

BAB II

LANDASAN TEORI

Untuk mengatasi permasalahan yang ada pada saat perancangan sistem informasi, sebaiknya mencari sumber-sumber tertentu sebagai informasi tambahan dan membantu untuk kedepannya pada saat perancangan.

2.1 Penelitian Terdahulu

Beberapa penelitian yang dengan berkaitan tema sejenis yang menjadi dasar penelitian ini dilakukan yaitu:

1. Fuad Rizky Novarin dan Sintya Sukarta, dalam penelitiannya yang berjudul “SISTEM INFORMASI PENJUALAN BERBASIS WEBSITE PADA BEHOLDER CLOTH” di Bandung, menjelaskan tentang sistem informasi penjualan pada Beholder Cloth yang bertujuan untuk merancang dan membangun sistem informasi yang dapat membantu mempromosikan produk yang dijual serta sebagai media transaksi penjualan secara online yang dapat mengotomatisasi proses transaksi penjualan dan tidak terkait jarak. Metode pengembangan yang digunakan pada penelitian ini menggunakan metode *prototype*, dan untuk metode penedekatannya menggunakan metode terstruktur [1].
2. Taufiq Hidayat dan Bella Hardiyana, dalam penelitiannya yang berjudul “SISTEM INFORMASI PEMBELIAN, PRODUKSI DAN PENJUALAN SINTONG TOPI BERBASIS WEB PADA TWIN’S PLAS”, menjelaskan

tentang sistem informasi pembelian, produksi dan penjualan sintong topi berbasis web pada Twin's Plas yang bertujuan untuk merancang dan mengimplementasikan sistem informasi pembelian, produksi dan penjualan berbasis web yang diusulkan untuk Twin's Plas dikarenakan sistem sebelumnya yang tradisional. Dalam kedua penelitian tersebut hanya membahas mengenai penjualan dan pembelian. Sementara itu, pada penelitian ini tidak hanya membahas mengenai penjualan dan pembelian saja, tetapi juga membahas mengenai pengelolaan data keluar masuk barang/bahan. Metode pengembangan yang digunakan pada penelitian ini menggunakan metode prototype, dan untuk metode penedekatannya menggunakan metode terstruktur [2].

2.2 Definisi

Definisi merupakan suatu batasan atau arti, bisa juga dimaknai kata, frasa, atau kalimat yang mengungkapkan makna, keterangan, atau ciri utama dari orang, benda, proses, atau aktivitas.

2.2.1 Konsep Dasar Sistem

Menurut Tata Sutabri “Sistem merupakan sebagai suatu kumpulan atau himpunan yang berasal dari unsur, komponen, variabel yang teroganisir, bergantung satu sama lain dan terpadu untuk satu tujuan. Model dasar yang membentuk sistem ini yaitu adanya masukan (*input*), pengolahan (*process*) dan keluaran (*output*)”. [3]

2.2.1.1 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, yaitu mempunyai komponen, batas sistem, lingkungan luar sistem, penghubung, masukan, keluaran, tujuan (Jogiyanto, 1999:3). Adapun pengertian dari masing-masing karakteristik Sistem tersebut adalah sebagai berikut:

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan.

2. Batasan Sistem

Batasan sistem (*boundary*) merupakan daerah yang membatasi antara suatu dengan Sistem yang lainnya atau dengan lingkungan luarnya.

3. Lingkungan Luar Sistem

Lingkungan luar sistem (*envronment*) dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi oprerasi sistem.

4. Penghubung Sistem

Penghubung (*interface*) merupakan media penghubung antara satu sub sistem dengan sub sistem yang lainnya.

5. Masukan Sistem

Masukan (*input*) energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran.

6. Keluaran Sistem

Keluaran (*output*) adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolahan Sistem

Suatu sistem dapat mempunyai suatu bagian pengolahan yang akan merubah masukan menjadi keluaran.

8. Sasaran atau Tujuan Sistem

Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang dihasilkan sistem. Suatu sistem pasti mempunyai tujuan atau sasaran, kalau tidak mempunyai sasaran maka operasi sistem tidak ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuannya.

2.2.2 Konsep Dasar Informasi

Menurut buku Elisabet Yunaeti Anggraeni dan Rita Irviani yang berjudul “Pengantar Sistem Informasi”. Secara umum informasi merupakan hasil dari pengolahan data yang lebih berarti dan berguna bagi para penerimanya [4].

1. Pengertian

Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang (Kadir, 2003:31). Data merupakan bentuk yang

masih mentah yang belum dapat bercerita banyak sehingga perlu diolah lebih lanjut. Data diolah melalui suatu model untuk dihasilkan informasi.

2. Kualitas Informasi

Kualitas dari suatu informasi tergantung dari tiga hal, tiga itu antara lain informasi harus akurat, tepat pada waktunya dan relevan. Akurat berarti informasi harus bebas dari kesalahan-kesalahan, tepat pada waktunya berarti informasi harus datang pada penerima tidak boleh terlambat, nilai mahal nya Masukan (data) Proses (Model) Keluaran (Informasi) Hasil Tindakan Data (ditangkap) Keputusan Tindakan Penerima Data Dasar informasi disebabkan cepatnya informasi tersebut didapat, sedangkan relevan berarti informasi mempunyai manfaat untuk pemakaiannya.

2.2.3 Pengertian Sistem Informasi

Menurut Muhammad Rajab Fachrizal dalam jurnal yang berjudul “PROTOTYPE SISTEM INFORMASI PENGELOLAAN AKADEMIK BERBASIS SMS GATEWAY DI SMA NEGERI 22 BANDUNG”. Penelitian tersebut menjelaskan bahwa “Pada dasarnya sistem informasi merupakan suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi.” [5]

Adapun Nilai Informasi, Menurut Amsyah (2001) nilai Informasi, ditentukan Oleh lima Karakteristik yaitu:

1. Ketelitian

Perbandingan dari informasi yang benar dengan jumlah seluruh informasi yang dihasilkan pada satu proses pengolahan data tertentu.

2. Ketepatan Waktu

Informasi yang terlambat tidak akan berguna walaupun informasi itu akurat karena keterlambatan membuat informasi sudah tidak berguna lagi.

3. Kelengkapan

Informasi yang kurang lengkap akan mengakibatkan ketertundaan pengambilan keputusan.

4. Ringkas

Informasi sangat bernilai jika disajikan dengan ringkas dan langsung ke sasaran yang diperlukan, tidak bertele–tele dan berlebihan.

