

BAB II TINJAUAN TEORI

2.1. Teori Umum

2.1.1. Konsep Dasar Perancangan

2.1.1.1. Pengertian Perancangan

Pengertian perancangan yang dipaparkan oleh Husda (2011: 137) bahwa perancangan sistem dapat diartikan sebagai berikut: (1) tahap setelah analisis dari siklus pengembangan sistem, (2) pendefinisian dari kebutuhan-kebutuhan fungsional, (3) persiapan untuk rancang bangun implementasi, (4) menggambarkan bagaimana suatu sistem dibentuk, (5) yang dapat berupa penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu keseluruhan yang utuh dan berfungsi.

Selanjutnya menurut Kristanto (2008: 61) bahwa Perancangan sistem adalah suatu fase dimana diperlukan suatu keahlian perancangan untuk elemen-elemen komputer yang akan menggunakan sistem yaitu pemilihan peralatan dan program komputer untuk sistem yang baru.

Berdasarkan definisi perancangan tersebut, maka penulis dapat menyimpulkan bahwa perancangan merupakan kemampuan untuk membuat beberapa alternatif pemecahan masalah dengan mendesain sistem yang baru.

2.1.2. Konsep Dasar Sistem

2.1.2.1. Pengertian Sistem

Menurut Mustakini (2009: 34) sistem dapat didefinisikan dengan pendekatan prosedur dan pendekatan komponen, sistem dapat didefinisikan sebagai kumpulan dari prosedur-prosedur yang mempunyai tujuan tertentu.

Selanjutnya Menurut Sutabri (2012: 10) secara sederhana, suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu.

Dari pendapat-pendapat yang dikemukakan di atas dapat disimpulkan bahwa sistem adalah suatu kumpulan atau kelompok dari elemen atau komponen yang saling berhubungan atau saling berinteraksi dan saling bergantung satu sama lain untuk mencapai tujuan tertentu.

2.1.3. Konsep Dasar Sistem Informasi

2.1.3.2. Definisi Informasi

Menurut Husda (2012: 13) informasi adalah data yang telah diproses/diolah ke dalam bentuk yang sangat berarti untuk penerimanya dan merupakan nilai yang sesungguhnya atau dipahami dalam tindakan atau keputusan yang sekarang atau nantinya.

Berdasarkan pendapat para ahli yang dikemukakan di atas dapat ditarik kesimpulan bahwa informasi adalah data yang sudah diolah menjadi sebuah bentuk yang berarti bagi pengguna, yang bermanfaat dalam pengambilan keputusan saat ini atau mendukung sumber informasi.

2.1.4. Definisi Sistem Informasi

Terdapat berbagai macam pengertian sistem informasi menurut beberapa ahli, diantaranya sebagai berikut :

Menurut Murhada dan Giap (2011: 143) sistem informasi adalah sistem yang mengumpulkan, menyimpan, mengolah, dan menyebarkan data dan informasi. Sistem informasi dibuat sesuai dengan keperluan organisasi dan tingkatan manajemennya.

selanjutnya pendapat dari Sutarman (2009: 13) sistem informasi adalah mengumpulkan, memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk tujuan tertentu.

Dari pendapat yang dikemukakan di atas dapat disimpulkan bahwa informasi adalah gabungan dari orang, hardware, software, jaringan komunikasi, sumber daya data, dan kebijakan dan prosedur yang menyimpan, mengumpulkan (mendapatkan kembali), memproses, dan mendistribusikan informasi untuk mendukung pengambilan dan pengontrolan keputusan dalam suatu organisasi.

2.1.5. Pengertian Absensi

Berdasarkan kamus besar bahasa Indonesia absen adalah tidak bekerjanya seorang karyawan pada saat hari kerja, karena sakit, izin, alpa, atau cuti. Berdasarkan sumber di atas penulis menyimpulkan bahwa absensi adalah daftar administrasi ketidakhadiran karyawan.

2.2. Tinjauan Teori Khusus

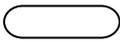
2.2.1. *Software* Pendukung

Merupakan alat yang digunakan untuk menggambarkan bentuk logika model dari suatu sistem dengan menggunakan simbol-simbol, lambang-lambang, diagram-diagram yang menunjukkan secara tepat arti dan fungsinya. Adapun peralatan pendukung (*tools system*) yang dijelaskan sebagai model sistem yang akan dirancang adalah sebagai berikut :

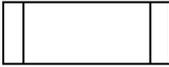
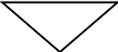
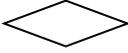
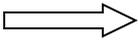
2.2.1.1. *Flowchart* (Diagram Alir)

Menurut (Murhada dan Giap, 2011: 112) flowchart adalah bagan-bagan yang mempunyai arus menggambarkan langkah-langkah penyelesaian suatu masalah, merupakan cara penyajian dari suatu algoritma.

