

## BAB II

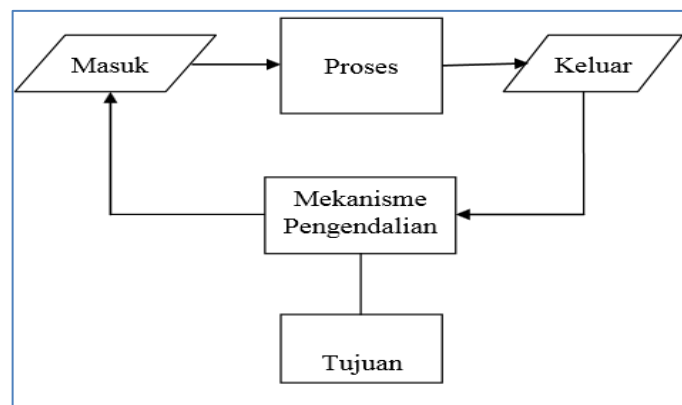
### LANDASAN TEORI

#### A. Pengertian Sistem Informasi

##### 1. Sistem

Abdul Kadir (2003:54) mendefinisikan sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan. Menurut Mulyanto (2009:2) sistem diartikan sebagai sekelompok komponen yang saling berhubungan, bekerja sama untuk mencapai tujuan bersama dengan menerima masukan serta menghasilkan keluaran dalam proses transformasi yang teratur.

Elemen sistem menurut Abdul Kadir (2003:54) antara lain tujuan, masukan, keluaran, proses, mekanisme pengendalian, dan umpan balik serta berinteraksi dengan lingkungan dan memiliki batas. Elemen sistem disajikan dalam gambar 2:



Gambar 2. Elemen Sistem

Data yang masuk melalui input (masukan) kemudian diproses dan dikeluarkan melalui output (keluaran) sesuai permintaan/tujuan. Mekanisme

pengendalian berupa umpan balik yang mengecek keluaran dengan melakukan perbandingan keluaran sistem dengan keluaran yang dikehendaki. Jika terdapat penyimpangan maka dilakukan pengiriman masukan untuk menyesuaikan proses supaya keluaran berikutnya mendekati standar. Dalam sistem harus dibuat batas dengan lingkungan untuk menentukan konfigurasi, ruang lingkup dan kemampuan sistem. Berdasarkan teori-teori tersebut, dapat disimpulkan bahwa sistem merupakan sekumpulan objek yang saling berhubungan untuk mencapai suatu tujuan.

## **2. Informasi**

Menurut Jogiyanto (1999:9), informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian (*event*) yang nyata (*fact*) yang digunakan untuk pengambilan keputusan.

## **3. Sistem Informasi**

Menurut Jogiyanto Hartono (1999:17) sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. Kenneth C. Laudon (2008:15) menyatakan bahwa sistem informasi secara teknis didefinisikan sebagai sekumpulan komponen yang saling berhubungan, mengumpulkan, dan mendistribusikan informasi untuk menunjang pengambilan keputusan, koordinasi, dan pengawasan. Informasi berarti data yang dibentuk menjadi sesuatu yang memiliki arti dan berguna bagi manusia.

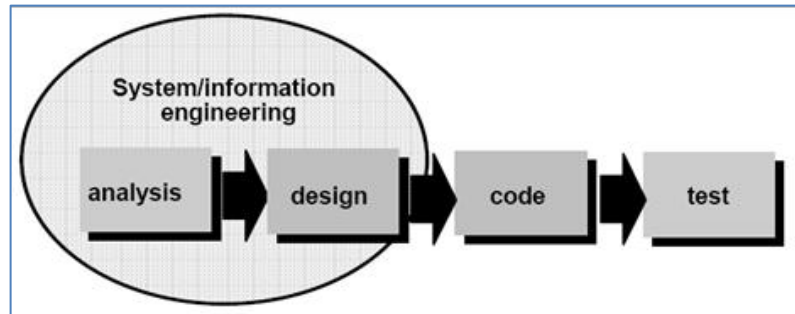
Terdapat tiga aktifitas di dalam sistem informasi yang akan memproduksi informasi yang dibutuhkan untuk membuat keputusan, mengendalikan operasi, menganalisis permasalahan, dan menciptakan produk baru, yaitu *input*, proses, dan *output*. *Input* merekap atau mengumpulkan data mentah. Pemrosesan mengubah data *input* mentah menjadi bentuk yang berarti. *Output* mengirimkan informasi yang telah diproses tersebut ke orang-orang yang akan menggunakan informasi tersebut. Sistem informasi juga memerlukan umpan balik, yang merupakan output yang dikembalikan ke sistem untuk mengevaluasi atau mengoreksi tahapan input.