5. Kesesuain

Informasi bernilai tinggi harus sesuai dengan keperluan pekerjaan atau keperluan manajemen.

2.2.4 Pengertian Sistem Informasi Penjualan

Menurut Suwinarno Nadjamuddin dalam jurnal yang berjudul “Sistem Informasi Penjualan, Pembelian Dan Jasa Grooming Pada Tera Petshop”, menjelaskan bahwa” Sistem informasi penjualan adalah sistem informasi yang mengatur tentang penjualan baik yang dilakukan secara piutang maupun secara tunai. Penjualan merupakan suatu kegiatan pelengkap atau suplemen dari pembeli, untuk memungkinkan terjadinya transaksi yang terdiri dari serangkaian kegiatan

yang meliputi permintaan (*demand*), mencari calon pembeli, negosiasi harga dan syarat pembayaran”. [6]

komponen-komponen dari sistem informasi penjualan secara umum terdiri dari:

- a. Pencatatan transaksi penjualan
- b. Pengecekan stok barang
- c. Kalkulasi jumlah dan harga
- d. Pembuatan dan pencetakan nota penjualan
- e. Pembuatan dokumen atau informasi penjualan untuk keperluan manajemen

Dalam sistem yang dibangun, komponen pencatatan transaksi penjualan, pengecekan stok barang, kalkulasi jumlah dan harga beserta pembuatan dan pencetakan laporan penjualan masuk dalam sub sistem point of sales. Sedangkan dokumen atau informasi penjualan untuk keperluan manajemen akan menjadi output dari sistem informasi yang dibangun.

2.2.5 Pengertian Pembelian

Pengertian pembelian pada umumnya adalah pengadaan barang atau jasa untuk keperluan konsumsi. Beberapa ahli mengemukakan tentang definisi pembelian antara lain.

Menurut Novrini Hasti, S. Si, MT dan Dani Setiadi, S. Kom dalam jurnal yang berjudul “SISTEM INFORMASI PENJUALAN DAN PEMBELIAN OBAT” menjelaskan bahwa, ”Pembelian merupakan kegiatan utama untuk menjamin kelancaran transaksi penjualan yang terjadi dalam suatu perusahaan. Dengan

adanya pembelian, perusahaan dapat secara mudah menyediakan sumber daya yang diperlukan organisasi secara efisien dan efektif”.[7]

Menurut Soemarso (2009:194) kegiatan pembelian dalam sebuah perusahaan dagang meliputi hal-hal sebagai berikut:

- a. Membeli barang dagang secara tunai atau kredit.
- b. Membeli aktiva produktif untuk digunakan dalam kegiatan perusahaan, contohnya kegiatan ini adalah pembelian kendaraan, peralatan kantor dan lain-lain.
- c. Membeli barang dan jasa-jasa lain sehubungan dengan kegiatan perusahaan,
- d. contohnya adalah biaya pengiriman, biaya listrik, air dan telepon.

Sehingga dapat diperoleh kesimpulan bahwa pembelian merupakan tindakan untuk memperoleh barang atau jasa secara tunai atau kredit yang digunakan dalam kegiatan perusahaan untuk memenuhi kebutuhan produksi

2.2.6 Rekapitulasi stok barang

Rekapitulasi Stok Barang adalah sekumpulan data-data barang perusahaan baik hasil penjualan maupun stok barang yang masih tersedia.

Stok barang ialah sebagai suatu aktivitas lancar yang meliputi barang-barang yang merupakan milik perusahaan dengan sebuah maksud supaya dijual dalam suatu periode usaha normal ataupun persediaan barang-barang yang masih dalam pekerjaan sebuah proses produksi maupun persediaan bahan baku yang juga menunggu penggunaannya di dalam suatu proses produksi (Assauri, 2016).

2.3 Piranti Pendukung

Jaringan komputer adalah sebuah sistem yang terdiri dari dua atau lebih komputer yang saling terhubung satu sama lain melalui media transmisi atau media komunikasi sehingga dapat saling berbagi data, aplikasi maupun berbagi perangkat keras komputer. Beberapa piranti pendukung yang digunakan dalam membangun aplikasi ini adalah:

2.3.1 Perangkat Keras

Pada seperangkat komputer terdapat beberapa komponen yang bekerja saling mendukung. Komponen tersebut dikenal dengan istilah hardware atau perangkat keras. Setiap komputer dirancang dengan spesifikasi perangkat keras tersendiri.

1. Laptop, sebagai alat yang digunakan dalam membangun aplikasi.
2. Mouse dan keyboard, sebagai alat masukan.
3. Monitor, sebagai alat keluaran.

2.3.2 Perangkat Lunak

Software (perangkat lunak) adalah istilah khusus untuk data yang diformat, dan disimpan secara digital, didalamnya termasuk program komputer, dokumentasinya, dan berbagai informasi yang dapat dibaca, dan ditulis oleh komputer. Perangkat lunak juga dapat dikatakan sebagai bagian sistem komputer yang tidak berwujud. Istilah ini digunakan untuk menonjolkan perbedaannya dengan Hardware (perangkat keras) komputer.

1. Windows: Sebagai sistem operasi yang digunakan oleh penulis.
2. Microsoft Word: Sebagai media penulisan.

3. XAMPP: Sebagai server web lokal.
4. Visual Studio Code : Sebagai penyunting kode-sumber.
5. Code Igniter 3 : *Framework Website*

2.3.3 Website

Dalam dunia teknologi yang pesat ini diperlukan suatu jaringan yang bias mempermudah serta mempercepat penyampaian informasi secara luas, dan dapat dengan mudah dan cepat oleh siapapun yang mendapatkan akses *internet*.

Menurut Beki (2015:35) menyimpulkan bahwa “*Website* merupakan kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman.”[8]

2.3.4 PHP

1. Sejarah PHP

Pada awalnya PHP merupakan kependekan dari Personal Home Page (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama Form Interpreted (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web.

- a. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilsan kode sumber ini menjadi

sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan PHP.

- b. Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini, interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.
- c. Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang PHP: Hypertext Preprocessing.
- d. Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut
- e. dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi.
- f. Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek. Server web bawaan ditambahkan pada

versi 5.4 untuk mempermudah pengembang menjalankan kode PHP tanpa menginstall software server.

- g. Versi terbaru dan stabil dari bahasa pemrograman PHP saat ini adalah versi 7.0.16 dan 7.1.2 yang resmi dirilis pada tanggal 17 Februari 2017.