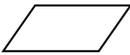
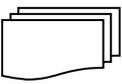
Tabel 2.1 Simbol-simbol *Flowchart*

Simbol	Nama	Fungsi
Processing Simbol		
	Terminator	Permulaan/akhir program
	Preparation	Proses inisialisasi/pemberian harga awal

Tabel 2.1 Lanjutan

	Proses	Proses perhitungan/proses pengolahan data
	Manual	Menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer
	Predefined Proses (Sub Program)	Permulaan sub program/proses menjalankan sub program
	Offline-storage	Menunjukkan bahwa data dalam symbol ini akan disimpan ke suatu media tertentu
	Manual input	Memasukkan data secara manual dengan menggunakan online keyboard
	Decision	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah sebelumnya
Flow Direction Symbol		
	Garis Alir (<i>Flow Line</i>)	Arah aliran program
	On Page Connector	Penghubung bagian flowchart yang berada pada satu halaman
	<i>Off Page</i> <i>Connector</i>	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda

Tabel 2.1 Lanjutan

Input/Output Simbol		
	<i>Input/Output</i>	Proses input/output data, parameter informasi
	<i>Document</i>	Mencetak keluaran dalam bentuk dokumen
	<i>Multiple Document</i>	Banyak dokumen
	<i>Display</i>	Mencetak keluaran dalam layar monitor

2.2.1.2. Unified Modeling Language (UML)

Menurut Rosa A. S dan M. Salahuddin (2013: 137) banyak orang telah membuat bahasa pemodelan pembangunan perangkat lunak sesuai dengan teknologi pemrograman yang berkembang pada saat itu, misalnya yang sempat berkembang dan digunakan banyak pihak adalah *Data Flow Diagram* (DFD) untuk memodelkan perangkat lunak yang menggunakan pemrograman procedural atau structural.

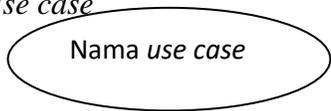
Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML).

UML muncul karena adanya kebutuhan model visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. Diagram yang akan dibahas pada penelitian ini hanya 4 diagram UML yg efektif biasa dipakai antara lain diagram *use case*, diagram *sequence*, diagram *class* dan diagram *activity*.

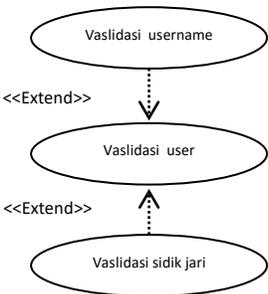
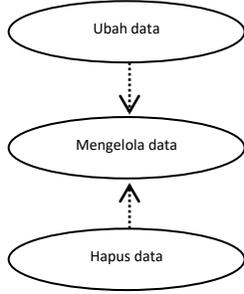
2.2.1.3. Use Case Diagram

Menurut Rosa A. S dan M. Salahuddin (2013: 155) *Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Secara umum, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut adalah simbol-simbol yang ada pada diagram *use case*:

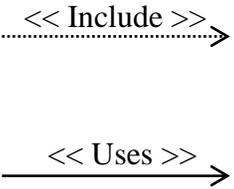
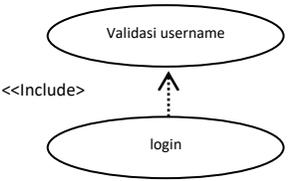
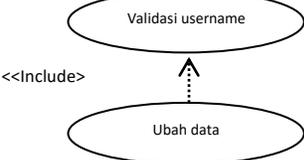
Tabel 2.2 Simbol-simbol pada diagram use case

Simbol	Deskripsi
<p><i>Use case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit atau actor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>.</p>
<p>Aktor / actor</p> 	<p>Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi itu sendiri, jadi walaupun simbol dari</p>

Tabel 2.2 Lanjutan

	actor adalah gambar orang tapi actor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama actor.
Asosiasi / <i>association</i> _____	Komunikasi antara actor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor.
Ekstensi / <i>extend</i> <<Extend>> Tabel 2.1 Lanjutan	<p>Relasi <i>use case</i> tamba han kesebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misalnya:</p>  <p>Arah panah mengarah pada <i>use case</i> yang ditambahkan, biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>
Generalisasi/ <i>generalization</i> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> di mana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:</p> <p>Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).</p>
Menggunakan / <i>include/uses</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana

