

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem Informasi Penerimaan Siswa Baru

2.1.1 Sistem Informasi

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Sedangkan informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

Sistem informasi adalah suatu sistem didalam organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan^[1].

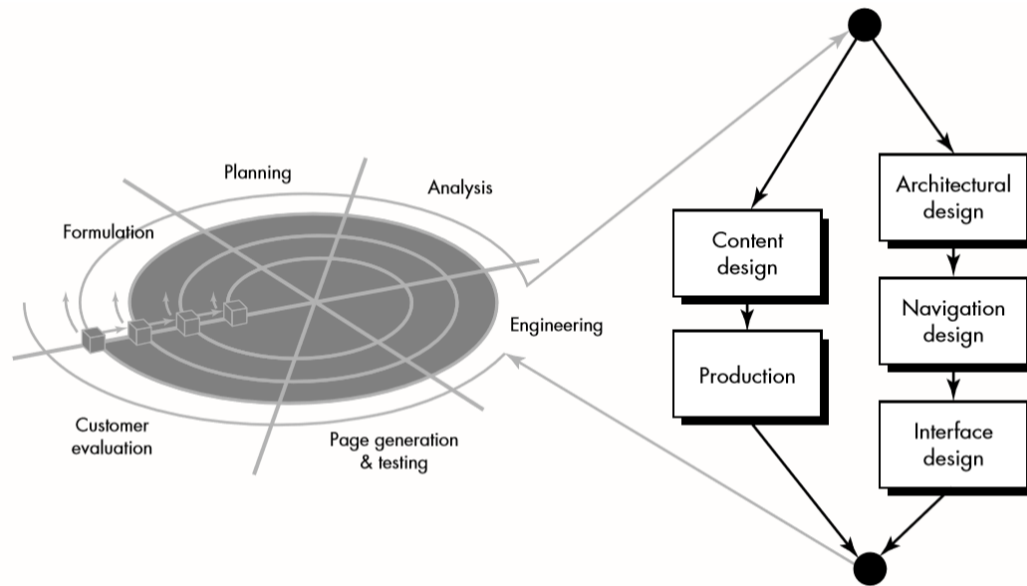
Pengertian sistem menurut Romney dan Steinbart (2015:3): Sistem adalah rangkaian dari dua atau lebih komponen-komponen yang saling berhubungan, yang berinteraksi untuk mencapai suatu tujuan. Sebagian besar sistem terdiri dari subsistem yang lebih kecil yang mendukung sistem yang lebih besar^[2].

2.1.2 Penerimaan Siswa Baru

Penerimaan siswa baru merupakan gerbang awal yang harus dilalui peserta didik dan sekolah didalam penyaringan objek-objek pendidikan. Peristiwa penting bagi suatu sekolah, karena peristiwa ini merupakan titik awal yang menentukan kelancaran tugas suatu sekolah. Kesalahan dalam penerimaan siswa baru dapat menentukan sukses tidaknya usaha pendidikan di sekolah yang bersangkutan. Penerimaan siswa baru dilakukan bukanlah hal yang ringan. Sekolah harus menyiapkan strategi-strategi yang tepat dalam menjalankannya, supaya dapat menarik siswa-siswa yang berkualitas yang mana input sekolah juga bisa lebih baik sehingga proses belajar bisa maksimal dan kualitas sekolah meningkat^[3].

2.2 Definisi Model Pengembangan Perangkat Lunak

Menurut Pressman (2001:769) *web engineering* adalah suatu proses yang digunakan untuk menciptakan suatu sistem aplikasi berbasis web yang berkualitas tinggi. Berikut tahapan dari *web engineering*:^[4].



Gambar 2.1
Tahapan Model Web Engineering

1. *Formulation* (Definisi Masalah)

Merupakan kegiatan yang bertujuan untuk merumuskan tujuan dan aplikasi berbasis web serta menentukan bahasan sistem. Memungkinkan pelanggan dan pengembang untuk mendirikan suatu set tujuan umum dan sasaran untuk membangun dan juga mengidentifikasi web itu sendiri.

Setelah melakukan tahap perumusan masalah, langkah selanjutnya adalah menentukan solusi yaitu membangun sistem.

