

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Pada penelitian dilakukan oleh Emily Wheeler pada tahun 2012 mengangkat topik mengenai manfaat *platform* teknologi POS terintegrasi pada restoran. Wheeler mengatakan bahwa dengan menggunakan teknologi POS yang terintegrasi maka akan lebih memberikan banyak informasi, integrasi lebih mudah dan pengendalian biaya yang lebih baik. Emily menganjurkan menggunakan POS terintegrasi berbasis cloud agar pemilik dapat melihat semua informasi POS dari manapun mereka berada, manajer juga dapat langsung melihat jumlah inventaris yang berbeda di tiap-tiap lokasi. Sedangkan bila dibandingkan dengan sistem POS yang konvensional sangat sulit mengumpulkan informasi dari masing-masing lokasi dan sulit untuk membandingkan jumlahnya secara akurat. Sistem POS terintegrasi juga harus dapat menawarkan fitur pelaporan yang dapat disesuaikan sehingga pemilik dapat menyelam jauh ke pengelolaan setiap lokasi. Entah itu untuk mengecek kebutuhan staf, membandingkan waktu pemesanan di seluruh lokasi serta laporan yang dapat disesuaikan untuk kebutuhan memperbaiki atau meningkatkan proses bisnis. Dengan adanya laporan yang bisa disesuaikan maka ada kemampuan untuk melihat seberapa baik penjualan mereka dan mereka dapat membuat penawaran yang khusus pada pelanggan untuk meningkatkan penjualan. Luasnya informasi yang diberikan oleh sistem POS terintegrasi juga menawarkan penghematan biaya. Dengan informasi yang lebih menyeluruh, operator dapat mengelola persediaan dan biaya tenaga kerja lebih efisien, serta mengidentifikasi masalah di toko tertentu dan berupaya menyelesaikannya dengan lebih cepat. Sistem POS adalah salah satu elemen paling vital untuk mencapai restoran yang sukses dan menguntungkan. Memilih sistem berbasis cloud yang terintegrasi memberikan informasi dan kontrol yang superior bagi operator, yang pada akhirnya mengurangi biaya dan meningkatkan kepuasan pelanggan (Wheeler, 2012).

Selanjutnya penelitian yang dilakukan oleh Arindam Banerjee dan Bibek Banerjee tahun 2000 tentang penggunaan sumber informasi *Point Of Sales* manajemen promosi ritel yang efektif di India membahas tentang pentingnya suatu sistem informasi point of sales dalam melakukan strategi promosi untuk menarik semakin banyak pelanggan dengan cara mengelola database pelanggan yang dapat ditambah secara efektif. Berinvestasi dengan sistem POS dalam dunia ritel di India telah menjadi salah satu kisah sukses di pasaran yang menyoroti beberapa aplikasi yang masuk akal dalam menuai keuntungan dari informasi pasar di masa depan. Pada sistem POS mengumpulkan informasi dalam toko ritel dalam setiap transaksi yang dilakukan seperti kuantitas barang yang terjual baik dalam jumlah maupun nilainya, harga transaksi yang dilakukan dan informasi tambahan mengenai program promosi yang disimpan oleh toko saat transaksi berlangsung. POS juga memberikan informasi perilaku langsung tentang apa yang konsumen beli. Penulis juga menyarankan agar organisasi ritel masa depan perlu melampaui integrasi teknologi informasi (TI) karena mereka akan mendapatkan keuntungan

dengan mengadopsi sikap proaktif dalam merancang dan mengembangkan gudang informasi berbasis TI untuk informasi pelanggan dan operasional, yang dapat secara signifikan mendorong pembuatan keputusan strategis di masa depan. Sebagai titik awal, penilaian jenis keputusan yang perlu dilakukan dalam tahap lanjutan dari siklus bisnis diperlukan. Pelajaran dari pasar maju di barat dan jenis masalah keputusan yang mereka hadapi dapat memberi petunjuk pada jenis persyaratan informasi di masa depan (Banerjee & Banerjee, 2000).

Persamaan penelitian diatas dengan yang dilakukan oleh penulis adalah bertujuan agar pemilik usaha dapat mendapatkan informasi yang lebih banyak dan dapat memantau semua informasi di tiap-tiap lokasi dimanapun mereka berada. Karena jika hanya menggunakan cara yang konvensional maka akan sulit mengumpulkan informasi dari masing-masing lokasi. Dengan adanya sistem POS yang terintegrasi maka akan memangkas banyak biaya yang dibutuhkan, lebih mudah mengidentifikasi masalah yang terjadi pada tiap-tiap lokasi dan memperbaiki proses bisnis. Dengan sistem POS terintegrasi juga dapat menggali apa yang diinginkan oleh konsumen dengan cara mengumpulkan informasi yang penting yang akan berperan dalam peningkatan kepuasan konsumen.

2.2 Konsep Dasar Sistem

Dalam membangun sistem informasi, sistem adalah suatu hal yang terpenting. Biasanya sistem informasi selalu ada pada organisasi dan perusahaan agar memudahkan dalam manajemen data. Dari sub-bab ini akan membahas mengenai definisi dan karakteristik sistem.

2.2.1 Definisi Sistem

Robert G. Mudrick dalam Gaol, mengemukakan bahwa “kumpulan dari bagian-bagian yang bergabung untuk suatu tujuan yang sama disebut pengertian dari sistem”. Dalam buku yang lain gaol juga menjelaskan bahwa “Hubungan satu unit dengan unit-unit lainnya yang saling berhubungan satu sama lain dan tidak dapat dipisahkan serta bertujuan sama dalam mencapai tujuan yang telah ditetapkan. Apabila suatu unit terganggu maka unit lain pun akan terganggu untuk mencapai suatu tujuan yang telah ditetapkan tersebut disebut pengertian dari sistem” (Gaol, 2008). Menurut Mustakini, sistem bisa didefinisikan dengan pendekatan prosedur atau dengan pendekatan komponen” (Mustakini, 2008).

