BAB II
TINJAUAN PUSTAKA

2.1. Sistem

Sistem adalah kumpulan atau himpunan dari unsur atau variable-variabel yang saling terkait, saling berinteraksi, dan saling tergantung satu sama lain untuk mencapai tujuan [3]

Sistem dapat didefinisikan sebagai kumpulan dari elemen-elemen berupa data, jaringan kerja dari prosedur-prosedur yang saling berhubungan, sumber daya manusia, teknologi baik *hadware* dan *software* yang saling berinteraksi sebagao kesatuan untuk mencapai tujuan atau sasaran tertentu yang sama [4]

Berdasarkan pengertian diatas dapat disimpulkan bahwa sistem adalah kumpulan dari komponen-komponen yang saling berkaitan satu dengan yang lain untuk mencapai tujuan dalam melaksanakan suatu kegiatan pokok perusahaan.

2.1.1. Karakteristik Sistem

Suatu sistem mempunyai beberapa karakteristik, yaitu:

1. Komponen Sistem (*Components*)

   Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian sistem, yang mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem keseluruhan.
2. Batas Sistem (*Boundary*)

   Batas sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.
3. Lingkungan Luar Sistem (*Environments*)

   Lingkungan luar (*environments*) dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan merugikan sistem.

4. Penghubung Sistem (*Interface*)

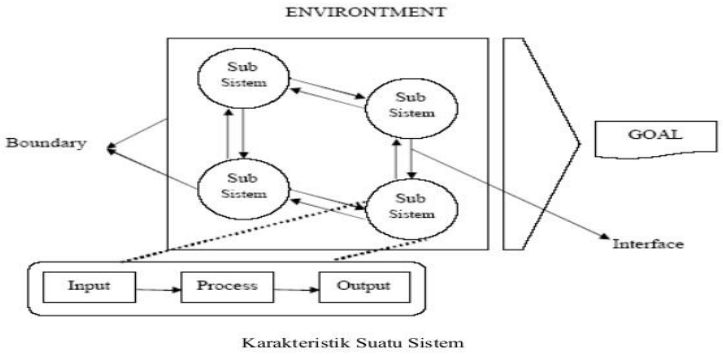
   Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan susbsistem lainnya sehingga memungkinkan sumber-sumber data mengalir antara subsistem yang satu dengan yang lain.

5. Masukan Sistem (*Input*)

   Masukan (*input*) adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*).

6. Keluaran Sistem (*Output*)

   Keluaran (*output*) adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supra sistem.



Gambar 2. 1 Karakteristik Sistem

Sumber : (https://nilamrosfalina.wordpress.com/2017/10/18/karakteristik-sistem-informasi-manajemen/)

7. Pengolah Sistem (*Process*)

   Pengolah Sistem merupakan bagian pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran Sistem (*Objectives*)

   Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

   Sistem merupakan suatu bentuk integrasi antara satu komponen dengan komponen lainnya. Karena sistem memiliki sasaran yang berbeda untuk

setiap kasus yang terjadi, maka sistem dapat diklasifikasikan ke dalam beberapa jenis. Berikut klasifikasi sistem menurut para ahli.

2.1.2. Klasifikasi Sistem

Sistem Informasi Konsep dan Aplikasi. Sistem dapat diklasifikasikan dari berbagai sudut pandang, diantaranya adalah sebagai berikut [5]:

1. Sistem abstak (*abstract system*) dan sistem fisik (*physical system*). Sistem abstrak (*abstract system*) adalah sistem yang berupa pemikiran atau gagasan yang tidak tampak secara fisik. Sedangkan sistem fisik (*physical system*) adalah sistem yang ada secara fisik dan dapat dilihat dengan mata.
2. Sistem alamiah (*natural system*) dan sistem buatan manusia (*human made system*). Sistem alamiah adalah sistem yang keberadaannya terjadi karena proses alam, bukan buatan manusia. Sedangkan sistem buatan manusia (*human made systems*) adalah sistem yang terjadi melalui rancangan atau campur tangan manusia.
3. Sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*). Sistem tertentu (*deterministic systems*) yaitu sistem yang operasinya dapat diprediksi secara cepat dan interaksi diantara bagian-bagiannya dapat dideteksi dengan pasti.