Tabel 2.2 Lanjutan

<p>  </p> <p>Tabel 2.1 Lanjutan</p> <p>  </p>	<p><i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.</p> <p>Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i>:</p> <ol style="list-style-type: none"> 1. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, missal pada kasus berikut: 2. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu melakukan pengecekan apakah <i>use case</i> 3. yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, missal pada kasus berikut: <p>  </p> <p>Kedua sudut pandang di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan kebutuhan.</p>
--	--

Sumber: Rosa A. S dan M. Salahuddin (2013: 137)

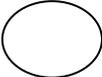
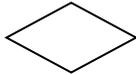
2.2.1.4. Activity Diagram

Menurut Rosa A. S dan M. Salahuddin (2013: 137) diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

1. Rancangan proses bisnis di mana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas:

Tabel 2.3 Simbol-simbol pada *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
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir

Tabel 2.3 Lanjutan

Swimlane		Nama swimlane	Nama swimlane
Nama swimlane	Nama swimlane		

Memisahkan organisasi bisnis yang bertanggungjawab terhadap aktivitas yang terjadi

Sumber: Rosa A. S dan M. Salahuddin (2013: 137)

2.2.1.5. Sequence Diagram

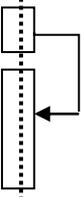
Menurut Rosa A. S dan M. Salahuddin (2013: 165) diagram sequence menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. oleh karena itu untuk menggambarkan diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Berikut adalah simbol-simbol yang ada pada diagram *sequence*:

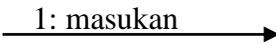
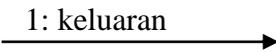
Tabel 2.4 Simbol-simbol pada *sequence* diagram

Simbol	Deskripsi
Aktor Atau  <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-top: 10px;"> <u>nama aktor</u> </div> Tanpa waktu aktif	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari actor adalah gambar orang, tapi actor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor

Tabel 2.4 Lanjutan

<p>Garis hidup/<i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek</p>
<p>Objek</p> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;">nama objek : nama kelas</div> <p>Waktu aktif</p> 	<p>Menyatakan objek yang berinteraksi pesan</p> <p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p> <p>1: Login 2: cekStatusLogin()</p>
<p>Pesan tipe create</p> <p><<create>></p> 	<p>Menyatakan suatu objek membuat objek lain, arah panah mengarah pada objek yang dibuat</p>
<p>Pesan tipe call</p> <p>1: nama_metode()</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p>  <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode yang dipanggil harus ada pada</p>

Tabel 2.4 Lanjutan

	diagram kelas sesuai dengan kelas objek yang berinteraksi
Pesan tipe send 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
Pesan tipe return 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
Pesan tipe destroy 	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy

2.2.1.6. Class Diagram

Menurut Rosa A. S dan M. Salahuddin (2013: 141) 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.

1. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis berikut:

1. Kelas main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan

2. Kelas yang menangani tampilan sistem (*view*)

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai

3. Kelas yang diambil dari pendefinisian *use case* (*controller*)

Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.

4. Kelas yang diambil dari pendefinisian data (*model*)

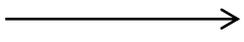
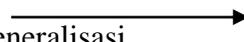
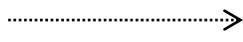
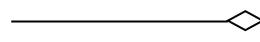
Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Berikut adalah simbol-simbol pada diagram kelas:

Tabel 2.5 Simbol-simbol pada class diagram

Simbol	Deskripsi			
<div style="border: 1px solid black; padding: 5px;"> <p>Kelas</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">nama_kelas</td> </tr> <tr> <td style="padding: 2px;">+atribut</td> </tr> <tr> <td style="padding: 2px;">+operasi</td> </tr> </table> </div>	nama_kelas	+atribut	+operasi	<p>Kelas pada struktur sistem</p>
nama_kelas				
+atribut				
+operasi				

Tabel 2.5 Lanjutan

Antarmuka / <i>interface</i>  nama_ <i>interfce</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
Asosiasi berarah/ <i>direct association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)
Kebergantungan / <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan kelas
Agresi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

Sumber: Rosa A. S dan M. Salahuddin (2013: 146)

