BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

A. Konsep Dasar Sistem Informasi

Menurut Taufiq (2013:17), mengatakan bahwa “Sistem Informasi adalah kumpulan dari sub-sub sistem yang saling terintegrasi dan berkolaborasi untuk menyelesaikan masalah tertentu dengan cara mengolah data dengan alat yang namanya komputer sehingga memiliki nilai tambah dan bermanfaat bagi pengguna”.

Menurut hutahaean (2014:13) mengatakan bahwa “Sistem informasi adalah sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung informasi bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan”.

Dari beberapa pendapat yang di atas dapat disimpulkan sistem informasi adalah sisitem yang terdiri dari dari pengumpulan, pemasukan, pemrosesan data, penyimpanan, pengolahan, pengendalian dan pelaporan segingga tercapai sebuah sasaran dan tujuannya.

B. SDLC (*Software Development Life Cycle*)

Menurut Rosa A. S dan M Shalahuding (2015:25), bahwa “ *Software Development Life Cycle*(SDLC), atau yang sering disebut juga *Software Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik)”.

1. Inisiasi (*initiation*)

Tahapan ini biasanya ditandai dengan pembuatan proposal proyek perangkat lunak.

2. Pengembangan konsep sistem (*system concept development*)

Mendefinisikan lingkup konsep termasuk dokumen lingkup sistem, analisis manfaat biaya, manajemen rencana, dan pembelajaran kemudahan sistem.

3. Perencanaan (*Planning*)

Mengembangkan rencana manajemen proyek dan dokumen perencanaan lainnya. Menyediakan dasar untuk mendapatkan sumber daya (*resources*) yang dibutuhkan untuk memperoleh solusi.

4. Analisis Kebutuhan (*requirements analysis*)

Menganalisis kebutuhan pemakai sistem perangkat lunak (*user*) dan mengembangkan kebutuhan *user*. Membuat dokumen kebutuhan fungsional.

5. Desain (*design*)

Mentransormasikan kebutuhan *detail* menjadi kebutuhan yang sudah lengkap, dokumen desain sistem fokus pada bagaimana dapat memenuhi fungsi-fungsi yang dibutuhkan.

6. Pengembangan (*development*)
Mengonversi desain ke sistem informasi yang lengkap termasuk bagaimana memperoleh dan melakukan instalasi lingkungan sistem yang dibutuhkan membuat basis data dan mempersiapkan prosedur kasus pengujian mempersiapkan berkas atau *file* pengujian, pengkodean, pengompilasian, memperbaiki dan membersihkan program peninjauan pengujian.
7. Integrasi dan Pengujian (*integration and test*)
Mendemonstrasikan sistem perangkat lunak bahwa telah memenuhi kebutuhan yang dispesifikasikan pada dokumen kebutuhan fungsional. Dengan oleh staf penjamin kualitas dan *user*. Menghasilkan laporan analisis pengujian.
8. Implementasi (*implementation*)
Termasuk pada persiapan implementasi perangkat lunak pada lingkungan produksi (lingkungan pada *user*) dan menjalankan resolusi dari permasalahan yang teridentifikasi dari fase integrasi dan pengujian.
9. Operasi (*operation*)
Mendeskripsikan pekerjaan untuk mengoperasikan dan memelihara sistem informasi pada lingkungan produksi (lingkungan pada *user*), termasuk implementasi pada proses peninjauan.
10. Disposisi (*disposition*)
Mendeskripsikan aktifitas akhir dari pengembangan sistem dan membangun data yang sebenarnya sesuai dengan aktifitas *user*.

Analisis dan Desain sering dikelompokkan sebagai proses sistem/rekayasa karena pada tahapan inilah informasi mengenai kebutuhan perangkat lunak banyak dikumpulkan dan diintegrasikan. Ada beberapa model SDLC yang dapat digunakan. Semuanya memiliki kelemahan dan kelebihan pada setiap model SDLC. Hal terpenting adalah mengenali tipe pelanggan dan memilih menggunakan model SDLC yang sesuai dengan karakter pelanggan dan sesuai dengan karakter pengembangan.

C. *Waterfall*

Menurut Rosa A.S dan M.Shalahudin (2015:28) Menyatakan bahwa :
“Model SDLC air terjun(*waterfall*) sering juga disebut model sekuensial linier (*sequintal linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dari analisis, desain, pengodean, pengujian dan tahap pendukung(*support*)”

1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*.

