

BAB II

LANDASAN TEORI

2.1 Pengertian E – Form

Electronic formulir atau disebut juga E-Form adalah formulir elektronik yang digunakan untuk pengisian formulir secara online untuk data tertentu.

2.2 Manfaat E – Form

Membantu memudahkan dalam melakukan pengisian formulir tertentu secara cepat dan efisien.

2.3 PHP

Menurut Sibero (2014:49), “ PHP adalah bahasa pemrograman *interpreter* yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan”. PHP disebut sebagai pemrograman *server side programming*, hal ini dikarenakan seluruh prosesnya dijalankan pada *server*. PHP adalah suatu bahasa dengan hak cipta terbuka atau yang juga dikenal dengan istilah *Open Source*, yaitu pengguna dapat mengembangkan kode-kode fungsi PHP sesuai dengan kebutuhan. Pemrograman PHP dapat ditulis dalam dua bentuk yaitu penulisan baris kode PHP pada file tunggal dan penulisan kode PHP pada halaman (*embedded*). Kedua penulisan tersebut tidak memiliki perbedaan, hanya menjadi kebiasaan gaya penulisan dari programmer.

HTML (Hypertext Markup Language)

2.4 HyperText Markup Language

(HTML) adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah Penjelajah web Internet dan formatting hypertext sederhana yang ditulis kedalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan kedalam format ASCII normal sehingga menjadi *home page* dengan perintah-perintah HTML. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML (Standard Generalized Markup Language), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh World Wide Web Consortium (W3C). HTML dibuat oleh kolaborasi Caillau TIM dengan Berners-Lee Robert

ketika mereka bekerja di CERN (lembaga penelitian fisika energi tinggi di Jenewa) pada tahun 1989.

2.5 Framework Laravel

Laravel adalah sebuah framework PHP yang dirilis dibawah lisensi MIT, dibangun dengan konsep MVC (*model view controller*). Laravel ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, jelas dan menghemat waktu. MVC adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari presentasi. MVC memisahkan aplikasi berdasarkan komponen-komponen aplikasi, seperti : manipulasi data, *controller*, dan *user interface*. **Model**, Model mewakili struktur data. Biasanya model berisi fungsi-fungsi yang membantu seseorang dalam pengelolaan basis data seperti memasukkan data ke basis data, pembaruan data dan lain-lain. **View**, *View* adalah bagian yang mengatur tampilan ke pengguna. Bisa dikatakan berupa halaman web. **Controller**, *Controller* merupakan bagian yang menjembatani model dan *view*.

Beberapa fitur yang terdapat di Laravel : Bundles, yaitu sebuah fitur dengan sistem pengemasan modular dan tersedia beragam di aplikasi. Eloquent ORM, merupakan penerapan PHP lanjutan menyediakan metode internal dari pola “active record” yang mengatasi masalah pada hubungan objek database. Application Logic, merupakan bagian dari aplikasi, menggunakan controller atau bagian Route. Reverse Routing, mendefinisikan relasi atau hubungan antara Link dan Route. Restful controllers, memisahkan logika dalam melayani HTTP GET and POST. Class Auto Loading, menyediakan loading otomatis untuk class PHP. View Composer, adalah kode unit logikal yang dapat dieksekusi ketika view sedang loading. IoC Container, memungkinkan obyek baru dihasilkan dengan pembalikan controller. Migration, menyediakan sistem kontrol untuk skema database. Unit Testing, banyak tes untuk mendeteksi dan mencegah regresi. Automatic Pagination, menyederhanakan tugas dari penerapan halaman.

2.6 MY SQL

MySQL adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengolahan datanya. (abdul kadir : 2013)

MySQL dikembangkan oleh perusahaan swedia bernama MySQL AB yang pada saat ini bernama Tcx DataKonsukt AB sekitar tahun 1994-1995, namun cikal bakal kodenya sudah ada sejak tahun 1979. Awalnya Tcx merupakan perusahaan pengembang software dan konsultan database, dan saat ini MySQL sudah diambil alih oleh Oracle Corp.