## **B. Metode Pengembangan Sistem**

Metode pengembangan sistem yang digunakan dalam pengembangan sistem adalah model pengembangan perangkat lunak *Waterfall*. Menurut Pressman (2002:37), metode *waterfall* adalah suatu proses pengembangan perangkat berurutan, dimana kemajuan dipandang sebagai terus mengalir ke bawah (seperti air terjun) melewati fase-fase perencanaan, pemodelan, implementasi (konstruksi), dan pengujian. Keterkaitan dan pengaruh antar tahap ini ada karena output sebuah tahap dalam *waterfall* merupakan *input* bagi tahap berikutnya. Oleh karena itu, ketidaksempurnaan hasil pelaksanaan tahap sebelumnya adalah awal ketidaksempurnaan tahap berikutnya.

Model pengembangan *waterfall* dipilih karena model ini memiliki kepraktisan rekayasa yang membuat kualitas perangkat lunak tetap terjaga. Model pengembangan ini merupakan model pengembangan yang terstruktur dan terawasi. Selain karena kepraktisan, juga karena terorganisirnya dokumen

pengembangan sistem di setiap fase. Setiap fase harus terselesaikan dengan lengkap sebelum melangkah ke fase berikutnya. Gambar 3 merupakan model pengembangan *waterfall*.



Gambar 3. Model Pengembangan Waterfall

Penjelasan dari masing-masing tahapan *Model Waterfall* adalah sebagai berikut:

1. Analisis Kebutuhan

Analisis kebutuhan bertujuan untuk memahami kebutuhan dari sistem yang akan dibangun sehingga dapat dikembangkan sistem yang memadai kebutuhan tersebut.

2. Perancangan

Tahap perancangan diperlukan untuk pengembangan sistem informasi yang diusulkan. Menurut Pressman (2010:260) tahap perancangan meliputi perancangan data, perancangan arsitektural, perancangan antarmuka, dan perancangan komponen.

3. Pemrograman

Pemrograman disebut sebagai tahap implementasi perangkat lunak atau *coding*. Dengan kata lain, pada tahap ini dilakukan implementasi hasil

rancangan ke dalam baris-baris kode program yang dapat dimengerti oleh komputer.

#### 4. Pengujian

Pengujian perangkat lunak merupakan proses menjalankan dan mengevaluasi suatu perangkat lunak baik secara manual maupun otomatis.

Hal ini dilakukan untuk menguji apakah perangkat lunak tersebut sudah memenuhi persyaratan atau belum dan untuk menentukan perbedaan antara hasil yang diharapkan dengan hasil yang sebenarnya.

### **C. Penjamin Kualitas Perangkat Lunak (*Software Quality Assurance*)**

Penjaminan Kualitas Perangkat Lunak (*Software Quality Assurance /SQA*) ditujukan pada suatu usaha untuk menjamin terciptanya perangkat lunak yang berkualitas. Menurut Pressman (2002:245) jaminan kualitas perangkat lunak adalah aktivitas pelindung yang diaplikasikan pada setiap langkah dalam proses perangkat lunak. SQA mencakup berbagai prosedur atau tolak ukur untuk menjamin kesesuaian dengan standar yang ada serta pengukuran dan mekanisme pelaporan.

Salah satu tolak ukur kualitas perangkat lunak adalah ISO/IEC 9126. ISO/IEC 9126 mendefinisikan kualitas produk perangkat lunak, model, karakteristik mutu, dan metrik terkait yang digunakan untuk mengevaluasi dan menetapkan kualitas sebuah produk software. Standar ISO/IEC 9126 telah dikembangkan dalam usaha untuk mengidentifikasi atribut-atribut kunci kualitas untuk perangkat lunak komputer. Faktor kualitas menurut ISO/IEC 9126 meliputi enam karakteristik kualitas sebagai berikut:

1. Fungsionalitas (*functionality*) yaitu tingkat kemampuan perangkat lunak untuk menyediakan fungsi sesuai kebutuhan pengguna dan tingkat keamanan sistem.
2. Keandalan (*Reliability*) yaitu kemampuan perangkat lunak untuk mempertahankan tingkat kinerja tertentu, ketika digunakan dalam kondisi tertentu.
3. Kebergunaan (*usability*) yaitu kemampuan perangkat lunak untuk dipahami, dipelajari, digunakan, dan menarik bagi pengguna, ketika digunakan dalam kondisi tertentu.
4. Efisiensi (*efficiency*) yaitu kemampuan perangkat lunak untuk memberikan kinerja yang sesuai dan relatif terhadap jumlah sumber daya yang digunakan pada saat keadaan tersebut.
5. Pemeliharaan (*maintainability*) yaitu kemampuan perangkat lunak untuk dimodifikasi. Modifikasi meliputi koreksi, perbaikan atau adaptasi terhadap perubahan lingkungan, persyaratan, dan spesifikasi fungsional.
6. Portabilitas (*portability*) yaitu kemampuan perangkat lunak untuk ditransfer dari satu lingkungan ke lingkungan lain.