2. Desain

Desain perangkat lunak adalah proses *multi* langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan ke representasi desain agar dapat diimplementasikan

menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3. Pembuatan kode program

Desain harus ditranslasikan kedalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan dan memastikan keluar yang dihasilkan sesuai dengan yang diinginkan.

5. Pendukung (*support*) atau pemeliharaan (*maintenance*)

Tidak menutupi kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada tapi tidak untuk membuat perangkat lunak baru.

Model air terjun adalah model yang paling sederhana, yang sangat cocok digunakan untuk kebutuhan pelanggan yang sudah sangat paham akan pengembangan perangkat lunaknya dengan spesifikasi yang tidak berubah-ubah. Hal positif dari model air terjun adalah struktur tahap

pengembangan sistem yang jelas, dokumentasi dihasilkan disetiap tahap pengembangan, dan sibuah tahap dijalankan setelah tahap sebelumnya selesai dijalankan, sehingga tidak akan adanya tumpang tidih saat pelaksanaanya. (Rosa dan M. Shalahudin, 2015:30).

D. *Unified Modelling Language (UML)*

Menurut Rosa dan Shalahuddin (2015:137), mengatakan bahwa "Padan perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk membangun perangkat lunak yang dibangun dengan menggunakan teknik pemograman berorientasi objek, yaitu *Unified Modelling Language (UML)*". UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangunkan, dan dokumentasi dari sistem perangkat lunak.

Pada UML 2.3 terdiri dari tiga belas (13) macam diagram yang dikelompokan dalam tiga (3) kategori menurut rosa dan shalahuddin (2015:141).
yaitu

1. *Structure Diagrams*

Yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.

2. *Behavior Diagrams*

Yang digunakan untuk menggambarkan kelakuan suatu sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

3. *Interaction Diagrams*

Yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antara sub sistem pada suatu sistem.

a. *Class Diagram*

Diagram ini menggambarkan struktur sistem dari segi penefisian kelas-kelas yang akan dibuat untuk membangun sistem. *Class* memiliki apa yang disebut atribut dan metode atau operasi.

b. *Object Diagram*

Diagram ini menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem. Pada diagram objek harus dipastikan semua kelas yang sudah didefinisikan pada diagram kelas harus dipakai objeknya, karena jika tidak, pendefinisian kelas itu tidak dapat dipertanggungjawabkan.

c. *Component Diagram*

Diagram ini dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada didalam sistem.

d. *Composite Structure*

Diagram Diagram ini digunakan untuk menggambarkan struktur dari bagian-bagian yang saling berhubungan maupun mendeskripsikan struktur pada saat berjalan (*runtime*) dari *instance* yang saling berhubungan.

e. *Package Diagram*

Diagram ini menyediakan cara mengumpulkan elemen-elemen yang saling terkait dalam diagram UML. Hampir sama diagram dalam UML dapat dikelompokkan menggunakan *package diagram*.

f. *Deployment Diagram*

Diagram ini menunjukkan konfigurasi komponen dalam aplikasi. Dapat digunakan untuk memodelkan sistem tambahan (*embedded system*) yang menggambarkan rancangan *device*, *node* dan *hardware*, *system client server*, sistem terdistribusi murni dan rekayasa ulang aplikasi.

g. *UseCase Diagram*

Diagram ini menunjukkan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use Case*, berguna untuk mengetahui fungsi apa saja yang ada pada sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

h. *Activity Diagram*

Menggambarkan aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak dan bukan yang dilakukan *actor*.

i. *State Machine Diagram*

Diagram ini menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem atau objek.

j. *Sequence Diagram*

Menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek message yang dikirim dan diterima antar objek.

k. Communication Diagram

Diagram ini menggambarkan interaksi antar objek atau bagian dalam bentuk urutan pengirim pesan. Diagram ini mempresentasikan informasi yang diperoleh dari class diagram, sequence diagram dan use case diagram untuk mendeskripsikan gabungan antara struktur statis dan tingkah laku dinamis dari suatu sistem.

l. Timing Diagram

Diagram yang fokus pada penggambaran terakit batasan waktu. Diagram ini digunakan untuk menggambarkan tingkah laku sistem dalam periode.

m. Interaction Overview

Diagram Diagram ini mirip dengan diagram aktivitas yang berfungsi untuk menggambarkan sekumpulan urutan aktivitas.

