

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. Tinjauan Pustaka

Dalam pembuatan sistem informasi, tinjauan pustaka dijadikan sebagai kajian dan acuan pada penelitian ini. Dengan dijadikannya sebagai tinjauan pustaka pada penelitian ini terdapat bahan pertimbangan yang sesuai dengan topik pembahasan yang akan dicantumkan dalam beberapa hasil penelitian-penelitian terdahulu yang pernah penulis baca diantaranya, yaitu:

Pada tinjauan pustaka pertama dalam penelitian yang berjudul “Sistem Informasi Klinik Berbasis Web Pada Klinik Umum dan Kecantikan Dokter Galuh Dwi Anandhita Jakarta”. Metode penelitian yang digunakan dalam sistem adalah teknik pengumpulan data meliputi observasi, wawancara, studi pustaka dan model pengembangan sistem meliputi analisa kebutuhan sistem, desain, *code generation*, *testing*, implementasi. Kesimpulan yang dihasilkan dari pembuatan sistem informasi klinik berbasis web pada klinik diharapkan dapat mempermudah proses *input* data, pencarian data, dan pembuatan laporan, serta memungkinkan layanan yang diberikan pada pasien akan menjadi lebih cepat dan akurat (Monalisa & Mahendra, 2017).

Pada tinjauan pustaka kedua dalam penelitian yang berjudul “Sistem Informasi Pelayanan Kesehatan Pada Klinik Kenten Medika Palembang Berbasis WEB”. Metode penelitian yang digunakan dalam jurnal adalah studi literatur, pengumpulan data, analisis sistem, pengembangan sistem, penarik kesimpulan, penyusunan laporan. Kesimpulan yang dihasilkan dari pembuatan sistem informasi pelayanan kesehatan pada klinik berbasis web yang menggunakan bahasa pemrograman PHP, dan HTML, serta MySQL sebagai media penyimpanan data atau *database*-nya, adalah sistem dapat membantu Klinik Kenten Medika untuk

mengelola data pendaftaran, pendataan dokter, rekam medis pasien, pemberian obat kepada pasien dan sistem ini juga dapat membantu pimpinan dalam mendapatkan laporan (Candra, Saputra, & Novita, 2018).

Pada tinjauan pustaka ketiga dalam penelitian yang berjudul “Perancangan Sistem Informasi Rekam Medis Pasien Rawat Jalan Berbasis Web Di Klinik Sehat Margasari Bandung”. Metode penelitian yang digunakan dalam sistem informasi adalah metode RUP dengan proses-proses seperti pemodelan bisnis, pengumpulan kebutuhan, analisis dan desain, implementasi, dan pengujian. Kesimpulan yang dihasilkan dari pembuatan sistem informasi rekam medis pasien rawat jalan berbasis web diharapkan terbangunnya sistem informasi rekam medis berbasis web untuk memudahkan Klinik Sehat Margasari dalam membantu pengolahan data pasien, obat, transaksi, rekam medis, tindakan medis pasien hingga pencetakan laporan sehingga dapat menyajikan informasi yang akurat dan efisien (Pasaribu & Sihombing, 2017).

Pada tinjauan pustaka keempat dalam penelitian yang berjudul “Rancang Bangun Sistem Informasi Klinik Praktik Dokter Berbasis Web”. Metode penelitian yang digunakan adalah metode pengumpulan data meliputi observasi, wawancara dan metode perancangan sistem meliputi analisis kebutuhan sistem, fase perancangan, perancangan basis data, perancangan antarmuka. Kesimpulan yang dihasilkan dari rancang bangun sistem informasi klinik berbasis web yang menggunakan cara manual untuk mencatat seluruh data kesehatan pasien, sehingga data kesehatan pasien sulit dikontrol mengakibatkan *human error* dan tidak dapat memberikan informasi yang akurat. Dari sistem ini diharapkan dapat meningkatkan efektifitas dan efisiensi kerja dalam mengelola data klinik secara cepat dan mudah dibandingkan dengan manual untuk menghemat tempat penyimpanan maupun pengelolaan seperti menambah data, mengubah data, dan menghapus data. Selain itu, sistem ini dibuat sebagai sarana informasi dalam menyajikan informasi mengenai data pasien, data rekam medis, data dokter, data obat dan data tindakan. Serta sistem ini memiliki fitur hak akses sehingga pihak-pihak tertentu yang dapat melihat data yang ada pada sistem sehingga kerahasiaannya terjamin. Sistem

informasi yang dirancang dengan menggunakan pemrograman PHP dan MySQL sebagai *database* (Permana, 2015).

Pada tinjauan pustaka kelima dalam penelitian yang berjudul “Aplikasi Penjualan Produk Kecantikan dan Konsultasi Berbasis Web di Klinik Vania Skincare”. Metode penelitian yang digunakan dari jurnal ini adalah metode *Software Development Life Cycle Waterfall* (SDLC) meliputi analisa kebutuhan, desain sistem, penulisan kode program, pengujian program, penerapan program. Kesimpulan yang dihasilkan dari sistem ini diharapkan dapat membantu konsumen dalam melihat katalog dan memesan produk secara online sehingga konsumen tidak perlu datang ke klinik dan dapat dilakukan diluar jadwal buka klinik. Selain itu, sistem ini dapat membantu pegawai untuk mengelola data konsumen dalam hal mencari, menghapus, dan mengubah. Dalam aplikasi ini juga dapat membantu pegawai untuk mengelola laporan transaksi dalam hal melihat dan mencetak - laporan transaksi. Dengan demikian pegawai tidak perlu mengelola data konsumen, data produk, dan laporan transaksi secara manual sehingga meminimalisir kemungkinan adanya duplikasi pencatatan, laporan data hilang atau rusak. Serta dalam sistem ini dapat membantu konsumen dan dokter dalam hal tanya-jawab konsultasi secara online sehingga proses konsultasi tidak perlu dilakukan di klinik dan dapat dilakukan diluar jadwal klinik (Febrianto, Budiwati, & Tambunan, 2017).

Pada tinjauan pustaka keenam dalam penelitian yang berjudul “Sistem Informasi Layanan Kesehatan Dengan Menggunakan *Codeigniter* Pada Puskesmas Bululawang”. Metode penelitian yang digunakan dalam metode pengembangan sistem ini adalah metode *prototype* meliputi analisa kebutuhan sistem, desain sistem, pengujian sistem, dan implementasi. Kesimpulan yang dihasilkan dari sistem ini diharapkan dapat membantu dalam pencatatan dan akses data pasien, pengolahan data rekam medis dan akses riwayat rekam medis pasien dapat diakses dengan mudah oleh petugas kesehatan dalam pembuatan laporan puskesmas yang lebih cepat, dapat meningkatkan pelayanan dari segi waktu yang lebih cepat dengan kenaikan yang signifikan. Selain itu, sistem informasi puskesmas dapat meningkatkan kualitas pelayanan kesehatan masyarakat oleh petugas kesehatan

kepada pasien. Pembuatan sistem informasi layanan kesehatan ini dengan menggunakan *codeigniter* dengan berorientasi objek. Adapun pengujian sistem menggunakan metode *blackbox* dengan menguji fungsionalitas sistem, sedangkan variabel pengujian lain dengan menggunakan waktu proses untuk menilai pengukuran metode kerja sistem (Yuniar & Muslim, 2018).

Pada tinjauan pustaka ketujuh dalam penelitian yang berjudul “Rancang Bangun Sistem Informasi Pengarsipan Data Pasien Klinik Cemara”. Metode penelitian yang digunakan dalam sistem ini adalah metode eksperimen yang memusatkan pengamatan pada variable dalam jumlah kecil meliputi identifikasi masalah, menentukan tujuan dan batasan, studi literatur, implementasi, pengujian, evaluasi hasil uji. Kesimpulan yang dihasilkan dari penelitian ini diharapkan sistem informasi pengarsipan data pasien yang berdasarkan hasil implementasi dan hasil evaluasi menunjukkan bahwa aplikasi ini bermanfaat dan membantu dalam pengolahan data pasien di Klinik Cemara. Pada perangkat lunak yang dirancang mempunyai fitur penambahan data diri pasien baru, menyimpan data diri pasien, menghapus data pasien yang sudah lama tidak berobat, serta meng-*update* data diri pasien, jika sewaktu mendaftar ada ketentuan yang belum terpenuhi. Analisis kebutuhan sistem menggunakan analisis berorientasi objek dengan menggunakan *use case* diagram untuk menggambarkan fungsionalitas sistem, dan diagram kelas untuk menggambarkan hubungan antar kelas. Pengujian dilingkungan pengguna menggunakan empat parameter yakni kebermanfaatan aplikasi, kemudahan dipelajari, kemudahan penggunaan, dan kepuasan pengguna (Subhiyakto, et al., 2017).

Pada tinjauan pustaka kedelapan dalam penelitian yang berjudul “Pembuatan Sistem Informasi Manajemen Klinik Dengan Rekam Medis: Studi Kasus Di Klinik Kebon Arum Boyolali”. Metode penelitian yang digunakan dalam sistem informasi ini adalah analisis kebutuhan, desain sistem, tahap *programming*, implementasi dan evaluasi. Kesimpulan yang dihasilkan dari pembuatan sistem informasi manajemen klinik bahwa sistem berbasis web yang dirancang dengan *framework codeigniter*, bahasa pemrograman PHP untuk menunjukkan kinerja

yang maksimal. Basis data MySQL mampu mengolah data dengan baik sesuai yang diharapkan. Aplikasi ini berjalan dengan baik dan dapat memenuhi fungsi-fungsi penting dalam manajemen klinik seperti manajemen obat, rekam medis, pemeriksaan, tarif dan lain-lain. Sistem ini siap untuk diimplementasikan di Klinik Kebon Arum Boyolali (Triaji, Kridalukmana, & Widiyanto, 2017).

Pada tinjauan pustaka kesembilan dalam penelitian yang berjudul “Pengembangan Sistem Informasi Data Kesehatan Kabupaten Banyuasin Berbasis *Model View Controller*”. Metode penelitian yang digunakan dalam sistem informasi adalah metode *waterfall* yang terdiri dari *requirements definitions, system and software design, implementation and unit testing, integration and system testing*, dan *operation and maintenance*. Kesimpulan yang dihasilkan dari pengembangan sistem informasi puskesmas diharapkan dapat membantu Dinas Kesehatan Kabupaten Banyuasin dalam pembuatan laporan data kesehatan setiap tahunnya, validitas data yang diperoleh dapat dipertanggungjawabkan kebenarannya, metode *waterfall* dapat digunakan dalam pengembangan sistem informasi data kesehatan. Adapun arsitektur *model view controller* dipilih karena arsitektur tersebut memiliki kelebihan seperti memudahkan sistem pengembangan dan memiliki modularitas yang tinggi (Fajri, 2017).

