

BAB II

Landasan Teori

2.1 Konsep Dasar Sistem Informasi

2.1.1 Sistem

Menurut Sutarman, 2012 sistem adalah kumpulan elemen yang saling berhubungan dan berinteraksi dalam satu kesatuan untuk menjalankan suatu proses pencapaian suatu tujuan utama.

Menurut Fathansyah, 2015 sistem adalah sebuah tatanan (keterpaduan) yang terdiri atas sejumlah komponen fungsional (dengan satuan fungsi dan tugas khusus) yang saling serhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses tertentu.

Dengan demikian, secara sederhana sistem dapat diartikan sebagai suatu kesatuan yang saling berinteraksi dan bekerja sama satu sama lain untuk mencapai suatu tujuan tertentu.

2.1.2 Informasi

Menurut Sutarman, (2012) informasi adalah sekumpulan fakta (data) yang diorganisasikan dengan cara tertentu sehingga mereka mempunyai arti bagi si penerima.

Menurut Ais Zakiyudin, (2012) informasi merupakan satu sumber daya yang sangat diperlukan dalam suatu organisasi. Informasi adalah data yang telah diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

Dengan demikian, secara sederhana informasi dapat didefinisikan sebagai sekumpulan data yang diolah sedemikian rupa sehingga memberikan suatu arti bagi penerimanya.

2.1.3 Sistem Informasi

Mohamad Subhan (2012) Sistem Informasi merupakan kumpulan dari perangkat keras dan perangkat lunak komputer serta perangkat manusia yang akan mengolah data menggunakan perangkat keras memegang peranan yang penting dalam sistem informasi.

Data yang akan dimasukkan dalam sebuah sistem informasi dapat berupa formulir-formulir, prosedur-prosedur dan bentuk data lainnya”.

Tujuan dari sistem informasi yaitu menyediakan informasi bagi semua tingkatan dalam organisasi kapan saja jika diperlukan. Sistem ini menyimpan, mengambil, mengubah, mengelola dan mengkomunikasikan informasi yang diterima dengan menggunakan sistem informasi. Berdasarkan teori-teori di atas, dapat disimpulkan bahwa sistem informasi merupakan kumpulan komponen-komponen, bisa berupa manusia, perangkat lunak, perangkat keras, jaringan komunikasi, dan sumber data yang saling berhubungan yang dapat menyampaikan informasi dengan menyediakan media untuk menyampaikan suatu tujuan.

2.1.4 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, yaitu mempunyai komponen-komponen (component), batas sistem (boundary), lingkungan luar (environments), penghubung (interface), masukan (input), keluaran (output), pengolahan (process), dan sasaran (objectives) atau tujuan (goal).

Adapun penjelasan dari karakteristik dari suatu sistem adalah sebagai berikut (Hanif, 2012):

a. **Komponen Sistem (Components)**

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk sub sistem. Setiap sistem dapat mempunyai system dapat mempunyai sistem yang lebih besar, yang disebut supra sistem.

b. **Batasan sistem (Boundary)**

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan dengan sistem lainnya atau sistem dengan lingkungan luarnya.

c. **Lingkungan luar sistem (Environment)**

Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem.

d. **Penghubung sistem (Interface)**

Sebagai media yang menghubungkan sistem dengan subsistem yang lain.

e. **Masukan sistem (Input)**

Masukan sistem adalah segala sesuatu yang masuk kedalam sistem dan selanjutnya menjadi bahan untuk diproses.

f. Keluaran sistem (Output)

Keluaran merupakan hasil dari pemrosesan. Keluaran dapat berupa suatu informasi, saran, cetakan laporan, dan lain sebagainya.

g. Pengolah sistem (Process)

Proses merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna.

h. Sasaran sistem (Objective) atau tujuan (Goal)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu system dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

2.2 Aplikasi

Menurut Sutabri (2012:147), Aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya.

Menurut Asropudin (2013:6), Aplikasi adalah software yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Ms.World, Ms.Excel.

2.2.1 Agenda Kerja

Agenda Kerja disebut juga agenda kegiatan yaitu suatu rencana kegiatan organisasi yang dibuat untuk jangka waktu tertentu yang disepakati oleh pengurus organisasi. Agenda kerja harus dibuat secara sistematis, terpadu, terperinci, serta terarah, sebab program kerja sebagai pegangan dalam mencapai tujuan organisasi.