E. ERD (*Entity Relationship Diagram*)

1. Pengertian ERD

Menurut Yanto (2016:32) menyatakan bahwa "ERD adalah suatu diagram untuk menggambarkan desain konseptual dari model konseptual suatu basis data relasional". ERD juga merupakan gambaran yang satu dengan objek yang lain dari objek di dunia nyata yang sering dikenal dengan hubungan antar entitas Contoh jika membuat ERD dari sistem perpustakaan maka bahan sebagai objek ERD bisa berupa anggota, buku, peminjaman, pengembalian dan sebagainya.

2. Rasio Kardinalitas (*Cardinality Ratio*)

Kardinalitas pemetaan atau rasio kardinalitas menunjukkan jumlah entitas yang dapat dihubungkan ke satu *entity* lain dengan satu relasi. Kardinalitas pemetaan meliputi :

a. Hubungan Satu ke Satu (*One to One*)

Setiap anggota entitas A hanya boleh berhubungan dengan satu anggota entitas B, begitu pula sebaliknya.

b. Hubungan Satu ke Banyak (*One to Many*)

Setiap anggota entitas A dapat berhubungan dengan lebih dari satu anggota entitas B, tetapi tidak sebaliknya.

c. Hubungan Banyak ke Satu (*Many to One*)

Setiap anggota entitas B dapat berhubungan dengan lebih dari satu anggota entitas A, tetapi tidak sebaliknya.

d. Hubungan Banyak ke Banyak (*Many to Many*)

Setiap entitas A dapat berhubungan banyak dengan entitas himpunan entitas B dan demikian pula sebaliknya.

F. *Logical Record Structure (LRS)*

Menurut Simarmata Dalam (Fridayanthie, E. W., & Mahdiati, T. 2016) “Logical Record Structured (LRS) adalah representasi dari struktur record-record pada tabel-tabel yang terbentuk dari hasil relasi antar himpunan entitas”. Menentukan kardinalitas, jumlah tabel, dan ForeignKey (FK).

Berikut adalah cara membentuk skema database atau LRS (Logical Record Strutured) berdasarkan Entity Relationship Diagram :

- a. Jika relasinya satu-ke-satu, maka foreign key diletakan pada salah satu dari dua entitas yang ada tau menyatukan kedua entitas tersebut.
- b. Jika relasinya satu-ke-banyak, maka foreign key diletakan pada entitas Many.
- c. Jika relasinya banyak-ke-banyak, maka dibua “file konektor” yang berisi dua foreign key yang berasal dari kedua entitas.

G. *Black Box Testing*

Menurut Rosa dan Shalahuddin (2013:275-276) *Black-box testing* adalah “Perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program”. Pengujian ini bermaksudkan untuk mengetahui fungsi-fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan dan pengujian dengan metode black-box testing memungkinkan pengembang software untuk membuat himpunan kondisi *input* yang akan melatih seluruh syarat-syarat fungsional suatu program. Menurut Mahdiati, dkk dalam Jurnal Khatulistiwa Vol. 5 No. 2 (2016) adapun beberapa kategori kesalahan yang diuji oleh *black-box testing* diantaranya yaitu :

1. Fungsi-fungsi yang salah atau hilang,
2. Kesalahan pada *interface*,
3. Kesalahan dalam struktur data atau akses database eksternal,
4. Kesalahan performa, dan
5. Kesalahan inisialisasi dan terminasi.

Dari beberapa pengertian di atas penulis dapat menarik kesimpulan bahwa *black box testing* adalah metode pengujian tanpa melihat struktur program.

H. Bahasa Pemrograman

1. HTML (*Hyper Text Markup Language*)

Menurut Anhar (2010:40) “HTML merupakan bahasa pemrograman *web* yang memiliki sintak atau aturan tertentu dalam menuliskan *script* atau kode-kode, sehingga *browser* dapat menampilkan informasi dengan membaca kode-kode HTML”. Untuk memudahkan kita dalam mempelajari dasar pemrograman HTML (*HyperText Markup Language*) dengan sekumpulan simbol-simbol atau tag-tag yang dituliskan dalam sebuah *file* yang digunakan untuk menampilkan halaman pada *web browser*. Tag-tag HTML selalu diawali dengan `<x>` dan diakhiri dengan `</x>` dimana *x* tag HTML itu seperti *b*, *i*, *u*, dll.