Menurut Ida, “Sistem adalah sekumpulan komponen di mana masing-masing komponen memiliki fungsi yang saling berinteraksi dan bergantung serta memiliki satu kesatuan yang utuh untuk bekerja mencapai tujuan tetentu.” Sistem berasal dari bahasa Latin (*systēma*) dan bahasa Yunani (*sustēma*) yaitu sekumpulan unsur ataupun elemen yang saling berkaitan juga mempengaruhi dalam melakukan kegiatan secara bersama-sama untuk mencapai suatu tujuan. Berdasarkan beberapa pengertian di atas sistem merupakan suatu kumpulan komponen-komponen dimana antara satu komponen dengan komponen lainnya memiliki keterkaitan dan saling mendukung satu sama lain dengan keberadaannya sehingga dapat digunakan untuk suatu tujuan tertentu (Nuraida, 2008).

2.2.2 Karakteristik Sistem

Suatu sistem tentu mempunyai karakteristik ataupun sifat-sifat tertentu yang dimiliki yaitu adanya komponen sistem (*components*), batas sistem (*boundary system*), lingkungan luar sistem (*environment system*), penghubung sistem (*interface system*), masukan (*input*), pengolah (*process*), keluaran (*output*), sasaran (*objective*) dan tujuan (*goal*). Adapun karakteristik suatu sistem yang dimaksud (Mustakini, 2008) adalah sebagai berikut:

2.2.2.1.1.1.1 Komponen Sistem (*Components*)

Terdiri dari komponen-komponen yang saling berinteraksi dan bekerja sama membentuk satu kesatuan. Komponen-komponen tersebut terdiri dari beberapa sub sistem, dimana setiap sub sistem tersebut memiliki fungsi khusus yang akan mempengaruhi proses sistem secara keseluruhan.

2.2.2.1.1.1.2 Batas Sistem (*Boundary System*)

Yaitu batas antara suatu sistem dengan sistem lainnya atau dengan lingkungan luarnya. *Boundary* ini memungkinkan suatu sistem dilihat sebagai satu kesatuan. Batas suatu sistem menunjukkan ruang lingkup dari sistem tersebut.

2.2.2.1.1.1.3 Luar Sistem (*Environment System*)

Yaitu apa saja diluar batas dari sistem yang dapat mempengaruhi operasi sistem. Lingkungan luar dapat menguntungkan dan merugikan. Lingkungan yang menguntungkan harus tetap dijaga dan dipelihara, sebaliknya yang merugikan harus ditahan dan dikendalikan, jika tidak ingin terganggu kelangsungan hidup sistem.

2.2.2.1.1.1.4 Penghubung (*Interface*)

Sebagai media penghubung (*interface*) antar sub sistem, yang memungkinkan adanya keterkaitan dari satu sub sistem ke subsistem lainnya. *Output* dari satu subsistem bisa menjadi *input* untuk sub sistem lainnya melalui *interface* untuk mengintegrasikan sub sistem - sub sistem menjadi satu kesatuan.

2.2.2.1.1.1.5 Masukan (*Input*)

Yaitu energi yang dimasukkan ke dalam sistem dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya sistem dapat beroperasi, sedangkan *signal input* adalah energi yang diproses untuk mendapatkan keluaran. Contohnya pada sistem komputer, program merupakan *maintenance input* yang dipakai untuk mengoperasikan komputer dan data adalah *signal input* untuk diolah menjadi informasi.

2.2.2.1.1.1.6 Keluaran (*Output*)

Yaitu hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. *Output* bisa berupa masukan untuk sub sistem lain. Contoh *output* yang tidak berguna adalah panas yang dihasilkan dari sistem komputer sedangkan *output* yang berguna dan dibutuhkan adalah informasi.

2.2.2.1.1.1.7 Pengolah (*Process*)

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah *input* menjadi *output*. Suatu sistem akan mengolah *input* berupa bahan baku menjadi *output* berupa barang siap pakai. Sebagai contoh sistem akuntansi mengolah data-data transaksi menjadi laporan-laporan keuangan dan laporan-laporan lain.

2.2.2.1.1.1.8 Sasaran (*Objectives*) dan Tujuan (*Goal*)

Tentunya sistem memiliki tujuan (*goal*) atau sasaran (*objective*). Jika sistem tidak memiliki sasaran maka sistem tersebut tidak berguna. Sasaran dari sistem sangat menentukan *input* yang dibutuhkan sistem dan *output* yang akan dihasilkan sistem. Suatu sistem dikatakan bersila bila mengenai sasaran atau tujuannya.

2.3 Pengertian Informasi

“Segala suatu hasil yang bermanfaat untuk para pengambil keputusan dalam mencapai tujuan organisasi yang sudah ditetapkan disebut informasi” (Gaol, 2008). “Data-data yang diolah menjadi bentuk yang lebih berguna bagi pemakainya disebut informasi” (Mustakini, 2008).

Menurut Maimunah dkk dalam jurnal CCIT (Henderi, et al., 2011) “Informasi merupakan data yang diolah sedemikian rupa menjadi sesuatu yang lebih berarti bagi penerimanya, dan bermanfaat dalam mengambil suatu keputusan”. Definisi lain dari informasi yaitu hasil dari data yang sudah diolah sebelumnya menjadi bentuk yang lebih berguna dan berarti bagi penerima yang menggambarkan suatu *event* yang nyata yang digunakan dalam pengambilan suatu keputusan.