2.2.2 Web

Menurut Asropudin (2013:109), Web adalah sebuah kumpulan halaman yang diawali dengan halaman muka yang berisikan informasi, iklan, serta program aplikasi.

Menurut Ardhana (2012:3), Web adalah suatu layanan sajian informasi yang menggunakan konsep hyperlink, yang memudahkan surfer (sebutan para pemakai komputer yang melakukan browsing atau penelusuran informasi melalui internet).

2.3 Pemodelan Sistem

2.3.1 *Unified Modelling Language (UML)*

Unified Modeling Language menurut Alan Dennis (2012) adalah kosakata umum berbasis objek dan diagram teknik yang cukup efektif untuk memodelkan setiap proyek pengembangan sistem mulai tahap analisis sampai tahap desain dan implementasi.

UML di implemetasikan untuk tujuan tertentu, antara lain :

- A. Merancang perangkat lunak
- B. Sarana komunikasi antara perangkat lunak dengan proses bisnis
- C. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan system
- D. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya

2.3.2 *Diagram-Diagram Unified Modelling Language (UML)*

Menurut Alan Dennis (2012), *UML is a standard set of diagramming techniques that provide a graphical representation rich enough to model any systems development project, from analysis through implementation.*


Jadi, *UML* merupakan suatu ketetapan standar dari teknik diagram yang menyediakan representasi grafis yang cukup banyak untuk berbagai model pengembangan sistem, dari analisis melalui implementasi diagram.

2.3.2.1 *Use Case Diagram*

Use Case Diagram merupakan inti fungsionalitas koheren yang diekspresikan sebagai transaksi-transaksi yang terjadi antara actor dan system.

Menurut Alan Dennis (2012) Use case diagram merupakan suatu diagram yang menangkap kebutuhan bisnis untuk menggambarkan interaksi antara sistem dengan lingkungannya.

Tabel 2.1 Simbol *Use Case* (Alan Dennis, 2012)

No	Simbol	Nama	Keterangan
1		Aktor	Seseorangatau sesuatu yang berinteraksi dengan sistem.


2		Include	Include memungkinkan Use Case untuk menggunakan fungsional yang di sediakan oleh Use Case lainnya.
3		Extends	Memungkinkan suatu Use Case memiliki kemungkinan memperluas fungsionalitas yang disediakan oleh use case lainnya.
4		Association	Untuk mendokumentasikan aliran -aliran logika dalam setiap Use Case.
5		System Boundary	Merupakan lingkup subjek , misalnya, sistem atau individu proses bisnis.
6		Use case	Menggambarkan bagaimana seseorang akan menggunakan sistem.



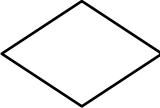
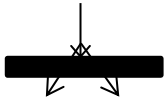
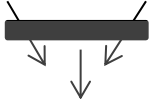

2.3.2.2 Activity Diagram

Secara grafis digunakan untuk menggambarkan rangkaian aliran aktivitas baik proses bisnis maupun use case.

Menurut Alan Dennis (2012) Pengertian diagram activity adalah yang menggambarkan alur kerja bisnis independen dari class, atau desain rinci sebuah metode.

Tabel 2.2 Simbol Activity Diagram (Alan Dennis, 2012)

No	Gambar	Nama	Fungsi
1		Node Awal	Merupakan tanda awal dari sebuah aktivitas.

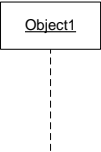

2		Node Akhir	Merupakan tanda berakhirnya sebuah aktivitas.
3		Aktivitas	merupakan sebuah gambaran aktivitas yang terjadi.
4		Keputusan	Pilihan untuk pengambilan keputusan.
5		Percabangan	Asosiasi ini bisa digunakan dalam percabangan menjadi 2 aktivitas
6		Penghubungan	Asosiasi ini bisa digunakan dalam penggabungan dari 2 aktivitas
7		Generalisasi	Relasi antar kelas dalam activity diagram


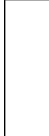

2.3.2.3 Sequence Diagram

Secara grafis menggambarkan bagaimana objek berinteraksi dengan satu sama lain melalui pesan pada sekuensi sebuah use case atau operasi.