Program yang digunakan untuk membuat dokumen HTML menggunakan HTML *editor*. Ada banyak HTML *editor* yang bisa digunakan, di antaranya: Notepad, MS. FrontPage, dan Dreamweaver. Setiap dokumen HTML harus diawali dan ditutup dengan tag HTML. Tag HTML memberitahu *browser* bahwa yang ada di dalam kedua tag tersebut adalah dokumen HTML. Bagian *header* dari dokumen HTML diapit oleh tag `<head></head>` di dalam bagian ini biasanya dimuat tag `<title></title>` yang menampilkan judul dari halaman pada *title browser*. Bagian *body* dari dokumen HTML diapit oleh tag `<body></body>`. Dokumen *body* digunakan untuk menampilkan *text*, *imagelink*, dan semua yang akan ditampilkan pada halaman *web*.

2. *PHP (Personal Home Page)*

Menurut Anhar (2010:1) menyatakan bahwa "PHP adalah singkatan dari *PHP Hypertext Preprocessor*. PHP merupakan bahasa pemrograman untuk membuat *web* yang bersifat *server-side scripting*". PHP memungkinkan Anda untuk membuat halaman *web* yang bersifat dinamis. PHP juga bersifat *open source* sehingga setiap orang dapat menggunakannya secara gratis.

Menurut Anhar (2010:2) menyimpulkan bahwa PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP bernama FI (*Form Interpreted*), pada saat tersebut PHP adalah sekumpulan *script* yang digunakan untuk mengolah data *form* dari *web*. Perkembangan selanjutnya adalah rumus melepaskan kode sumber tersebut dan menamakannya PHP/FI, pada saat tersebut kepanjangan dari PHP/FI adalah *Personal Home Page/Form Interpreter*, dengan pelepasan kode sumber ini menjadi *open source*, maka banyak *programmer* yang tertarik untuk ikut mengembangkan PHP. Pengembangan versi-versi php selanjutnya dikembangkan oleh kelompok pengembang, termasuk didalamnya Andi Gutschman dan Zeev Suraski, yang selanjutnya mendirikan Zend Technologies.

3. *JQuery (JavaScript Library)*

Menurut Sigit (2011:1) "jQuery adalah *library* atau kumpulan kode JavaScript siap pakai. Keunggulan menggunakan jQuery di bandingkan dengan *JavaScript* standar, yaitu menyederhanakan kode *JavaScript* dengan cara memanggil fungsi-fungsi yang disediakan oleh jQuery". *JavaScript*

sendiri merupakan bahasa scriptin yang berkerja idisisi *Client/Browser* sehingga *website* bisa lebih interaktif.

4. CSS (*Cascading Style Sheet*)

Menurut Madcoms (2011:100) “CSS (*Cascading Style Sheet*) Adalah sebuah fitur yang diperkenalkan sejak HTML versi 4.0 dan berfungsi untuk menangani masalah tampilan pada HTML seperti jenis, ukuran, dan warna *font*, posisi *teks*, batas tulisan atau *margin*, warna *background*, dan sebagainya”.

Dengan CSS, pembuat website dapat mengontrol/mengatur seluruh tampilan halaman web dengan hanya sebuah dokumentasi CSS. Ekstensi untuk file CSS adalah *.css* (*dot css*). Selain itu, CSS juga dapat menggantikan fungsi table dan dapat meminimalkan ukurandile HTML.

I. Basis Data

Menurut Fathansyah (2012:2) Basisdata terdiri dari 2 kata, yaitu basis dan data. Basis dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep keadaan dan sebagainya, yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, dan kombinasinya.

1. MySQL

Menurut Anhar (2010:44) “Program yang digunakan untuk mengolah dan mengelola *database* adalah MySQL”. Sebuah *website* yang dinamis membutuhkan tempat penyimpanan data agar pengunjung dapat memberikan komentar, saran, dan masukan atas *website* yang dibuat. Tempat penyimpanan data berupa informasi dalam sebuah tabel disebut *database*.

MySQL yang memiliki sekumpulan prosedur dan struktur sedemikian rupa sehingga mempermudah dalam menyimpan, mengatur, dan menampilkan data. Agar memudahkan kita dalam mempelajari dasar pemrograman MySQL.

MySQL (*My Structure Query Language*) merupakan salah satu *Database Management System* (DBMS) dari sekian banyak DBMS seperti *Oracle*, *MS SQL*, *Postgre SQL*, dan lainnya. MySQL berfungsi untuk mengolah *database* menggunakan bahasa SQL. MySQL bersifat *open source* sehingga kita bisa menggunakannya secara gratis. Pemrograman PHP juga sangat mendukung dengan *database* MySQL.