Menurut Gordon B. Davis dalam buku Gaol (Gaol, 2008), “Informasi yaitu data yang diproses atau diolah menjadi sesuatu yang lebih berarti untuk penerimanya dan merupakan nilai yang sesungguhnya atau dipahami dalam tindakan atau keputusan yang sekarang atau nantinya. Berdasarkan pengertian dari sejumlah ahli sekumpulan data yang telah diolah sedemikian rupa sehingga bermanfaat dan bermakna sehingga dapat digunakan oleh penggunanya untuk kepentingan pengambilan keputusan, sedangkan data itu sendiri merupakan sekumpulan objek maupun fakta-fakta yang terjadi disebut sebagai definisi dari informasi.

2.3.1 Kualitas Informasi

Agar informasi dapat lebih berharga maka menurut Jogiyanto (Mustakini, 2008), informasi harus didukung tiga pilar untuk dapat berguna antara lain tepat sasaran atau relevan yaitu informasi yang diberikan harus mempunyai manfaat untuk penggunanya. Relevan informasi untuk tiap-tiap orang satu dengan yang lain berbeda-beda. Sebab itu harus benar-benar terasa manfaatnya bagi yang membutuhkan. Selain itu tepat waktu makudnya informasi yang datang pada penerimaan tidak boleh terlambat dan usang karena jika informasi usang maka tidak akan memiliki nilai lagi, karena informasi berperan sebagai alat atau tindakan dalam mengambil sebuah keputusan. Apabila pengambilan keputusan terlambat maka akan berakibat fatal bagi perusahaan atau organisasi. Yang terakhir yaitu

akurat yang berarti informasi tidak boleh mengandung kesalahan-kesalahan. Informasi juga seharusnya jelas mencerminkan maksudnya, tidak ambigu.

Selain 3 pilar diatas, menurut John Burch dan Gary Grudnitski dalam buku karangan Ladjamudin, informasi juga harus memenuhi kriteria tambahan yaitu ekonomis yang berarti informasi yang diberikan harus bernilai jual yang tinggi dan memberikan dampak yang luas bagi laju pertumbuhan ekonomi dan teknologi informasi serta dalam menghasilkan informasi perlu meminimalkan biaya operasional. Lalu efisien dalam arti agar lebih berkualitas, informasi harus memiliki kalimat yang sederhana namun mudah dipahami dan memberikan makna yang dalam. Yang terakhir yaitu dapat dipercaya karena sumber yang terpercaya dan sudah teruji kebenarannya akan menghasilkan informasi yang berkualitas (Ladjamudin, 2005).

2.4 Pengertian Sistem Informasi

Agar organisasi atau perusahaan dapat mengolah segala kebutuhan yang diperlukan maka perlu adanya suatu sistem informasi. Pada sistem informasi dibutuhkan keterkaitan antar data yang merupakan suatu nilai yang berdiri sendiri dengan informasi yaitu data yang sudah diolah sehingga menghasilkan sesuatu yang lebih bermanfaat bagi penggunaanya dalam mengambil suatu keputusan untuk mencapai satu kesatuan informasi yang bermanfaat. Sistem sendiri merupakan kumpulan komponen-komponen yang saling terkait dan mendukung satu sama lain dan digunakan untuk pengambilan keputusan dengan tujuan yang sama. Menurut Agus Mulyanto (Mulyanto, 2009), "Sistem informasi merupakan suatu komponen terdiri dari manusia, teknologi informasi dan prosedur kerja yang memproses, menyimpan, menganalisis dan menyebarkan informasi untuk mencapai suatu tujuan". Dari penjelasan diatas dapat disimpulkan bahwa sistem informasi adalah suatu sistem yang diciptakan oleh manusia yang terdiri dari kumpulan komponen dalam suatu organisasi atau perusahaan bertujuan untuk memenuhi kebutuhan informasi dalam mengambil suatu keputusan.

2.4.1 Komponen Sistem Informasi

Sistem Informasi juga memiliki komponen-komponen penting yang berperan dalam membangun keseluruhan sistem. Menurut Jogiyanto (Mustakini, 2008), "Sistem informasi mempunyai enam buah komponen, yaitu komponen masukan, komponen model, komponen keluaran, komponen teknologi, komponen basis data dan komponen kontrol atau pengendalian". Diperlukan keenam komponen yang disebutkan oleh Jogiyanto dalam membentuk satu kesatuan untuk menjadi suatu sistem. Dimana jika salah satu atau lebih komponen tidak ada, maka sistem informasi tersebut tidak bisa menjalankan fungsinya. Komponen sistem informasi yang pertama adalah komponen masukan atau biasa disebut input yang merupakan bahan dasar dalam pengelolaan informasi ada pada komponen ini. Jika komponen ini tidak ada maka tidak akan ada informasi yang masuk dan diolah. Dalam mengelola sistem informasi tentunya mengharuskan adanya suatu data yang harus dimasukkan untuk nantinya diolah agar menjadi suatu informasi yang

berharga. Setelah melalui proses input suatu data, maka informasi didapat dari data yang tersimpan pada basis data dengan cara atau metode tertentu untuk dapat menghasilkan *output* yang sesuai dengan harapan, hal tersebut merupakan komponen model. Proses dalam memanipulasi data input tersebut terdiri atas kombinasi prosedur, model matematis serta logika.