Berikut ini adalah metode pengukuran untuk setiap aspek karakteristik kualitas software.

1. Aspek Fungsionalitas (*Functionality*)

Menurut ISO/IEC 9126, *functionality* merupakan tingkat kemampuan perangkat lunak dalam menyediakan fungsi-fungsi sesuai yang telah ditentukan.

Menurut Simarmata (2010: 317), *functionality* juga meliputi kejelasan dari jenis

fungsi-fungsi perangkat lunak serta operasi *back-end* seperti keamanan dan bagaimana meningkatkan sistem. Beberapa subkarakteristik yang ada dalam aspek *functionality* diantaranya adalah subkarakteristik *suitability* dan subkarakteristik *security*. Subkarakteristik *suitability* yaitu kesesuaian dari fungsi-fungsi yang ada dengan tugas dan tujuan. Subkarakteristik *security* yaitu kemampuan sistem untuk melindungi informasi dan data sehingga orang atau sistem yang tidak sah tidak dapat membaca atau memodifikasi sistem.

Pengujian subkarakteristik *suitability* menggunakan instrumen *test case*. Menurut Williams (2006:44), format *test case* yang disarankan dalam pengujian *functionality* adalah: 1) *test id* adalah *identifier* dari *test case*, dimana *identifier* tiap *test case* harus unik; 2) *description* menggambarkan serangkaian langkah-langkah atau *input* dari tes yang akan dijalankan; 3) *expected result* adalah hasil yang diharapkan berdasarkan pada *input* yang dijelaskan dalam kolom deskripsi; 4) *actual result* berisi hasil setelah tes dijalankan yang dapat berisi sukses dan gagal. Tabel 1 adalah contoh format *test case login*:

Tabel 1. *Test Case Login*

No	Test ID	Description (langkah atau input)	Expected Result	Actual Result	
				Sukses	Gagal
1	<b>Login</b>	Mengetikkan url skn.web.id	Menampilkan halaman <i>login</i> .		
		Memasukkan <i>username</i> dan <i>password</i> yang benar.	Berhasil <i>login</i> kedalam sistem.		
		Memasukkan <i>username</i> atau <i>password</i> yang salah.	Tidak dapat masuk kedalam sistem dan sistem menampilkan pesan bahwa <i>username</i> atau <i>password</i> yang dimasukkan salah.		

Pengujian dilakukan oleh ahli pemrograman (*programmer/developer*) dengan rumus analisis sebagai berikut:

$$X = 1 - \frac{A}{B}$$

keterangan: X = *functionality*  
A = banyaknya fungsi yang tidak valid  
B = banyaknya seluruh fungsi

Berdasarkan rumus pengukuran implementasi *functionality* tersebut, *functionality* dianggap baik jika x mendekati 1 ( $0 \leq x \leq 1$ ).

Pengujian subkarakteristik *security* menggunakan *software Acunetix Web Vulnerability Scanner (WVS)*. *Acunetix WVS* adalah alat pengujian keamanan untuk aplikasi *web* seperti *SQL injections*, *cross site scripting* dan eksploitasi keamanan yang lain. Pengujian dilakukan dengan menganalisis hasil *scan result Acunetix WVS* yang berupa *web alert*. Dari *web alert* ini, keamanan sebuah sistem dapat dikategorikan berdasarkan empat *level* yaitu *severity high*, *severity medium*, *severity low*, dan *severity info*. Aspek *security* dikatakan berbahaya jika terdapat pada *level severity high* (Acunetix, 2013:40).

## 2. Aspek Efisiensi (*Efficiency*)

ISO/IEC 9126 mendefinisikan *efficiency* sebagai hubungan antara tingkat kinerja perangkat lunak dan jumlah sumber daya yang digunakan di bawah kondisi tertentu. *Efficiency* menilai kemampuan perangkat lunak memberikan waktu respon pengolahan yang sesuai dalam melakukan fungsinya (*time behavior*). *Efficiency* juga menilai kemampuan perangkat lunak dalam menggunakan sumber daya yang dimilikinya ketika melakukan fungsi yang ditentukan (*resource utilization*).