MySQL diciptakan oleh Michael “Monty” Widenius pada tahun 1979, seorang proramer asal sewdia yang mengmbangkan sebuah sistem database sederhana yang dinamakan UNIREG yang menggunakan koneksi low-level ISAM database engine dengan indexing.

Adapun kelebihan MySQL dalam penggunaannya dalam database adalah:

1. Free atau gratis sehingga MySQL dapat dengan mudah untuk mendapatkannya
2. MySQL stabil dan tangguh dalam pengoprasiannya
3. MySQL mempunyai sistem keamanan yang cukup baik
4. Sangat mendukung transaksi dan mempunyai banyak dukungan dari komunitas
5. Sangat fleksibel dengan berbagai macam program.

2.7 Metode PIECES

Menurut James Wetherbe 2012, Metode yang digunakan untuk membangun sistem ini adalah metode analisis PIECES (*Performance, Information, Economy, Control, Efficiency dan Services*). Adapun yang dilakukan pada analisis PIECES ini ada beberapa tahapan, yaitu Analisis *Performance*, Analisis *Information*, Analisis *Economy*, Analisis *Control*, Analisis *Efficiency*, dan Analisis *Services*

2.7 Systems Development Life Cycle (SDLC)

Systems development life cycle (SDLC) adalah proses memahami bagaimana sebuah informasi sistem (IS) dapat mendukung kebutuhan bisnis dengan merancang suatu sistem, membangun, dan memberikannya kepada pengguna (Dennis, dkk : 2012).

2.8 UML (*Unified Modeling Language*)

UML singkatan dari *Unified Modeling Language*, UML adalah kosakata umum berbasis objek dan diagram teknik yang cukup efektif untuk memodelkan setiap proyek

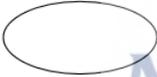
pengembangan sistem mulai tahap analisis sampai tahan perancangan dan tahap implementasi. (:Alan Dennis, 2012 513)

2.9.1 Use Case Diagram

Use case diagram adalah suatu model yang sangat fungsional dalam sebuah sistem yang menggunakan *actor* dan *use case*. Sedangkan pengertian *use case* sendiri adalah layanan atau fungsi-fungsi yang tersedia pada sistem untuk penggunaannya. *Use case* diagram menggambarkan efek fungsionalitas yang telah diharapkan oleh sistem. Kesimpulannya *use case* merupakan gambaran fungsional dari suatu sistem sehingga antara konsumen dan pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun. (Alan Dennis, 2012).

Berikut simbol *use case* diagram (Alan Dennis, 2012) :

Tabel 2. 1Use Case Diagram

Simbol	Nama simbol	Keterangan
	aktor / role	Peran yang pengguna mainkan ketika berinteraksi dengan use case.
	Use case	Deskripsi dari urutan aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
	Subjek	Merepresentasikan runglingkup subjek sistem atau proses bisnis.
	Association	Menghubungkan objek yang satu dengan objek yang lain.
	Extends	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.

	Include	Memungkinkan use case untuk menggunakan fungsional yang disediakan oleh use case lainnya.
	Generalisasi	Memperlihatkan aktor atau use case memiliki sesuatu yang bersifat umum.

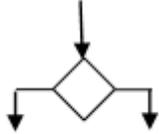
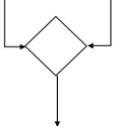
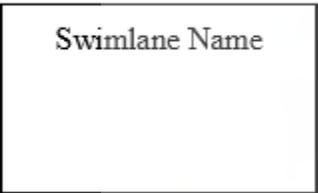
2.9.2 Activity Diagram

Activity diagram digunakan untuk memodelkan perilaku dalam independen proses bisnis benda. *Activity diagram* dapat dipandang sebagai diagram alir data yang digunakan dalam hubungan analysis terstruktur. Tidak seperti diagram aliran data, *activity diagram* termasuk notasi yang membahas pemodelan paralel, kegiatan bersamaan dan proses. (Alan Dennis, 2012).