Pada tinjauan pustaka kesepuluh dalam penelitian yang berjudul “Sistem Informasi Penjadwalan Dokter Berbasis Web Dengan Menggunakan *Framework Codeigniter* (Studi Kasus: Rumah Sakit Yukum Medical Centre)”. Metode penelitian yang digunakan dalam penelitian ini adalah kerangka penelitian, tahapan penelitian, metode pengumpulan data yang terdiri dari observasi (*observation*), wawancara (*interview*), tinjauan pustaka (*literature review*), dan dokumentasi (*documentation*), kerangka pengujian. Kesimpulan yang dihasilkan dari penelitian ini diharapkan dapat mempermudah bagian informasi dan pasien. Terutama admin dan dokter dapat menginputkan jadwal praktek dokter ter-*update*. Sistem ini diuji dengan menggunakan 3 metode pengujian, yaitu *whitebox testing, software tester*, dan *blackbox testing* (ISO 9126) dengan melakukan pengujian kuisisioner kepada bagian informasi, dokter, dan pasien. Dalam pembuatan aplikasi sistem informasi

penjadwalan dokter berbasis web dengan menggunakan *database MySQL* dan *tools Dreamweaver* dengan menggunakan *framework codeigniter* dan bahasa pemrograman PHP. Sistem ini juga dirancang dengan alat pengembangan sistem UML (*Unified Modelling Language*) yang terdiri dari *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram* serta menggunakan *entity relationship diagram* dan relasi antar tabel. Selain itu, sistem informasi penjadwalan dokter ini melakukan analisis dengan pengumpulan data seperti wawancara dan observasi. Dalam pengembangan sistem menggunakan *waterfall*. Analisis PIECES digunakan untuk melakukan perbandingan antara sistem yang lama dengan sistem baru yang akan dibuat serta menentukan analisis kebutuhan *fungsiional*, kebutuhan *non fungsiional*, dan analisis kelayakan untuk mengidentifikasi analisis-analisis sistem (Destiningrum & Adrian, 2017).

Ditinjau dari beberapa penelitian yang sebelumnya sudah ada dilakukan, bahasa pemrograman PHP dan *database MySQL* telah banyak diterapkan dalam membangun sistem berbasis *website*. Dengan begitu, pada penelitian ini akan dibangun sistem konsultasi dan rekapitulasi data pasien berbasis *web* menggunakan *framework codeigniter* pada klinik sejahtera. Program ini dibangun bertujuan untuk memberikan kemudahan bagi setiap pasien dalam melakukan proses konsultasi karena pasien tidak perlu harus datang ke klinik bila hanya ingin bertanya-tanya, dan untuk memudahkan staff klinik atau admin dalam proses pengolahan data.

Dalam pengembangan sistem *website* ini dengan menggunakan metode *waterfall*. Program ini dibangunnya sistem *website* yang menggunakan sistem *template bootstrap*. *Bootstrap* adalah sebuah *library framework CSS* yang dibuat khusus untuk bagian pengembangan *front-end website*. *Bootstrap* merupakan salah satu *framework HTML, CSS* dan *Javascript* yang populer dibagian *web developer* untuk *website* yang *responsive*. *Bootstrap* juga memiliki kelebihan pada tampilan yang lebih menarik dan optimal serta menyesuaikan sesuai dengan ukuran layar monitor *device* dan dapat diakses melalui segala perangkat dengan memiliki kecepatan *loading* yang lebih cepat karena *bootstrap* dibuat dengan sangat terstruktur. Pada pembuatan *website* ini untuk memudahkan *developer* dan *designer*

dengan menggunakan Bahasa pemrograman PHP, *Javascript*, *CSS*, dan dalam pengolahan *database* menggunakan *MySQL*.

Keunggulan *website* sistem konsultasi dan rekapitulasi data pasien berbasis web menggunakan *framework codeigniter* pada klinik sejahtera adalah memiliki fitur rekam medis data pasien (*medical record*), fitur print yang berfungsi untuk mencetak data dalam bentuk PDF, dan fitur konsultasi melalui via *whatsapp* untuk memudahkan staff layanan klinik dalam melakukan proses pengolahan data pasien, mempermudah laporan dan pencatatan dalam dokumentasi hasil konsultasi pasien agar datanya lebih akurat dan detail, membantu meningkatkan kinerja staff layanan klinik dalam memantau pasien yang berkonsultasi, dan mempermudah pasien dalam melakukan proses konsultasi dengan dokter.

2.2. Landasan Teori

2.2.1. Fasilitas Pelayanan Kesehatan

Berdasarkan sesuai dengan Undang-Undang Republik Indonesia No. 36 Tahun 2009 tentang Kesehatan, yang dimaksud dengan fasilitas pelayanan kesehatan merupakan suatu alat dan atau tempat yang digunakan untuk menyelenggarakan upaya pelayanan kesehatan baik berupa upaya promotif, preventif, kuratif, maupun rehabilitatif yang dilakukan oleh pemerintah, pemerintah daerah, dan masyarakat. Berdasarkan jenis pelayanan yang diberikan, fasilitas pelayanan kesehatan tersebut dikelompokkan menjadi:

- 1) Pelayanan kesehatan tingkat pertama yakni pelayanan kesehatan yang diberikan oleh fasilitas kesehatan dasar.
- 2) Pelayanan kesehatan tingkat kedua yakni pelayanan kesehatan yang diberikan oleh fasilitas kesehatan spesialis.
- 3) Pelayanan kesehatan tingkat ketiga yakni pelayanan kesehatan yang diberikan oleh fasilitas kesehatan subspecialistik.

Pelayanan kesehatan adalah sebuah konsep yang digunakan dalam memberikan layanan kesehatan kepada masyarakat. Definisi pelayanan kesehatan

menurut Prof. Dr. Soekidjo Notoatmojo (2008) adalah sebuah sub sistem pelayanan kesehatan yang tujuan utamanya adalah pelayanan preventif (pencegahan) dan promotif (peningkatan kesehatan) dengan sasaran masyarakat.

Menurut Levey dan Loomba (1973) menyatakan bahwa, pelayanan kesehatan adalah upaya yang diselenggarakan sendiri atau secara bersama-sama dalam suatu organisasi untuk memelihara dan meningkatkan kesehatan dan mencegah dan menyembuhkan penyakit serta memulihkan kesehatan perorangan, keluarga, kelompok, ataupun masyarakat.

Definisi kesehatan menurut Undang-Undang no. 36/2009 tentang Kesehatan terdiri dari dua unsur yaitu “upaya kesehatan” dan “sumber daya kesehatan”. Sumber daya kesehatan yang dimaksud, terdiri dari sumber daya manusia kesehatan (tenaga kesehatan yaitu dokter, apoteker, bidan, perawat) dan sarana kesehatan (antara lain rumah sakit, puskesmas, poliklinik, tempat praktik dokter).

2.2.2. Pengertian Klinik

Menurut Kamus Besar Bahasa Indonesia (KBBI) klinik adalah suatu lembaga kesehatan atau balai pengobatan khusus atau organisasi kesehatan yang bergerak dalam penyediaan pelayanan kesehatan kuratif (diagnosis dan pengobatan), biasanya terhadap satu macam gangguan kesehatan yang dijadikan tempat orang berobat dan memperoleh advis medis serta tempat mahasiswa kedokteran melakukan pengamatan terhadap kasus penyakit yang diderita pada pasien (Pusat Bahasa, 2008).

Berdasarkan Peraturan Menteri Kesehatan Republik Indonesia Nomor 028/Menkes/Per/2011, pengertian klinik adalah fasilitas pelayanan kesehatan yang menyelenggarakan pelayanan kesehatan perorangan yang menyediakan pelayanan medis dasar dan spesialisik, diselenggarakan oleh lebih dari satu jenis tenaga kesehatan dan dipimpin oleh seorang tenaga medis (Menkes RI, 2011).

Dengan demikian, sebuah klinik harus menentukan pelayanan yang akan disediakan, karena bias terbatas pada pelayanan media dasar atau pelayanan

spesialistik atau keduanya. Keputusan ini akan dapat mempengaruhi strata klinik yang diselenggarakan dalam pelayanan.

Menurut Peraturan Menteri Republik Indonesia Nomor 028/Menkes/Per/2011, klinik berdasarkan pelayanannya dibagi menjadi 2, yaitu:

1. Klinik Pratama

Strata klinik yang terbatas pada klinik yang melayani pelayanan medik dasar.

2. Klinik Utama

Strata klinik yang dapat menyelenggarakan pelayanan medik spesialistik atau pelayanan media dasar dan spesialistik.

Penyelenggaraan klinik harus memperhatikan beberapa persyaratan meliputi:

1. Syarat Lokasi

Lokasi yang harus disesuaikan dengan tata ruang daerah

2. Syarat Bangunan dan Ruangan

Bangunan dan ruangan yang permanen, yang harus memenuhi persyaratan lingkungan sehat dengan memperhatikan fungsi-fungsi keamanan, kenyamanan dan kemudahan dalam pemberian layanan, perlindungan dan keselamatan bagi semua orang termasuk penyandang cacat, anak-anak, orang usia lanjut. Bangunan minimal harus memiliki: (a) ruang pendaftaran/ruang tunggu, (b) ruang konsultasi dokter, (c) ruang administrasi, (d) ruang tindakan, (e) ruang farmasi, (f) kamar mandi, dan (g) ruang lainnya sesuai dengan kebutuhan pelayanan.

3. Sarana dan Prasarana

Sarana peralatan medis dan nonmedis yang memadai dan memenuhi standar mutu, keamanan, dan keselamatan.

Prasarana yang terpelihara dan berfungsi dengan baik, yaitu: instalasi air, listrik, sirkulasi udara, sarana pengolahan limbah, pencegahan dan penanggulangan kebakaran, ambulans dll.

4. Peralatan Ketenagakerjaan

Peralatan ketenagakerjaan yang terdiri atas tenaga medis, tenaga kesehatan lain dan tenaga non kesehatan. Tenaga medis dalam menjalankan kegiatan prakteknya harus memiliki Surat Tanda Registrasi (STR) dan Surat Izin Praktik (SIP) sesuai fasilitas pelayanan kesehatan tempatnya berpraktik. Demikian pula halnya untuk tenaga kesehatan lain diwajibkan mempunyai surat izin sebagai tanda registrasi (STR) dan Surat Izin Kerja (SIK) atau Surat Izin Praktik Apoteker (SIPA).

2.2.3. Pengertian Konsultasi

Konsultasi menurut KBBI adalah pertukaran pikiran untuk mendapatkan kesimpulan (nasihat, saran dan sebagainya) yang sebaik-baiknya.

Layanan konsultasi merupakan proses dalam suasana kerja sama dan hubungan antar pribadi dengan tujuan memecahkan suatu masalah dalam lingkup professional dari orang yang meminta konsultasi. Dalam konsultasi ada tiga unsur, yaitu klien, orang yang minta konsultasi dan konsultan (Mu'awanah & Hidayah, 2009).

Menurut KBBI berdasarkan medis, konsultasi yaitu perundingan antara penerima layanan kesehatan yang bertujuan mencari penyebab timbulnya penyakit dan menentukan cara pengobatannya. Konsultasi merupakan suatu bentuk komunikasi pribadi secara langsung antara dokter dengan pasien. Tujuan dari konsultasi adalah menjalankan tindakan pencegahan untuk menghentikan berkembangnya berbagai macam penyakit bagi pasien yang memiliki faktor resiko.

Beberapa manfaat yang didapatkan oleh pasien dalam menjalani konsultasi umum, seperti:

1. Pasien dapat membangun hubungan dekat dengan tim spesialis kesehatan yang dapat membantu mengatasi masalah, baik saat ini maupun ke depannya.