Komponen yang ketiga adalah komponen keluaran yaitu hasil dari sistem informasi berupa informasi yang berguna dan berkualitas bagi para penggunanya. Jika suatu sistem informasi tidak memiliki keluaran maka sistem tersebut gagal karena tidak memenuhi tujuan dibangunnya suatu sistem informasi. Lalu komponen keempat adalah komponen teknologi, dalam sistem informasi merupakan komponen yang penting, karena dengan adanya teknologi maka dalam mendapatkan input, menjalankan model, menyimpan dan mengakses basis data serta menghasilkan keluaran akan dapat dilakukan tepat sesuai pada waktunya. Selanjutnya ada komponen basis data, basis data yaitu kumpulan data yang saling berhubungan dan tersimpan pada *hardware* komputer dan bisa dimanipulasi menggunakan suatu paket *software* yang disebut dengan DBMS (*database management system*). Data-data yang sudah didapatkan perlu dikumpulkan dalam basis data agar bisa diolah dengan baik bila ingin mendapatkan keluaran yang berkualitas. Yang terakhir yaitu komponen control, keluaran yang dihasilkan oleh sistem informasi dapat dijamin oleh komponen kontrol agar bisa menghasilkan informasi yang akurat dan berkualitas (Mustakini, 2008).

2.5 Point Of Sales

Point of sales (POS) dapat diartikan sebagai sebuah sistem yang dapat melakukan proses transaksi. POS dapat digunakan dimana saja asalkan untuk transaksi penjualan, contohnya supermarket, restoran, hotel dan toko-toko lainnya. Dari penjelasan yang sudah disebutkan, POS dapat diartikan sebagai proses untuk pelayanan transaksi dalam sebuah toko retail. Kesimpulan yang didapat dari penjabaran tersebut adalah POS dapat diartikan sebagai sistem yang untuk melakukan transaksi dan termasuk penggunaan mesin kasir.

Menurut Rahman (Rahman, 2013), *software* yang biasanya digunakan pada usaha retail seperti minimarket, swalayan, apotek, café dan sebagainya disebut *point of sales* (POS). Secara umum, proses-proses yang biasanya berkerja pada sistem *point of sales* yang terdapat di perusahaan-perusahaan antara lain *point of sales* (transaksi penjualan), *inventory control* (pengendalian persediaan barang), pembacaan *barcode*, manajemen toko, retur penjualan dan pelaporan.

2.5.1 Manfaat Point Of Sales

Menurut Rahman (Rahman, 2013), beberapa keuntungan jika memakai aplikasi *point of sales* antara lain peningkatan kualitas layanan dengan adanya POS maka perusahaan akan dengan mudah dalam menjalankan proses transaksi yang tepat, cepat dan sistematis. Hal tersebut mendukung orientasi pelayanan usaha terhadap konsumen serta meningkatkan *market interest*. Selain itu dapat pula

meningkatkan citra usaha sebab para konsumen akan memandang usaha sebagai sebuah *market interest*. Lalu ada *competitive advantage*, dengan penerapan teknologi informasi dapat meningkatkan daya saing di kancah bisnis dalam efisiensi waktu terlebih di jaman sekarang dalam menghadapi era *global market*. Manfaat lainnya yaitu memudahkan proses mengontrol dan mengambil keputusan. Biasanya proses *controlling* dapat dengan mudah dilakukan sebab semua laporan dapat tersedia dengan cepat, sehingga memudahkan proses pengambilan keputusan baik secara kolektif maupun personal.

2.6 Konsep Dasar Basis Data

Basis data merupakan suatu komponen untuk melakukan penyimpanan dan pemanipulasian data. Pada sub-bab ini akan membahas tentang pengertian basis data dan operasi-operasi apa saja yang bisa dilakukan dalam basis data secara lebih lengkap.

2.6.1 Pengertian Basis Data

Tabel-tabel yang menyusun sekumpulan data yang saling berkaitan antara satu dengan yang lainnya yang disimpan pada suatu media disebut dengan basis data. Basis data dapat digunakan oleh satu atau lebih program aplikasi secara optimal, data disimpan tanpa mengalami ketergantungan pada program yang akan menggunakannya (Kadir, 1999). Basis data merupakan salah satu komponen yang sangat dibutuhkan dalam membangun suatu sistem karena bertujuan sebagai penyimpanan dan memanipulasi data. Yang melengkapi berdirinya suatu sistem ialah karena adanya suatu basis data. Agar suatu basis data dapat dikatakan baik maka harus memenuhi suatu kriteria. Kriteria penting agar suatu basis data dapat dikatakan baik diantaranya adalah data harus terpadu atau saling terkait, berorientasi pada data bukan program, tidak adanya inkonsistensi data yaitu pada field yang sama selalu muncul data yang tidak konsisten padahal menggunakan kunci yang sama, tidak menyulitkan dalam mengembangkan struktur data serta volumenya, kebutuhan sistem-sistem yang baru dapat dipenuhi dengan mudah dan menjunjung tinggi keamanan maksudnya memiliki wewenang khusus dalam mengakses basis data.

2.6.2 Operasi Dasar Basis Data

Kumpulan data pada basis data disimpan dalam suatu file atau table oleh karena itu pada basis data tidak hanya memiliki satu table saja namun ada banyak yang nantinya bisa diolah untuk dapat memanipulasi data-data yang ada. Adapun dalam melakukan pemanipulasian data diperlukan suatu operasi dasar dalam basis data yaitu: *create database*, *drop database*, *create table*, *drop table*, *insert*, *retrieve/search*, *update* dan *delete*. Pada *create database* yaitu menciptakan lemari sendiri. Jika ingin membuat basis data maka hal pertama yang dilakukan adalah membuat arsip yang baru. Lalu pada operasi *drop database* yaitu dilakukan untuk melenyapkan arsip atau menghapus basis data serta semua *table* berisi data yang ada didalamnya. Setelah membuat *database* baru maka operasi yang selanjutnya dilakukan adalah *create table* yaitu proses membuat *table* baru

kedalam basis data, yaitu map arsip baru yang ditambahkan dalam arsip yang sudah tersedia. Jika ingin menghapus *table* dari basis data atau merusak map arsip lama yang ada didalam sebuah arsip maka operasi yang dilakukan adalah *drop table*. Untuk menambah *table* baru atau lembaran arsip ke sebuah map arsip maka melakukan operasi *insert* dan bila ingin mengubah data yang ada pada *table* dalam rangka perbaikan isi lembaran arsip yang ada di sebuah map arsip maka operasi yang dilakukan adalah *update*. Lalu ada operasi *delete* untuk menghapus data yang ada pada *table* dari map arsip dan retrieve/search untuk pengambilan atau pencarian dari sebuah *table* di sebuah basis data, yang identik dengan pencarian lembaran arsip ke map arsip (Nugroho, 2011).