Menurut Alan Dennis (2012) Diagram *Sequence* merupakan urutan model dinamis yang menggambarkan contoh *class* yang berpartisipasi dalam *use case* dan pesan yang lewat di antara mereka dari waktu ke waktu.

Tabel 2.3 Simbol *Sequence Diagram* (Alan Dennis, 2012)

No	Gambar	Nama	Fungsi
1		<i>Object</i>	Objek merupakan <i>instance</i> dari sebuah <i>class</i> dan dituliskan tersusun secara horizontal.
2		<i>Actor</i>	Aktor juga dapat berkomunikasi dengan objek, maka aktor juga dapat diurutkan sebagai kolom.

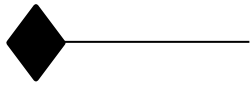

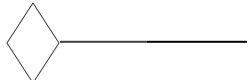
3		<i>Lifeline</i>	<i>Lifeline</i> mengartikan keberadaan sebuah objek dalam basis waktu.
4		<i>Activation</i>	<i>Activation</i> dinotasikan sebagai sebuah kotak segi panjang yang digambar pada sebuah <i>lifeline</i> , mengindikasikan sebuah obyek yang akan melakukan sebuah aktifitas
5		<i>Message</i>	<i>Message</i> dilakukan dengan anak panah horizontal antara <i>activation message</i> mengindikasikan komunikasi antara objek

2.3.2.4 Class Diagram

Menggambarkan struktur *object system*. Diagram ini menunjukkan *class object* yang menyusun sistem dan juga hubungan antara *class object* tersebut. Menurut Alan Dennis (2012:) Diagram kelas adalah model statis yang menunjukkan kelas dan hubungan di antara kelas yang tetap konstan dalam sistem dari waktu ke waktu.

Tabel 2.4 Simbol *Class Diagram* (Alan Dennis, 2012)

Nama Komponen	Keterangan	Simbol				
<i>Class</i>	Kelas pada struktur sistem.	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>Class name</td> </tr> <tr> <td>+Atribut</td> </tr> <tr> <td>+Atribut</td> </tr> <tr> <td>+Method</td> </tr> </table>	Class name	+Atribut	+Atribut	+Method
Class name						
+Atribut						
+Atribut						
+Method						
<i>Association</i>	Asosiasi merupakan sebuah relationship paling umum antara 2 class dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 class. Garis ini bisa	<p>1..1 Owned by 1</p> <hr style="width: 50%; margin-left: auto; margin-right: auto;"/>				

	melambangkan tipe-tipe relationship (Contoh <i>One-to-one</i> , <i>one-to-many</i> , <i>many-to-one</i>)	
<i>Composition</i>	Jika sebuah <i>class</i> tidak bisa berdiri sendiri dan harus menjadi bagian dari <i>class</i> yang lain maka <i>class</i> tersebut memiliki relasi <i>composition</i> terhadap <i>class</i> tempat dia bergantung tersebut.	
<i>Dependency</i>	Untuk kondisi tertentu sebuah <i>class</i> menggunakan <i>class</i> yang lain. Hal ini disebut <i>dependency</i> .	
<i>Aggregation</i>	<i>Aggregation</i> mengindikasikan keseluruhan bagian <i>relationship</i> dan biasanya disebut sebagai relasi	

2.4 Software Development Life Cycle (SDLC)

Setiap perangkat lunak memiliki siklus hidup pengembangan sistem (SDLC) yaitu suatu proses periode pembuatan dan perubahan sistem yang disertai dengan metodologi yang digunakan untuk mengembangkan sistem. Rekayasa perangkat lunak adalah suatu proses penelusuran, perencanaan, pemodelan, pengembangan yang dilaksanakan dan dikelola untuk membangun sebuah sistem perangkat lunak. Setiap sistem perangkat lunak memiliki siklus hidup (life cycle) dimana periode dilakukan pengembangan perangkat lunak untuk menyelesaikan masalah baru yang timbul, umumnya pengembangan sistem memiliki 5 tahapan yaitu investigasi dan analisis, modeling, development, implementasi, dan pengelolaan.

Dalam pengembangan suatu sistem, sulit diprediksi hal-hal apa saja yang akan terjadi saat pengembangan berlangsung. Keinginan pemilik produk (product owner) dapat berubah sewaktu-waktu. Gagal beradaptasi dengan perubahan menjadi salah satu penyebab kegagalan suatu proyek. Menurut Elvis C. Foster (2014), siklus hidup pengembangan, sistem memiliki banyak variasi model.