2. *Planning* (Perencanaan)

Merupakan tahap penggabungan *requirement* (kebutuhan) dan informasi dari user dan perencanaan teknis serta menanggapi respon (tanggapan) dari user. Perencanaan teknis dilakukan dengan mengidentifikasi perangkat lunak

maupun perangkat keras apa saja yang dibutuhkan, respon pengguna dapat dilakukan dengan cara melakukan wawancara terhadap stakeholder.

3. *Analysis (Analisis)*

Pada tahap ini dilakukan analisa kebutuhan dan spesifikasi lengkap tentang isi, jenis skrip yang digunakan, menentukan *web* statis atau dinamis, penggunaan multimedia dan kebutuhan-kebutuhan lainnya. Pada tahap ini harus menerangkan sejelas-jelasnya terhadap situs *web* yang akan dibangun, alat dan bahan yang akan digunakan.

4. *Engineering (Design)*

Tahap perancangan adalah untuk merancang, menghasilkan, dan memperoleh semua teks, grafik, audio, video yang akan disatukan ke dalam aplikasi *web*.

5. *Page Generation and Testing*

Pada tahap ini dilakukan pengujian terhadap program. Kegiatan ini dilakukan untuk mewujudkan desain menjadi website dan menguji aplikasi berbasis web yang telah selesai dibuat. Proses pengujian berfokus pada logika internal software untuk memastikan bahwa semua pernyataan sudah diuji serta pada fungsi eksternal yaitu, mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa dengan input yang terbatas akan didapatkan hasil yang sesuai dengan yang dibutuhkan.

6. *Customer Evaluation*

Pada tahap ini antara *customer* dan tim programmer melakukan evaluasi terhadap sistem yang telah dibangun tersebut. Evaluasi ini dilakukan secara berkala, selama sistem tersebut masih digunakan maka evaluasi akan terus dilakukan sesuai dengan perkembangan teknologi.

2.3 Metode Berorientasi Obyek

Melihat laju perkembangan teknologi informatika yang demikian pesat dewasa ini, maka pendekatan berarah objek diperkirakan akan menggantikan pendekatan Struktural dalam merancang sistem di masa yang akan datang. Hal ini dikarenakan pendekatan berarah objek memiliki beberapa kelebihan diantaranya:

Aplikasi yang akan dibangun didukung oleh perkembangan teknologi Software Engineering yang baru menuju ke konsep Object Oriented.

1. Tehnologi dengan pendekatan object-oriented ini memungkinkan komponen dari program dapat dipakai ulang sehingga mempermudah dalam pengembangan sistem selanjutnya serta program yang dihasilkan pun makin berkualitas baik.
2. Aplikasi yang dibangun dengan pendekatan ini mudah dalam perawatan karena strukturnya mengalami proses inheritance. Sehingga teknologi dengan pendekatan object-oriented ini dapat mengurangi efek samping atau resiko ketika terjadi perubahan pada struktur.

2.3.1 UML (*Unified Modelling Language*)

UML singkatan dari *Unified Modelling Language* yang berarti bahasa pemodelan standar. Chonoles mengatakan sebagai bahasa, berarti UML memiliki sintaks dan semantik. Ketika kita membuat model menggunakan konsep UML ada aturan-aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat berhubungan satu dengan lainnya harus mengikuti standar yang ada UML bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya^[5].

UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk:

1. Merancang perangkat lunak
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis
3. Menjabarkan sistem secara rinci untuk analisis dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

Adapun diagram-diagram UML, Penulis menyediakan 13 macam diagram untuk memodelkan aplikasi perangkat lunak berorientasi objek, Yaitu :

2.3.2 *Activity Diagram*

Diagram Aktivitas (*Activity Diagram*), bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama

penting dalam suatu sistem serta pemodelan fungsi-fungsi suatu sistem dan memberikan tekanan pada aliran kendali antara obyek.

2.3.3 Analisa Dokumen Keluaran

Analisa dokumen keluaran merupakan analisa mengenai keluaran-keluaran dokumen yang dihasilkan melalui proses-proses yang ada didalam sistem berjalan.

2.3.4 Analisa Dokumen Masukan

Analisa dokumen masukan untuk mengetahui dokumen-dokumen apa saja yang digunakan sebagai masukan data pengelolaan sistem permohonan bantuan hukum yang berjalan.

2.3.5 Use Case Diagram

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 aktor dengan sistem informasi yang akan dibuat . Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*^[6].