2.7 Bahasa Pemrograman

Bahasa pemrograman merupakan bahasa yang adalah bahasa tertentu yang digunakan oleh para *programmer* untuk membuat suatu aplikasi atau *software* pada komputer, dasar-dasar yang dipakai adalah prinsip kerja algoritma yang sudah dipelajari sebelumnya, bahasa pemrograman itu sendiri sudah ditemukan pada tahun setelah diketemukannya mesin komputer, mulai generasi yang paling pertama kali yaitu kode mesin sampai generasi sekarang yang banyak digunakan sebagai bahasa pemrograman dalam pembuatan berbagai aplikasi, dalam konteksnya adalah setiap bahasa pemrograman mempunyai yang namanya kode, kode tersebut akan mendasari berbagai pemrosesan tiap langkah berdasarkan tipe data yang dimasukkan oleh programmer.

2.7.1 HTML (*Hypertext Markup Language*)

Hyper Text Markup Language (HTML) adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah penjelajah web Internet dan pemformatan hiperteks sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegerasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan dalam format ASCII normal sehingga menjadi halaman web 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 (Willard, 2006).

Hyper Markup Language (HTML) merupakan salah satu dari banyak dokumen pada internet yang sanggup dibaca oleh *web browser*. Terdapat dua fitur utama yang menjadikan HTML sebagai standar dokumen pada internet. Kedua buah fitur tersebut adalah, pertama, HTML dapat membuat link pada halaman web yang dapat mengantarkan pada halaman web lainnya. Kedua, dokumen HTML dapat disimpan sebagai dokumen teks yang hampir semua komputer dapat membacanya (Willard, 2006).

2.7.2 PHP

PHP adalah Bahasa *server-side-scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis (Arief, 2011). Karena PHP merupakan *server-side-scripting* maka sintaks dan perintah-perintah PHP akan dieksekusi diserver kemudian hasilnya akan dikirimkan ke browser dengan 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 atau singkatan dari Personal Home Page merupakan bahasa skrip yang tertanam dalam HTML untuk dieksekusi bersifat server side (Nugroho, 2006). 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 berjalan di banyak sistem operasi yang beredar saat ini, diantaranya: Sistem Operasi Microsoft Windows (semua versi), Linux, Mac Os, Solaris. PHP dapat dibangun sebagai modul 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. Salah satu keunggulan yang dimiliki PHP adalah kemampuannya untuk melakukan koneksi ke berbagai macam software sistem manajemen basis data atau *Database Management System (DBMS)*, sehingga dapat menciptakan suatu halaman web dinamis. PHP mempunyai koneksitas yang baik dengan beberapa DBMS seperti Oracle, Sybase, mSQL, MySQL, Microsoft SQL Server, Solid, PostgreSQL, Adabas, FilePro, Velocis, dBase, Unix dbm, dan tidak terkecuali semua database ber-interface ODBC. Hampir seluruh aplikasi berbasis web dapat dibuat dengan PHP. Namun kekuatan utama adalah konektivitas basis data dengan web. Dengan kemampuan ini kita akan mempunyai suatu sistem basis data yang dapat diakses.

2.7.3 CSS (*Cascading Style Sheets*)

CSS adalah suatu dokumen yang berdiri sendiri dan dapat dasumsikan dalam kode HTML atau sekedar menjadi rujukan oleh HTML dalam pendefinisian style (Sulistiyawan, Rubianto, & Rahmad Saleh, 2008). CSS menggunakan kode-kode yang tersusun untuk menetapkan style pada halaman HTML atau dapat juga digunakan untuk membuat style baru yang biasa disebut class. CSS atau *Cascading Style Sheet* merupakan bahasa pemrograman yang berfungsi dalam pengaturan penampilan dokumen yang ditulis dengan *markup language* pada sebuah halaman web. Pada umumnya CSS digunakan untuk mendesain pada halaman HTML maupun XHTML, namun kini CSS dapat diaplikasikan untuk segala jenis dokumen XML, termasuk SVG, XUL bahkan Android (Ollie, 2008).

Tujuan utama penggunaan CSS antara lain untuk membedakan konten (isi) dari dokumen dan tampilan dokumen. Dengan adanya perbedaan tersebut

memudahkan pembuatan atau pemrograman ulang web. Bagian yang termasuk dalam desain web meliputi warna, ukuran dan *formatting*.

2.7.4 SQL (Structured Query Language)

SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengakses data didalam sebuah *database* relasional. SQL sering juga disebut dengan istilah *query* dan bahasa SQL secara praktiknya digunakan sebagai bahasa standar untuk manajemen *database* relasional. Hingga saat ini hampir seluruh *server database* atau *software database* mengenal dan mengerti bahasa SQL. Dalam penggunaan SQL terdapat beberapa perintah yang berguna untuk mengakses dan memanajemen data yang terdapat dalam *database*. Jenis peringah SQL secara umum dibagi kepada tiga sub perintah, yaitu DDL (*Data Definition Language*), DML (*Data Manipulation Language*), dan DCL (*Data Control Language*) (Anonymous, 2016). Ketiga sub perintah tersebut sangat perlu untuk dipahami bagi anda yang ingin menguasai bahasa SQL dan mahir dalam pembuatan *database*.