Instrumen yang digunakan dalam pengujian *resource utilization* adalah *YSlow* dengan *ruleset default* dari *GTMetrix* yaitu V2. Tabel 2 menyajikan daftar 23 *testable rules* dari V2:

Tabel 2. *Testable Rules YSlow*

No	Test	Grade
1	<i>Use a Content Delivery Network (CDN)</i>	A sampai F
2	<i>Use cookie-free domains</i>	A sampai F
3	<i>Make fewer HTTP requests</i>	A sampai F
4	<i>Avoid URL redirects</i>	A sampai F
5	<i>Avoid empty src or href</i>	A sampai F
6	<i>Add Expires headers</i>	A sampai F
7	<i>Compress components with gzip</i>	A sampai F
8	<i>Minify JavaScript and CSS</i>	A sampai F
9	<i>Make AJAX cacheable</i>	A sampai F
10	<i>Put CSS at top</i>	A sampai F
11	<i>Remove duplicate JavaScript and CSS</i>	A sampai F
12	<i>Put JavaScript at bottom</i>	A sampai F
13	<i>Avoid AlphaImageLoader filter</i>	A sampai F
14	<i>Avoid HTTP 404 (Not Found) error</i>	A sampai F
15	<i>Reduce the number of DOM elements</i>	A sampai F
16	<i>Do not scale images in HTML</i>	A sampai F
17	<i>Use GET for AJAX requests</i>	A sampai F
18	<i>Avoid CSS expressions</i>	A sampai F
19	<i>Reduce DNS lookups</i>	A sampai F
20	<i>Reduce cookie size</i>	A sampai F
21	<i>Make favicon small and cacheable</i>	A sampai F
22	<i>Configure entity tags (ETags)</i>	A sampai F
23	<i>Make JavaScript and CSS external</i>	A sampai F

Instrumen yang digunakan dalam pengujian *time behavior* adalah *pageSpeed Insights* yang dikembangkan oleh Google Developer. Tabel 3 menyajikan parameter pengujian *time behavior*:

Tabel 3. *Parameter Time Behaviour PageSpeed Insights*

No	Test	Grade
1	<i>Specify a chace validator</i>	A sampai F
2	<i>Specify image dimensions</i>	A sampai F
3	<i>Avoid a character set in the meta tag</i>	A sampai F
4	<i>Leverage browser caching</i>	A sampai F
5	<i>Server scaled images</i>	A sampai F

No	Test	Grade
6	<i>Defer parsing of JavaScript</i>	A sampai F
7	<i>Minify HTML</i>	A sampai F
8	<i>Minify CSS</i>	A sampai F
9	<i>Specify a character set early</i>	A sampai F
10	<i>Optimize images</i>	A sampai F
11	<i>Avoid bad requests</i>	A sampai F
12	<i>Avoid landing page redirect</i>	A sampai F
13	<i>Enable gzip compression</i>	A sampai F
14	<i>Enable Keep-alive</i>	A sampai F
15	<i>Inline smal CSS</i>	A sampai F
16	<i>Inline small JavaScript</i>	A sampai F
17	<i>Minify JavaScript</i>	A sampai F
18	<i>Minimize redirect</i>	A sampai F
19	<i>Minimize request size</i>	A sampai F
20	<i>Optimize the order of styles and scripts</i>	A sampai F
21	<i>Put CSS in the document head</i>	A sampai F
22	<i>Remove query string from static resources</i>	A sampai F
23	<i>Serve resources from a consistent URL</i>	A sampai F
24	<i>Specify a Vary: Accept-Encoding header</i>	A sampai F
25	<i>Avoid CSS @import</i>	A sampai F
26	<i>Combine images using CSS sprites</i>	A sampai F
27	<i>Prefer asynchronous resources</i>	A sampai F

Pengujian *resource utilization* dan *time behavior* menghasilkan nilai yang menunjukkan tingkat performance masing-masing halaman *web* dalam rentang A sampai F. Total nilai sistem adalah rata-rata dari pengujian setiap halaman. Nilai A merupakan nilai dengan tingkat performance paling baik, sedangkan Nilai F adalah sebaliknya. Rentang nilai ini dijelaskan dalam Gambar 4.

(A) $90 \leq S \leq 100$	(B) $80 \leq S \leq 90$	(C) $70 \leq S \leq 80$	(D) $60 \leq S \leq 70$	(E) $50 \leq S \leq 60$	(F) $0 \leq S \leq 50$
-----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	---------------------------

Gambar 4. *Yslow Ruleset Matrix*  
Sumber: Yslow (2014)

Menurut Dhiauddin dkk. (2014:19) waktu respon aplikasi *web* yang baik adalah kurang dari atau sama dengan 5 detik.

### 3. Aspek Keandalan (*Reliability*)

ISO-9126 mendefinisikan *reliability* adalah kemampuan perangkat lunak untuk mempertahankan tingkat kinerja tertentu ketika digunakan dalam kondisi tertentu. Dalam hal ini, perangkat lunak diharuskan mampu menyediakan ketersediaan layanan yang dibutuhkan oleh pengguna. Sedangkan menurut McCall, Richard, dan Walters yang dikutip oleh Roger S. Pressman, *reliability* sebuah perangkat lunak adalah seberapa baik perangkat lunak memberikan hasil yang akurat, tanpa kegagalan. Selain berapa lama perangkat lunak dapat dioperasikan sebelum terjadinya kegagalan, *reliability* juga tentang penyediaan hasil yang benar, penanganan deteksi kesalahan, dan *recovery* untuk menghindari kegagalan.