- a. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau actor

2.3.6 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) digunakan untuk pemodelan basis data rasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD, ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh peter Chen), Barker (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi Crow's foot, dan beberapa notasi lainnya

2.3.7 Logical Record Structure (LRS)

LRS dibentuk dengan nomor dari tipe record digambarkan oleh kotak empat persegi panjang dengan nama yang unik. LRS juga terdiri dari hubungan antara tipe record. Dua metode yang dapat dikonversikan ke LRS. Metode ini dimulai dengan ERD dan langsung dikonversikan ke LRS.

Logical Record Structure dibentuk dengan nomor dari tipe record. Beberapa tipe record digambarkan oleh kotak empat persegi panjang dan dengan nama yang unik. Beda LRS dan ERD nama tipe record berada diluar kotak field tipe record ditempatkan.

2.3.8 Transformasi ERD ke LRS

Kolom relasi atau tabel menunjukkan atribut file yang merupakan suatu data field. Kumpulan pada data field disebut dengan dominan, sedangkan masing-masing baris dari record didalam data tersebut dengan tuple. Setiap tuple atau record mempunyai kunci yang unik (key field) yang terdiri dari primary key (kunci primer) atau foreign key (kunci asing).

2.3.9 Tabel/Relasi

Secara umum tabel adalah koleksi objek yang terdiri dari sekumpulan elemen yang diorganisasi secara continue, artinya memori yang dialokasikan antara satu elemen yang lainnya mempunyai address yang berurutan. Pada tabel, pengertian perlu dipahami adalah:

- a. Keseluruhan tabel (sebagai koleksi) adalah kontainer yang menampung seluruh elemen.
- b. Indek tabel yang menunjukkan address dari sebuah elemen.
- c. Elemen tabel yang dapat dipacu melalui indeknya bertipe tertentu yang sudah terdefinisi.

2.3.10 Spesifikasi Basis Data

Sistem basis data adalah sistem yang tujuan utamanya adalah memelihara data yang sudah diolah informasi dan membuat tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.

2.3.11 Rancangan Layar

Rancangan layar merupakan bentuk tampilan sistem dilayar komputer sebagai antar muka dengan pemakai yang akan dihasilkan dari sistem yang dirancang.

2.3.12 Sequence Diagram

Sequence Diagram menggambarkan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sequen maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstasikan menjadi objek

2.3.13 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang ada dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

- a. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
- b. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Diagram kelas dibuat agar pembuatan program atau *programmer* membuat kelas-kelas sesuai rancangan didalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. Kelas-kelas ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuatan perangkat lunak atau *programmer* dapat membuat kelas-kelas dalam program perangkat lunak sesuai dengan perancangan diagram kelas . susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas 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 disebutkan 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. Semua tabel yang dibuat dibasis data dapat dijadikan kelas, namun untuk tabel dari hasil relasi atau atribut *multivalued* pada ERD dapat dijadikan kelas tersendiri dapat juga tidak asalkan pengaksesannya dapat dipertanggung jawabkan atau tetap ada di dalam perancangan kelas.

2.3.14 *Sequence Diagram*

Sequence diagram menggambarkan kelakuan obyek pada use case dengan mendeskripsikan waktu hidup obyek dan *message* yang dikirimkan dan diterima antar obyek. Oleh karena itu, untuk menggambar *sequence diagram* maka harus diketahui obyek-obyek yang terlihat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Membuat *sequence diagram* juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Banyaknya *sequence diagram* yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada *sequence diagram* sehingga semakin banyak *use case* yang didefinisikan maka *sequence diagram* yang harus dibuat juga semakin banyak^[7].

2.3.15 *Package Diagram*

Package diagram menyediakan cara mengumpulkan elemen-elemen yang saling terkait dalam diagram UML. Hampir semua diagram dalam UML dapat dikelompokkan menggunakan package diagram.

2.3.16 *Deployment Diagram*

Diagram *Deployment (Deployment Diagram)*, bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time*). Memuat simpul-simpul beserta komponen-komponen yang ada di dalamnya. Diagram *deployment* berhubungan erat dengan diagram komponen dimana diagram ini memuat satu atau lebih komponen-komponen. Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing*).