2.7.4.1.1.1 DDL (*Data Definition Language*)

DDL adalah sub perintah dari bahasa SQL yang digunakan untuk membangun kerangka sebuah *database*, dalam hal ini *database* dan table. Terdapat tiga perintah penting dalam DDL, yaitu *CREATE*, *ALTER*, *DROP*. *CREATE* merupakan perintah yang digunakan untuk membuat, termasuk di dalamnya membuat *database* baru, tabel baru view baru, dan kolom baru. Perintah *ALTER* berfungsi untuk mengubah struktur tabel yang telah dibuat. Mencakup didalamnya mengubah nama tabel, menambah kolom, mengubah kolom, menghapus kolom dan memberikan atribut pada kolom. Sedangkan perintah *DROP* berfungsi untuk menghapus *database* atau tabel.

2.7.4.1.1.2 DML (*Data Manipulating Language*)

DML adalah sub perintah dari bahasa SQL yang digunakan untuk memanipulasi data dalam *database* yang telah dibuat. Terdapat empat perintah penting dalam DML, yaitu *INSERT*, *SELECT*, *UPDATE*, dan *DELETE*. Perintah *INSERT* digunakan untuk memasukkan data baru ke dalam sebuah tabel. Perintah ini tentu saja bisa dijalankan ketika *database* dan tabel sudah dibuat. Perintah *SELECT* digunakan untuk mengambil dan menampilkan data dari tabel atau bahkan dari beberapa tabel dengan penggunaan relasi. Perintah *UPDATE* digunakan untuk memperbaharui data pada sebuah tabel. Perintah *DELETE* digunakan untuk menghapus data dari sebuah tabel.

2.7.4.1.1.3 DCL (*Data Control Language*)

DCL adalah sub bahasa SQL yang berfungsi untuk melakukan pengontrolan data dan server databasenya, seperti manipulasi user dan hak akses (*priviledges*). Yang termasuk perintah dalam DCL ada dua, yaitu *GRANT* dan *REVOKE*. Perintah *GRANT* digunakan untuk memberikan hak akses oleh admin ke salah satu user atau pengguna. Hak akses tersebut bisa berupa hak membuat (*CREATE*), mengambil data (*SELECT*), menghapus data (*DELETE*), mengubah data (*UPDATE*), dan hak

khusus lainnya yang berhubungan dengan sistem database. Sedangkan perintah *REVOKE* digunakan untuk mencabut hak akses yang telah diberikan kepada user.

2.7.5 JavaScript

JavaScript adalah bahasa pemrograman yang bisa disisipkan ke HTML seperti halnya PHP akan tetapi javascript berjalan di sisi client. Javascript adalah bahasa scripting yang populer di internet dan dapat bekerja di sebagian besar browser populer seperti Internet Explorer (IE), Mozilla Firefox, Netscape dan Opera. Kode *Javascript* dapat disisipkan dalam halaman web menggunakan tag *script* (Sunyoto, 2007). JavaScript pada awal perkembangannya berfungsi untuk membuat interaksi antara user dengan situs web menjadi lebih cepat tanpa harus menunggu pemrosesan di *web server*. Sebelum *javascript*, setiap interaksi dari user harus diproses oleh *web server*. Dalam perkembangannya, JavaScript mengalami permasalahan yang sama seperti kode pemrograman web yang bersifat *client side* seperti CSS, yakni bergantung kepada implementasi *web browser*.

Kode JavaScript yang kita buat, bisa saja tidak bekerja di Internet Explorer, karena web browser tersebut tidak mendukungnya. Sehingga programmer harus bekerja extra untuk membuat kode program agar bisa “mengakali” dukungan dari *web browser*. Karena hal tersebut, JavaScript pada awalnya termasuk bahasa pemrograman yang rumit, karena harus membuat beberapa kode program untuk berbagai *web browser*. Namun, beberapa tahun belakangan ini, JavaScript kembali bersinar berkat kemudahan yang ditawarkan oleh komunitas programmer yang membuat library JavaScript seperti *jQuery*. *Library* ini memudahkan kita membuat program JavaScript untuk semua web browser, dan membuat fitur-fitur canggih yang sebelumnya membutuhkan ribuan baris kode program menjadi sederhana.

2.8 MySQL Database

MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem database (DBMS) dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya. Sebagai database server, MySQL dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query data*. Hal ini terbukti untuk query yang dilakukan oleh single user, kecepatan query MySQL bisa sepuluh kali lebih cepat dari *PostgreSQL* dan lima kali lebih cepat dibandingkan *Interbase* (Ratna, 2014).

MySQL memiliki beberapa keistimewaan yaitu portabilitas dalam arti MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux,

FreeBSD, Mac Os X Server, Solaris, Amiga dan masih banyak lagi. Lalu MySQL didistribusikan secara *open source*, dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma. MySQL juga bersifat *multiuser* yang berarti dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik. Keistimewaan selanjutnya yaitu *performance tuning* yaitu memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu. Selain keistimewaan yang disebutkan diatas MySQL memiliki tipe kolom yang sangat kompleks, seperti *signed / unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain (Ratna, 2014).