   Sedangkan sistem tidak tentu (*probabilistic systems*) yaitu sistem yang hasilnya tidak dapat diprediksi karena mengandung unsur probabilitas.
4. Sistem tertutup (*closed system*) dan sistem terbuka (*open system*). Sistem tertutup (*closed systems*) yaitu sistem yang tidak berhubungan dengan lingkungan di luar sistem. Sistem ini tidak berinteraksi dan tidak dipengaruhi oleh lingkungan luar. Sistem ini juga bekerja secara otomatis tanpa adanya campur tangan dari pihak luar. Dalam kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah sistem yang relatif tertutup (*relative closed system*). Sistem relatif tertutup biasanya mempunyai masukan dan keluaran yang tertentu serta tidak terpengaruh oleh keadaan di luar sistem. Sedangkan sistem terbuka (*open system*) adalah sistem yang berhubungan dengan lingkungan luar dan dapat terpengaruh dengan keadaan lingkungan luar. Sistem terbuka menerima

input dari subsistem lain dan menghasilkan output untuk subsistem lain. Sistem ini mampu beradaptasi dan memiliki sistem pengendalian yang baik karena lingkungan luar yang bersifat merugikan dapat mengganggu jalannya proses di dalam sistem.

2.2. Informasi

Informasi merupakan kumpulan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima. Tanpa suatu informasi, suatu sistem tidak akan berjalan dengan lancar dan akhirnya bisa mati. Suatu organisasi tanpa adanya suatu informasi maka organisasi tersebut tidak bisa berjalan dan tidak bisa beroperasi [5].

2.3. Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi organisasi yang bersifat manajerial dalam kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan [5].

2.4. *E-Document*

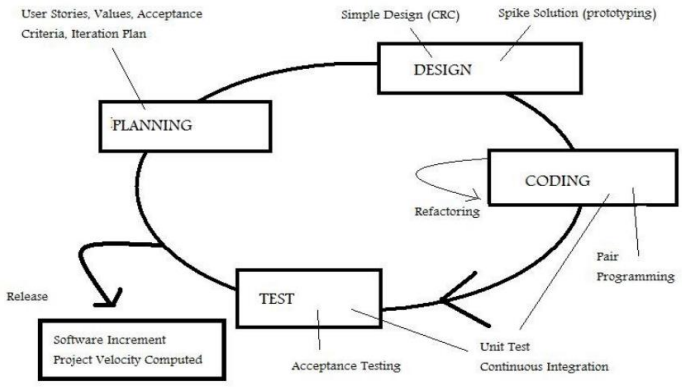
E-Document adalah segala bentuk file digital yang hanya terdapat dalam bentuk/format elektronik seperti data yang disimpan dikomputer, jaringan *(network) back-up* ke dalam *Compact Disk* (CD) ataupun DVD, atau media penyimpanan lainnya. Contoh bentuk dokumennya bisa berupa : *email, voice mail, instans messages (IM), e-calender, audio-file, video*, data-data hasil transfer *handphone*, animasi, grafik, foto, presentasi, *web pages*, dan data digital lainnya [6].

2.5. Metode Pengembangan Sistem *Extreme Programming (XP)*

Extreme Programming (XP) adalah metodologi pengembangan perangkat lunak yang ditujukan untuk meningkatkan kualitas perangkat lunak dan tanggap terhadap perubahan kebutuhan pelanggan. Jenis pengembangan perangkat lunak semacam ini dimaksudkan untuk meningkatkan produktivitas dan memperkenalkan pos pemeriksaan di mana persyaratan pelanggan baru dapat diadopsi [7].

Tahapan-tahapan dari *Extreme Programming* terdiri dari *planning* seperti memahami kriteria pengguna dan perencanaan pengembangan, *designing* seperti perancangan *prototype* dan tampilan, *coding* termasuk pengintegrasian, dan yang terakhir adalah *testing*. Unsur-unsur lain dari *Extreme Programming* meliputi paired programming pada tahapan *coding*, *unit testing* pada semua kode, penghindaran pemrograman fitur kecuali benar-benar diperlukan, struktur manajemen yang datar, kode yang sederhana dan jelas, dan seringnya terjadi komunikasi antara programmer dan pelanggan ketika terjadi perubahan kebutuhan pelanggan seiring berlalunya waktu berlalu.