Seperti diantaranya:

1. Waterfall Model
2. Phased Prototype Model
3. Iterative Development Model
4. Rapid Prototype Model
5. Formal Transformation Model
6. Component-Based Model
7. Agile Development Model

Menurutnya Terlepas dari model apa yang di gunakan, pada umumnya siklus hidup pengembangan sistem memiliki 5 tahapan yaitu :

SDLC Phase	Related Deliverable(s)
Investigation and Analysis	Initial System Requirements; Requirements Specification
Design (Modeling)	Design Specification
Development (Construction)	Actual Software System; Product Documentation
Implementation	
Management	Enhanced Software System; Revised Documentation

Gambar 2.1 lima tahapan utama SDLC

Sumber: Elvis C. Foster (2014:9)

2.4.1 Metode Pengembangan *Waterfall*

Menurut Pressman (2015) “Waterfall merupakan salah satu model klasik dalam perancangan piranti lunak. Penyusun memilih model waterfall, karena langkah-langkahnya berurutan dan sistematis”. Metode Waterfall disebut juga dengan *Classic Life Cycle*. Metode ini membutuhkan pendekatan dalam pengembangan perangkat lunak dan mengumpulkan kebutuhan secara lengkap yang diantaranya, *Communication, Planning, Modelling, Construction dan Development*.

a. Communication

Langkah ini merupakan analisis terhadap kebutuhan software, dan tahap untuk mengadakan pengumpulan data dengan melakukan pertemuan dengan customer, amupun mengumpulkan data-data tambahan baik yang ada di jurnal, artikel, maupun dari internet.

b. Planning

Proses planning merupakan lanjutan dari proses communication. Tahapan ini menghasilkan dokumen user requirement atau bisa dikatakan sebagai data yang berhubungan dengan keinginan user dalam pembuatan software, termasuk rencana yang akan dilakukan.

c. Modelling

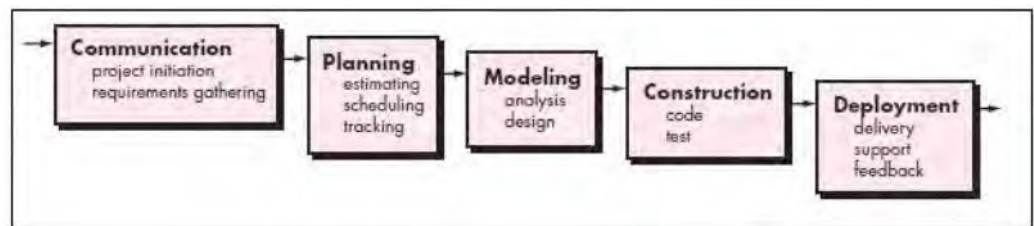
Proses modelling ini akan menerjemahkan syarat kebutuhan ke sebuah perancangan software yang dapat diperkirakan sebelum dibuat coding. Proses ini berfokus pada rancangan struktur data, arsitektur software, representasi interface, dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut software requirement.

d. Construction

Construction merupakan proses membuat kode. Coding atau pengkodean merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. Programmer akan menerjemahkan transaksi yang diminta oleh user. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu software, artinya penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan testing terhadap sistem yang telah dibuat tadi. Tujuan testing adalah menemukan kesalahan-kesalahan terhadap sistem tersebut untuk kemudian bisa diperbaiki.

e. Deployment

Tahapan ini bisa dikatakan final dalam pembuatan sebuah software atau sistem. Setelah melakukan analisa, desain dan pengkodean maka sistem yang sudah jadi akan digunakan oleh user. Kemudian software yang telah dibuat harus dilakukan pemeliharaan secara berkala.