Berikut simbol pada *activity diagram* (Alan Dennis, 2012) :

Tabel 2. 2 Activity Diagram

Simbol	Nama simbol	Keterangan
	Initial node	Digunakan sebagai tanda awal dari sebuah tindakan atau aktivitas.
	Final-activity node	Digunakan sebagai tanda berakhirnya sebuah tindakan atau aktivitas.
	Activity	Merupakan gambaran dari aktivitas yang terjadi.

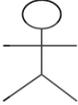
	Decision node	Sebuah pilihan untuk pengambilan keputusan.
	Merge node	Digunakan untuk memastikan bahwa aliran kontrol atau objek mengalir pada satu jalur.
	Object flow	Menunjukkan aliran objek dari satu tindakan ke tindakan yang lain.
	Control flow	Menunjukkan urutan eksekusi.
	Swimlane	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktifitas yang terjadi.

2.9.3 Sequence Diagram

Sequence diagram merupakan salah satu jenis diagram interaksi. *Sequence* menggambarkan benda-benda yang ada didalam kasus penggunaan dan pesan yang melewati antara mereka dari waktu ke waktu pada satu use case. *Sequence diagram* merupakan model dinamis yang menggambarkan urutan eksplisit pesan yang melewati objek dalam interaksi yang didefinisikan. Karena urutan diagram menekankan pemesanan berbasis waktu yang terjadi di antara set benda, *sequence* sangat membantu dalam memahami spesifikasi real-time dan kompleks menggunakan kasus (Alan Dennis, 2012).

Berikut simbol *sequence diagram* (Alan Dennis, 2012):

Tabel 2. 3 *Sequence Diagram*

Simbol	Nama symbol	Keterangan
	Actor	Orang atau sistem yang berasal dari manfaat ke sistem yang berpartisipasi secara berurutan dengan mengirim dan / atau menerima pesan.
	Object	Berpartisipasi secara berurutan dengan mengirim dan / atau menerima pesan yang ditempatkan diatas diagram.
	Message	Pesan yang menjadi komunikasi antar objek.
	Return message	Pesan yang dikirim untuk diri sendiri ditampilkan sebagai panah putus-putus.
	Excecution occurence	Menyatakan interaksi pesan dan objek dalam keadaan aktif
	Lifeline	Menyatakan kehidupan suatu objek di dalam sequence
	Frame	Menunjukkan isi sequence diagram.

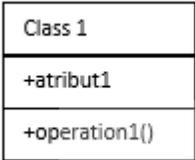
2.9.4 *Class Diagram*

Class diagram merupakan model statis yang menunjukkan kelas dan hubungan antar kelas yang tetap konstan dalam sistem dari waktu ke waktu. *Class diagram* menggambarkan

kelas yang mencakup baik perilaku dan negara dengan hubungan antar kelas. (Alan Dennis, 2012).

Berikut simbol pada *class diagram* (Alan Dennis, 2012) :

Tabel 2. 2 Class Diagram

Simbol	Nama simbol	Keterangan
	Class	Merupakan jenis orang, tempat atau hal mengenai sistem yang dibutuhkan untuk menangkap dan menyimpan informasi.
Attribute name / derived attribute name	Attribute	Properti yang menjelaskan keadaan suatu objek.
Operation name ()	Operation	Merupakan aksi atau fungsi yang dapat dilakukan kelas.
	Agregation	Merupakan sebuah logika dari hubungan antara kelas dan kelas itu sendiri.
	Generalization	Merupakan hubungan antara beberapa class