2.9 Model-View-Controller (MVC)

Pola MVC atau *Model-View-Controller* merupakan suatu metode untuk membuat suatu perangkat lunak dengan cara memisahkan data, tampilan dan kontrol. Sama hal dengan namanya pola ini dibagi menjadi 3 bagian komponen yaitu *model, view* dan *controller*. Model merupakan bagian yang berhubungan dengan data yang tersimpan di database. Tidak hanya dapat melakukan *select* data, namun pada model juga dapat memanipulasi data dengan melakukan operasi *insert, update, delete* dan *search*. Model tidak dapat langsung berhubungan dengan *view*. Jika ingin berhubungan dengan *view* maka *controller*lah sebagai perantaranya.

Pada bagian *view* berfungsi untuk menampilkan atau mempresentasikan data dan informasi. Biasanya file *view* berupa file template HTML yang terkadang didalamnya terdapat penggalan kode PHP. Terakhir merupakan komponen *controller* yang tugasnya mengatur hubungan antar *model* dan *view*. *Controller* juga disebut sebagai otak logika yang dapat menjalankan permintaan dan data dari pengguna lalu yang menentukan apa yang akan diproses. Secara singkatnya, *model* untuk mengatur alur database, *view* bertugas menampilkan web serta *controller* mengatur alur kerja antara *model* dan *view*.

Kelebihan dari pola MVC yaitu kode program dapat secara otomatis mengikuti pada struktur file yang ada pada framework tersebut sehingga memudahkan dalam melakukan manajemen kode program. Jika ada kelebihan tentu ada kekurangan, kekurangan dari pola ini yaitu adanya peningkatan kompleksitas sehingga pada aplikasi kecil yang tidak membutuhkan loose coupling pada Model yang menjadi blok penghalang dalam pola MVC ini sendiri.

2.10 CodeIgniter Framework

CodeIgniter merupakan salah satu bingkai kerja atau lebih dikenal dengan framework yang mengusung prinsip MVC (*Model-View-Controller*). *Framework* ini banyak digunakan oleh para programmer dan perusahaan sebab dengan menggunakan konsep MVC CodeIgniter dapat dikembangkan sesuai dengan kemampuan developer sebagai contoh model dan controller ditangani oleh programmer sedangkan untuk bagian *view* ditangani oleh designer. Hal tersebut

membuat peningkatan maintainability dan organisasi kode. CodeIgniter juga tergolong *framework* yang *open source*.

CodeIgniter atau biasa disebut dengan CI sangat cocok untuk pengembangan aplikasi berbasis website yang menggunakan bahasa pemrograman PHP (Hypertext Preprocessor) sebab CI memiliki fitur yang mumpuni. CI dapat berjalan dengan menggunakan PHP4 dan PHP5. Dari segi ukuran CI tidak memakan begitu besar hanya sekitar 2.2 MB dan sudah termasuk panduan ebook yang lengkap. Kelebihan lain dari CI yaitu dapat diterapkan di berbagai hosting standar sebab yang dibutuhkan oleh CI adalah hosting yang menyediakan 5.1.6. Selain itu CI mampu untuk melakukan pengolahan database dan juga memberikan kebebasan kepada para penggunanya dalam membuat kode dan penamaannya sendiri (Griffiths, 2010).

CI juga memiliki beberapa kelebihan dibanding *framework* PHP lain, antara lain performa sangat cepat berbeda saat tidak menggunakan *framework* maka eksekusinya akan lebih lambat dan bahkan CI memungkinkan menjadi *framework* yang paling cepat dibanding yang lain. Konfigurasi yang sangat minim (nearly zero configuration) hanya merubah beberapa file saja pada folder config. Memiliki banyak komunitas CI yang memudahkan dalam berinteraksi dengan yang pengguna lain. Dokumentasi yang sangat lengkap, karena pada saat melakukan instalasi sudah langsung terdapat *user guide* yang lengkap untuk dijadikan petunjuk awal dan mudah dipahami. CI juga telah mengaplikasikan *Active Record* yang dapat mempermudah user untuk menulis sintaks SQL sehingga aplikasi dapat berjalan sesuai dengan keinginan user. *Active Record* mengizinkan pengguna CI untuk melakukan penulisan sintaks SQL dalam bahasa PHP (Griffiths, 2010).

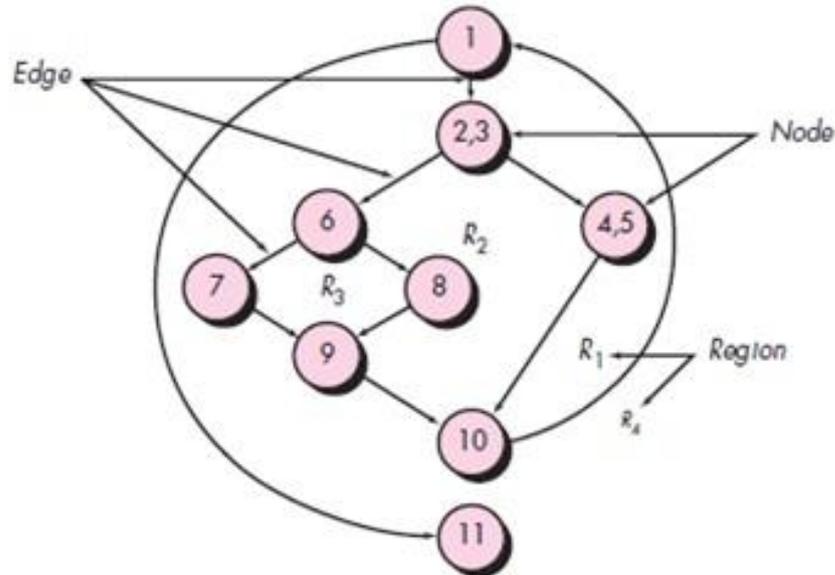
2.11 Pengujian Perangkat Lunak

Pengujian perangkat lunak biasanya menjadi tahap terakhir yang dilakukan dalam pengembangan perangkat lunak. Pengujian dilakukan untuk mendeteksi kesalahan pada program menyebabkan pengujian menjadi sesuatu hal yang penting yang harus dilakukan agar dapat mempertahankan kesenangan *customer*, meningkatkan kualitas dan mengurangi biaya dalam perbaikan yang akan datang jika tidak diuji. Pada penelitian ini akan menggunakan *whitebox testing* dan *blackbox testing* yang akan dijelaskan pada sub-bab ini.