2. PHP

PHP sering dipakai para programmer untuk membuat situs web yang bersifat dinamis karena gratis dan berguna dalam merancang aplikasi web. Supono dan Putratama (2016:3) mengemukakan bahwa "PHP (*PHP:Hypertext Preprocessor*) adalah suatu bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang berbasis server-side yang dapat ditambahkan ke dalam HTML". [9]

2.3.4.1 OOP

OOP (*Object Oriented Programming*) adalah suatu metode pemrograman yang berorientasi kepada objek. Tujuan dari OOP diciptakan adalah untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada di kehidupan sehari-hari. Jadi setiap bagian dari suatu permasalahan adalah objek, nah objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil lagi. Saya ambil contoh Pesawat, Pesawat adalah sebuah objek. Pesawat itu sendiri terbentuk dari beberapa objek yang lebih kecil lagi seperti mesin, roda, baling-baling, kursi, dll. Pesawat sebagai objek yang terbentuk dari objek-objek yang lebih kecil saling berhubungan, berinteraksi, berkomunikasi dan saling mengirim pesan kepada objek-objek yang lainnya. Begitu juga dengan program,

sebuah objek yang besar dibentuk dari beberapa objek yang lebih kecil, objek-objek itu saling berkomunikasi, dan saling berkiripesan kepada objek yang lain.

2.3.4.2 Konsep OOP

1. Abstrak *Class*

- a. Kelas merupakan deskripsi abstrak informasi dan tingkah laku dari sekumpulan data.
- b. Kelas dapat diilustrasikan sebagai suatu cetak biru(*blueprint*) atau prototipe yang digunakan untuk menciptakan objek.
- c. Kelas merupakan tipe data bagi objek yang mengenkapsulasi data dan operasi pada data dalam suatu unit tunggal.
- d. Kelas mendefinisikan suatu struktur yang terdiri atas data kelas (data field), prosedur atau fungsi (*method*), dan sifat kelas (*property*).

2. *Encapsulation*

- a. Istilah enkapsulasi sebenarnya adalah kombinasi data dan fungsionalitas dalam sebuah unit tunggal sebagai bentuk untuk menyembunyikan detail informasi.
- b. Proses enkapsulasi memudahkan kita untuk menggunakan sebuah objek dari suatu kelas karena kita tidak perlu mengetahui segala hal secara rinci.
- c. Enkapsulasi menekankan pada antarmuka suatu kelas, atau dengan kata lain bagaimana menggunakan objek kelas tertentu.

- d. Contoh: kelas mobil menyediakan antarmuka fungsi untuk menjalankan mobil tersebut, tanpa kita perlu tahu komposisi bahan bakar, udara dan kalor yang diperlukan untuk proses tersebut.

3. *Inheritance*

- a. Kita dapat mendefinisikan suatu kelas baru dengan mewarisi sifat dari kelas lain yang sudah ada.
- b. Penurunan sifat ini bisa dilakukan secara bertingkattingkat, sehingga semakin ke bawah kelas tersebut menjadi semakin spesifik.
- c. Sub kelas memungkinkan kita untuk melakukan spesifikasi detail dan perilaku khusus dari kelas supernya.
- d. Dengan konsep pewarisan, seorang programmer dapat menggunakan kode yang telah ditulisnya pada kelas super berulang kali pada kelas-kelas turunannya tanpa harus menulis ulang semua kodekode itu.

4. *Polymorphism*

- a. Polimorfisme merupakan kemampuan objek-objek yang berbeda kelas namun terkait dalam pewarisan untuk merespon secara berbeda terhadap suatu pesan yang sama.
- b. Polimorfisme juga dapat dikatakan kemampuan sebuah objek untuk memutuskan method mana yang akan diterapkan padanya, tergantung letak objek tersebut pada jenjang pewarisan.
- c. *Method overriding*.
- d. *Method name overloading*.

2.3.5 CodeIgniter

CodeIgniter adalah sebuah *web application network* yang bersifat *opensource* yang digunakan untuk membangun aplikasi php dinamis. *CodeIgniter* menjadi sebuah *framework* PHP dengan model MVC (*Model, View, Controller*) untuk membangun website dinamis dengan menggunakan PHP yang dapat mempercepat pengembang untuk membuat sebuah aplikasi web. Selain ringan dan cepat, *CodeIgniter* juga memiliki dokumentasi yang super lengkap disertai dengan contoh implementasi kodenya. Dokumentasi yang lengkap inilah yang menjadi salah satu alasan kuat mengapa banyak orang memilih *CodeIgniter* sebagai *framework* pilihannya.

CodeIgniter pertama kali dikembangkan pada tahun 2006 oleh Rick Ellis. Dengan logo api yang menyala, *CodeIgniter* dengan cepat “membakar” semangat para *web developer* untuk mengembangkan web dinamis dengan cepat dan mudah menggunakan *framework* PHP yang satu ini.[14]

2.3.5.1 Fungsi CodeIgniter

1. Mempercepat dan mempermudah kita dalam pembuatan website.
2. Menghasilkan struktur pemrograman yang sangat rapi, baik dari segi kode maupun struktur file phpnya.
3. Memberikan standar coding sehingga memudahkan kita atau orang lain untuk mempelajari kembali system aplikasi yang dibangun.

2.3.5.2 Kelebihan CodeIgniter

1. Berukuran sangat kecil. File downloadnya hanya sekitar 2MB, itupun sudah include dokumentasinya yang sangat lengkap.
2. Dokumentasi yang bagus. Saat anda mendownloadnya, telah disertakan dengan dokumentasi yang berisi pengantar, *tutorial*, bagaimana panduan penggunaan, serta referensi dokumentasi untuk komponen-komponennya.
3. Kompatibilitas dengan *Hosting*. CodeIgniter mampu berjalan dengan baik pada hampir semua *platform hosting*. CodeIgniter juga mendukung database database paling umum, termasuk MySQL.
4. Tidak ada aturan coding yang ketat. Terserah anda jika anda hanya ingin menggunakan *Controller*, tanpa *View*, atau tidak menggunakan Model, atau tidak salah satu keduanya. Namun dengan menggunakan ketiga komponennya adalah pilihan lebih bijak.
5. Kinerja yang baik. Codeigniter sangat cepat bahkan mungkin bisa dibilang merupakan *framework* yang paling cepat yang ada saat ini.
6. Sangat mudah diintegrasikan. CodeIgniter sangat mengerti tentang pengembangan berbagai *library* saat ini. Karenanya CodeIgniter memberikan kemudahan untuk diintegrasikan dengan *library-library* yang tersedia saat ini.
7. Sedikit Konfigurasi. Konfigurasi CodeIgniter terletak di folder *application/config*. CodeIgniter tidak membutuhkan konfigurasi yang rumit, bahkan untuk mencoba menjalankannya, tanpa melakukan konfigurasi sedikitpun ia sudah bisa berjalan.