Metode ini membawa unsur-unsur yang menguntungkan dari praktek rekayasa perangkat lunak tradisional ke tingkat *"ekstrem",* sehingga metode ini dinamai *Extreme Programming*. Unsur-unsur yang menjadi karakteristik metodologi adalah kesederhanaan, komunikasi, umpan balik, dan keberanian. Gambar tahapan XP dapat dilihat pada gambar 2.1 :



Gambar 2. 2 Tahapan *Extreme Programming*

Sumber : [7]

Dibawah ini adalah penjelasan tahapan *Extreme Programming* yaitu :

1. *Planning*

Pada *Planning* berfokus untuk mendapatkan gambaran fitur dan fungsi dari perangkat lunak yang akan dibangun. Aktivitas *planning* dimulai dengan membuat kumpulan gambaran atau cerita yang telah diberikan oleh klien yang akan menjadi gambaran dasar dari perangkat lunak tersebut. Kumpulan gambaran atau cerita tersebut akan dikumpulkan dalam sebuah indeks dimana setiap poin memiliki prioritasnya masing-masing. Tim pengembang aplikasi juga akan menentukan perkiraaan waktu serta biaya yang dibutuhkan untuk masing-masing indeks. Setelah semua kebutuhan terpenuhi, tim XP akan menentukan alur dari pengembangan aplikasi sebelum memulai pengembangan tugas.

2. *Design*

Aktivitas *design* dalam pengembangan aplikasi ini, bertujuan untuk mengatur pola logika dalam sistem. Sebuah desain aplikasi yang baik adalah desain yang dapat mengurangi ketergantungan antar setiap proses pada sebuah sistem. Jika salah satu fitur pada sistem mengalami kerusakan, maka hal tersebut tidak akan mempengaruhi sistem secara keseluruhan.

Tahap *Design* pada model proses *Extreme Programming* merupakan panduan dalam membangun perangkat lunak yang didasari dari cerita klien sebelumnya yang telah dikumpulkan pada tahap *planning*. Dalam XP, proses *design* terjadi sebelum dan sesudah aktivitas *coding* berlangsung. Artinya, aktivitas *design* terjadi secara terus-menerus selama proses pengembangan aplikasi berlangsung.

3. *Coding*

Setelah menyelesaikan gambaran dasar perangkat lunak dan menyelesaikan *design* untuk aplikasi secara keseluruhan, XP lebih merekomendasikan tim untuk membuat modul unit tes terlebih dahulu yang bertujuan untuk melakukan uji coba setiap cerita dan gambaran yang diberikan oleh klien.

Setelah berbagai unit tes selesai dibangun, tim barulah melanjutkan aktivitasnya ke penulisan *coding* aplikasi. XP menerapkan konsep *Pair Programming* dimana setiap tugas sebuah modul dikembangkan oleh dua orang *programmer*. XP beranggapan, 2 orang akan lebih cepat dan baik dalam

menyelesaikan sebuah masalah. Selanjutnya, modul aplikasi yang sudah selesai dibangun akan digabungkan dengan aplikasi utama.

4. *Testing*

Walaupun tahapan uji coba sudah dilakukan pada tahapan *coding*, XP juga akan melakukan pengujian sistem yang sudah sempurna. Pada tahap *coding*, XP akan terus mengecek dan memperbaiki semua masalah-masalah yang terjadi walaupun hanya masalah kecil. Setiap modul yang sedang dikembangkan, akan diuji terlebih dahulu dengan modul unit tes yang telah dibuat sebelumnya.

Setelah semua modul selesai dan dikumpulkan ke dalam sebuah sistem yang sempurna, maka tim XP akan melakukan pengujian penerimaan atau *acceptance test*. Pada tahap ini, aplikasi akan langsung diuji coba oleh *user* dan klien agar mendapat tanggapan langsung mengenai penerapan gambaran dan cerita yang telah dilakukan sebelumnya.

