program ini tersedia di bawah GNU (*General Public License*) dan bebas, adalah mudah untuk menggunakan web server yang dapat melayani tampilan halaman web yang dinamis. Menurut.[10]

2.11 Pengujian Black – Box

Pendekatan pengujian *Black-Box* adalah metode pengujian di mana data tes berasal dari persyaratan fungsional yang ditentukan tanpa memperhatikan struktur program akhir. Karena hanya fungsi dari modul perangkat lunak yang menjadi perhatian, pengujian *Black-Box* juga mengacu pada uji fungsional, metode pengujian menekankan pada menjalankan fungsi dan pemeriksaan inputan dan data output.[11]

Pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut :

- a. Fungsi – fungsi yang tidak benar atau hilang,
- b. Kesalahan interface
- c. Kesalahan dalam struktur data atau akses eksternal
- d. Kesalahan kinerja
- e. Inisialisasi dan kesalahan terminasi

Pada *black box testing* terdapat jenis teknik design tes yang dapat dipilih berdasarkan pada tipe testing yang akan digunakan, diantaranya sebagai berikut:

1. *Equivalence Class Partitioning*
2. *Boundary Value Analysis*
3. *State Transitions Testing*

4. Cause-Effect Graphing

2.12 Penelitian Sebelumnya

Tabel 2. 5 Penelitian Sebelumnya

No	Nama Penelitian	Judul Penelitian
1	Mustar Aman,2021	RANCANG BANGUN SISTEM E-MARKETING DENGAN PENDEKATAN SISTEM BERORIENTASI OBJECT PADA PT. KHAULA PRIMA [12]
2	Hari Prima, Nurrahman,2019	SISTEM INFORMASI PEMASARAN DAN MANAJEMEN PADA PT. CITRA MANDIRI DWI PRATAMA BERBASIS WEB [13]
3	Efta Kharistya Kisnanto Ramos Somya, S.Kom., M.Cs.2019	PERANCANGAN SISTEM INFORMASI <i>WEB DATABASE</i> PERUMAHAN KECAMATAN SIDOMUKTI SALATIGA MENGGUNAKAN <i>FRAMEWORK NETTE</i> [14]