2.4 Tools Pengembangan Perangkat Lunak

2.4.1 ERD (*Entity Relationship Diagram*)

ERD (*Entity Relationship Diagram*) adalah pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram* (ERD). ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Richard Barker, Ian Palmer, Harry Ellis), notasi *Crow's Foot*, dan beberapa notasi lain. Namun yang banyak digunakan adalah notasi dari Chen.

2.4.2 Transformasi ERD ke LRS

Berikut tahapan transformasi ERD ke LRS menurut Wulandari :

1. Konversi ERD ke LRS, *Entity Relationship Diagram* harus diubah ke bentuk LRS (struktur *record* secara *logic*). Dari bentuk LRS inilah yang nantinya dapat ditransformasikan ke bentuk relasi tabel.
2. Konversi ERD ke LRS sebuah model sistem yang digambarkan dengan sebuah model sistem yang digambarkan dengan sebuah ERD akan mengikuti pola pemodelan tertentu. Dalam kaitannya dengan konversi ke LRS, untuk perubahan yang terjadi adalah mengikuti aturan-aturan berikut:
 - a. Setiap entitas diubah ke bentuk kotak dengan nama entitas berada diluar kotak dan atribut berada didalam kotak.
 - b. Sebuah *relationship* kadang disatukan, dalam sebuah kotak bersama entitas, kadang sebuah kotak bersama-sama dengan entitas, kadang disatukan dalam sebuah kotak tersendiri.
 - c. Konversi LRS ke relasi tabel atau tabel adalah bentuk pernyataan data secara grafis dimensi, yang terdiri dari kolom dan baris. Relasi adalah bentuk visual dari sebuah *file* , dan tiap *tuple* dalam sebuah *field*, atau dalam bentuk lingkaran *Diagram Entity Relationship* dikenal dengan sebutan atribut. Konversi dari logical structure, dilakukan dengan cara:
 - 1) Nama *logical record structure* menjadi nama relasi
 - 2) Tiap atribut menjadi sebuah kolom didalam relasi

2.4.3 LRS (*Logical Record Structure*)

LRS (*Logical Record Structure*) dibentuk dengan nomor tipe *record*. Beberapa tipe *record* digambarkan oleh kotak empat persegi panjang dan dengan nama yang unik". Perbedaan LRS dengan ERD dan tipe *record* ditempatkan. LRS terdiri dari *link-link* diantara tipe *record* . *Link* ini menunjukkan arah dari satu tipe *record* lainnya. Banyak *link* dari LRS yang diberi tanda *field-field* yang kelihatan pada kedua *link type record*. Penggambaran LRS mulai dengan menggunakan model yang dimengerti. Dua metode yang dapat digunakan, dimulai dengan

hubungan kedua model yang dapat dikonversikan ke LRS. Metode yang lain dimulai dengan *Entity Relationship Diagram* dan langsung dikonversikan ke LRS^[8].

2.4.4 Spesifikasi Basis Data

Spesifikasi Basis Data merupakan suatu kumpulan data yang berhubungan secara logis dan deskripsi data tersebut, yang dirancang untuk memenuhi informasi yang dibutuhkan organisasi. Artinya, basis data merupakan tempat penyimpanan data yang besar, dimana dapat digunakan oleh banyak pengguna. Bentuk dari spesifikasi basis data sendiri secara umum berupa tabel yang menyajikan informasi *field* untuk seluruh tabel yang digunakan.

Sedangkan sistem berbasis data adalah suatu sistem penyusunan dan pengelolaan *record-record* dengan menggunakan komputer dengan tujuan untuk menyimpan atau merekam serta melihat data operasional lengkap pada seluruh organisasi, sehingga mampu menyediakan informasi yang diperlukan untuk kepentingan proses pengambilan keputusan.

2.5 Definisi Software Pengembangan Perangkat Lunak

2.5.1 Website

Website adalah keseluruhan halaman-halaman web yang terdapat dalam sebuah domain yang mengandung informasi. Sebuah website biasanya dibangun atas banyak halaman web yang saling berhubungan. Dapat juga diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk suatu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman.