2. Deteksi dini meningkatkan kemungkinan pasien dapat melawan penyakit dan memungkinkan tim spesialis kesehatan mengendalikan penyakit akut ataupun kronis.
3. Dokter dapat melakukan tindakan pencegahan dan program lainnya yang dapat mendorong pendekatan yang lebih proaktif untuk mencegah, mengobati dan menangani suatu penyakit.

Dalam menjalankan konsultasi terdapat beberapa tahapan proses konsultasi diantaranya meliputi:

1. *Provision*

Konsultan memberikan pelayanan langsung kepada *konsultee* yang tidak memiliki waktu ataupun keterampilan dalam menyelesaikan masalahnya. Disini konsultan memberikan solusi dan *konsultee* bebas menentukan cara menyelesaikan masalahnya.

2. *Prescription*

Konsultan memberikan nasehat dan tidak ikut turut dalam membantu proses penyelesaian masalah yang sedang dihadapi oleh *konsultee*.

3. *Mediation*

Konsultan berperan sebagai mediator dalam menyelesaikan masalah yang dihadapi oleh *konsultee*.

4. *Collaboration*

Konsultan bersama dengan *konsultee* menyelesaikan masalah yang dihadapi.

Layanan di internet dapat membantu bentuk konsultasi antara konsultan atau dokter dan konsultee atau pasien secara langsung. Layanan yang mendukung bentuk konsultasi antara lain: *electronic mail (e-mail)*, *mailing list*, dan *chatting*. Dalam sistem konsultasi online ini menggunakan pesan dan chatting untuk layanan konsultasi.

1. Pesan, merupakan layanan yang memungkinkan secara pribadi antara seorang dokter dan pasien berkomunikasi. Karena bersifat pribadi jadi tidak ada orang lain yang tahu apa yang dikomunikasikan, seperti sosial media yang sering digunakan *whatsapp* dan e-mail lebih cepat dan murah.
2. *Chatting*, layanan di internet untuk “ngobrol” atau bicara. Bentuk layanan ini cara berkomunikasi tidak bertatap muka tapi dengan *chatting* ini menjadi komunikasi yang interaktif. Antara *user chatting* dapat saling menanggapi, hal ini yang menjadi menarik. Dikalangan masyarakat, dunia internet yang menyediakan layanan *chatting* sudah banyak di alamat *website*.

2.2.4. Pengertian Internet

Menurut Edy (2007:2) internet diartikan sebagai jaringan komputer luas dan besar yang mendunia, yaitu dengan menghubungkan pengguna komputer dari satu negara ke negara lain diseluruh dunia yang didalamnya terdapat bermacam-macam sumber daya informasi dari mulai yang statis maupun dinamis dan bersifat interaktif.

Menurut Mulyanto (2009:113), internet atau International Network merupakan rangkaian jaringan terbesar di dunia dimana semua jaringan yang berada pada semua organisasi dihubungkan dengan suatu jaringan terbesar sehingga dapat saling berkomunikasi.

Internet dapat menghubungkan jutaan manusia di seluruh dunia, tanpa mereka mengetahui keberadaan lawan komunikasinya. Informasi dapat dikirim dengan berbagai bentuk, seperti suara, gambar, teks, data, maupun kombinasinya (Sutarman, 2009).

Sejarah jaringan komputer bermula dari lahirnya konsep jaringan computer pada tahun 1940-an di Amerika yang digagas oleh sebuah proyek pengembangan komputer MODEL I di laboratorium Bell dan group riset Universitas Harvard yang dipimpin professor Howard Aiken. Pada mulanya proyek tersebut hanyalah ingin memanfaatkan sebuah perangkat komputer yang harus dipakai bersama. Untuk

mengerjakan beberapa proses tanpa banyak membuang waktu kosong dibuatlah proses beruntun (*Batch Processing*), sehingga beberapa program bias dijalankan dalam sebuah komputer dengan kaidah antrian.

Di tahun 1950-an, ketika komputer mulai membesar sampai tercipta superkomputer, maka sebuah komputer harus melayani beberapa terminal. Oleh sebab itu, ditemukan konsep distribusi proses berdasarkan waktu yang dikenal dengan TSS (*Time Sharing System*), dan untuk pertama kali terbentuklah jaringan (*network*) komputer pada lapis aplikasi.

Pada sistem TSS beberapa terminal terhubung ke sebuah host komputer. Dalam proses TSS mulai nampak perpaduan teknologi komputer dan teknologi telekomunikasi yang pada awalnya berkembang sendiri-sendiri.

Pada tahun 1957 *Advanced Research Agency* (ARPA) dibentuk oleh *Departement of Defence* (DoD) USA. Internet mulai dibangun pada tahun 1969 pada saat Departemen Pertahanan Amerika, *United States (U.S.) Defense Advanced Research Project Agency* (DARPA) memutuskan untuk mengadakan riset tentang bagaimana caranya untuk menghubungkan sejumlah komputer sehingga membentuk jaringan organik. Program riset ini dikenal dengan nama ARPANET. Memasuki tahun 1970, dalam membentuk sebuah jaringan internet yang telah lebih dari 10 komputer yang berhasil dihubungkan satu sama lain, sehingga mereka dapat saling berkomunikasi.

Tahun 1972, Roy Tomlinson telah berhasil menyempurnakan riset pada program e-mail yang ia ciptakan setahun yang lalu untuk perkembangan ARPANET. Program e-mail mudah berkembang dan begitu populer di kalangan masyarakat. Ditahun 1972, icon “@” diperkenalkan sebagai lambang untuk menunjukkan “at” atau “pada”. Pada tahun 1973, jaringan komputer ARPANET mulai dikembangkan keluar Amerika Serikat. Komputer University College di London merupakan salah satu komputer pertama yang menjadi anggota jaringan ARPANET diluar Amerika. Vinton Cerf dan Bob Kahn yang merupakan ahli komputer menyatakan bahwa mempresentasikan sebuah gagasan yang lebih besar

tentang asal mula pemikiran internet yang dipresentasikan pertama kali di Universitas Sussex.

Pada peristiwa 26 Maret 1976, saat Ratu Inggris berhasil mengirimkan e-mail dari Royal Signals and Radar Establishment di Malvern. Tahun 1977, sudah lebih dari 100 komputer yang bergabung di ARPANET membentuk sebuah jaringan atau *network*. Para ahli yaitu Tom Truscott, Jim Ellis, dan Steve Bellovin menciptakan penemuan baru newsgroups pertama yang dikenal USENET di tahun 1979. France Telecom menciptakan gebrakan baru dengan meluncurkan telpon televisi pertama, yang mana orang bias saling berkomunikasi dengan menelpon yang terhubung pada video *link*.

Banyaknya peredaran komputer yang berkembang di masyarakat yang membentuk jaringan maka dibutuhkan protocol resmi yang diakui oleh semua jaringan. Pada tahun 1982 dibentuk *Transmission Control Protocol* (TCP) dan Internet Protocol atau IP yang terpopuler. Sementara itu, di benua Eropa muncul jaringan komputer yang dikenal sebagai EUNET, dengan menyediakan jaringan komputer di negara Belanda, Inggris, Denmark dan Swedia. Jaringan EUNET menyediakan jasa e-mail dan newsgroups USENET. (Proof reader b. indo)

Untuk menyelaraskan alamat di jaringan komputer maka pada tahun 1984 diperkenalkan sistem nama domain yang dikenal dengan *Domain Name System* (DNS). Perkembangan komputer yang terhubung dengan jaringan telah lebih dari 1000 komputer. Pada 1987, jumlah komputer yang tersambung dengan jaringan sudah meningkat 10 kali lipat menjadi 10.000 lebih.

Tahun 1988, Jarko Oikarinen dari Finland menciptakan penemuan baru yang diperkenalkan sebagai *Internet Relay Chat* (IRC). Setahun kemudian, jumlah komputer yang saling berhubungan meningkat 10 kali lipat dalam setahun. Lebih dari 100.000 jumlah komputer yang saling berhubungan membentuk jaringan. Pada tahun 1990, terdapat penemuan baru yang diperkenalkan oleh Tim Berners Lee yaitu program editor dan browser yang bias menjelajah antara satu komputer ke komputer lain yang membentuk jaringan disebut dengan *World Wide Web* (www).

Tahun 1992, jumlah komputer yang saling terhubung membentuk jaringan melampaui sejuta komputer, dan muncul istilah *surfing the internet*. Tahun 1994, perkembangan situs internet tumbuh menjadi 3000 alamat halaman, dan muncul untuk pertama kalinya *virtual-shopping* atau *e-retail* muncul di dunia internet. Yahoo! berdiri di tahun 1994 dan melahirkan *Netscape Navigator 1.0*.

2.2.5. Pengertian Sistem

Definisi dari sistem adalah serangkaian subsistem yang saling terkait dan bergantung pada satu sama lain, bekerja bersama-sama untuk mencapai tujuan dan sasaran yang sudah diterapkan sebelumnya. Sistem mempunyai *input*, proses, output, dan umpan balik (Kendall & Kendall, 2006).

Pengertian sistem dapat diartikan dalam 2 pendekatan, yaitu sistem yang menekankan pada prosedur dan sistem yang menekankan pada elemen komponen. Menurut Jogiyanto dalam bukunya yang berjudul “Analisa dan Desain Sistem Informasi (2005:8) yang termasuk dalam sistem yang menekankan pada prosedur menyatakan bahwa ”Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau penyelesaian suatu sasaran tertentu”. Sedangkan sistem yang menekankan pada elemen, yakni: “Sistem adalah suatu seri dari komponen-komponen yang saling berhubungan satu sama lain, dengan bekerja sama dalam suatu kerangka kerja pada tahapan yang terpadu untuk menyelesaikan, demi mencapai sasaran yang telah ditetapkan sebelumnya”.

Berdasarkan definisi yang telah di jelaskan diatas maka dapat ditarik sebuah kesimpulan bahwa sistem adalah suatu jaringan kerja yang saling berhubungan dan berkumpul bersama-sama untuk mencapai suatu jaringan tertentu.

2.2.6. Pengertian Informasi

Informasi dapat didefinisikan sebagai suatu hasil dari pengolahan data sehingga dapat menjadi bentuk yang penting bagi penerimaannya dan memiliki kegunaan sebagai dasar dalam pengambilan keputusan yang dapat dirasakan

akibatnya baik yang secara langsung saat itu juga ataupun secara yang tidak langsung pada saat mendatang (Sutanta, 2003).

Menurut Drs. Zulkifly Amsyah (2005) menyatakan bahwa informasi adalah suatu data yang telah dikelola, dibentuk, atau dimanipulasi sesuai dengan keperluan yang dibutuhkan tertentu. Pengertian dari data yaitu data yang telah ditulis dalam bentuk catatan atau direkam dalam bentuk media (contohnya: komputer).

2.2.7. Pengertian Website

Menurut Hidayat (2010:2), *website* atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam maupun bergerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dan masing-masing saling berhubungan dengan jaringan-jaringan halaman (*hyperlink*)”.

Website merupakan serangkaian kumpulan dari halaman-halaman yang berhubungan dengan file-file lain yang saling terkait. Dalam sebuah *website* terdapat satu halaman yang dikenal dengan sebutan *home-page*. *Homepage* adalah sebuah halaman yang pertama kali dilihat ketika pengguna mengunjungi sebuah *website* (Jhonsen, 2004).