Pengujian aspek *reliability* dilakukan dengan menggunakan menggunakan parameter dasar WAPT 8.1, yaitu *failed session*, *failed pages*, dan *failed hits*. Kemudian nilai *reliability* ditentukan menggunakan model Nelson dengan perhitungan:

$$R1 = 1 - \frac{ne}{n}$$

Keterangan:  $R1$  = nilai *reliability*  
 $ne$  = banyaknya input yang gagal  
 $n$  = banyaknya input

Selanjutnya, untuk mengetahui tingkat *reliability* sistem, maka hasil nilai *reliability* dicocokkan dengan standar telcordia GR 282. Menurut Asthana dan Olivieri (2009), jika 95% test case lolos maka sistem memenuhi aspek *reliability*.

Tabel 4 merupakan tabel instrumen pengujian pada aspek *reliability* menggunakan WAPT 8.1

Tabel 4. Instrumen Pengujian Aspek *Reliability*

No	Metrik	Sukses	Gagal	Presentase	Hasil
1.	<i>Session</i>				
2.	<i>Pages</i>				
3.	<i>Hits</i>				

#### 4. Aspek Kebergunaan (*Usability*)

ISO/IEC 9126 mendefinisikan aspek *usability* sebagai tingkat usaha yang diperlukan dalam penilaian individual dari penggunaan sistem yang dinyatakan oleh pengguna. Menurut *IEEE Standard Glossary of Software Engineering Technology* yang dikutip Simarmata (2010:297) “*Usability*: atribut yang menunjukkan tingkat kemudahan pengoperasian perangkat lunak”. Dari dua definisi tersebut aspek *usability* dapat diartikan sebagai seberapa mudah fungsi-fungsi perangkat lunak untuk dimengerti, dipelajari, dan digunakan oleh pengguna. *Usability* memiliki subkarakteristik- subkarakteristik sebagai berikut:

- a. *Understandability* menilai kemudahan fungsi-fungsi sistem untuk dipahami.
- b. *Learnability* menilai kemudahan fungsi-fungsi sistem untuk dipelajari oleh pengguna, baik pengguna ahli maupun awam.
- c. *Operability* menilai kemudahan perangkat lunak untuk dioperasikan oleh pengguna.

Pengujian aspek *usability* menggunakan kuesioner terstandar SUS (*Software Usability Scale*) yang dibuat oleh John Brooke pada tahun 1986. Kuesioner SUS mengukur aspek *usability* pada subkarakteristik *learnability* dan

*usability*. Pertanyaan pada nomor 4 dan 10 merupakan pertanyaan yang mencirikan subkarakteristik *usability*, selebihnya merupakan subkarakter *learnability*. Tabel 5 merupakan daftar pertanyaan dari kuesioner SUS.

Tabel 5. Kuesioner SUS (*Software Usability Scale*)

No	Pernyataan	Sangat tidak setuju		Sangat setuju		
		1	2	3	4	5
1	Saya akan sering menggunakan aplikasi ini.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	Saya merasa bahwa aplikasi ini tidak komplek.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	Saya merasa bahwa aplikasi ini mudah digunakan	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	Saya membutuhkan bantuan teknis (panduan) untuk dapat menggunakan aplikasi ini.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	Saya merasa bahwa fungsi dalam aplikasi ini terintegrasi dengan baik.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	Saya menemukan banyak inkonsistensi ketika menggunakan aplikasi ini.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	Saya merasa setiap orang dapat mempelajari dengan cepat bagaimana menggunakan aplikasi ini.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8	Saya merasa bahwa aplikasi ini sulit untuk digunakan.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9	Saya merasa nyaman menggunakan aplikasi ini.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10	Saya perlu banyak belajar sebelum menggunakan aplikasi ini.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

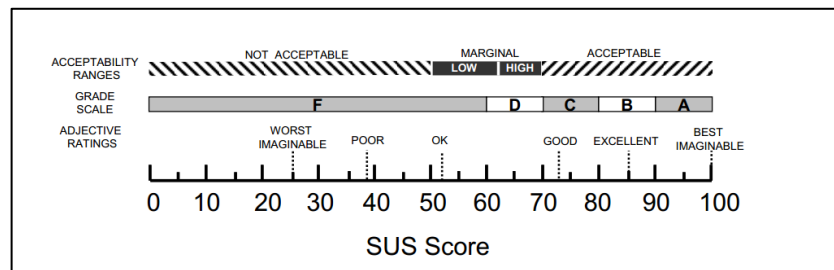
Pada kuesioner SUS, setiap nomor memiliki nilai skala antara 0 sampai 4.