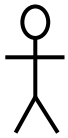
2.6. *Unified Modelling Language* (UML)

Bahasa Pemodelan Pengembangan Sistem (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek [8]. Beberapa jenis diagram UML antara lain sebagai berikut:

2.6.1. Use Case Diagram

Use case diagram 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 [8], simbol-simbol yang ada pada diagram *use case* dapat dilihat pada gambar 2.1 di bawah ini:

Tabel 2. 1 Simbol Diagram *Use Case*

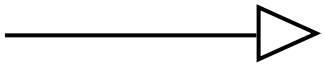
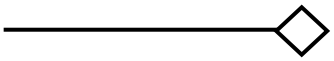
Simbol	Deskripsi
Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama *use case*
Aktor/*actor* 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama *actor*
Asosiasi/*association*	Komunikasi antara aktor dan *use case* yang berpatisipasi pada *use case* atau *use case* memiliki interaksi dengan *actor*
Ekstensi/*extend* <<*extend*>> 	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu mirip dengan prinsip *inheritance* pada pemrograman berorientasi objek biasanya *use case* tambahan memiliki nama depan
Generalisasi/***generalization*** 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan/*Include/uses* <<*include*>> 	Relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya

Sumber: [8]

2.6.2. Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi [8]**,** simbol-simbol yang ada pada diagram kelas pada tabel *class diagram* 2.2 di bawah ini:

Tabel 2. 2 Simbol *Class Diagram*

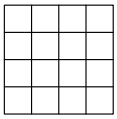
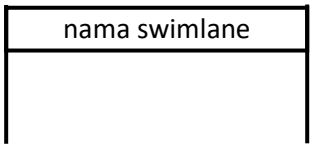
Simbol	Deskripsi
Kelas **nama_kelas** +atribut +operasi()	Kelas pada struktur sistem
Antarmuka/*Interface* **nama_*interface*** 	Sama dengan konsep *interface* dalam pemrograman berorientasi objek
Asosiasi/*association* 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*
Asosiasi berarah/*directed association* 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan *multiplicity*
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi(umum khusus)
Kebergantungan/*dependecy* 	Relasi antar kelas dengan makna kebergantungan antar kelas
Agregasi/*agregation* 	Relasi antar kelas dengan makna semua bagian (*whole-part*)

Sumber : [8]

2.6.3. Activity Diagram

Activity diagram atau diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem [8]**,** simbol-simbol yang ada pada *activity diagram* dapat dilihat pada tabel 2.3 di bawah ini :

Tabel 2. 3 Simbol *Activity Diagram*

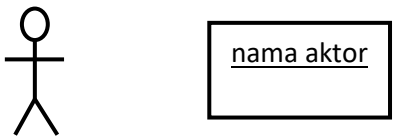
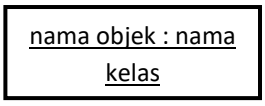
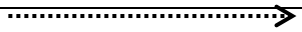
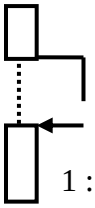
Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem,sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/*decision* 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan/*join*	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Tabel 	Suatu file komputer dari mana data bisa dibaca atau direkam selama kejadian bisnis
Dokumen 	Menunjukan dokumen sumber atau laporan
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

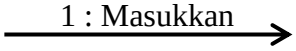
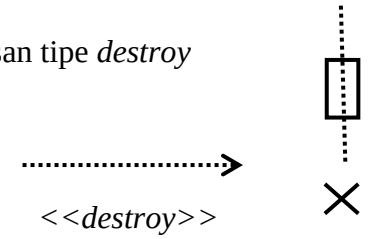
Sumber : [8]

2.6.4. Sequencial Diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram *sequen*ce maka harus diketahui objek-objek yang terlibat dalam sebuah *use case.* simbol-simbol yang ada pada *sequence* digram dapat dilihat pada tabel 2.4 di bawah ini :