8. Mudah dipelajari. Disamping dokumentasi yang lengkap, ia juga memiliki berbagai forum diskusi.

2.3.5.3 Kekurangan CodeIgniter

1. CodeIgniter tidak ditujukan untuk pembuatan web dengan skala besar.
2. Library yang sangat terbatas. Hal ini dikarenakan sangat sulit mencari *plugin* tambahan yang terverifikasi secara resmi, karena pada situsnya CodeIgniter tidak menyediakan plugin-plugin tambahan untuk mendukung pengembangan aplikasi dengan CI.
3. Belum adanya editor khusus CodeIgniter, sehingga dalam melakukan *create project* dan modul-modulnya harus berpindah-pindah folder.

2.3.5.4 CodeIgniter 3

Pada bulan Juli 2013, EllisLab mengumumkan bahwa mereka mencari pemilik baru untuk CodeIgniter karena internal mereka sendiri tidak memiliki cukup fokus untuk terus mengembangkan CodeIgniter. Akhirnya pada bulan Oktober 2014, kepemilikan CodeIgniter berpindah tangan kepada *British Columbia Institute of Technology*, salah satu sekolah tinggi teknologi di Kanada. Setelah hampir lima bulan lamanya sejak peralihan kepemilikan, BCIT akhirnya merilis CodeIgniter 3.0. Dan berikut adalah perubahan codeigniter 2 menjadi codeigniter 3:

1. Codeigniter 3 memerlukan PHP versi 5.1.6 atau di atasnya.
2. Penamaan model, controller Codeigniter 3 harus diawali huruf besar.

3. Driver databasenya kini memiliki *refactoring* yang lebih luas. Sekarang *default database driver* nya menggunakan *mysqli*, tidak lagi menggunakan *mysql*.
4. Penambahan user agent Windows 7, Windows 8, Windows 8.1, Android, Blackberry, iOS dan PlayStation 3.
5. Update perbaikan di *mimes.php*.
6. Update penulisan class dengan PHP 5 style.
7. Pindah path halaman *error* di *application/view/errors*.
8. Pindah Log Class di *application/core*.
9. Update perbaikan di beberapa *Library dan Helper*.
10. Perbaikan file dokumentas.

2.3.6 MVC

MVC adalah konsep dasar yang harus diketahui sebelum mengenal CodeIgniter. MVC (*Model View Controller*) merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web, berawal pada bahasa pemrograman Small Talk, yang memisahkan bisnis logic (alur piker), *data logic* (penyimpanan data) dan *presentation logic* (antarmuka aplikasi) atau secara sederhana adalah memisahkan antara desain, data dan proses.[15] Ada 3 komponen yang membangun suatu MVC yaitu:

1. Model, biasanya berhubungan dengan data dan interaksi ke *database* atau *webservice*. Model juga merepresentasikan struktur data dari aplikasi yang bisa berupa basis data maupun data lain, misalnya dalam bentuk file teks, file XML maupun *webservice*. Biasanya didalam model akan berisi *class*

dan fungsi untuk mengambil, melakukan *update* dan menghapus data website. Sebuah aplikasi web biasanya menggunakan basis data dalam menyimpan data, maka pada bagian Model biasanya akan berhubungan dengan perintah-perintah *query* SQL.

2. *View*, merupakan bagian yang menangani *presentation logic*. Pada suatu aplikasi web bagian ini biasanya berupa file template HTML, yang diatur oleh *controller*. *View* berfungsi untuk menerima dan merepresentasikan data hasil dari model dan *controller* kepada user. *View* tidak memiliki akses langsung terhadap bagian model.
3. *Controller*, merupakan bagian yang mengatur hubungan antara bagian model dan bagian *view*. Pada controller terdapat *class-class* dan fungsi-fungsi yang memproses permintaan dari *View* ke dalam struktur data di dalam model. *Controller* juga tidak boleh berisi kode untuk mengakses basis data Karena tugas megakses data telah diserahkan kepada model. Tugas *controller* adalah menyediakan berbagai variable yang akan ditampilkan di *view*, memanggil model untuk melakukan akses ke basis data, menyediakan penanganan kesalahn/error, mengerjakan proses logika dari aplikasi serta melakukan validasi atau cek terhadap input.

2.3.7 Framework

Framework atau dalam bahasa indonesia dapat diartikan sebagai "kerangka kerja" merupakan kumpulan dari fungsi-fungsi/prosedur-prosedur dan class-class untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah

dan mempercepat pekerjaan seorang programmer, tanpa harus membuat fungsi atau class dari awal.[16]

Alasan mengapa menggunakan Framework:

1. Mempercepat dan mempermudah pembangunan sebuah aplikasi web.
2. Relatif memudahkan dalam proses maintenance karena sudah ada pola tertentu dalam sebuah framework (dengan syarat programmer mengikuti pola standar yang ada).
3. Umumnya *framework* menyediakan fasilitas-fasilitas yang umum dipakai sehingga kita tidak perlu membangun dari awal (misalnya validasi, ORM, pagination, *multiple database*, *scaffolding*, pengaturan *session*, *error handling*, dll).
4. Lebih bebas dalam pengembangan jika dibandingkan CMS.

2.3.8 HTML

Proses tampilnya sebuah halaman website di browser melibatkan HTML. HyperText Markup Language (HTML) tergolong dalam salah satu format yang digunakan dalam pembuatan dokumen yang terbaca oleh web.

Menurut Solichin (2016:10) [10] mengemukakan bahwa “HTML merupakan bahasa pemrograman web yang memberitahukan peramban web (web browser) bagaimana menyusun dan menyajikan konten di halaman web”.

Berdasarkan teori dari para ahli di atas, maka hypertext markup language (HTML) merupakan bahasa pemrograman yang dikenal oleh browser untuk menampilkan informasi lebih menarik di halaman web melalui web browser.

2.3.9 CSS

Cascading style sheet (CSS) digunakan untuk menampilkan sebuah web dengan tampilan yang menarik, memperindah tampilan web dan mudah digunakan.

Menurut Sulistiyawan, dkk (2008:32) [11] mengemukakan bahwa “*cascading style sheet* adalah suatu bahasa stylesheet yang digunakan untuk mengatur style suatu dokumen. Pada umumnya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML”.