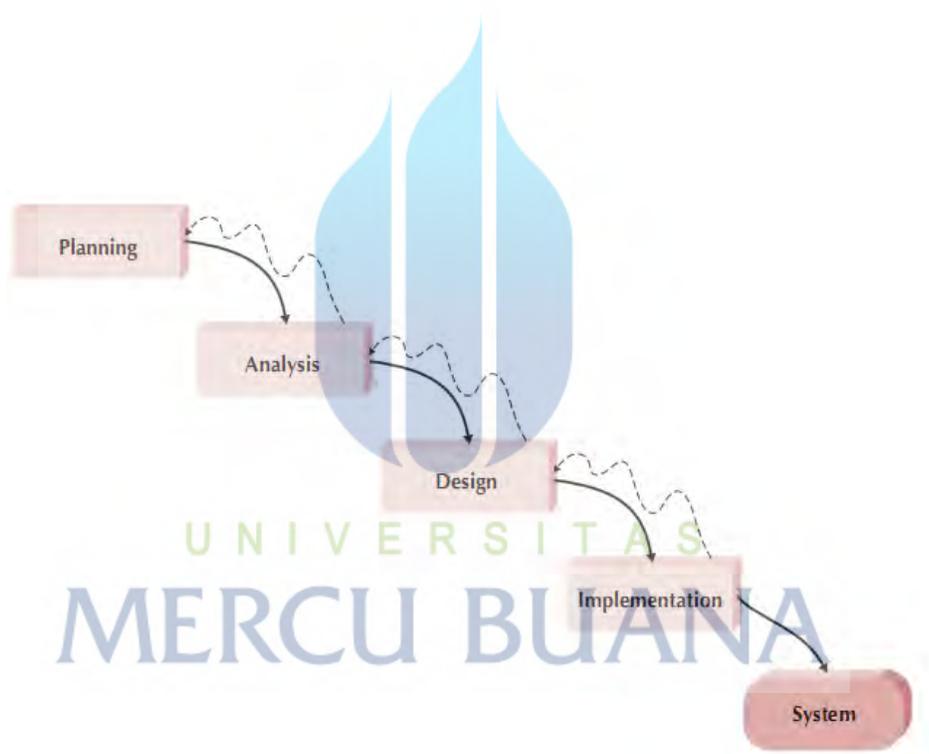
2.10 Xampp

Xampp adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (localhost), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU General Public

License dan bebas, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis. Untuk mendapatkannya dapat mendownload langsung dari web resminya <https://www.apachefriends.org/index.html>

2.11 Metode *Waterfall*

Menurut Dennis *dkk.* (2012:8), “*Waterfall* adalah metodologi berbasis pengembangan air terjun”. Dengan metodologi pengembangan air terjun para analis dan pengguna melanjutkan secara berurutan dari satu tahap ke tahap berikutnya. Berikut 4 fase langkah SDLC dalam metode *waterfall*:



Gambar 2.1. Metode *Waterfall* (Dennis, *dkk.* : 2012)

1. *Planning*

Planning atau perencanaan adalah proses dasar untuk memahami mengapa sistem informasi harus dibangun, dan menentukan bagaimana tim proyek akan membangun

sistem tersebut serta menjelaskan dan mengumpulkan informasi tentang permasalahan serta persyaratannya, kemudian menentukan kriteria dan pembatasan permasalahan serta memberikan alternatif jalan keluarnya.

2. *Annalysis*

Dari tahapan perencanaan, kemudian dilanjutkan dengan proses analisa yang lebih menekankan pada siapa, apa, kapan dan dimana sebuah sistem akan digunakan. Tujuan utama dari tahap analisis adalah untuk memahami dan mendokumentasikan kebutuhan bisnis dan persyaratan proses dari sistem yang baru, serta menganalisa kebutuhan sebagai bahan dalam membuat spesifikasi di tahap selanjutnya.

3. *Design*

Merupakan suatu tahapan bagaimana mencoba mencari solusi permasalahan yang didapat dari tahap analisis. Tahapan mengubah kebutuhan yang masih berupa konsep menjadi spesifikasi sistem yang riil untuk diimplementasikan. Jika pada tahapan analisis (*from requirement to specification*), maka tahapan desain adalah (*from specification to implementation*). Jadi, bagaimana pembuatan spesifikasi yang detail untuk bisa diimplementasikan.

4. *Implementation*

Merupakan tahap bagaimana mengimplementasikan perancangan sistem ke situasi nyata atau desain harus diterjemahkan ke dalam bentuk mesin yang bisa dibaca. Di sini kita mulai berurusan dengan pemilihan perangkat keras dan penyusunan perangkat lunak aplikasi (*pengkodean/coding*).