Tabel 2. 4 Simbol *Sequence Diagram*

Simbol	Deskripsi
Aktor 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri,
Atau **nama aktor** tanpa waktu aktif	jadi walaupun simbol dari aktor gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda
Garis hidup/*lifeline*	Menyatakan kehidupan suatu objek
Objek 	Menyatakan objek yangberinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan
Pesan tipe *create* <<*create*>> 	Menyatakan suatu objek membuat objek lain, arah panah objek yang dibuat
Pesan tipe *call* 1 : nama_metode() 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri 

Simbol	Deskripsi
Pesan tipe *send* 	Menyatakanbahwa suatu objek mengirimkandata/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirimi
Pesan tipe *destroy* 	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada *create* maka ada *destroy*

2.7. Pengertian MySQL

MySQL adalah server database relasional yang menawarkan berbagai mekanisme untuk memproses data yang dikenal sebagai mesin penyimpanan [9].

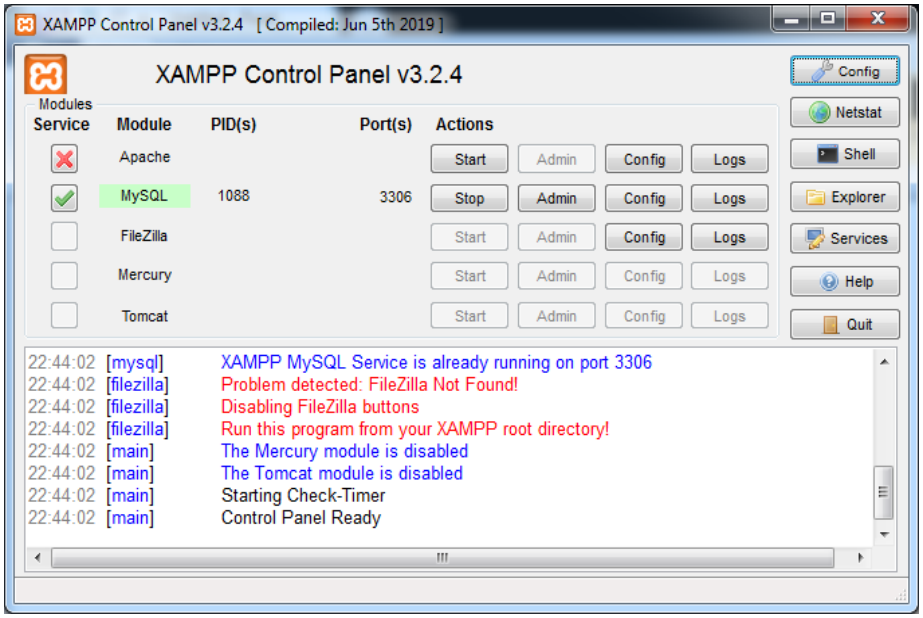
MySQL sama seperti sistem DBMS lainnya, seperti *Oracel*, DB2, dan SQL *Server*. Memungkinkan untuk mengakses, memanipulasi, melindungi, dan memelihara metadata yang diperlukan untuk menentukan data yang disimpan [10].

Jadi dapat disimpulkan bahwa *MySQL* adalah seabuah relasional *database server* suatu bahasa yang digunakan untuk mengelola data pada *Relation* DBMS.

2.8. XAMPP

XAMPP merupakan pengembangan dari LAMP (Linux, Apache, MySQL, PHP and PERL), XAMPP ini merupakan *project* nonprofit yang di kembangkan oleh *Apache Friends* yang didirikan Kai 'Oswalad' Seidler dan Kay Vogelgesang pada tahun 2002, *project* mereka ini bertujuan mempromosikan pengunaan *Apache web server* [10].

XAMPP merupakan paket program web lengkap yang dapat Anda gunakan untuk mempelajari pemrograman web khususnya PHP dan MySQL[10]. Berikuti ini adalah tampilan XAMPP dapat dilihat pada Gambar 2.2



Gambar 2. 3 Tampilan XAMPP

Kelebihan Xampp adalah performa tinggi, stabil, banyak fitur, mudah dikonfigurasi, hanya menggunakan sedikit *resource* di server, tidak bergantung pada *thread* untuk melayani klien.