2. XAMPP

XAMPP singkatan X (empat *system* operasi apapun), *Apache*, *MySQL*, *PHP*, *Perl*. Merupakan *tool* menyediakan paket perangkat lunak dalam satu paket. Menurut Dukom (2011:43) “XAMPP adalah perangkat lunak yang disediakan secara gratis, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program”. Untuk menjalankan fungsinya sebagai *server* yang berdiri sendiri, yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penterjemahan bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. Program ini tersedia dalam *GNU “General Public License”* dan bebas, merupakan *web server* yang mudah digunakan yang mampu melayani halaman dinamis. *Software* XAMPP bisa kita *download* secara gratis di *internet*.

2.2. Penelitian Terkait

Dalam tinjauan jurnal ini penulis menggunakan beberapa jurnal yang berhubungan dengan proses untuk mendukung isi dari penulisan skripsi ini diantaranya:

Menurut (Kosasi, 2015), “Tujuan penelitian untuk menghasilkan sebuah rancangan aplikasi sistem E-learning berbasis web untuk membantu meningkatkan motivasi belajar guru dan siswa pada SMK Immanuel Pontianak. Meminimalisasi kesulitan komunikasi antara guru dan siswa dalam proses pembelajaran, memahami dan mengulang setiap materi pelajaran sekolah. Melalui aplikasi berbasis web memungkinkan guru dan siswa dapat berinteraksi dimanapun mereka berada tanpa batasan ruang dan waktu.”

Menurut (Diah, 2015). “Salah satu TPQ yang ada di Kartasura ialah TPQ Al-Fadhillah, merupakan salah satu TPQ yang membutuhkan website sekaligus e-learning. Kegunaan website pada TPQ Al- Fadhillah yaitu agar mudah dalam berkomunikasi serta mencari dan mendapat informasi. Sedangkan e-learning sebagai solusi untuk memberikan suasana pembelajaran yang berbeda dan menarik dan proses belajar mengajar agar lebih mudah dalam menyampaikan materi-materi, seperti iqro’, do’a-do’a harian, dan sejarah Islam.”

Menurut (Andini, 2015). “*E-Learning* ini memberikan beberapa keuntungan kepada para penggunanya terutama para mahasiswa prodi teknik informatika, mahasiswa dapat dengan mudah mengakses media pembelajaran ini, dapat mentransformasi ilmu pengetahuan mengenai metode numerik tanpa tatap muka sehingga dapat memotivasi belajar mahasiswa dan yang pasti juga bisa meningkatkan kompetensi mahasiswa.”

Menurut (Setiawan, 2018). “*E-learning* idealnya membutuhkan dukungan infrastruktur IT yang baik. Berkembangnya layanan *web* dan sosial media ternyata dapat digunakan untuk membangun sebuah mekanisme *e-learning* yang murah dan efisien.

Menurut (Setiyorini, 2018). “Dengan adanya sistem informasi *e-learning* berbasis *web* ini, diharapkan mampu membantu proses pembelajaran di sekolah terasa lebih menyenangkan, efisien dan berjalan lebih cepat jika dibandingkan dengan metode- metode pembelajaran secara manual.”

Menurut (Noor, 2017). “*E-learning* memiliki manfaat bagi lembaga pendidikan untuk menarik lebih banyak peserta didik dibanding metode pembelajaran konvensional. *E-learning* tidak hanya dapat meningkatkan jumlah peserta didik yang ingin mendaftar saja, namun juga memberikan kesempatan peserta didik yang sebelumnya dinyatakan tidak akan mampu mengikuti pendidikan menjadi dapat melakukannya.”

Menurut (Sihotang, 2017). “*E-learning* berisi materi pelajaran yang disampaikan oleh pengajar kepada siswa dan siswa dapat mendownloadnya dengan format PDF dan Document, melihat teman satu kelas, melihat mata pelajaran dan mengerjakan tugas atau quiz. Materi pelajaran dan tugas atau kuis yang ada di dalam sistem *e-learning* dapat disesuaikan dengan kelas yang diduduki.”

Menurut (Yuniva, 2017). “Sistem informasi *Elektronik Learning* berbasis web untuk memudahkan para siswa dalam mempelajari materi yang berkaitan dengan agama islam seperti Aqidah Akhlak, Fiqih dan Qur’an Hadits. Siswa dapat mendownload materi, mengerjakan soal latihan dan melihat hasil nilai quiz. Dalam rancangan sistem ini juga Guru dapat mengupload materi tambahan dan mengupload soal quiz.”