Gambar 2.2 Model Pengembangan Waterfall

Sumber : *Software Engineering, Pressman, R.S (2015)*

2.5 Database

Basis data (*Database*) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis, sehingga data digunakan oleh suatu program komputer untuk memperoleh informasi dari basis data tersebut. Basis data adalah sekumpulan data yang terhubung satu sama lain secara logika dan suatu deskripsi data yang dirancang untuk memenuhi kebutuhan informasi suatu organisasi. [Verdi Yasin, 2012:274]

Menurut Rosa .A.S & Shalahuddin[2014:43] basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Menurut Fathansyah [2012:2-3] basis data (*Database*) dapat didefinisikan dalam sejumlah sudut pandang seperti :

- Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak data dimanfaatkan kembali dengan cepat dan mudah.
- Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
- Kumpulan file yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

2.6 Hypertext Preprocessor (PHP)

Menurut (Ardhana 2012:88), PHP merupakan bahasa pemrograman berbasis *server-side* yang dapat melakukan *parsing script php* menjadi *script web* sehingga dari sisi *client* menghasilkan suatu tampilan yang menarik. PHP pertama kali dibuat oleh Rasmus Ledorf pada tahun 1995 yang diberi nama FI (*Form Interpreted*) dan

digunakan untuk mengelola *form* dan *web*. Pada perkembangannya, kode tersebut dirilis ke umum sehingga mulai banyak dikembangkan oleh programmer diseluruh dunia. Keunggulan yang dimiliki oleh php adalah sebagai berikut :

1. *Open Source*

PHP adalah *open source*, dikembangkan dan diperbaharui oleh komunitas pengembang dari seluruh dunia, yang memastikan bahwa bahasa ini bukan hanya relevan namun juga bisa sinkron dengan tren teknologi terbaru.

2. Kompatibel dengan berbagai *OS*

PHP bisa berjalan dengan baik di Linux, UNIX, Mac dan Windows, sangat sempurna ketika digabungkan dengan MySQL.

3. Fleksibel dan Dinamis

Bahasa ini menawarkan lebih banyak fleksibilitas dibanding dengan para pesaingnya dan dapat disesuaikan untuk memenuhi kebutuhan pengembangan *web* yang sesuai dengan yang diinginkan.

4. *Resource* besar

Terdapat banyak sekali *resource* yang bisa dimanfaatkan oleh para pengembang PHP sesuai dengan fungsi dan fiturnya. Mulai dari *framework*, *dependency managers*, *editor* dan alat untuk dokumentasi, PHP memiliki berbagai *tool* berkeualitas yang siap untuk digunakan.

5. *User Friendly*

PHP sangat sederhana untuk dipahami dibandingkan dengan bahasa pemrograman lainnya. Selain itu PHP juga bahasa yang mudah namun kuat, sudah banyak sekali digunakan untuk membuat aplikasi berbasis *web* yang membutuhkan fungsionalitas maksimal dengan kode yang minimal.

6. Skalabilitas

Dalam dunia IT, kata skalabilitas seperti emas. Apapun yang Anda lakukan baik itu berurusan *database*, *hosting* atau hal lain seperti *programming*, maka skalabilitas tidak menjadi masalah jika Anda menggunakan PHP, karena Anda dapat dengan mudah meningkatkan ukuran cluster Anda misalnya dengan menambah lebih banyak server jika proyek Anda semakin tumbuh besar..

2.7 MySQL

Menurut Sibero (2013:97) *MySQL* atau dibaca *My Sekuel* adalah suatu *Relational Database Management System* yaitu aplikasi sistem yang menjalankan fungsi pengolahan data. MySQL sendiri pertama kali diciptakan dan juga dikembangkan pada tahun 1979, oleh seorang *programmer* bernama Michael Widenius yang berasal dari Swedia. Pada

awalnya, MySQL dibuat dengan nama UNIREG, yang merupakan suatu sistem pengolahan database sederhana. Berdasarkan pendapat yang dikemukakan di atas dapat ditarik kesimpulan bahwa MySQL adalah suatu *software* atau program yang digunakan untuk membuat sebuah *database* yang bersifat *open source*

2.8 Teknik Pengujian Perangkat Lunak

2.8.1 Metode White Box

Dalam pengujian *white-box*, anda bisa tahu bagaimana sebuah metode bekerja. Anda kemudian menggunakan pengetahuan lebih untuk merancang sebuah tes untuk mencoba membuat sebuah metode rusak. Pengujian *white-box* memiliki keuntungan bahwa anda tahu bagaimana sebuah metode bekerja, jadi anda bisa mencoba untuk memilih kasus pengujian yang sulit.