2.11.1 White-box Testing

Pengujian kotak putih atau biasa disebut dengan *white-box testing* merupakan sebuah perancangan *test case* menggunakan struktur kontrol sebagai bagian dari perancangan tingkat komponen untuk menghasilkan *test case*. *Test case* yang diperoleh jika menggunakan pengujian ini antara lain menjamin semua jalur independen dalam modul sudah dieksekusi sedikitnya sekali dan melaksanakan semua *loop* dalam batas-batas operasional sistem. *Basis path testing* merupakan salah satu metode dari *white-box testing* yang memungkinkan perancang *test case* untuk menurunkan ukuran kompleksitas logis dari suatu rancangan dan menggunakan ukuran ini sebagai dasar dalam menentukan rangkaian dasar jalur

eksekusi. Dimulai dengan menggambarkan grafik alir (*flow graph*), *flow graph* memungkinkan perancang untuk menelusuri jalur program dengan cara yang lebih baik (Pressman, 2012).



Gambar 0.1 Grafik Alir/ Flow graph

Sumber: (Pressman, 2012)

Mengacu pada gambar 2.1 merupakan contoh *flow graph*, pada bentuk lingkaran disebut *node* sedangkan panah disebut *edge*. Sebuah *edge* harus berhenti pada sebuah *node*. Area yang dibatasi oleh *node* dan *edge* disebut *region*. Pada saat ingin menghitung *region*, maka juga harus memasukkan area luar grafik sebagai *region*. Tahap setelah menggambar *flow graph* adalah menentukan jalur yang biasa disebut *independent path* yaitu jalur pada program yang menunjukkan paling sedikit satu proses atau kondisi baru (Pressman, 2012). Diambil dari gambar 2.1 *independent path* yang terbentuk antara lain:

Jalur 1: 1-11

Jalur 2: 1-2-3-4-5-10-1-11

Jalur 3: 1-2-3-6-8-9-10-1-11

Jalur 4: 1-2-3-6-7-9-10-1-11

Setelah mendapatkan *independent path* maka langkah selanjutnya adalah melakukan perhitungan kompleksitas siklomatik (*cyclomatic complexity*). *Cyclomatic complexity* adalah rumusan perhitungan yang menyediakan ukuran kompleksitas dari suatu program, semakin banyak *independent path* dalam modul maka kompleksitas ujinya semakin tinggi. Kompleksitas dihitung dengan tiga cara berikut (Pressman, 2012):

$$V(G) = E - N + 2$$

$$V(G) = P + 1$$

$V(G)$ = Jumlah Region

Keterangan:

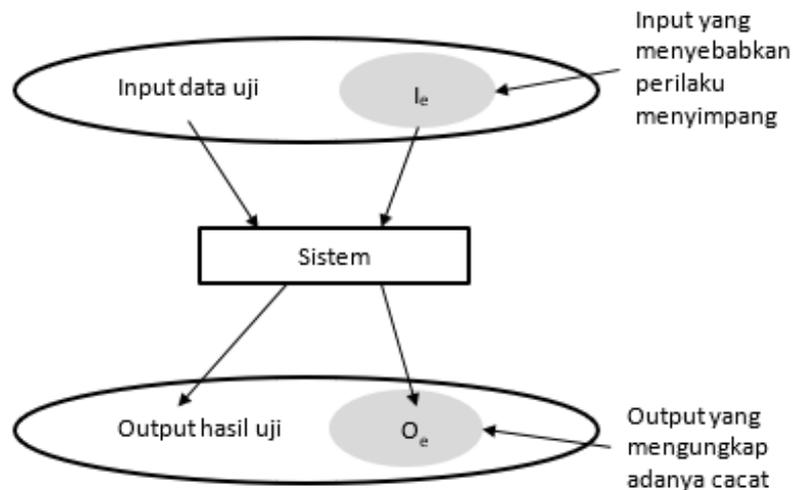
E = Jumlah *edge* pada *flow graph*

N = Jumlah *node* pada *flow graph*

P = Jumlah *node* predikat pada *flow graph*

2.11.2 Black-box Testing

Pengujian fungsional atau pengujian kotak hitam (*black-box testing*) adalah suatu pendekatan pengujian yang ujinya diturunkan dari spesifikasi program atau komponen. Sistem disebut dengan 'kotak hitam' karena perilakunya hanya dapat ditentukan dari mempelajari masukan (*input*) dan keluaran (*output*) yang berkaitan. Nama lain dari *blackbox testing* yaitu pengujian fungsional, disebut pengujian fungsional sebab penguji hanya mengacu pada fungsionalitasnya, bukan implementasi sistemnya (Sommerville, 2003). Pada gambar 2.2 mengilustrasikan model sistem yang diasumsikan oleh *blackbox testing*. Penguji memberikan input kepada sistem dan meneliti output yang dihasilkan. Jika output yang dihasilkan tidak sesuai dengan yang diramalkan maka uji tersebut telah berhasil mendeteksi masalah dengan sistem tersebut.



Gambar 0.2 Blackbox testing

Sumber: (Sommerville, 2003)