Maka dari itu, *cascading Style Sheet* (CSS) merupakan bahasa pemrograman web yang digunakan untuk mengatur konten dalam sebuah halaman web yang ditulis dalam bahasa markup agar halaman web tersebut lebih menarik dan terstruktur.

2.3.10 Bootstrap

Menurut Alatas (2013) [12] Bootstrap adalah *framework* atau *tools* yang digunakan untuk membuat aplikasi *web* ataupun situs *web responsive* secara cepat, mudah, dan gratis. Bootstrap terdiri dari CSS dan HTML untuk menghasilkan *grid, layout, typography, table, form, navigation*, dan lain-lain. Didalam bootstrap juga sudah terdapat *jquery plugins* untuk menghasilkan komponen *user interface* yang cantik seperti *transitions, modal, dropdown, scrollspy, tooltip, tab, popover, alert, button, carousel*, dan lain-lain. Kegunaan Bootstrap adalah membuat *responsive website* dengan cepat, mudah dan dapat berjalan pada *web browser* umum seperti Chrome, Firefox, Safari Opera, dan Internet Explorer.

2.3.11 Database

Sebuah sistem *database* dapat memiliki beberapa basis data. Setiap basis data berisi objek basis data seperti table, indeks, dan lain-lain. Selain berisikan data, *database* juga dapat menyimpan definisi struktur (baik untuk *database* maupun objek-objek secara rinci). Berikut merupakan jenis-jenis basis data:

a. MySQL

MySQL merupakan basis data yang paling populer dan banyak digunakan untuk aplikasi berbasis web seperti website dinamis dan e-commerce. Penggunaan MySQL sebagai basis data untuk aplikasi website seringkali dipadukan dengan PHP sebagai Bahasa skrip berorientasi objek. MySQL merupakan komponen penting dari web service solution stack LAMP, yaitu platform pengembangan web dimana Linux sebagai sistem operasi, Apache sebagai Web Server, MySQL sebagai basis data, dan PHP sebagai Bahasa skrip.

b. MariaDB

MariaDB merupakan sistem basis data relational *open source*. Kode yang digunakan oleh MariaDB dirilis di bawah lisensi GPL, LGPL, atau BSD. MariaDB dapat dikatakan adalah *fork* dari basis data MySQL. *Fork* yang dimaksud adalah sebagai proyek terkait yang dapat dianggap sebagai versi lain dari MySQL standar. MariaDB merupakan versi dari pengembangan MySQL.

c. Oracle

Oracle merupakan basis data yang dimiliki oleh Oracle Corporation. Oracle telah tersedia dalam berbagai konfigurasi dengan cakupan tool yang dapat digunakan untuk perusahaan skala kecil, menengah hingga besar.

2.3.12 Jaringan Komputer

Jaringan komputer adalah himpunan “interkoneksi” antara 2 komputer autonomous atau lebih yang terhubung dengan media transmisi kabel atau tanpa kabel (*wireless*). Bila sebuah komputer dapat membuat komputer lainnya restart, shutdown, atau melakukan kontrol lainnya, maka komputer-komputer tersebut bukan *autonomous* (tidak melakukan kontrol terhadap komputer lain dengan akses penuh). [13, p. 6]

2.3.12.1 Jaringan Komputer yang digunakan

Local Area Network (LAN)

Sebuah LAN adalah jaringan yang dibatasi oleh area yang relatif kecil, umumnya dibatasi oleh area lingkup, seperti sebuah kantor pada sebuah gedung, atau tiap-tiap ruangan pada sebuah sekolah. Biasanya jarak antarnode tidak lebih jauh dari sekitar 200m. [13]

2.3.12.2 Arsitektur Jaringan

Arsitektur jaringan salah satu aspek fisik dalam memetakan lokasi dari perangkat jaringan. Jenis perangkat yang akan digunakan pada design arsitektur seperti penempatan lokasi yang aman serta jenis server, kabel, lokasi keamanan fisik. Bagaimana perangkat jaringan saling terhubung, jenis antar muka dan kecepatan, serta spesifikasi perangkat yang digunakan (McCabe,2007). Seperti halnya dengan design konseptual arsitektur jaringan enterprise yang dibangun

biasanya sesuai dengan kebutuhan strategi distribusi data, aplikasi serta sharing data antara unit kerja dalam suatu organisasi. Saat ini model desain arsitektur perancangan dan manajemen jaringan merupakan salah satu bentuk dalam membuat panduan /dokumen sebagai pedoman dalam membangun jaringan kedepannya. Dengan dibuatnya rancangan dan pedoman infrastruktur jaringan dapat dijadikan sebagai panduan / dokumen pendukung bagi institusi dalam membangun dan mengembangkan suatu jaringan. Hierarchical network model atau model jaringan hirarki merupakan model desain arsitektur jaringan berbentuk hirarki. Model ini menyediakan cara pandang yang bervariasi mengenai sebuah jaringan (network), sehingga mempermudah kita dalam mendesain dan membangun jaringan yang terskala.[17]

1. Pengertian Arsitektur Jaringan

Arsitektur Jaringan dapat diartikan sebagai rancangan arus komunikasi media elektronik. Arsitektur jaringan merupakan sebuah himpunan layer (lapisan) dan protokol. Dimana layer bertujuan memberi layanan ke layer yang ada di atasnya.

2. Pembagian Dasar Arsitektur Jaringan

- a. LAN (*Local Area Network*)
- b. MAN (*Metropolitan Area Network*)
- c. WAN (*Wide Area Network*)

3. Pembagian Topologi Jaringan

Pembagian topologi secara garis besar terbagi tiga namun pada saat ini telah banyak terdapat kombinasi dari topologi. Tiga topologi Umum yaitu:

- a. Topologi Star
- b. Topologi Bus
- c. Topologi Ring

4. Arsitektur Terminalogi Jaringan

- a. CAN (*Campus Area Network*): Jaringan yang menghubungkan bangunan pada suatu kompleks.
- b. Intranet: Jaringan yang dimiliki oleh suatu tempat yang dapat diakses oleh pengguna sah atau yang diizinkan.
- c. Internet: Jaringan yang menghubungkan jutaan komputer
- d. MAN (*Metropolitan Area Network*): Jaringan yang dirancang untuk sebuah kota.
- e. SAN (*Storage Area Network*): Digunakan Untuk menghubungkan perangkat yang berkaitan seperti RAID penyimpan file server dan sistem tape.
- f. VLAN (*Virtual Local Area Network*): Jaringan yang memungkinkan komputer pada jaringan fisik yang terpisah seolah-olah terhubung jaringan yang sama.
- g. *Client-Server*: Jaringan sistem yang melayani sistem pemberi dan menerima.