Sayangnya ia memiliki kelemahan yang anda tahu bagaimana sebuah metode bekerja, jadi anda melewati beberapa kasus pengujian yang anda anggap bekerja. Misalnya, anda mungkin mengetahui bahwa sebuah metode akan dibingungkan dengan string kosong. Tapi anda tahu bahwa ketika anda menulis kode, jadi anda yang mengatasinya. Masalahnya, anda mungkin tidak menanganinya dengan benar. Jika anda menangani semuanya dengan benar, maka tidak akan ada *bug* dan anda tidak akan perlu melakukan pengujian secara keseluruhan. Menggunakan pengujian *white-box* untuk membuat tes akan merepotkan, tapi jangan melewati tes yang anda “tahu” metodenya anda dapat tangani (RodStephend, 2015:188)

2.8.2 Metode Black-Box

Dalam pengujian *black-box*, anda berpura-pura bahwa metode ini adalah *black-box* yang anda tidak dapat melihat kedalam. Anda tahu apa yang seharusnya dilakukan, tapi anda tidak tahu bagaimana ini bekerja. Anda kemudian mencoba segala macam masukan untuk melihat apa yang akan dilakukannya. Anda dapat memulai pengujian *black-box* dengan mengirim sekelompok masukan acak. Ingat bahwa anda perlu untuk melakukan tes ini hanya beberapa kali, tidak setiap kali program berjalan, jadi anda bisa menguji banyak nilai acak.

Contohnya, anda bisa mencoba untuk menebak nilai yang mungkin mengacaukannya. Biasanya, itu melibatkan nilai spesial seperti 0 untuk angka dan kosong untuk string. Mereka juga boleh termasuk nilai terbesar dan terkecil yang memungkinkan. Untuk string yang mungkin maksudnya adalah string yang semuanya kosong atau semua karakter. Misalnya, biasanya quicksort adalah salah satu algoritma pengurutan tercepat, tapi ia memberikan kinerja yang buruk jika item yang diurutkan memiliki nilai yang

sama. Jika metode mengambil sejumlah variabel masukan, pastikan ia dapat mengatasi masukan 0 dan sejumlah angka yang sangat besar untuk masukan. Jika dibutuhkan sebuah array atau daftar parameter, lihat apakah ia lakukan jika array atau daftar parameter kosong atau hilang. Akhirnya, lihat pada batasan nilai. Jika metode menerapkan parameter float antara 0.0 dan 1.0, maka pastikan ia dapat menangani kedua nilai itu (Rod Stephend,2015:187).

2.9 Studi Pustaka (*Literatur Review*)

Penggalian dari penelitian terdahulu dilakukan sebagai upaya untuk memperjelas penelitian yang telah dilakukan serta membedakan penelitian ini dengan penelitian sebelumnya. Beberapa penelitian yang telah dilakukan terkait dengan Sistem Informasi Monitoring, akan dibahas dibawah ini :

Judul	:	PENGEMBANGAN SISTEM BASIS DATA UNTUK APLIKASI PEMBUATAN DN MONITORING TARGET LETTER STUDI KASUS PT ASURANSI ALLIANZ
Penulis	:	Ayuliana, Rusdianto, Steven Daniel, Steffen Ayuliana_st@binus.ac.id , gembels_89@yahoo.com , calavon@yahoo.com , st3ff3n2002@yahoo.com
Tahun Terbit	:	Jurnal Ilmiah Fifo P-ISSN 2085-4315/E-ISSN 2502-8332
Pembahasan	:	<ul style="list-style-type: none"> ✓ Pembuatan aplikasi monitoring target letter ✓ Database menggunakan Microsoft SQL Server 2005 dan program menggunakan PHP ✓ Merubah sistem dari manual menjadi sistem aplikasi (menggunakan database Microsoft SQL Server 2005) ✓ pengolahan monitoring target letter menjadi terkomputerisasi
Kesimpulan	:	✓ aplikasi ini mempermudah karyawan dalam melakukan pembuatan monitoring dan perubahan target
Saran	:	✓ perlu dilakukan pengembangan dari segi rancangan agar semakin memudahkan pengoperasian dan lebih fleksibel
Perbedaan	:	✓ pada jurnal ini, aplikasi ini dibuat untuk menentukan target-target penjualan karyawan dan sistem yang berlangung menggunakan flowchat Sedangkan pada penelitian ini penulis membuat program aplikasi untuk melihat kegiatan pekerjaan serta dapat menilai kinerja kerja karyawan (KPI)