Website dapat diartikan sebagai fasilitas sarana layanan internet yang menghubungkan dokumen dengan lingkup lokal maupun jarak jauh. Dokumen yang ada pada *website* dapat disebut dengan *web page* dan *link* dalam *website* yang dapat memudahkan pengguna dalam melakukan *browsing* dari satu *page* ke *page* lainnya (*hyper text*), baik halaman yang disimpan dalam *server* yang sama maupun *server* yang ada diseluruh dunia. Halaman yang diakses dan dibaca melalui *browser* seperti *Google Chrome*, *Netscape Navigator*, *Mozilla Firefox*, *Internet Explorer* dan aplikasi *browser* yang lainnya (Yuhefizar, 2008).

Berdasarkan uraian diatas, penulis menyimpulkan bahwa web adalah perangkat lunak yang digunakan untuk menampilkan dokumen-dokumen pada suatu halaman web yang membuat pengguna agar dapat mengakses dan membaca

internet melalui software browser seperti *Mozilla Firefox*, *Google Chrome*, *Internet Explorer* dll yang terkoneksi dengan internet dengan lingkup lokal maupun jarak jauh.

Yuliatmoko (2010) berpendapat bahwa secara garis besar, *website* bisa digolongkan menjadi tiga bagian adalah sebagai berikut:

1. *Website Statis* adalah web yang mempunyai halaman tidak berubah.
2. *Website Dinamis* adalah *website* yang memerlukan *update* sesering mungkin. Ccontoh *website dinamis* yaitu web berita atau web portal yang didalamnya terdapat fasilitas berita, *polling* dan sebagainya.
3. *Website Interaktif* adalah web yang saat ini memang sedang *booming user* bisa berinteraksi dan beradu *argument*. Salah satu contoh *website* interaktif yaitu blog dan forum.

2.2.8. Pengertian Sistem Informasi

Menurut Leitch dalam Mujiati dan Sukandi (2016:12) berpendapat bahwa sistem informasi adalah suatu sistem yang ada didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategis dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. Menurut Ladjamaddin Al-bahra-Bin (2005) menyatakan bahwa sistem informasi adalah suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi.

Berdasarkan uraian penjelasan diatas, penulis menyimpulkan bahwa sistem informasi adalah kumpulan komponen didalam suatu organisasi yang berfungsi sebagai pengolahan untuk menghasilkan sebuah laporan-laporan yang disajikan kepada pihak tertentu.

2.2.9. Metode MVC

Model View Controller (MVC) menurut Dudi Rahmadiansyah dan Dedy Irwan (2012:2) adalah sebuah metode dalam pembuatan aplikasi dengan memisahkan data dari tampilan *user interface* dan cara memprosesnya. MVC

pertama kali diperkenalkan oleh peneliti di XERO PARAC yang bekerja pada tahun 1970 – 1980 dalam pembuatan bahasa pemrograman *Smalltalk*. MVC memisahkan dalam pengembangan aplikasi berdasarkan komponen utama untuk membangun aplikasi diantaranya seperti manipulasi data, *user interface*, dan kontrol aplikasi.

Metode MVC dalam membangun aplikasi dibagi menjadi 3 komponen utama yaitu:

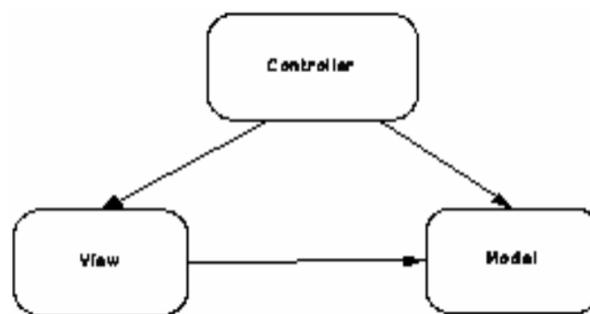
1. *Model*, berfungsi sebagai pengelola perilaku dan data pada domain aplikasi, dengan melakukan tanggapan terhadap permintaan informasi dan merespons instruksi untuk merubah suatu kondisi (*state*).
2. *View*, berfungsi untuk menerjemahkan informasi yang berasal dari model ke dalam sebuah bentuk yang sesuai untuk berinteraksi dengan *user*. Biasanya berupa satu atau lebih elemen antarmuka *user*.
3. *Controller*, menerima masukan dari *user* dan memicu *respons* dengan membuat pemanggilan ke objek-objek model.

Terdapat beberapa kelebihan dari metode MVC adalah sebagai berikut:

1. Bagian model memberikan penempatan detil data yang terpisah dan tidak disebar di dalam keseluruhan aplikasi sehingga meningkatkan kecepatan dan fleksibilitas dalam proses pemeliharaan aplikasi.
2. Pemisahan model juga membuat objek model dapat digunakan oleh aplikasi lain dengan kebutuhan yang sama (*reuse*).
3. Pemisahan view memudahkan perakitan atau integrasi dengan komponen aplikasi lainnya tanpa harus memperhatikan detail proses. Desainer hanya berkonsentrasi pada bentuk dan tampilan antarmuka pemakai (*user interface*).
4. Penggunaan *Controller* memungkinkan untuk perubahan proses tanpa harus mengganggu antarmuka pemakai. Detail proses disembunyikan oleh *Controller* sehingga tidak mengganggu presentasi ke *user* maupun pengelolaan data atau informasi (manajemen database).

Selain terdapat kelebihan seperti yang dijelaskan diatas, terdapat juga kekurangan yang dimiliki dari metode MVC, yaitu:

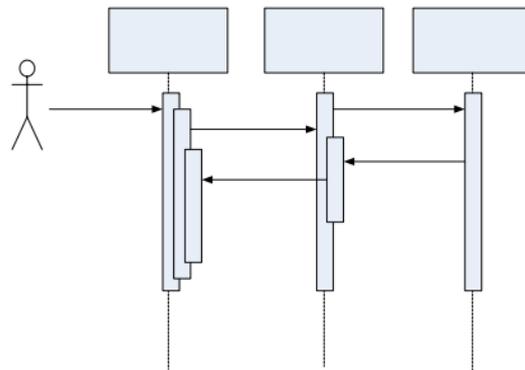
1. Peningkatan kompleksitas aplikasi karena arsitektur aplikasi yang terbagi menjadi tiga bagian.
2. Menimbulkan *loosely coupled* (Komponen aplikasi yang membuat atau mengacu pada komponen lainnya dengan sedikit atau tanpa informasi detail dari komponen yang diacu tersebut).



Gambar 2. 1 Konsep MVC

Beberapa jenis MVC yang dikenal ada di dalam aplikasi berbasis web, yaitu:

1. Server Side MVC yang mana seluruh proses bisnis dilakukan pada sisi server.
2. Campuran *Server Side* dan *Client Side* MVC, yang mana proses dilakukan di sisi *client* dan sisi server, bisa menggunakan atau tidak menggunakan model dalam koneksi ke server dan biasanya memiliki kompleksitas yang tinggi.
3. *Rich Internet Application* (RIA), disebut juga *Fat Client*, merupakan aplikasi web yang memiliki kemampuan dan fungsi yang mirip aplikasi desktop, memiliki bagian yang mengambil data sendiri (MVC tersendiri) dan hanya bagian model yang ada dibagian server.



Gambar 2. 2 Prinsip Kerja Metode MVC

2.2.10. *Framework Codeigniter*

Menurut Dudi Rahmadiansyah dan Dedy Irwan (2012:2) yang menjelaskan mengenai *framework* adalah abstraksi didalam sebuah perangkat lunak yang menyediakan fungsi yang *generic* sehingga dapat dirubah oleh kode yang dibuat *user* sehingga dapat menyediakan perangkat lunak untuk aplikasi tertentu. Metode MVC adalah sebuah arsitektur yang dapat diimplementasikan secara bebas dengan atau tanpa bahasa pemrograman berorientasi objek. Dengan demikian, metode MVC dapat diimplementasikan dalam sebuah *framework*. *Codeigniter* merupakan sebuah *framework* pemrograman web dengan menggunakan bahasa php. *Framework* ini ditulis dengan menggunakan bahasa php versi 4 dan versi 5 oleh Rick Ellislab yang menjadi CEO Ellislab, Inc. dan dipublikasikan dengan lisensi dibawah *Apache* atau *BSD Open Source*. Jadi *Codeigniter* adalah *framework php* dan *Open Source*.

Adapun beberapa manfaat kelebihan yang didapat dalam menggunakan *framework Codeigniter*, antara lain sebagai berikut:

1. Gratis, didalam *support Open Source* yang dapat digunakan dan dikembangkan secara bersama-sama. Yang dapat di-*download* secara gratis, bebas yang digunakan secara persyaratan yang sesuai dengan persetujuan lisensi (*licence agreemen*) yang dapat dilihat pada *website*.
2. Dapat ditulis dengan menggunakan bahasa PHP 4 (versi 1.x.x) dan versi 5 (versi 2.x.x) sehingga mendukung pemrograman dengan bahasa php.

3. Menggunakan metode MVC sebagai prinsip kerjanya sehingga dapat digunakan untuk mengembangkan aplikasi secara efisien dan dinamis serta lebih memudahkan dalam melakukan pemeliharaan aplikasi.
4. Menggunakan URL (*Uniform Resource Locator*) yang sederhana, bersih, dan SEF (*Search Engine Friendly*).
5. Mempunyai paket *library* yang lengkap, mendukung fungsi-fungsi *database, html, web, e-mail, session, pagination* dan lain sebagainya.
6. Terdapat dokumentasi yang lengkap dan jelas, serta dalam *website* resminya dan dapat di-*download* secara bersama-sama dengan *framework*-nya.
7. Di dalam komunitas, banyak pengguna dan pengembang yang mendukung *framework* ini, Waupun mulanya dikembangkan oleh Ellislab, Inc.
8. Bahasa pemrograman php yang mendukung bersifat portabel dan dapat di jalankan pada berbagai *platform*.

Selain kelebihan yang dijelaskan diatas, *framework* juga mempunyai kekurangan-kekurangan antara lain sebagai berikut:

1. Adanya toleran dalam penerapan aturan MVC, sehingga pemrograman masih diberikan peluang untuk dapat melanggar kaidah-kaidah pada MVC.
2. Pada konsep *Object Relational Model* (ORM) tidak didukung yang merupakan metode pengaksesan *database* yang menggunakan relasi antar objek maka pemrogram tidak butuh penulisan atau mengetahui sintaks bahasa SQL.
3. Meskipun dikembangkan oleh komunitas, jumlah pada pengembangnya tidak sebanyak *framework* php lainnya misalnya *CakePHP*, dimana masih dalam bawahan koordinasi Ellislab, Inc, sehingga *update core engine*-nya memakan waktu yang lebih lama daripada *framework Open Source* lainnya.

4. Sebagai *framework Open Source*, *Codeigniter* tidak menyediakan dukungan (*support*) secara khusus kecuali melalui forum pengguna.

Framework Codeigniter yang mengikuti metode MVC, memiliki tiga bagian utama, yaitu sebagai berikut:

1. *Model*

Pada bagian ini berisi kode yang berguna untuk koneksi dan mengakses database.

2. *View*

Berisi kode-kode HTML dan php yang digunakan untuk menampilkan informasi ke layar *browser*. Isi kode-kode yang berhubungan dengan format tampilan, contohnya huruf, *font*, warna dan lain sebagainya.