Nilai pada nomor ganjil adalah nilai skala pada nomor tersebut dikurangi 1 ( $x_i - 1$ ), sedangkan nilai pada nomor genap adalah 5 dikurangi nilai skala pada nomor tersebut ( $5 - x_i$ ). Nilai kuesioner SUS diperoleh dari penjumlahan keseluruhan nilai nomor dikalikan 2,5. Hal ini dijelaskan dalam rumus berikut:

$$\begin{aligned}
 Ne &= (5 - x_i) \\
 No &= (x_i - 1) \\
 Ns &= \left( \sum No + \sum Ne \right) \times 2,5
 \end{aligned}$$

Keterangan:  $x_i$  = nilai pada nomor ke-i  
 $No$  = Nilai pada nomor ganjil  
 $Ne$  = Nilai pada nomor genap  
 $Ns$  = SUS Score

Nilai SUS (SUS Score) ada diantara 0-100. Menurut Bangor, Kortum, dan Miller (2009:121), semakin mendekati 100 maka tingkat *usability* perangkat lunak semakin baik. Gambar 5 menunjukkan rentang nilai SUS beserta kategorinya.



Gambar 5. Rentang Nilai SUS beserta Kategorinya

#### 5. Aspek Pemeliharaan (*Maintainability*)

ISO/IEC 9126 mendefinisikan aspek *maintainability* adalah kemampuan perangkat lunak untuk dimodifikasi, mencakup koreksi, perbaikan, adaptasi terhadap perubahan lingkungan dan spesifikasi fungsional. *Maintainability* memiliki subkarakteristik- subkarakteristik sebagai berikut:

- Analyzability* adalah kemampuan perangkat lunak mendiagnosis kekurangan atau penyebab kegagalan.
- Changeability* adalah kemampuan perangkat lunak untuk dimodifikasi tertentu.
- Stability* adalah kemampuan perangkat lunak untuk meminimalkan efek tak terduga dari modifikasi perangkat lunak.
- Testability* adalah kemampuan perangkat lunak untuk divalidasi perangkat lain.

Pengujian aspek *maintainability* dilakukan dengan menggunakan serangkaian metrik yang digunakan untuk mengetes aplikasi secara operasional. Menurut Land (2002), suatu perangkat lunak dikatakan memiliki kualitas *maintability* yang baik jika hasil pengujian menunjukkan perangkat lunak memenuhi aspek *instrumentation*, *consistency*, dan *simplicity*. Metrik tersebut dijelaskan dalam tabel 6.

Tabel 6. Instrumen Pengujian Pada Aspek *Maintainability*

<b>Aspek</b>	<b>Aspek yang dinilai</b>	<b>Kriteria lolos</b>
<i>instrumentation</i>	Terdapat peringatan pada sistem pengelolaan data untuk mengidentifikasi kesalahan.	Ketika ada kesalahan yang dilakukan oleh user, maka sistem akan mengeluarkan peringatan untuk mengidentifikasi kesalahan.
<i>consistency</i>	Penggunaan satu bentuk rancangan tampilan dan bahasa pada seluruh sistem	Bentuk rancangan sistem informasi mempunyai satu bentuk yang sama. Hal ini dapat dilihat pada bagian implementasi sistem
<i>simplicity</i>	Kemudahan dalam pengelolaan, perbaikan, dan pengembangan sistem.	Mudah dikelola, diperbarui, dan dikembangkan. Hal ini dapat dilihat pada tahap-tahap proses penulisan kode.

#### 6. Aspek Portabilitas (*Portability*)

ISO/IEC 9126 mendefinisikan *portability* sebagai kemampuan *software* untuk ditransfer (dipindahkan) dari satu lingkungan ke yang lain. Pengujian Portabilitas dilakukan dengan menggunakan sistem pengelolaan berbasis *web* ke berbagai browser dan perangkat. Menurut Simarmata (2010:264), *portability* adalah usaha yang diperlukan untuk memindahkan perangkat lunak dari sebuah perangkat keras satu ke perangkat keras lain dan dari sebuah perangkat lunak satu ke perangkat lunak lainnya.

Berdasarkan pengertian diatas, pengujian portabilitas *web* dilakukan dengan menjalankan perangkat lunak pada berbagai macam *browser* yang umum digunakan. Browser tersebut adalah *Google Chrome, Mozilla Firefox, Mobile Gadget (Nokia Mobile, Blackberry, Android)*. Instrumen pengujian aspek *portability* disajikan pada tabel 7.