- h. *Peer-to-peer*: Jaringan dimana semua komputer berlaku sama tidak ada *clien* dan server.

5. Pengertian Protokol Jaringan Komputer

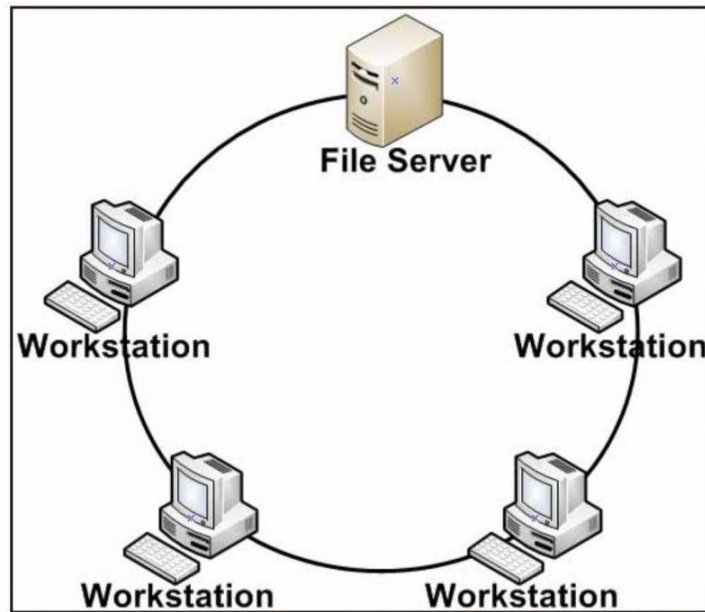
Menurut Wikipedia Indonesia, Protokol adalah sebuah aturan atau standar yang mengatur atau mengijinkan terjadinya hubungan, komunikasi, dan perpindahan data antara dua atau lebih titik komputer. Protokol dapat diterapkan pada perangkat keras, perangkat lunak atau kombinasi dari keduanya. Pada tingkatan yang terendah, protokol mendefinisikan koneksi perangkat keras. *Protocol* digunakan untuk menentukan jenis layanan yang akan dilakukan pada internet.

2.3.12.3 Topologi Jaringan

Topologi jaringan komputer adalah suatu cara menghubungkan komputer yang satu dengan komputer lainnya sehingga membentuk jaringan. Cara yang saat ini banyak digunakan adalah bus, token ring, dan star. Dalam suatu jaringan komputer jenis topologi yang dipilih akan mempengaruhi kecepatan komunikasi. Untuk itu maka perlu dicermati kelebihan/keuntungan dan kekurangan/kerugian dari masing - masing topologi berdasarkan karakteristiknya [20].

Berikut merupakan 4 macam contoh topologi jaringan:

1. Topologi jaringan ring



Gambar 2. 1 Topologi jaringan ring

Topologi ring atau topologi cincin adalah Topologi jaringan yang rangkaiannya berupa titik yang mana masing-masing titik bagian kanan dan kirinya terhubung ke dua titik lainnya sampai ke komputer yang pertama dan akhirnya membentuk cincin atau lingkaran.

Titik yang ada pada topologi cincin tersebut berfungsi memperkuat sinyal di setiap rangkaiannya atau sebagai repeater. Cara kerjanya yaitu: komputer 1 berencana mengirim file ke komputer 4? supaya terkirim maka file harus melewati komputer 2 dan 3? baru kemudian komputer 4 menerima file tersebut.

Dengan metode seperti itu, sinyal dan aliran data tetap akan stabil. Arah aliran datanya bisa searah jarum jam atau berlawanan dengan jarum jam, tergantung dengan kebutuhan.

a. Kelebihan Topologi Ring

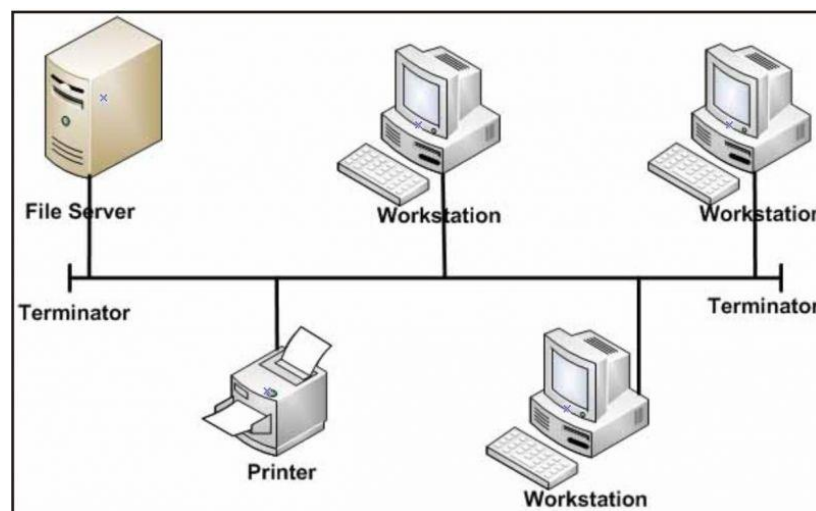
- 1) Mudah dalam perancangan dan pembuatan.

- 2) Performa sinyal dan aliran data stabil, bahkan lebih baik dari topologi bus, meskipun mengalirkan data yang berat.
- 3) Jika terjadi masalah, mudah untuk dilakukan konfigurasi ulang atau pemasangan baru.
- 4) Hemat dalam penggunaan kabel.
- 5) Hemat dalam biaya.

b. Kekurangan Topologi Ring :

- 1) Jika ada satu komputer yang eror, maka keseluruhan jaringan juga akan ikut eror, solusinya adalah dengan penggunaan topologi ring ganda
- 2) Performa aliran lalu lintas data bergantung pada jumlah komputer pada jaringan tersebut

2. Topologi bus



Gambar 2. 2 Topologi Bus

Topologi bus adalah jaringan yang hanya memakai satu kabel (*coaxial*) untuk media transmisi dan kabel tersebut sebagai pusat bagi seluruh server yang terhubung. Masing-masing komputer dihubungkan ke kabel utama dengan menggunakan konektor BNC, lalu diakhiri dengan terminator apabila Konektor BNC sudah terhubung dengan kabel *Coaxial*.