3. *Controller*

Dalam bagian ini, berisi kode-kode *script* yang menjalankan kegunaan aturan bisnis aplikasi dan menjadi perantara antara Model dan View serta seluruh sumber daya yang diperlukan untuk memproses permintaan layanan HTTP untuk menampilkan halaman web.

2.2.11. JavaScript

JavaScript mempunyai definisi, yaitu bahasa pemrograman *script* yang berdasarkan objek yang memperbolehkan pengguna untuk mengendalikan banyak aspek interaksi pengguna pada suatu dokumen HTML. Objek tersebut dapat seperti *window*, *frame*, *url*, *dokumen*, *form*, *button* dan lain sebagainya (Suryana & Koesheryatin, 2014)

2.2.12. PHP

Budi Raharjo (2015:3) berpendapat bahwa, PHP merupakan singkatan rekursif dari *Hypertext Preprocessor*, dimana bahasa pemrograman skrip yang paling sering digunakan untuk tujuan umum dalam mengembangkan aplikasi web, sama seperti bahasa pemrograman skrip lainnya, seperti: *C*, *C++*, *Pascal*, *Phyton*, *Perl*, *Ruby*, dan lain sebagainya. Selain itu, PHP juga dapat digunakan untuk mengembangkan aplikasi-aplikasi *Command Line Interface* (CLI) dan *desktop*

(GUI). Pada tahun 1994, PHP pertama kali diciptakan oleh Rasmus Lerdorf. Awalnya, PHP singkatan dari “*Personal Home Page Tools*” yang kemudian diganti dengan FI (*Forms Interpreter*). Sejak munculnya versi 3.0, nama bahasa pemrograman ini diganti menjadi PHP (*Hypertext Preprocessor*). PHP versi terbaru yaitu versi ke-5. Sejak kemunculan PHP 5, terdapat fitur baru yang ditambahkan ke dalam bahasa PHP yang berkaitan dengan pemrograman yang berorientasi objek seperti: *trait*, *interface*, *magic method*, dan sebagainya. Untuk kebutuhan modularisasi kode program, PHP juga menambahkan fitur *namespace* ke dalam bahasanya. Hingga sampai saat ini, PHP memiliki dukungan penuh terhadap konsep pemrograman berorientasi objek, seperti: Java, C++, dan C#.

Menurut Budi Raharjo (2015) yang menyatakan bahwa, pengembangan bahasa pemrograman PHP memfokuskan lebih kepada proses pembuatan aplikasi web yang sering disebut: *server-side scripting*, tapi sebenarnya PHP memiliki kemampuan yang lebih ideal dan diterapkan menjadi 3 tipe aplikasi, yaitu:

1. Aplikasi web (*server-side scripting*). Terdapat 3 hal yang dibutuhkan dalam menjalankan kode PHP sebagai aplikasi *web*, yaitu PHP Parser atau *software* PHP itu sendiri, *server web* (contohnya seperti *IIS*, *Apache*, *lighttpd*, *nginx* dan sebagainya), dan *web browser* (*Mozilla Firefox*, *Google Chrome*, *IE*, *Opera* dan sebagainya). Dengan tujuan untuk kode dapat dieksekusi oleh *PHP Interpreter*, *server web* harus dalam keadaan aktif. Akses halaman *web* sendiri dapat menggunakan aplikasi *web browser*.
2. Program CLI (*command-line scripting*). Pada jenis aplikasi ini, kode PHP dapat dieksekusi hanya dengan *PHP Interpreter* tanpa ada *server web* ataupun *web browser* dan dilakukan melalui *command prompt* atau terminal. Program CLI digunakan untuk melakukan tugas regular yang dilakukan dibelakang layar seperti proses *backup* data. Kode PHP dapat dijalankan secara terjadwal dan otomatis melalui program *cron* (*nix dan Linux) atau *Task Scheduler* (Windows).

3. Aplikasi *desktop* (GUI). Dalam mengembangkan PHP pada aplikasi *desktop* ini, harus menggunakan pustaka yang disediakan oleh pihak ketiga, seperti PHP-GTK dan wxPHP.

Ciri-ciri khusus yang dimiliki oleh kode PHP diantaranya sebagai berikut:

- a. Hanya dapat dijalankan menggunakan *web server*, contoh: *Apache*.
- b. Kode PHP diletakkan dan dieksekusi di *web server*.
- c. Kode PHP dapat digunakan untuk mengakses *database*, misal: MySQL.
- d. Termasuk dalam *software* yang bersifat *open source*.
- e. Dapat digunakan dan di *download* secara gratis.
- f. Mempunyai sifat *multiplatform*, yaitu dapat di eksekusi dengan menggunakan sistem operasi, seperti: *Linux*, *Unix*, *Windows* dan sebagainya.

PHP dapat diintegrasikan (*embedded*) ke dalam *web server* dan berperan sebagai program CGI yang terpisah. Karakteristik yang paling unggul dan paling kuat dalam PHP adalah lapisan integrasi *database* (*database integration layer*). *Database* yang didukung PHP adalah: *Oracle*, *Adabas-D*, *Sybase*, *FilePro*, *mSQL*, *Velocis*, *MySQL*, *Informix*, *Solid*, *dBase*, *ODBC*, *Unix dbm*, dan *PostgreSQL* (Schwendiman, 2001).

2.2.13. HTML

Pendapat Suryana dan Koesheryatin (2014:29) yang menyatakan bahwa HTML merupakan kepanjangan dari *HyperText Markup Language* adalah bahasa yang digunakan dalam menulis halaman web. Suatu pengembangan dari standar pemformatan dokumen teks yang sering disebut HTML, yaitu *Standard Generalized Markup Language* (SGML). Pada dasarnya HTML merupakan dokumen ASCII atau teks biasa, yang di rancang agar tidak tergantung pada suatu sistem operasi tertentu.

HTML memiliki fungsi utama, yaitu dapat memberikan perintah pada *browser* dalam melakukan manipulasi tampilan dengan melalui tag-tag yang ditulis pada HTML. Dengan demikian, *web browser* akan menghasilkan suatu tampilan

yang sesuai dengan perintah-perintah yang sudah dirancang, dibuat atau ditetapkan lebih dulu (Koesheryatin, 2014).

Pada perintah browser terdapat daftar tag yang sering digunakan dalam mempelajari pembuatan web, diantaranya sebagai berikut:

a. Tag “!..”

Berfungsi untuk menandai sebuah komentar suatu kode (*script*) dengan tanda ini, maka *browser* ini tidak akan menerjemahkannya.

b. Tag “a”

Kepanjangan dari *anchor*, tag ini biasanya untuk membuat sebuah tautan (*link*) antar *web*. Link mempunyai ciri-ciri, yaitu:

1. Bila link belum pernah diakses (klik) maka akan berwarna biru dan bergaris bawah.
2. Bila sudah pernah diakses (klik) maka akan berubah menjadi ungu dan tetap bergaris bawah.
3. Bila sedang aktif (diklik) maka akan berwarna merah dan bergaris bawah.

c. Tag “b”

Berfungsi untuk membuat teks cetak tebal.

d. Tag “big”

Berguna untuk membuat sebuah tulisan agar nampak lebih besar. Hal ini berbeda untuk cetak tebal (*bold*).

e. Tag “body”

Termasuk dalam bagian dari struktur HTML yang digunakan untuk menandai badan (*body*) suatu dokumen HTML.

f. Tag “br”

Berfungsi untuk mengakhiri suatu baris tulisan, lalu pindah ke baris baru dibawahnya.

g. Tag “button”

Tag ini digunakan untuk membuat tombol di *web browser*. Selain itu digunakan untuk *trigger* suatu proses, contohnya untuk menyimpan atau menghapus data. Tag ini biasanya diletakkan dalam sebuah *form*.

h. Tag “caption”

Tag ini berfungsi untuk membuat judul sebuah table. Penggunaannya tag ini biasanya bersamaan dengan deklarasi table.

i. Tag “center”

Biasanya digunakan untuk menengahkan suatu teks halaman.

j. Tag “div”

Tag ini berfungsi untuk mendefinisikan bagian dari suatu halaman *web* dengan format atau *style* tertentu.

k. Tag “font”

Berfungsi untuk memformat bentuk *font*, ukuran, warna, ukuran dari sebuah teks.

l. Tag “form”

Tag yang berfungsi untuk memasukkan data yang dilakukan oleh pengguna. Tag ini biasanya dibantu oleh beberapa komponen lain, seperti: *input*, *button*, *option* dan sebagainya.

m. Tag “frame” dan “frameset”

Tag ini biasanya berfungsi untuk menampilkan halaman web lain dalam sebuah halaman web yang dibuat. Halaman dapat dibuat terpisah dengan menampilkan halaman web lain yang berbeda-beda.

n. Tag “head”

Berfungsi untuk menampung elemen-elemen *header* dari sebuah halaman *web*. Penggunaannya seperti pada contoh tag “big”.

o. Tag “h1” sampai dengan “h6”

Berfungsi untuk mendefinisikan tulisan header. Untuk mengetahui perbedaannya, perhatikan ukuran setiap teks hasil di *browser*.

p. Tag “hr”

Berfungsi untuk membuat garis horizontal

- q. Tag “html”
Berfungsi untuk memberitahu *browser* bahwa kode yang dituliskan merupakan dokumen HTML. Penggunaannya seperti pada contoh tag “big”.
- r. Tag “i”
Berfungsi untuk membuat teks cetak miring.
- s. Tag “img”
Berfungsi untuk memasukkan gambar didalam dokumen HTML.
- t. Tag “input”
Berfungsi untuk berinteraksi dengan pengguna dalam hal memasukan data ke *server*. Banyak tipe *input* yang dimiliki, yaitu *button, checkbox, file, hidden, image, password, radio, submit, dan text*. Penggunaannya seperti pada contoh tag “form”
- u. Tag “li”
Berfungsi untuk menampilkan symbol data berurut, contohnya simbol *bullet* dan nomor. Dalam urutan bernomor diawali dengan tag “ol”, sedangkan untuk simbol *bullet* diawali tag “ul”
- v. Tag “link”
Berfungsi untuk mendefinisikan hubungan antara halaman web yang dibuat dengan data atau aturan dari luar. Tag ini biasanya digunakan pada saat web menggunakan file *css*.
- w. Tag “ol”
Berfungsi untuk urutan data bernomor atau numerik, dan juga alphabet. Penggunaannya seperti pada contoh tag “li”.

2.2.14. Bootstrap

Bootstrap dapat didefinisikan sebagai *framework front-end* yang intuitif dan *powerful* untuk pengembangan aplikasi web agar lebih cepat dan mudah. *Bootstrap* menggunakan bahasa pemrograman, yaitu HTML, CSS, dan *Javascript*. *Bootstrap* dikembangkan oleh Mark Otto dan Jacob Thornton dari Twitter. *Framework* ini diluncurkan sebagai produk *open source* pada Agustus 2011 di GitHub. *Bootstrap* mempunyai fitur-fitur komponen *interface* yang bagus seperti *Typography, Forms,*

Buttons, Tables, Navigations, Dropdowns, Alerts, Modals, Tabs, Accordion, Carousel, dan lain sebagainya (Enterprise, 2016).

Salah satu kelebihan yang dipunyai *bootstrap* adalah *framework* ini berisi kumpulan *tool* yang gratis untuk membuat *layout web* yang fleksibel dan responsif. *Framework* ini juga memiliki komponen interface bagus lainnya.