Tabel 7. Instrumen Pengujian Pada Aspek *Portability*

No	<i>Browser</i>	Tampilan	Error
1	Mozila Firefox		Tidak ada
2	Google Chrome		Tidak ada
3	Internet Explorer		Tidak ada
4	Opera Mini		Tidak ada
5	Safari		Tidak ada

#### D. Kerangka Kerja (*Framework*) Aplikasi Web

##### 1. *Hypertext Preprocessor (PHP)*

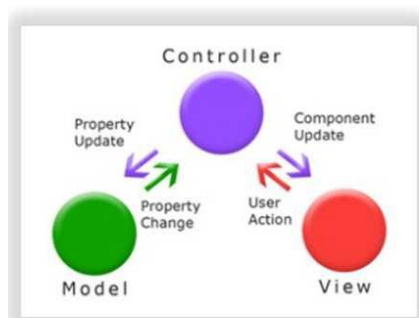
PHP (*Hypertext Preprocessor*) adalah bahasa scripting yang menyatu dengan HTML dan dijalankan pada sisi server. Menurut Nugroho (2004: 140), PHP merupakan bahasa standar yang digunakan dalam dunia *website*. PHP adalah bahasa program yang berbentuk *script* yang diletakkan di dalam server *web*. Semua *script* yang diberikan sepenuhnya dijalankan pada server sedangkan yang dikirimkan ke *browser* hanya hasilnya saja. PHP merupakan sebuah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan situs *web*. Bahasa pemrograman ini digunakan untuk membangun *web* yang dinamis.

##### 2. *Framework CodeIgniter*

Menurut Pratama (2010:10), *framework* atau kerangka kerja merupakan sebuah set kode program atau *library* yang menyediakan fungsionalitas secara

umum untuk keseluruhan kelas (*class*) aplikasi. *Framework* merupakan kumpulan kelas (*class*) dan fungsi (*function, method*) yang disusun secara sistematis berdasarkan kegunaan atau fungsionalitas tertentu untuk mempermudah pembuatan atau pengembangan perangkat lunak atau aplikasi.

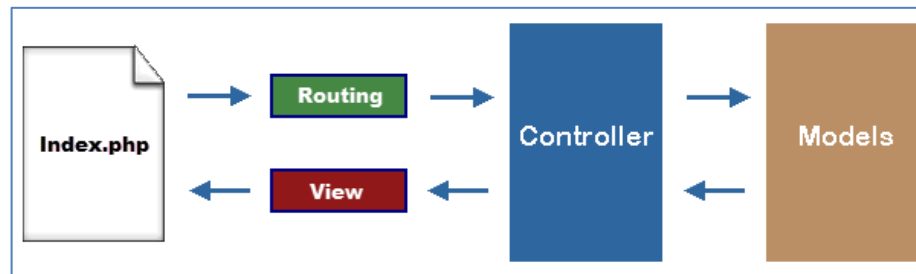
Menurut Ibnu (2011:3), *CodeIgniter* merupakan salah satu *framework* yang dikembangkan oleh Ellislab,Inc yang menggunakan bahasa pemrograman PHP. PHP merupakan suatu bahasa pemrograman dari sisi server yang dapat digunakan untuk membangun aplikasi *web* dinamis. *Framework* aplikasi *web* biasanya mengimplementasikan pola desain *Model View Controller* atau sering disingkat *MVC*, tak terkecuali *CodeIgniter*. Desain *MVC* atau arsitektur *MVC* merupakan salah satu dari sejumlah model infrastruktur aplikasi *web* yang melakukan pemisahan antarmuka-antarmuka pengguna dari fungsionalitas-fungsionalitas aplikasi *web* dan memisahkannya juga dengan isi-isi yang bersifat informasional (Pressman:2010, 470). Teknik pemrograman *MVC* disajikan pada gambar 6.



Gambar 6. Teknik Pemrograman MVC

*Model* merupakan bagian dari aplikasi yang mengimplementasikan logika untuk domain data aplikasi. Objek *model* digunakan untuk mengambil data dari *database* atau menyimpan data ke *database*. *View* adalah komponen yang

digunakan untuk menampilkan antarmuka untuk pengguna aplikasi. *Controller* merupakan komponen untuk menangani interaksi pengguna. *Controller* membaca data dari komponen *view*, mengontrol masukan pengguna, dan mengirim data masukan untuk komponen *model*. Konsep pemrograman *MVC* berjalan menggunakan alur seperti pada gambar 7.



Gambar 7. *Flowchart Codeigniter*

Menurut Sidik (2012:31-32), *index* berfungsi sebagai *controller* di depan yang menginisialisasi sumber yang dibutuhkan untuk menjalankan *Codeigniter*. *Router* akan memeriksa *request* HTTP untuk menentukan apa yang harus dilakukan terhadap *request* tersebut. *Controller* akan memuat *model*, *library* inti, *helper*, dan sumber daya lainnya yang diperlukan untuk memproses *request* tersebut. *View* yang sudah diproses dikirim ke *browser* sebagai suatu hasil yang dapat dilihat. *View* merupakan tampilan terakhir yang digunakan untuk berinteraksi dengan pengguna.