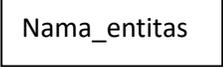
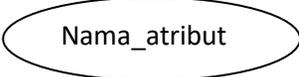
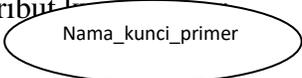
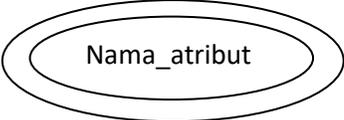
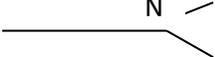
2.2.1.7. *Entity Relationship Diagram (ERD)*

Menurut Rosa A. S dan M. Salahuddin (2013: 50) pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram (ERD)*. ERD digunakan untuk pemodelan bisnis data relasional, berikut adalah simbol-simbol yang digunakan pada ERD:

Tabel 2.6 Simbol-simbol pada *entity relationship diagram*

Simbol	Deskripsi
Entitas / <i>entity</i>	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda

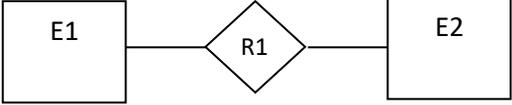
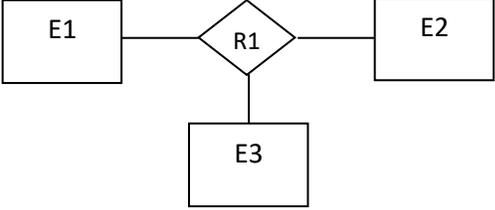
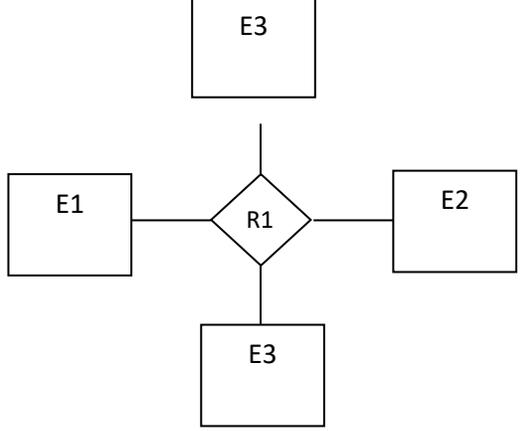
Tabel 2.6 Lanjutan

	<p>yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama label</p>
<p>Atribut</p> 	<p><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas</p>
<p>Atribut kunci primer</p> 	<p><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama)</p>
<p>Atribut multi nilai / <i>multi value</i></p> 	<p><i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu</p>
<p>Relasi</p> 	<p>Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja</p>
<p>Asosiasi / <i>association</i></p> 	<p>Penghubung antara relasi dan entitas dimana kedua ujungnya memiliki multiplicity kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan <i>one to many</i> menghubungkan entitas A dan entitas B.</p>

Sumber: Rosa A. S dan M. Salahuddin (2013: 146)

ERD biasanya memiliki hubungan binary (satu relasi menghubungkan dua buah entitas). Beberapa metode perancangan ERD menoleransi hubungan relasi *ternary* (satu relasi menghubungkan banyak entitas), tetapi banyak metode perancangan ERD yang tidak mengizinkan hubungan *ternary* atau *N-ary*. Berikut adalah contoh bentuk hubungan relasi dalam ERD.

Tabel 2.7 Simbol-simbol pada hubungan relasi dalam ERD

Nama	Gambar
<i>Binary</i>	
<i>Ternary</i>	
<i>N-ary</i>	

Sumber: Rosa A. S dan M. Salahuddin (2013: 146)

2.2.1.8. Bahasa Pemrograman PHP

Menurut Aditya (2011: 1) *Hypertext Preprocessor* (PHP) adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak digunakan untuk memprogram situs web dinamis. PHP dapat digunakan untuk membuat sebuah CMS ataupun sistem yang berbasis pada web.

Beberapa kelebihan PHP dari bahasa pemrograman web, antara lain:

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. Web server yang mendukung PHP dapat ditemukan dimana-mana dari mulai *apache, IIS, Lighttpd*, hingga *Xiatami* dengan konfigurasi yang relative murah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan developer yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi banyak.
5. PHP adalah bahasa *open source* yang dapat dijalankan *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

2.2.1.9. MySQL Databases

Menurut Aditya (2011: 62) MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis di bawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial.