Bootstrap memiliki kelebihan-kelebihan antara lain sebagai berikut:

1. Menghemat waktu – *Bootstrap* sudah menyediakan berbagai desain template dan kelas, sehingga dapat menghemat waktu dan tenaga.
2. Fitur yang responsif – Pada *bootstrap* dapat dengan mudah membuat desain yang responsif. Dengan fitur- fitur ini, halaman web akan tampil responsif pada perangkat yang berbeda tanpa perlu adanya perubahan kode *markup*.
3. Desain yang konsisten – Semua komponen *bootstrap* mempunyai desain *template* dan *style* yang sama sehingga membuat tampilan web menjadi konsisten.
4. Mudah digunakan – Bootstrap sangat mudah digunakan oleh siapa saja yang mempunyai kemampuan dasar HTML dan CSS.
5. Didukung oleh semua *browser* populer – *Bootstrap* dapat digunakan pada semua *browser modern* seperti *Mozilla Firefox, Google Chrome, Safari, Internet Explorer*, dan *Opera*.
6. Gratis – *Bootstrap* merupakan *framework open source* yang dapat digunakan secara gratis.

2.2.15. CSS

Menurut Suryana dan Koesheryatin (2014:101) yang berpendapat bahwa *Cascading Style Sheet* (CSS) adalah suatu bahasa pemrograman *stylesheet* yang digunakan untuk mengatur tampilan suatu *website* yang mencakup tata letak, jenis huruf, warna, dan semua yang berhubungan dengan tampilan. Secara umum, CSS digunakan untuk memformat halaman web yang ditulis dengan HTML atau XHTML.

Cascading Style Sheet merupakan singkatan dari CSS, adalah bahasa pemrograman web yang biasanya digunakan untuk mengendalikan beberapa komponen dalam sebuah web, sehingga dapat menjadi lebih terstruktur dan seragam (Slamet, 2009).

Untuk menentukan sebuah penyajian visual pada dokumen HTML, CSS menggunakan sebuah aturan *style* yang terintegrasi dengan halaman web dengan melakukan berbagai cara, antara lain sebagai berikut:

a. *Internal style*

CSS disimpan di area *<head>* pada sebuah halaman HTML dan *style* hanya digunakan untuk halaman tersebut.

b. *External style*

CSS disimpan di file terpisah yang berformat *.css* dan *style* dapat digunakan untuk beberapa halaman web.

c. *Inline style*

Style hanya diterapkan pada elemen tertentu dalam sebuah dokumen.

2.2.16. MySQL

MySQL mempunyai pengertian yaitu *Database Management System* (DBMS) atau suatu perangkat lunak pada manajemen sistem berbasis data SQL. MySQL adalah *Relational Database Management System* (RDBMS) yang di kirim secara gratis dengan lisensi *General Public License* (GPL). Dimana setiap orang bebas untuk menggunakan MySQL, tetapi tidak boleh dijadikan produk turunan yang bersifat *close source* atau komersial.

Sesungguhnya MySQL adalah salah satu turunan konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*). SQL diartikan sebagai sebuah konsep pengoperasian database, yang biasanya untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data di kerjakan dengan mudah secara otomatis. Pada suatu sistem database (DBMS) memiliki andalan yang diketahui cara kerja *optimizer*-nya dalam melakukan proses perintah SQL, yang dibuat oleh pengguna maupun program aplikasi lainnya. Dalam

database server, MySQL dapat disebut lebih unggul daripada *database server* lainnya dalam *query* data. Oleh sebab itu, *query* dapat dilakukan untuk *single user*, dimana kecepatan query MySQL dapat sepuluh kali lebih cepat daripada *PostgreSQL* dan lima kali lebih cepat dibandingkan *Interbase*. Sehingga kemampuan ini cukup mengejutkan untuk perangkat lunak yang gratisan (Jogianto, 2003).

MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah David Axmark, Allan Larsson, dan Michael “Monty” Widenius. MySQL merupakan *database* yang pertama kali didukung oleh bahasa pemrograman *script* untuk internet (PHP dan Perl). MySQL dan PHP dianggap sebagai pasangan software pengembangan aplikasi web yang ideal. MySQL lebih sering digunakan untuk membangun aplikasi berbasis web, umumnya pengembangan aplikasinya menggunakan bahasa pemrograman *script* PHP (Arief, 2011).

MySQL mempunyai beberapa keunggulan diantaranya sebagai berikut:

1. Struktur Direktori MySQL. Software MySQL secara default akan diletakkan pada direktori C:\MySQL jika diinstal pada sistem operasi windows. Apabila instalasi dilakukan dengan menggunakan software PHP Triad (paket software yang menggabungkan tiga aplikasi yaitu Apache Web Server, PHP, dan MySQL), maka software MySQL terletak dalam direktori C:\Apache\MySQL.
2. Fleksibel. MySQL dapat digunakan untuk mengembangkan aplikasi desktop maupun aplikasi web. MySQL memiliki fleksibilitas terhadap teknologi yang akan digunakan sebagai pengembang aplikasi seperti PHP, JSP, Java, Delphi, C++ dan lain sebagainya.
3. Bersifat *open source* atau gratis

4. Proteksi data yang handal. MySQL menyediakan mekanisme yang powerful untuk menangani perlindungan terhadap keamanan, yaitu dengan menyediakan fasilitas manajemen *user*, enkripsi data dan lain sebagainya.

2.2.17. PhpMyAdmin

Untuk memajemen *database MySQL* secara visual dan *Server MySQL* terdapat *tools* yang sering digunakan dengan mudah, sehingga kita tidak perlu lagi harus menulis *query* SQL pada setiap akan melakukan perintah operasi database hal ini merupakan definisi dari *PhpMyAdmin*. Tools ini cukup terkenal dan populer, anda akan mendapatkan fasilitas ini ketika menginstall paket *triad PhpMyAdmin*, kerana termasuk dalam *xampp* yang sudah di *install*. *PhpMyAdmin* juga memiliki arti, yaitu *tools MySQL Client* berlisensi *Freeware*. *PhpMyAdmin* harus di eksekusi di sisi *server web* dan pada komputer harus tersedia *PHP*, karena berbasis web (Nugroho, 2013).

2.2.18. Web Server

Menurut Kurniawan (2008:2), *Web server* adalah sebuah perangkat lunak server yang memiliki fungsi untuk menerima permintaan HTTP (*Hyper Text Transfer Protocol*) atau HTTPS (*Hyper Text Transfer Protocol Secure*) dari klien yang dikenal sebagai *web browser* dan mengirimkan kembali hasilnya pada halaman-halaman web yang umumnya dalam bentuk format dokumen HTML. *Web server* yang dimaksud disini merupakan simulasi dari sebuah *web server* secara fisik. *Web server* biasanya disebut HTTP server karena menggunakan protokol HTTP sebagai basisnya. Adapun beberapa *web server* yang sering digunakan oleh pengguna antara lain, yaitu *PWS*, *IIS*, *Apache* dan lain sebagainya.

Menurut Wikipedia, *Server web* atau peladen web dapat merujuk baik pada perangkat keras (*hardware*) ataupun perangkat lunak (*software*) yang menyediakan layanan akses kepada pengguna melalui protokol komunikasi HTTP atau HTTPS atas berkas-berkas yang terdapat pada suatu situs web dalam layanan ke pengguna dengan menggunakan aplikasi tertentu seperti peramban web.

Penggunaan *server web* pada umumnya adalah untuk menempatkan situs web, tetapi pada prakteknya penggunaannya diperluas sebagai tempat penyimpanan data maupun untuk menjalankan sejumlah aplikasi kelas bisnis.

2.2.19. XAMPP

Menurut Puspitasari (2011:1), berpendapat bahwa pengertian dari *xampp* adalah sebuah perangkat lunak *web server apache* yang didalamnya sudah tersedia *database server MySQL* dan *support PHP programming*. *Xampp* merupakan perangkat lunak yang mudah digunakan secara gratis dan mendukung instalasi di *linux* dan *windows*. Keuntungan lainnya adalah cuma menginstal satu kali sudah tersedia *apache web server, MySQL database server, PHP support (PHP4 dan PHP5)* dan beberapa modul lainnya hanya bedanya kalau versi *windows* selalu dalam bentuk instalasi grafis dan yang *linux* dalam bentuk file terkompresi *tar.gz*. Kelebihan lainnya yang membedakan dari versi untuk *windows* adalah memiliki fitur untuk mengaktifkan sebuah *server* secara grafis, sedangkan *linux* masih berupa perintah-perintah didalam *console*. Oleh karena itu, versi untuk *linux* sulit untuk dioperasikan.

Menurut Nugroho (2013:6) yang menjelaskan bahwa terdapat dalam folder utama *xampp*, ada beberapa folder penting yang harus diketahui oleh pengguna. Terdapat fungsi folder penting pada *xampp*, untuk lebih lanjut dalam memahami disetiap fungsinya, dibawah ini dapat dilihat Tabel 2. 1 penjelasan mengenai fungsi folder *xampp*, antara lain sebagai berikut:

Tabel 2. 1.Fungsi Folder *Xampp*

| Folder | Keterangan |
|---------------|---|
| <i>Apache</i> | Salah satu folder utama dari <i>Apache Webserver</i> . |
| <i>Htdocs</i> | Termasuk bagian dari folder utama yang berguna untuk menyimpan data-data latihan web, baik PHP maupun HTML biasa. |
| <i>Manual</i> | Dalam folder ini berisi <i>subfolder</i> yang terdapat manual program dan <i>database</i> , termasuk manual <i>PHP</i> dan <i>MySQL</i> . |
| <i>MySQL</i> | Folder utama yang berfungsi untuk <i>database MySQL Server</i> . |

| Folder | Keterangan |
|--------|--|
| PHP | Folder ini berisi folder utama yang berguna untuk program PHP. |

(Sumber: Nugroho, 2013:7)

2.2.20. Sublime Text

Sublime Text adalah sebuah perangkat lunak aplikasi *editor* yang biasanya digunakan untuk kode dan teks yang dapat berjalan di berbagai *platform operating system* dengan menggunakan teknologi *Python API*. Terciptanya kemunculan aplikasi ini terinspirasi dari aplikasi Vim. Aplikasi tersebut sangatlah fleksibel dan *powerfull*. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan *sublime-packages*. *Sublime Text* bukanlah aplikasi *open source*, yang artinya aplikasi ini membutuhkan lisensi (*license*) yang harus dibeli. Akan tetapi, terdapat beberapa fitur pengembangan fungsionalitas (*packages*) dari aplikasi ini merupakan hasil dari temuan dan mendapat dukungan penuh dari komunitas serta memiliki lisensi (*license*) aplikasi gratis.

Sublime Text mendukung berbagai bahasa pemrograman dan mampu menyajikan fitur *syntax highlight* hampir di semua bahasa pemrograman yang didukung ataupun dikembangkan oleh komunitas antara lain: *C, C++, C#, CSS, D, Dylan, Erlang, HTML, Groovy, Haskell, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, MATLAB, OCaml, Perl, PHP, Python, R, Ruby, SQL, TCL, Textile* dan *XML*. Biasanya bagi bahasa pemrograman yang didukung atau belum didukung secara *default* dapat lebih dimaksimalkan atau *di-support* dengan menggunakan *add-ons* yang bisa *di-download* sesuai kebutuhan pengguna (Haughee, 2013).