Kekurangan Xampp adalah tidak mendukung IPV6, *update / patch* terbaru sudah lama, *Fast*-CGI-nya tidak berfungsi maksimal, penggunanya tidak sebanyak aktivitas paket program lainnya.

2.9. PHP

PHP (PHP: *Hypertext apareprocessor*) adalah bahasa *server-side scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Karena PHP merupakan *server-side scripting* maka sintaks dan perintah-perintah PHP akan dieksekusi di server kemudian hasil nya dikirimkan ke *browser* dalam format HTML. Dengan demikian kode program yang ditulis dalam PHP tidak akan terlihat oleh *user* sehingga keamanan halaman web lebih terjamin. PHP dirancang untuk membuat halaman web yang dinamis, yaitu halaman web yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman web. PHP termasuk dalam *Open Source Product*, sehingga *source code* PHP dapat diubah dan didistribusikan secara bebas. PHP juga dapat berjalan pada

berbagai web server seperti IIS (*Internet Information Server*), PWS (*Personal Web Server*), Apache, Xitami. PHP juga mampu lintas platform. Artinya PHP dapat berjalan dibanyak sistem operasi yang beredar saat ini, di antaranya : Sistem Operasi *Microsoft Windows* (semua versi), *Linux, Mac OS, Solaris*. PHP dapat dibangun sebagai modul pada *web server Apache* dan sebagai *binary* yang dapat berjalan sebagai CGI (*Common Gateway Interface*). PHP dapat mengirim HTTP header, dapat mengatur *cookies*, mengatur *authentication* dan *redirect user* [11].

2.10. Pengkodean

Pengkodean adalah suatu tahap dari analisa kebutuhan sistem dan desain sistem yang dituliskan dalam suatu bahasa pemrograman komputer tertentu yang biasanya oleh pabrik komputer sudah ditentukan spesifikasinya [5].

Dalam [12] disebutkan bahwa angka merupakan simbol yang banyak digunakan pada sistem kode akan tetapi kode yang berbentuk angka lebih dari 6 digit akan sangat sulit untuk di ingat kode numerik (*numeric code*) menggunakan 10 macam kombinasi angka di dalam kode. Kode alphabetik (*alphabetic code*) menggunakan 26 kombinasi huruf untuk kodenya. Kode alphanumerik (*alphanumeric code*) merupakan kode yang menggunakan gabungan angka, huruf, dan karakter-karakter khusus meskipun kode numerik, alphabetik dan alphanumerik merupakan kode yang paling banyak digunakan di dalam sistem informasi, tetapi kode yang lain juga mulai banyak digunakan, seperti misalnya kode batang (*bar code*). Ada beberapa macam tipe dari kode yang dapat digunakan di dalam sistem informasi, diantaranya adalah kode mnemonik (*mnemonic code*), kode urut (*sequential code*), kode blok (*block code*), kode grup (*group code*), dan kode desimal (*decimal code*), masing-masing tipe dari kode tersebut mempunyai kebaikan dan kelemahannya tersendiri. Dalam praktek, tipe-tipe kode yang ada dapat dikombinasikan:

1. Kode Mnemonik (*mnemonic code*) digunakan untuk tujuan supaya mudah di ingat. Kode mnemonik dibuat dengan dasar singkatan atau mengambil sebagian karakter dari item yang akan di wakili dengan kode ini. Sebagai contoh kode “P” untuk mewakili pria dan kode “W’ untuk wanita akrab untuk

mudah diingat. Umumnya kode *mnemonic* menggunakan huruf, akan tetapi dapat juga menggunakan gabungan huruf dan angka misalnya barang dagangan komputer IBM PC dengan ukuran memori 640 Kb, colour monitor, dapat dikodekan menjadi K-IBM- PC-640-CO supaya lebih mudah diingat. Kebaikan dari kode ini adalah mudah diingat dan kelemahannya adalah kode dapat menjadi terlalu panjang.