2.5.2 PHP

PHP (*Hypertext Preprocessor*) adalah skrip bersifat *server-side* yang ditambahkan ke dalam HTML. PHP sendiri merupakan singkatan dari *Personal Home Page Tools*. Skrip ini akan membuat suatu aplikasi dapat diintegrasikan ke dalam HTML sehingga suatu halaman *web* tidak lagi bersifat statis, namun menjadi bersifat dinamis. Sifat *server side* berarti pengerjaan kode program dilakukan di *server*, baru kemudian hasilnya dikirim ke *browser*". Berdasarkan pengertian di atas, dapat ditarik kesimpulan bahwa PHP (*PHP: Hypertext Preprocessor*) adalah suatu bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang bersifat *server-side* yang dapat ditambahkan ke dalam HTML^[9].

2.5.3 MySQL

MySQL adalah salah satu *Database Management System*(DBMS) dari sekian banyak DBMS seperti *Oracle*, *MS SQL*, *Postagre SQL*, dan lainnya. *MySQL* berfungsi untuk mengolah database menggunakan bahasa *SQL*. *MySQL* bersifat *open source* sehingga kita bisa menggunakannya secara gratis. Pemrograman PHP sangat mendukung/support dengan database *MySQL*^[9].

2.5.4 Database

Database adalah sebuah struktur yang umumnya terbagi dalam 2 hal, yaitu sebuah *database flat* dan sebuah *database relational*. *Database relational* lebih muda dipahami dari pada *database flat* karena *database relational* mempunyai bentuk yang sederhana serta mudah dilakukan operasi data^[10].

2.5.5 XAMPP

XAMPP adalah sebuah software yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolah data *MySQL* di komputer lokal. *XAMPP* berperan sebagai *server web* pada komputer anda. *XAMPP* juga dapat disebut sebuah *Cpanel server virtual*, yang dapat membantu anda melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus online atau terakses dengan internet^[11].

2.6 Tinjauan Penelitian Terdahulu

Dalam penulisan ini, penulis juga mendapatkan beberapa sumber yang berasal dari penelitian yang terdahulu. Berikut adalah beberapa penelitian terdahulu yang dijadikan referensi:

1. Penelitian Yan Irawan (2016)

Penelitian dengan judul “PENDAFTARAN PESERTA DIDIK BARU DENGAN MENGGUNAKAN METODE SIMPLE ADDITIVE WIGHTING (SAW)”. Tujuan penelitian ini adalah mempermudah panitia pendaftaran untuk memutuskan sebuah keputusan yang akurat mengenai penerimaan siswa baru pada Sekolah Menengah Pertama^[12].

2. Penelitian Usman Ependi (2013)

Penelitian dengan judul “PENGEMBANGAN SISTEM INFORMASI ONLINE MENGGUNAKAN METODE WEB ENGINEERING. Tujuan pengembangan sistem ini untuk mengamankan informasi secara digital. Model yang digunakan yaitu web engineering, penulis menganalisa web engineering untuk menganalisis permasalahan serta kebutuhan untuk pemecahan masalah. Hasil dari penelitian ini yaitu dengan adanya media online diharapkan pengguna

bias memperoleh informasi yang diinginkan mengenai mengenai sistem informasi online pada program studi teknik informatika Universitas Bina Darma. Keterkaitan antara penelitian ini dengan penelitian penulis laukakan adalah sama-sama menggunakan model Web Engineering dan menggunakan bahasa pemrograman PHP dengan database MySQL^[13].

3. Penelitian Adi Saputra, Mulyadi dan Martono (2014)

Penelitian dengan judul “PERANCANGAN SISTEM INFORMASI PENERIMAAN SISWA BARU BERBASIS WEB PADA SMK N 6 MUARO JAMBI”. Tujuan untuk mengetahui informasi pendaftaran, melakukan proses pendaftaran hingga melihat pengumuman hasil seleksi. Hal ini mengakibatkan seringkali terjadi antrian disekolah pada saat jadwal penerimaan. Metode pengembangan sistem yang digunakan adalah waterfall. Model Waterfall adalah metode System Development Life Cycle yang paling sederhana. Model ini hanya cocok untuk pengembangan perangkat lunak dengan spesifikasi yang tidak berubah - ubah. Hasil dari penelitian ini dengan adanya sistem penerimaan berbasis web proses penerimaan akan menjadi lebih mudah dan cepat^[14]. Keterkaitan antara penelitian ini dengan penelitian penulis laukakan adalah sama-sama membahas tentang Penerimaan Siswa Baru Berbasis Web^[14].