Kelebihan dan Kekurangan Jaringan Bus

a. Kelebihan:

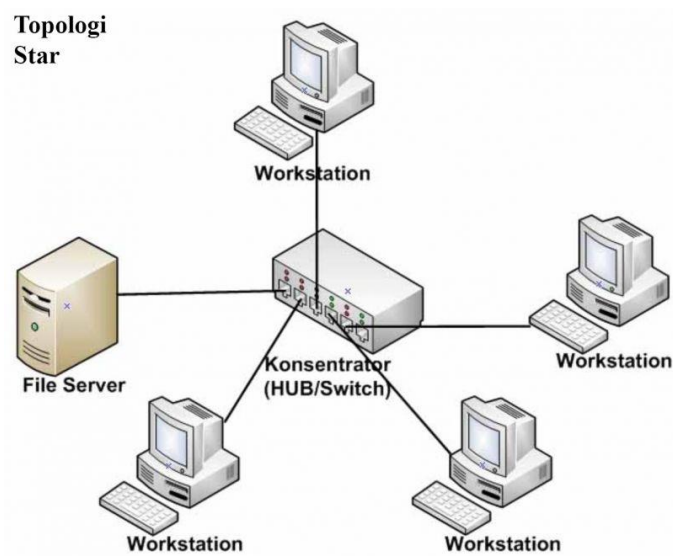
- 1) Lumayan sederhana.
- 2) Apabila ingin menambah server baru lumayan mudah, karena pemasangan tidak memutus keseluruhan jaringan sehingga tidak mengganggu server yang lain.
- 3) Hemat dalam penggunaan kabel dan biaya karena cuma menggunakan satu kabel utama.

b. Kekurangan:

- 1) Bila kabel utama mengalami gangguan, maka semua jaringan juga mengalami gangguan.
- 2) Jalur lalu lintas lumayan padat karena bolak-balik.
- 3) Apabila jarak jauh diperlukan repeater.
- 4) Susah melacak tempat trouble.

- 5) Jika komputer banyak yang terhubung maka performa sinyal dan lalu lintas akan menurun.

3. Topologi star



Gambar 2. 3 Topologi star

Topologi star merupakan bentuk jaringan yang mana terdapat satu penghubung (*Hub/Switch*) sebagai pusat dan setiap komputer terhubung ke penghubung tersebut. *Hub/Switch* ini posisinya ada di *central* dan berfungsi

untuk menghubungkan satu komputer ke setiap komputer yang terhubung dan juga menghubungkan komputer ke *File Server*.

Cara kerjanya yaitu apabila komputer berkirim data antara satu dengan yang lainnya maka data tersebut harus mengalir ke *Hub/Switch* terlebih dahulu baru kemudian menuju ke komputer yang dituju.

Kelebihan dan Kelemahan Topologi Star

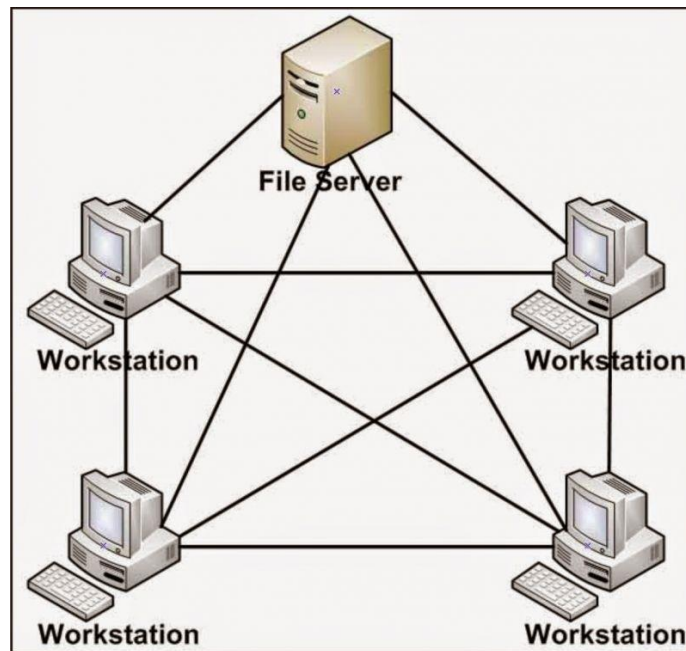
a. Kelebihan:

- 1) Tingkat keamanan tergolong tinggi.
- 2) Mudah dalam penambahan komputer baru yang ingin disambungkan.
- 3) Apabila ada yang eror mudah untuk dideteksi.
- 4) Paling fleksibel diantara topologi jaringan yang lainnya.
- 5) Kemudahan dalam mengontrol karena sistemnya yang terpusat.

b. Kekurangan:

- 1) Apabila *Hub/Switch* rusak maka keseluruhan jaringan juga rusak.
- 2) Lumayan boros dalam penggunaan kabel.
- 3) Hub sangat sensitif, karena dia sebagai tempat *central* bagi jaringan.
- 4) Tergantung spesifikasi Hubnya, apabila rendah maka performa sistem jaringan juga rendah.
- 5) Biaya lebih mahal.

4. Topologi Mesh



Gambar 2. 4 Topologi Mesh

Topologi Mesh atau mudahnya dinamai topologi jala adalah bentuk topologi jaringan yang mana semua *Workstation* bisa terhubung satu sama lain secara acak atau tidak teratur.

Karena *Workstation* langsung terhubung dengan *Workstation* yang dituju maka arus data bisa dilakukan dengan cepat tanpa harus melalui workstation lain.

Masing-masing *Workstation* setidaknya memiliki 2 jenis sambungan yaitu pertama kabel yang terhubung dengan workstation lain dan terakhir terhubung dengan File Server.

Topologi jaringan jala cocoknya digunakan pada jaringan komputer yang kecil bukan yang besar. Alasannya adalah selain karena sambungan antar *workstation* yang berlebihan tetapi juga sangat sulit untuk mengendalikannya.

Kelebihan dan Kekurangan Topologi Jaringan Mesh

a. Kelebihan:

- 1) Arus lalu lintas data cepat diantara topologi jaringan yang lain karena memiliki jalur masing masing.
- 2) Terjaminnya kapasitas channel komunikasi.
- 3) Jika ada gangguan pada satu koneksi jaringan, maka tidak mengganggu koneksi yang lain.
- 4) Terjaminnya keamanan dan privasi karena akses langsung dapat dilakukan antara dua komputer tanpa melibatkan komputer yang lain.
- 5) Kemudahan dalam mengidentifikasi titik trouble.

b. Kekurangan:

- 1) Biaya yang dikeluarkan termasuk tinggi.
- 2) Boros dalam pemakaian kabel karena semua komputer atau Workstation seminimal-minimalnya memiliki dua penghubung.
- 3) Diperlukan ruangan yang lumayan besar dalam membangun jaringan komputer tersebut.