2. Kode Urut (*Sequential Code*) merupakan kode yang nilainya urut antara satu kode dengan kode berikutnya. Contoh kode urut adalah sebagai berikut:
   001 Kass
   002 Piutang Dagang
   003 Persediaan Produk Selesai
   004 Persediaan Produk Dalam Proses
   005 Persediaan Bahan Baku
   006 Biaya Dibayar Dimuka
3. Kode Blok (*block code*) mengklasifikasikan item ke dalam kelompok blok tertentu yang mencerminkan suatu klasifikasi tertentu atas dasar pemakaian maksimum yang diharapkan. Contoh kode blok adalah sebagai berikut: Rekening-rekening dalam buku besar dapat diberi kode dengan mengklasifikasikannya ke dalam kelompok rekening utama sebagai berikut:
   BLOK KELOMPOK 1000-1999
   AKTIVA LANCAR 2000-2999
   AKTIVA TETAP 3000-3999
   HUTANG LANCAR 3500-3999
   HUTANG JANGKA PANJANG 4000-4999
   MODAL
4. Kode Desimal (*desimal code*) mengklasifikasikan kode atas dasar 10 unit angka desimal dimulai dari angka 0 sampai dengan angka 9 atau dari 00 sampai dengan 99 tergantung dari banyaknya kelompok.
5. Kode Grup (*group code*) merupakan kode yang berdasarkan field-field, dan tiap field-fieldnya mempunyai arti.
6. Kode Batang (*barcode*) Sebagai kumpulan kode yang berbentuk garis, dimana masing-masing ketebalan setiap garis berbeda sesuai dengan isi kodenya.

2.11. Pengujian *Black Box*

Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program [13]. Pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut :

1. Fungsi – fungsi yang tidak benar atau hilang,
2. Kesalahan *interface*
3. Kesalahan dalam struktur data atau akses eksternal
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi

Kelebihan *Black Box* adalah *black box testing* dapat menguji keseluruhan fungsionalitas perangkat lunak. *Black box testing* dapat memilih *subset test* yang secara efektif dan efisien dapat menemukan cacat. Dengan cara ini *black box testing* dapat membantu memaksimalkan testing investment [13].

2.12. Hasil Penelitian Sebelumnya

Berikut ini adalah beberapa literatur yang digunakan dalam penelitian, dapat dilihat pada berikut :

Tabel 2. 5 Penelitian Sebelumnya

No	Penulis (Tahun)	Judul	Masalah	Metode yang digunakan	Hasil
1	Solikin and Putra (2018) [1]	Aplikasi *E-Document* Pada Kantor Kepala Desa Tugu Jaya Berbasis Website	Kesulitan didalam hal pencarian dokumen ketika dibutuhkan, resiko kehilangan dokumen disebabakan penyusunan	*Waterfall*	Aplikasi *e-document* pada kantor kepala desa Tugu Jaya Kabupaten Ogan Komering Ilir berbasis web yang bertujuan mempermudah penyimpanan, meningkatkan

No	Penulis (Tahun)	Judul	Masalah	Metode yang digunakan	Hasil
			dokumen yang tidak tersusun dengan rapi, bahkan kerusakan dokumen yang diakibatkan lamanya dokumen tersimpan atau oleh hewan		keamanan dokumen atau agar dokumen tidak rusak, mempermudah pencari dokumen bila diperlukan sewaktu-waktu
2	Susanto (2021) [2]	Implementasi Sistem Informasi *e-Document* Pada Dinas Pekerjaan Umum Kota Semarang	Pengarsipan dokumen masih dilakukan secara manual sehingga lama dalam proses pencarian dan pembuatan surat	*Enterprise Architecture Planning*	Aplikasi sistem informasi pada Dinas pekerjaan umum kota semarang yang diantaranya adalah sistem Informasi pelayanan publik, sistem Informasi *E-Document*, Sistem Informasi Gudang, Sistem Informasi Keuangan dan Sistem Informasi Kepegawaian
3	Bagaskara, Purwaningsih and Budiawan (2020) [14]	Perancangan Sistem Manajemen Dokumen Elektronik Berbasis Web Dengan Metode Sdlc Studi Kasus Teknik Industri	Proses aktivitas dari mata kuliah Kerja Praktek (KP), Kuliah Kerja Industri (KKI) dan Tugas Akhir (TA) yang masih berjalan secara manual	*Development Life Cycle* / SDLC	Sistem informasi manajemen dokumen elektronik berbasis web dalam bentuk *Electronic Document Management System* (EDMS) dengan multiuser dari proses