2.2.21. Unified Markup Language

Unified Modelling Language (UML) adalah suatu standar bahasa yang sering digunakan di dalam dunia industri dengan tujuan untuk menginterpretasikan requirement, membuat analisis dan rancangan desain, serta mendeskripsikan arsitektur dalam pemrograman berorientasi objek. *Unified Modelling Language* (UML) merupakan salah satu bahasa pemrograman visual yang biasanya digunakan untuk pemodelan dan sebagai bahasa komunikasi yang membahas tentang sebuah sistem dengan menggunakan teks-teks pendukung dan diagram.

Dengan adanya kebutuhan pemodelan visual yang menyebabkan kemunculan *Unified Modelling Language* (UML) yang digunakan pengguna untuk menggambarkan, menspesifikasikan, merancang, membangun dan dokumentasi dari sebuah sistem perangkat lunak (*software*). Fungsi dari UML hanya untuk melakukan pemodelan. Maka dari itu, penggunaan UML tidak dibatasi pada metodologi tertentu, meskipun pada kenyataannya UML yang paling banyak digunakan untuk metodologi berorientasi objek (A.S & Shalahudin, 2015).

Diagram UML

Menurut Rosa A.S dan M. Shalahudin (2015: 140) yang mengemukakan bahwa UML mempunyai 13 macam diagram yang dikelompokkan terdiri dalam 3 kategori. Dibawah ini pembagian kategori yang dijelaskan dengan singkat sebagai berikut:

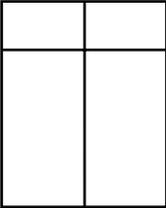
1. *Structure diagram*, yakni serangkaian diagram yang digunakan untuk mendeskripsikan suatu struktur statis dari sistem yang dimodelkan. *Structure diagram* memiliki beberapa diagram yang terdiri dari *class diagram*, *object diagram*, *component diagram*, *composite structure diagram*, *package diagram* dan *deployment diagram*.
2. *Behavior diagram*, yakni serangkaian diagram yang digunakan untuk mendeskripsikan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem. *Behavior diagram* memiliki beberapa diagram yang terdiri dari *Use case diagram*, *State Machine System*, *Activity diagram*.
3. *Interaction diagram*, yakni serangkaian diagram yang digunakan untuk mendeskripsikan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem. *Interaction diagram* mempunyai beberapa rangkaian diagram yang terdiri dari *Communication diagram*, *Sequence diagram*, *Timing diagram*, *Interaction Overview diagram*.

UML yang digunakan dalam pengembangan *website* sistem konsultasi dan rekapitulasi data pasien, antara lain:

a. *Activity Diagram*

Menurut Rosa A.S dan M. Shalahudin (2015:161) yang mendeskripsikan mengenai diagram aktivitas atau *activity diagram* yang menggambarkan *workflow* (aliran kerja) atau proses bisnis atau aktivitas dari sebuah sistem atau menu yang ada pada perangkat lunak. Dalam *activity diagram* ada hal yang butuh diperhatikan, yaitu bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, tetapi aktivitas yang dapat dilakukan oleh sebuah sistem. Di dalam *activity diagram* memiliki simbol-simbol yang dapat dilihat pada Tabel 2.2.

Tabel 2. 2 Simbol-Simbol Dalam *Activity Diagram*

| No. | Gambar | Nama | Keterangan |
|-----|---|--|---|
| 1 |  | <i>Swimlane</i> | Menampilkan organisasi bisnis dengan memisahkan siapa yang bertanggung jawab dalam setiap langkah terhadap aktivitas yang terjadi dalam activity diagram. |
| 2 |  | Status Awal (<i>Initial node</i>) | Dalam status awal suatu sistem untuk melakukan aktifitas pada sebuah aliran diagram aktifitas mempunyai sebuah status awal |
| 3 |  | Aktivitas (<i>Action</i>) | Langkah-langkah aktivitas yang dilakukan dalam sistem, aktivitas diawali dengan kata kerja antar aliran diagram aktivitas. |
| 4 |  | Percabangan (<i>Merge node</i>) | Pada asosiasi percabangan yang merupakan pilihan aktivitas lebih dari satu |

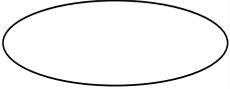
| No. | Gambar | Nama | Keterangan |
|-----|---|---------------------------------------|---|
| 5 |  | Penggabungan (<i>Join node</i>) | Suatu assosiasi penggabungan yang menggabungkan beberapa aliran aktivitas digabungkan menjadi satu aliran. |
| 6 |  | Status Akhir (<i>Final node</i>) | Sebuah sistem yang melakukan status akhir membutuhkan dengan tujuan untuk mengakhiri sebuah aliran diagram aktifitas. |

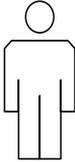
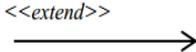
(Sumber: Rosa A.S dan M. Shalahudin, 2015:162)

b. *Use case Diagram*

Menurut Rosa A.S dan M. Shalahudin (2015:155) yang mendefinisikan tentang Use Case atau diagram use case merupakan suatu bentuk pemodelan untuk tingkah laku (*behavior*) yang di dalam sistem informasi yang akan dibuat. Pada *Use Case* diagram yang mendeskripsikan sebuah bentuk interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Pada umumnya, *use case* biasanya digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak untuk dapat menggunakan fungsi-fungsi tersebut. Dalam diagram *use case* terdapat simbol-simbol yang ada di *use case* diagram dapat dijelaskan pada Tabel 2.3 dibawah ini sebagai berikut:

Tabel 2. 3 Simbol Yang Terdapat Pada *Use Case Diagram*

| No. | Gambar | Nama | Keterangan |
|-----|---|-----------------|--|
| 1. |  | <i>Use Case</i> | Suatu sistem menyediakan fungsionalitas sebagai unit-unit yang saling bertukar pesan antar aktor dan unit, pada kenyataan biasanya digunakan sebagai kata kerja yang diawal frase nama <i>use case</i> . |

| No. | Gambar | Nama | Keterangan |
|-----|---|--|---|
| 2. |  | Aktor (<i>Actor</i>) | Suatu simbol yang berfungsi sebagai orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat oleh sistem itu sendiri, jadi walaupun simbol dari aktor yaitu gambar orang, tetapi aktor belum tentu merupakan orang, biasanya menggunakan kata benda yang diawali dengan frase nama aktor. |
| 3. |  | Asosiasi (<i>Association</i>) | Sebagai fungsi salah satu komunikasi antar aktor dan <i>Use Case</i> yang berpartisipasi di dalam diagram <i>use case</i> atau <i>use case</i> yang mempunyai suatu interaksi dengan aktor. |
| 4. |  | Ekstensi (<i>Extend</i>) | Sebuah relasi <i>use case</i> tambahan ke dalam sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa sebuah <i>use case</i> tambahan itu, hal ini sama dengan prinsip <i>inheritance</i> yang ada pada pemrograman berorientasi objek, umumnya <i>use case</i> tambahan mempunyai nama depan yang sama dengan <i>use case</i> yang ditambahkan. |
| 5. |  | Generalisasi (<i>Generalization</i>) | Suatu hubungan generalisasi dan spesialisasi antara dua buah <i>Use Case</i> yang mana fungsi yang satu lebih umum dari yang lainnya. |

| No. | Gambar | Nama | Keterangan |
|-----|--------|--|---|
| 6. | | Menggunakan (<i>Include/Use Case</i>) | Sebuah relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan membutuhkan use case ini untuk dapat mengeksekusi fungsinya atau sebagai syarat untuk menjalankan use case tersebut. Include yang artinya use case yang ditambahkan akan selalu dipanggil saat use case tambahan dapat dijalankan. |

(Sumber: Rosa A.S dan M. Shalahudin, 2015:156)

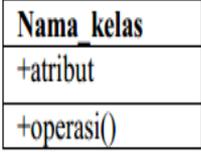
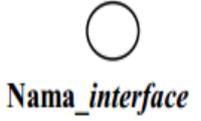
c. *Class Diagram*

Menurut Rosa A.S dan M. Shalahudin (2015:141) yang mengemukakan bahwa class diagram atau diagram kelas adalah suatu diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sebuah sistem. Pada diagram kelas mempunyai apa yang disebut atribut dan *method* atau operasi. Dibawah ini penjelasan dari atribut dan *method* antara lain sebagai berikut:

1. Atribut adalah variabel-variabel yang dimiliki oleh suatu kelas.
2. *Method* atau operasi merupakan fungsi-fungsi yang dimiliki oleh suatu kelas.

Pada diagram kelas ada simbol-simbol yang dijelaskan dibawah ini dapat dilihat pada tabel 2.4 sebagai berikut:

Tabel 2. 4 Simbol-Simbol Dalam *Class Diagram*

| No. | Gambar | Nama | Keterangan |
|-----|---|---|---|
| 1. |  | Kelas | Simbol kelas yang ada pada struktur sistem dalam diagram kelas. |
| 2. |  | Antarmuka (<i>Interface</i>) | Pada simbol ini bermakna sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek. |
| 3. |  | Asosiasi (<i>Association</i>) | Suatu simbol ini melambangkan relasi antarkelas dengan makna umum, asosiasi biasanya disertai dengan <i>multiplicity</i> . |
| 4. |  | Asosiasi berarah (<i>Directed Association</i>) | Suatu simbol ini melambangkan relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> . |
| 5. |  | Generalisasi | Suatu simbol ini melambangkan relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus) |
| 6. |  | Kebergantungan (<i>Dependency</i>) | Suatu simbol ini melambangkan relasi antarkelas dengan makna kebergantungan antarkelas. |

| No. | Gambar | Nama | Keterangan |
|-----|---|------------------------------------|---|
| 7. |  | Agregasi (<i>Aggregation</i>) | Suatu simbol ini melambangkan relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>). |

(Sumber: Rosa A.S dan M. Shalahudin, 2015:146)

d. ERD

Menurut Sutanta (2011:91) yang menjelaskan mengenai bahwa “*Entity Relationship Diagram* (ERD) merupakan bentuk suatu model data yang dikembangkan berdasarkan objek”. *Entity Relationship Diagram* (ERD) biasanya digunakan pengguna untuk mendeskripsikan hubungan antar data dalam basis data kepada pengguna secara logis yang dapat memudahkan pengguna dalam memahami penggunaannya. *Entity Relationship Diagram* (ERD) didasarkan pada suatu persepsi bahwa *real world* terdiri dari objek-objek dasar. Dalam penggunaan *Entity Relationship Diagram* (ERD) relatif mudah dipahami, bahkan oleh pengguna yang awam pun dapat memahaminya. Bagi analis sistem atau perancang, *Entity Relationship Diagram* (ERD) yang bermanfaat untuk memodelkan sistem yang nantinya, basis data akan dikembangkan. Pada model ini juga mampu membantu analis sistem atau perancang yang pada saat melakukan analis dan perancangan basis data karena model ini dapat menunjukkan macam data yang dibutuhkan dan kerelasian antardata yang ada didalamnya. Berikut ini keterangan simbol-simbol yang ada pada ERD dapat dilihat pada Tabel 2.5.