MySQL sebenarnya merupakan turunan salah satu konsep utama dari basis data yang telah ada sebelumnya, SQL (*Structure Query Language*). SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

2.2.1.10. Adobe Dreamweaver CS5

Dreamweaver adalah sebuah HTML editor profesional untuk mendesain *web* secara visual dan mengelola situs atau halaman *web* (Madscom, 2011: 2). Saat ini terdapat software dari kelompok Adobe yang belakangan banyak digunakan untuk mendesain situs *web*. Pada Dreamweaver CS5, terdapat beberapa kemampuan bukan hanya sebagai software untuk desain web saja tetapi juga untuk menyunting kode serta pembuatan aplikasi *web* dengan menggunakan berbagai bahasa pemrograman *web* antara lain: JSP, PHP, ASP, dan *ColdFusion*.

2.2.1.11 AppServ

AppServ adalah salah satu paket software webserver yang terdiri dari Apache, PhpMyAdmin. (Madscom, 2011: 206).

2.2.1.12. Pengujian Perangkat Lunak

Menurut Rosa A. S dan M. Salahuddin (2013: 272) pengujian adalah satu set aktivitas yang direncanakan dan sistematis untuk menguji atau mengevaluasi kebenaran yang diinginkan. Aktifitas pengujian terdiri dari satu set atau sekumpulan langkah yang dapat menempatkan desain kasus uji yang spesifik dan metode pengujian. Secara umum pola pengujian pada perangkat lunak sebagai berikut:

1. Pengujian dimulai dari level komponen hingga integrasi antar komponen menjadi sebuah sistem.
2. Teknik pengujian berbeda-beda sesuai dengan berbagai sisi atau unit uji dalam waktu yang berbeda-beda pula bergantung pada pengujian bagian mana yang dibutuhkan.
3. Pengujian dilakukan oleh pengembang perangkat lunak, dan jika untuk proyek besar, pengujian bisa dilakukan oleh tim uji yang tidak terkait dengan tim pengembang perangkat lunak (*Independent Test Group*) ITG.

4. Pengujian dan penirkutan (*debugging*) merupakan aktifitas yang berbeda, tetapi penirkutan (*debugging*) harus diakomodasi pada berbagai strategi pengujian.

Menurut Rosa A. S dan M. Salahuddin (2013: 275) pengujian kota hitam (*Black - Box Testing*) yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program.

Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus proses login maka kasus uji yang dibuat adalah:

1. Jika user memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
2. Jika user memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau salah keduanya.

2.3. Penelitian Terdahulu

Penelitian ini merupakan penelitian lanjutan, sebagai bahan pertimbangan maka peneliti mencantumkan beberapa jurnal ilmiah sebagai berikut:

Tabel 2.8 Penelitian Terdahulu

No	Nama	Judul	Metode	Kesimpulan
1.	Aditya Tegar Satya	Rancang Bangun Sistem Absensi Karyawan Online Berbasis Web menggunakan Framework PHP Codeigniter & MySQL (Studi Kasus: PT Starone Mitra Telekomunikasi)	<i>Waterfall Model</i>	Absensi kehadiran karyawan menggunakan sistem baru terbukti lebih efisien, tepat guna serta memudahkan karyawan untuk mengisi absen datang dan pulang. Pihak manajemen perusahaan mampu memonitoring kinerja dan kehadiran karyawan dengan lebih akurat.
2.	Warkim, Hafiz Novanda Ichwan, dan Husnul Kamal. Z	Analisa dan Deain Sistem Kehadiran Pegawai Pada Pusat Penelitian Perkembangan IPTEK Lembaga Ilmu Pengetahuan	SDLC	Sistem Informasi Kehadiran Pegawai dapat mempermudah dan mempercepat proses data kehadiran sehingga dapat menghemat waktu dan lebih efisien karena semua proses dilakukan secara online
3.	Pitri Candrawati dan Rinawati	Sistem Informas Absensi Karyawan Pada PT. Harja Gunatama Lestari Bandung	<i>Waterfall Model</i>	Dengan adanya sistem informasi yang berbasis computer yang dapat memberikan solusi lebih cepat optimal dan ketepatan dalam pengolahan data absensi karyawan (kehadiran karyawan, sakit, alpa, cuti) serta mengurangi tingkat kesalahan pada waktu proses pengabsensian.
4.	Rachman Mulyadi dan Cynthia Ayu Wulan	Aplikasi Absensi Pegawai	-	Bagian Personalia mempunyai media penyimpanan/database

Tabel 2.8 Lanjutan

	Dini, 2013	Kecamatan Bataceper Tangerang Dalam Meningkatkan Akurasi Informasi		khusus yang digunakan untuk menampung semua data yang menjadi tugas bagian Personalia/Umum. Juga membantu mempercepat kegiatan yang ada pada Personalia.
--	------------	--	--	--

Sumber: Data Penelitian (2016)