No	Penulis (Tahun)	Judul	Masalah	Metode yang digunakan	Hasil
		Universitas Diponegoro			aktivitas mata kuliah Kerja Praktek (KP), Kuliah Kerja Industri (KKI) dan Tugas Akhir (TA) di bagian akademik program studi Teknik Industri Universitas Diponegoro
4	Alfia and Rahman (2021) [15]	Perancangan *E-Document System* Berbasis Web Sebagai Upaya Penerapan Lean Proses Dalam Administrasi Dokumen (Studi Kasus : Jurusan Teknik Industri Its)	Masih dilakukan secara manual	*Waterfall*	Hasil dari penelitian diharapkan membantu perguruan tinggi dalam pengelolaan dokumen yang lebih baik dengan adanya penyimpanan salinan fisik dokumen ke dalam media elektronik, mempercepat proses pencarian dokumen yang dilakukan secara elektronik dengan akses melalui internet, efesiensi dalam penyediaan sarana dan prasarana dalam penyimpanan dokumen.

No	Penulis (Tahun)	Judul	Masalah	Metode yang digunakan	Hasil
5	Astuti dan Sunarni (2017) [16]	Sistem Penjualan *Online* Dengan Metode *Extreme Programming*	Pengolahan data yang tidak terintegrasi dengan baik, mengakibatkan pelanggan menunggu terlalu lama untuk membeli produk yang *ready stock.*	*Extreme programming*	Hasil penelitian yaitu aplikasi yang bisa menangani penjualan dan penyediaan barang dan membantu pelanggan untuk bisa melakukan reservasi secara *online* dengan cepat tanpa harus menunggu lama. Penulis mengembangkan sistem *e-commerce* menggunakan pendekatan metodologi *Extreme Programming,* yang dianggap tepat saat ini karena semuanya membutuhkan waktu yang cepat
6	Kiplie *et al* (2018) [17]	*System Development for Document Management System System Development for Document Management System*	*This includes the process of internal development in order to customize the systems, and creating the database system as well.*	*SDLC*	*In order to develop a good and success system, it required to choose suitable software in ensuring that the user can use or browse through the system and view the document in the database. There is some*

No	Penulis (Tahun)	Judul	Masalah	Metode yang digunakan	Hasil
					possible vagueness or misperception when describing information system and digitization system definitions
7	Febriani, Ochi Marshella Wahyuni, Tri (2017) [18]	Perancangan Sistem E-Document Administrasi Logbook Penelitian Pada Unit Layanan Di Bandar Lampung	Masih dilakukan secara manual	*SDLC*	Hasil penelitian adalah Sistem E-Document Administrasi Logbook Penelitian Pada Unit Layanan Di Bandar Lampung
8	Karnila, Sri Meiliza Irianto, Suhendro Y (2018) [19]	Perancangan Strategis Sistem Informasi pada Yayasan Al Falah Pesawaran Menggunakan Enterprise Architecture Planning	berjalannya upaya Yayasan dalam pengembangan M.Ts Al Falah sekaligus pembangunan MA Al Falah yang baru saja beroperasi, menyebabkan adanya overload beban pekerjaan yang dapat berujung pada ketidak teraturan pada proses sistem informasi yang ada	*Watrefall*	Hasil penelniliat adalah Sistem Informasi pada Yayasan Al Falah Pesawaran

No	Penulis (Tahun)	Judul	Masalah	Metode yang digunakan	Hasil
9	Rahadi, Agus (2019) [20]	Perbandingan Metode Analytical Hirearchy Process Dengan Metode Simple Additive Weighting Untuk Perekrutan Dosen Pada IBI Darmajaya Lampung	Kurang efektifnya penilaian dosen	*AHP*	Hasil penelitian yaitu sistem pengolahan data yang menghasilkan alternatif masukan dalam bentuk gambar atau grafik merupakan manfaat dari sistem penunjang keputusan (*decision support systems*) yang dilakukan pada suatu organisasi dan dapat juga diterapkan pada institusi pendidikan tinggi