2.3.13 White Box dan Black Box Testing

1. *White Box*

White Box Testing adalah salah satu cara untuk menguji suatu aplikasi atau software dengan melihat modul untuk memeriksa dan menganalisis kode program ada yang salah atau tidak. Jika modul ini dan telah diproduksi dalam *output* yang tidak memenuhi persyaratan, kode akan

dikompilasi ulang dan diperiksa lagi sampai mencapai apa yang diharapkan singkatnya *White Box Testing* ini menguji dengan cara melihat *Pure Code* dari suatu aplikasi/*software* yang diuji tanpa memperdulikan Tampilan atau UI dari aplikasi tersebut.

Teknik White-box Testing

a. *Basis Path Testing*

Basis path testing merupakan metode yang memungkinkan perancang testcase untuk membuat pengukuran kompleksitas logikal dari rancangan prosedural dan menggunakan pengukuran ini sebagai panduan untuk mendefinisikan himpunan basis dari jalur eksekusi. Test case yang dibuat untuk menguji himpunan basis dijamin akan mengeksekusi setiap statement di dalam program sekurangnya sekali pada saat pengujian.

b. *Flow Graph*

Flow graph merupakan notasi sederhana untuk merepresentasi control flow.

c. *Cyclomatic Complexity*

Cyclomatic complexity digunakan untuk mengetahui jumlah jalur yang perlu dicari. *Cyclomatic complexity* adalah metric software yang menyediakan ukuran kuantitatif dari kompleksitas logikal program. Nilai yang dihitung bagi *cyclomatic complexity* menentukan jumlah jalur-jalur yang independen dalam kumpulan basis suatu program dan memberikan jumlah tes minimal yang harus dilakukan

untuk memastikan bahwa semua pernyataan telah dieksekusi sekurang-kurangnya satu kali.

Cyclomatic complexity mempunyai fondasi dalam teori *graph* dan dapat dihitung dengan satu dari tiga cara:

- 1) Jumlah region sama dengan cyclomatic complexity.
- 2) *Cyclomatic complexity*, $V(G)$, untuk sebuah *flow graph*, G , didefinisikan sebagai: $V(G) = E - N + 2$ E adalah jumlah *edge* pada *flow graph*, dan N adalah jumlah node pada *flow graph*.
- 3) *Cyclomatic complexity*, $V(G)$, untuk *flow graph*, G , juga didefinisikan sebagai: $V(G) = P + 1$ P adalah jumlah *predicate nodes* yang terdapat pada *flow graph* G .

d. *Graph Matrix*

Prosedur untuk membuat *flow graph* dan menentukan himpunan *basis path* dapat diterima berdasarkan mekanisme. Untuk mengembangkan software yang membantu pengujian *basis path*, sebuah struktur data yang disebut *graph matrix*, dapat sangat bermanfaat. *Graph matrix* adalah matriks kotak yang ukurannya (jumlah baris dan kolom) sama untuk jumlah node pada *flow graph*. Setiap baris dan kolom berhubungan dengan node yang teridentifikasi, dan data matriks berhubungan dengan koneksi (*edge*).[18]

e. Kelemahan *White Box-Testing*:

- 1) Sangat mahal untuk dilakukan karena membutuhkan tester yang terampil untuk melakukan pengujian.

- 2) Pada perangkat lunak yang jenisnya besar, metode *white box testing* ini dianggap boros karena melibatkan banyak sumberdaya untuk melakukannya.
- 3) Tidak mempedulikan Tampilan UI aplikasinya.

2. *Black Box*

Pada *Black Box Testing* dilakukan pengujian yang didasarkan pada detail aplikasi seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi, dan kesesuaian alur fungsi dengan bisnis proses yang diinginkan oleh customer. *Black-box Testing* ini lebih menguji ke Tampilan Luar (*Interface*) dari suatu aplikasi agar mudah digunakan oleh *Customer*. Pengujian ini tidak melihat dan menguji *source code* program. *Black-box Testing* bekerja dengan mengabaikan struktur control sehingga perhatiannya hanya terfokus pada informasi domain.[19]

a. Keuntungan dari *Black-box Testing*:

- 1) Penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu.
- 2) Pengujian yang dilakukan berdasarkan sudut pandang user agar dapat mengungkapkan inkonsistensi atau ambiguitas dalam spesifikasi.
- 3) Programmer dan tester memiliki ketergantungan satu sama lain.

b. Kekurangan *Black-box Testing*:

- 1) Uji kasus sulit disain tanpa spesifikasi yang jelas.

- 2) Kemungkinan memiliki pengulangan tes yang sudah dilakukan oleh programmer.
- 3) Beberapa bagian back end tidak diuji sama sekali.

c. Teknik *Black-box Testing*

1) *Equivalence Partitioning*

Cara kerja teknik ini adalah dengan melakukan partition atau pembagian menjadi beberapa partisi dari input data.

2) *Boundary Value Analysis*

Teknik ini lebih fokus kepada *boundary*, dimana adakah *error* dari luar atau sisi dalam *software*, minimum, maupun *maximum* nilai dari *error* yang didapat.

3) *Fuzzing*

Fuzzing merupakan teknik untuk mencari *bug* / gangguan dari *software* dengan menggunakan injeksi data yang terbilang cacat ataupun sesi semi-otomatis.

4) *Cause-Effect Graph*

Ini adalah teknik testing dimana menggunakan *graphic* sebagai pacuannya. Dimana dalam grafik ini menggambarkan relasi diantara efek dan penyebab dari *error* tersebut.

5) *Orthogonal Array Testing*

Dapat digunakan jika input domain yang relatif terbilang kecil ukurannya, tetapi cukup berat untuk digunakan dalam skala besar.

6) *All Pair Testing*

Dalam teknik ini, semua pasangan dari *test case* di desain sedemikian rupa agar dapat di eksekusi semua kemungkinan kombinasi diskrit dari seluruh pasangan berdasar input parameteranya. Tujuannya testing ini adalah memiliki pasangan *test case* yang mencakup semua pasangan tersebut.

7) *State Transition*

Testing ini berguna untuk melakukan pengetesan terhadap kondisi dari mesin dan navigasi dari UI dalam bentuk grafik.