5. *System*

Pada tahapan sistem dilakukan pengujian (*testing*) dan pemeliharaan, yang dapat digunakan untuk menentukan apakah sistem/perangkat lunak yang kita buat sudah sesuai dengan kebutuhan pengguna atau belum. Jika belum, proses selanjutnya adalah bersifat iteratif, yaitu kembali ke tahap – tahap sebelumnya. Tahap pemeliharaan/perawatan di mana kita mulai melakukan pengoperasian sistem dan jika diperlukan melakukan perbaikan – perbaikan kecil. Kemudian jika waktu penggunaan sistem habis, maka kita akan masuk lagi pada tahap perencanaan (*design*).

2.12 Perangkat Lunak

Menurut Verdi Yasin (2012:2) Perangkat lunak adalah seluruh perintah yang digunakan untuk memproses informasi. Perangkat lunak dapat berupa program atau prosedur. Program adalah kumpulan perintah yang dimengerti oleh komputer sedangkan prosedur adalah perintah yang dibutuhkan oleh pengguna dalam.

2.13 Rekayasa Perangkat Lunak

Menurut Verdi Yasin (2012:2) Rekayasa perangkat lunak adalah suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal yaitu analisa kebutuhan, menentukan spesifikasi dari kebutuhan pengguna, desain, pengkodean, pengujian sampai pemeliharaan sistem setelah digunakan.

2.14 Basis Data (Database)

Menurut Dewi Kusumawati (2015:1) Basis data atau juga sering disebut database adalah kumpulan dari item data yang saling berhubungan satu dengan yang lainnya yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, tersimpan di hardware komputer dan software untuk melakukan manipulasi untuk kegunaan tertentu. Software untuk melakukan manipulasi pada database disebut Database Management System (DBMS). Database juga dapat disebut dengan jantung sebuah sistem informasi atau komponen yang sangat penting dalam sistem informasi atau komponen yang sangat penting dalam sistem informasi.

2.15 Database Management System (DBMS)

Database management system adalah suatu sistem piranti lunak yang memungkinkan pengguna untuk mendefinisikan, membuat, memelihara, serta mengatur akses terhadap basis data. *Database management system* merupakan suatu sistem perangkat lunak yang digunakan untuk membuat, merawat, dan menyediakan pengontrolan akses pada pengguna basis data (Verdi Yasin, 2012:276).

2.16 Kamus Data (Data Dictionary)

Menurut Verdi Yasin (2012:127) Kamus data merupakan suatu tempat penyimpanan (gudang) dari data dan informasi yang dibutuhkan oleh suatu sistem informasi. Kamus data digunakan untuk mendeskripsikan rincian dari aliran data atau informasi yang mengalir

dalam sistem, elemen-elemen data, file maupun basis data (tempat penyimpanan) dalam Data flow diagram (DFD). Simbol penulisan dalam kamus data adalah sebagai berikut :

- = : sama dengan atau terdiri dari atau terbentuk dari
- + : dan
- [] : pilih salah satu
- { } : intersi atau pengulangan
- * : komentar
- | : pemisah

2.17 *Black Box Testing*

Pendekatan *black-box* merupakan pendekatan pengujian untuk mengetahui apakah semua fungsi perangkat lunak telah berjalan dengan semestinya dengan kebutuhan fungsional yang telah didefinisikan (Jiang, 2012). Kasus uji ini bertujuan untuk menunjukkan fungsi perangkat lunak tentang cara beroprasinya. Teknik pengujian ini berfokus pada domain informasi dari perangkat lunak, yaitu melakukan kasus uji dengan mempartisi domain input dan output program. Metode *black-box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan persyaratan fungsional untuk suatu program. Pengujian ini berusaha menemukan kesalahan dalam kategori fungsi-fungsi yang tidak benar atau hilang, kesalahan interface, kesalahan dalam struktur data atau akses basis data eksternal, kesalahan kinerja, dan inisialisasi dan kesalahan terminasi. Yang menjamin penerapan perangkat lunak benar-benar sesuai dengan fungsinya. Sementara validasi merupakan kumpulan aktivitas yang berbeda yang memastikan bahwa perangkat lunak yang dibangun dapat memenuhi keperluan pelanggan.