Tabel 2. 5 Keterangan Simbol ERD

| No. | Gambar | Nama | Keterangan |
|-----|---|---------|---|
| 1. |  | Entitas | Kumpulan dari objek yang dapat diidentifikasi secara unik. |
| 2. |  | Relasi | Hubungan yang terjadi antara satu atau lebih entitas. Jenis hubungan antara lain; satu ke satu, banyak ke banyak, dan banyak ke banyak. |
| 3. |  | Atribut | Karakteristik dari entity atau relasi yang merupakan penjelasan detail tentang entitas. |
| 4. |  | Garis | Hubungan antara entity dengan atributnya dan himpunan entitas dengan himpunan relasi. |

Komponen *Entity Relationship Diagram* (ERD)

Komponen *Entity Relationship Diagram* (ERD) menurut Sutanta (2011:91) adalah sebagai berikut:

a. Entitas

Entitas merupakan suatu objek yang dapat dibedakan dari lainnya yang dapat diwujudkan dalam basis data. Objek dasar dapat berupa orang, benda, atau hal yang keterangannya perlu disimpan didalam basis data. Untuk menggambarkan sebuah entitas digunakan aturan sebagai berikut ini:

1. Entitas dinyatakan dengan simbol persegi panjang
2. Nama entitas dituliskan didalam simbol persegi panjang

3. Nama entitas berupa kata benda, tunggal.
4. Nama entitas sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

b. Atribut

Atribut merupakan keterangan-keterangan yang terkait pada sebuah entitas yang perlu disimpan dalam basis data. Atribut berfungsi sebagai penjelas pada sebuah entitas. Untuk menggambarkan atribut digunakan aturan sebagai berikut:

1. Atribut digambarkan dengan simbol ellips.
2. Nama atribut dituliskan didalam simbol ellips.
3. Nama atribut merupakan kata benda, tunggal.
4. Nama atribut sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan makna dengan jelas.

c. Relasi

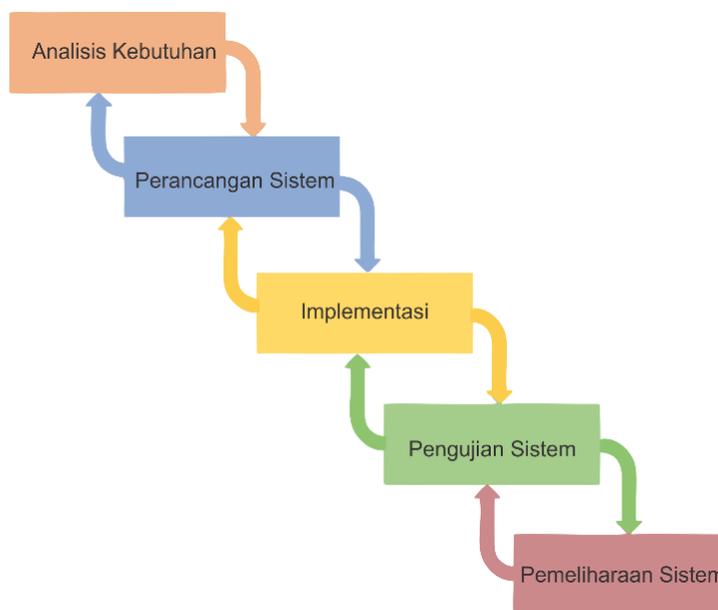
Relasi merupakan hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Aturan penggambaran relasi adalah sebagai berikut:

1. Relasi dinyatakan dengan simbol belah ketupat.
2. Nama relasi dituliskan didalam simbol belah ketupat.
3. Nama relasi berupa kata kerja aktif.
4. Nama relasi sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

2.2.22. Metode *Waterfall*

Metode *waterfall* atau yang biasanya sering disebut dengan metode air terjun merupakan suatu siklus hidup klasik (*classic life cycle*), yang mana metode ini menggambarkan suatu pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak (*software*), mulai diawali dengan spesifikasi kebutuhan pengguna yang kemudian berlanjut melalui tahapan-tahapan perencanaan (*planning*), permodelan (*modeling*), konstruksi (*construction*), serta penyerahan sistem ke para pelanggan/pengguna (*deployment*), yang diakhiri dengan

dukungan pada perangkat lunak yang lengkap dihasilkan. Pada metode waterfall ini memiliki tahapan yang dapat ditunjukkan pada gambar yang dibawah ini sebagai berikut (Pressman, 2012).



Gambar 2. 3 Langkah Penelitian Dengan Metode *Waterfall*

Terdapat beberapa tahapan yang berurutan dalam pengembangan yang dimiliki metode *waterfall*, yaitu *requirement* (analisis kebutuhan), *design system* (desain sistem), *coding* (pengkodean) dan *testing* (pengujian), penerapan program serta tahapan terakhir pemeliharaan. Dibawah ini penguraian dari tahapan-tahapan dalam metode *waterfall* sebagai berikut:

1. Analisis Kebutuhan.

Tahapan analisis ini adalah tahapan pengembangan pertama yang harus dilakukan oleh *developer*. Pada tahap ini komunikasi yang baik sangat dibutuhkan dalam metode *waterfall*. Pengembang sistem memerlukan komunikasi yang bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi ini biasanya dapat diperoleh melalui wawancara, diskusi atau survei langsung. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna. Tahapan analisis merupakan penentu tahapan awal untuk melanjutkan ke tahapan selanjutnya.

2. Perancangan Sistem.

Tahapan selanjutnya adalah tahapan perancangan sistem. Tahapan ini memiliki tujuan untuk memberikan gambaran yang seharusnya dikerjakan dan bagaimana tampilan sistem yang akan dibuat. Spesifikasi kebutuhan dari tahap sebelumnya akan dipelajari dalam fase ini dan desain sistem disiapkan. Perancangan sistem ini membantu dalam menentukan perangkat keras (*hardware*) dan sistem persyaratan serta juga dapat membantu dalam mendefinisikan arsitektur sistem secara keseluruhan.

3. Implementasi.

Pada tahap ketiga selanjutnya yaitu implementasi. Tahap ini, sistem pertama kali dikembangkan di program kecil yang sering dikenal dengan sebutan *unit*, yang terintegrasi dalam tahap selanjutnya. Proses pengkodean pada suatu sistem mulai dari *unit* terkecil. Setiap *unit* dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai *unit testing*.

4. Pengujian Sistem.

Di tahap ini, semua unit yang dikembangkan dalam tahap implementasi program diintegrasikan ke dalam sistem setelah pengujian yang dilakukan masing-masing unit. Kemudian, diintegrasikan semua unit sistem diuji untuk mengecek setiap kegagalan dan kesalahan (*error*).

5. Pemeliharaan Sistem

Tahap akhir dalam metode *waterfall* adalah tahap pemeliharaan sistem. Jika keseluruhan tahapan telah diselesaikan dan telah menjadi sebuah sistem perangkat lunak (*software*), lalu akan dijalankan serta dilakukan pemeliharaan. Pemeliharaan ini termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah testing sebelumnya. Pada tahapan ini lebih mengutamakan pemeliharaan, karena uji coba sangat menentukan apakah sistem akan berhasil atau tidak untuk memenuhi kebutuhan. Perbaikan implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru.

Kelebihan Metode *Waterfall*

Kelebihan dari menggunakan metode air terjun (*waterfall*) adalah metode ini memungkinkan untuk departementalisasi dan pengontrol. Dalam pengontrolan proses pengembangan model fase *one by one*, sehingga meminimalis kesalahan yang mungkin akan terjadi. Pengembangan bergerak dari suatu konsep, yaitu melalui sebuah desain, implementasi, pengujian, instalasi, penyelesaian masalah, dan berakhir di operasi serta juga pemeliharaan. Selain itu, dalam metode *waterfall* juga klasik dan mudah dipahami untuk semua orang dalam melakukan pengembangan.

Kekurangan Metode *Waterfall*

Kekurangan dari menggunakan metode air terjun (*waterfall*) adalah metode ini tidak memungkinkan untuk banyak revisi jika terjadi kesalahan dalam prosesnya. Karena setelah aplikasi ini dalam tahap pengujian, susah untuk dapat kembali lagi ke tahapan awal dan mengubah sesuatu yang tidak terdokumentasi dengan baik dalam tahap konsep sebelumnya. Karena tahapan ini dilakukan sangat berurutan (*step by step*). Selain itu, kekurangan metode *waterfall* juga dalam persoalan biaya, pembengkakan biaya karena memakan waktu yang terlalu lama.

2.2.23. Pengujian *Black Box*

Menurut Roger S. Pressman (2010:597) yang menyatakan bahwa *Black Box Testing* (Pengujian Kotak Hitam) atau biasanya juga sering disebut pengujian perilaku, dimana berfokus pada persyaratan fungsional dari suatu perangkat lunak. Yang artinya, teknik pengujian kotak hitam (*Black Box Testing*) yang memungkinkan anda untuk dapat membuat sekumpulan kondisi masukan yang sepenuhnya akan menjalankan semua persyaratan kebutuhan fungsional untuk suatu program. Pengujian kotak hitam (*Black Box Testing*) ini yang menfokuskan pada perangkat lunak (*software*) yang memungkinkan *programmer* untuk mendapatkan *input* yang sepenuhnya melakukan persyaratan fungsional untuk

program. Pengujian kotak hitam (*Black Box Testing*) tidak cocok pada teknik alternatif untuk kotak hitam. Sedangkan, pengujian ini adalah pendekatan pelengkap yang mungkin dilaksanakan untuk membuka kelas kesalahan yang berbeda dari yang dibuka oleh metode pengujian kotak putih (*White Box Testing*). Pengujian kotak hitam (*Black Box Testing*) ini memiliki beberapa upaya untuk menemukan kesalahan dalam kategori sebagai berikut:

1. Fungsi yang tidak benar atau fungsi yang hilang.
2. Kesalahan antarmuka (*interface*).
3. Kesalahan dalam struktur data atau akses *database* eksternal.
4. Kesalahan kinerja atau performansi.
5. Kesalahan inisialisasi dan pemutusan kesalahan

Pengujian kotak hitam (*Black Box Testing*) memiliki kelebihan dan kelemahan yang diuraikan dibawah ini sebagai berikut:

1. Kelebihan *Black Box Testing*

Meskipun dalam pelaksanaannya *testing* kita dapat menguji keseluruhan fungsional perangkat lunak namun formal *black box testing* yang sebenarnya kita dapat memilih *subset test* yang secara efektif dan efisien dapat menentukan cacat. Dengan cara ini, *black box testing* dapat membantu memaksimalkan *testing investment*.

2. Kelemahan *Black Box Testing*

Ketika *tester* melakukan *black box testing*, *tester* tidak akan pernah yakin apakah perangkat lunak (*software*) yang telah diuji telah benar-benar lolos pengujian. Hal ini terjadi karena kemungkinan masih ada beberapa jalur eksekusi yang belum pernah diuji oleh *user*. Untuk menentukan cacat perangkat lunak (*software*) menggunakan *black box testing*, *tester* seharusnya membuat setiap kemungkinan kombinasi data *input* baik yang valid maupun yang tidak valid.

Ada beberapa langkah-langkah untuk menjalankan dalam pengujian kotak hitam (*Black Box Testing*) antara lain sebagai berikut:

1. Analisa kebutuhan dan spesifikasi
2. Pemilihan *input*
3. Pemilihan *output*-nya
4. Seleksi *input*
5. Pengujian
6. *Review* hasil