#### **E. Pengertian Penilaian Berbasis Kurikulum 2013**

Penilaian (*assesment*) adalah proses pengumpulan dan pengolahan informasi untuk mengukur pencapaian hasil belajar peserta didik. Menurut Calongesi (1995:21) penilaian adalah keputusan tentang nilai. Oleh karena itu, langkah selanjutnya setelah melaksanakan pengukuran adalah penilaian. Penilaian

dilakukan setelah siswa menjawab soal-soal yang terdapat pada tes. Hasil jawaban siswa tersebut ditafsirkan dalam bentuk nilai.

Penilaian berbasis Kurikulum 2013 berupa penilaian autentik yang merupakan penilaian yang dilakukan secara komprehensif untuk menilai mulai dari masukan (*input*), proses, dan keluaran (*output*) pembelajaran, yang meliputi ranah sikap, pengetahuan, dan keterampilan. Penilaian autentik menilai kesiapan peserta didik, serta proses dan hasil belajar secara utuh. Keterpaduan penilaian ketiga komponen (input-proses-output) tersebut akan menggambarkan kapasitas, gaya, dan hasil belajar peserta didik, bahkan mampu menghasilkan dampak instruksional (*intructional effect*) dan dampak pengiring (*nurturant effect*) dari pembelajaran.

#### **F. Pengolahan Nilai Berbasis Kurikulum 2013**

Untuk mengumpulkan informasi tentang kemajuan belajar peserta didik, terkait sikap, pengetahuan, dan keterampilan perlu adanya langkah-langkah yang harus dilakukan. Langkah tersebut meliputi perencanaan, pelaksanaan dan pengolahan penilaian hasil belajar. Adapun penjelasan dari pengolahan penilaian pada masing-masing aspek sebagai berikut:

##### **1. Penilaian Sikap**

Penilaian sikap dilakukan oleh guru kelas (termasuk guru muatan pelajaran) menggunakan teknik observasi yang ditulis dalam bentuk jurnal. Penilaian diri dan penilaian antarteman dilakukan oleh peserta didik sesuai kebutuhan guru sebagai alat konfirmasi. Contoh pengisian jurnal seperti pada Gambar 8.

No	Waktu	Nama Siswa	Catatan Perilaku	Butir Sikap	Ket.
1.	21/07/14	Bahtiar	Tidak mengikuti sholat Jumat yang diselenggarakan di sekolah.	Ketaqwaan	Spiritual
		Andreas	Menolong orang lanjut usia untuk menyeberang jalan di depan sekolah.	Kepedulian	Sosial
2.	22/09/14	Burhan	Mempengaruhi teman untuk tidak masuk sekolah.	Kedisiplinan	Sosial
		Andreas	Mengingatkan temannya untuk melaksanakan sholat Dzuhur di sekolah.	Toleransi beragama	Spiritual

Gambar 8. Contoh Pengisian Jurnal

Data hasil penilaian dalam jurnal dibahas minimal dua kali dalam satu semester. Pembahasan hasil penilaian menghasilkan deskripsi nilai sikap peserta didik seperti pada Gambar 9.

Nama Peserta Didik	: Arora	Kelas	: I-A
NISN/NIS	: 30401540/1415001	Semester	: I (Satu)
Nama Sekolah	: SD Bagimu Negeri	Tahun Pelajaran	: 2015 / 2016
Alamat Sekolah	: Jl. Ahmad Yani No. 45 Balikpapan		
<b>A. SIKAP</b>			
<b>Deskripsi</b>			
<b>1. Sikap Spiritual</b>	Arora sangat taat beribadah, berperilaku syukur, dan selalu berdoa sebelum melakukan kegiatan. Dengan bimbingan dan pendampingan yang lebih, Arora akan mampu meningkatkan sikap toleransi beragama .		
<b>2. Sikap Sosial</b>	Arora sangat jujur, percaya diri, santun, peduli dan tanggung jawab. Dengan bimbingan dan pendampingan yang lebih, Arora akan mampu meningkatkan sikap disiplin.		

Gambar 9. Contoh Hasil Deskripsi Nilai Sikap

## 2. Penilaian Pengetahuan dan Keterampilan

Nilai pengetahuan dan keterampilan diolah secara kuantitatif dengan menggunakan angka dengan skala 0 sampai 100 serta dibuatkan deskripsi capaian kemampuan peserta didik. Deskripsi tersebut berupa kalimat positif terkait capaian kemampuan peserta didik dalam setiap muatan pelajaran yang mengacu pada setiap KD pada muatan mata pelajaran.