

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan. [1]

Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building block*) yaitu: [1]

1. Blok masukan (*input block*)

Input mewakili data yang masuk ke dalam sistem informasi. Input disini termasuk metode-metode dan media yang digunakan untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen dasar.

2. Blok model (*model block*)

Blok ini terdiri dari kombinasi prosedur, logika dan metode matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang sudah diinginkan.

3. Blok keluaran (*output block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok teknologi (*technology block*)

Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian diri secara keseluruhan. Teknologi terdiri dari unsur utama:

a. Teknisi (*human ware* atau *brain ware*)

b. Perangkat lunak (*software*)

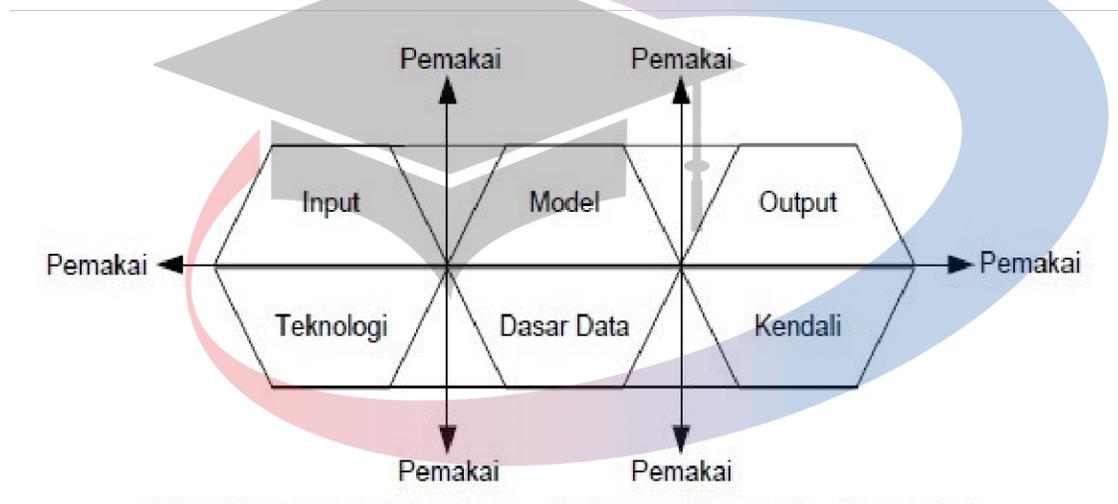
c. Perangkat keras (*hardware*)

5. Blok basis data (*data base block*)

Blok basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan diperangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

#### 6. Blok kendali (*control block*)

Banyak faktor yang dapat merusak sistem informasi, misalnya bencana alam, api, temperatur tinggi, air, debu, kecurangan-kecurangan, kejanggalkan sistem itu sendiri, kesalahan-kesalahan ketidakefisienan, sabotase dan sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah atau bila terlanjur terjadi kesalahan dapat diatasi.



Gambar 2.1 Blok Sistem Informasi

## 2.2 Perancangan Sistem Informasi

Perancangan sistem adalah tahapan dari siklus pengembangan sistem yang dapat didefinisikan sebagai tahap pendefinisian kebutuhan-kebutuhan fungsional dan menggambarkan bagaimana suatu sistem dibentuk. Penggambaran dapat berupa sketsa yang terdiri atas perancangan entitas (Entity Relationship Diagram) juga perancangan antarmuka. [2]

## 2.3 Analisis Sistem

Analisis sistem digunakan untuk menciptakan dan mengelola sistem dalam melakukan fungsi dasar bisnisnya. Terdapat banyak pendekatan untuk analisis sistem dan pada dasarnya semuanya mempunyai tujuan yang sama, yaitu memahami sistem yang rumit kemudian

melakukan modifikasi dengan beberapa cara. Hasil modifikasi dapat berupa subsistem baru, komponen baru atau serangkaian transformasi baru dan lain-lain. Tujuan utamanya adalah memperbaiki sistem organisasi melalui penerapan software yang dapat membantu karyawan mencapai tugas bisnis utamanya secara mudah dan efisien. Analisis sistem dapat terbagi dalam beberapa tahapan, yaitu analisis sistem berjalan, analisis masalah, dan analisis kebutuhan sistem. [3]

a. Analisis Sistem Berjalan.

Dalam tahapan ini, akan dilakukan tahapan analisis mengenai sistem yang sedang berjalan saat ini. Analisis dapat menggunakan *tools* seperti FOD atau DFD.

b. Analisis Masalah.

Analisis masalah adalah proses untuk mempelajari permasalahan-permasalahan sistem yang sudah diidentifikasi pada proses identifikasi permasalahan sistem. Pada proses ini sistem analisis dan tim pengembang akan mempelajari apakah solusi yang telah diidentifikasi benar-benar bisa memecahkan masalah atau apakah sistem yang baru layak untuk dikembangkan atau tidak. Tujuan dari aktivitas ini adalah untuk mempelajari dan memahami permasalahan sistem dan menentukan solusi yang akan digunakan serta menentukan apakah sistem baru layak untuk dibangun atau tidak.

c. Analisis Kebutuhan Sistem.

Analisis kebutuhan sistem menjelaskan kondisi atau kemampuan yang harus dipenuhi oleh sistem sesuai dengan spesifikasi yang diinginkan oleh pemakai. Analisis kebutuhan sistem terbagi 2, yaitu [2]:

1. Analisis Kebutuhan Fungsional.

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik serta sesuai dengan kebutuhan. Tools yang digunakan dalam analisis kebutuhan fungsional adalah DFD Fisik.

2. Analisis Kebutuhan Non-Fungsional.

Analisis kebutuhan non fungsional merupakan kebutuhan sistem yang menitikberatkan pada properti perilaku yang dimiliki oleh sistem, diantaranya kebutuhan perangkat lunak, perangkat keras, serta pengguna sistem (user) sebagai bahan analisis kekurangan dan kebutuhan yang harus dipenuhi dalam perancangan

sistem yang akan diterapkan. Tools yang digunakan dalam analisis kebutuhan non fungsional adalah PIECES.

## 2.4 Website

*Website* atau disingkat *web*, dapat diartikan sekumpulan halaman yang terdiri atas beberapa laman yang berisi informasi dalam bentuk data digital, baik berupa teks, gambar, video, audio dan animasi lainnya yang disediakan melalui jalur koneksi internet. Lebih jelasnya, *website* merupakan halaman-halaman yang berisi informasi yang ditampilkan oleh browser, seperti Mozilla Firefox, Google Chrome atau yang lainnya, sedangkan internet adalah jaringan yang digunakan untuk mengirim informasi pada *website* [2].

Dalam perkebangannya hingga era saat ini, teknologi berbasis *web* telah memiliki kurang lebih tiga *phase* perkembangan/generasi yakni dimulai dari *Web 1.0*, *Web 2.0*, *Web 3.0/web semantic* [2].

### 1. *Web 1.0*

Merupakan teknologi *Web* generasi pertama yang merupakan revolusi baru di dunia Internet karena telah mengubah cara kerja dunia industri dan media. Secara umum, *Web 1.0* dikembangkan untuk pengaksesan informasi dan memiliki sifat yang sedikit interaktif. Secara garis besar, sifat *Web 1.0* adalah *Read*.

### 2. *Web 2.0*

Istilah *Web 2.0* pertama kalinya diperkenalkan oleh O'Reilly Media pada tahun 2004 sebagai teknologi *Web* generasi kedua yang mengedepankan kolaborasi dan *sharing* informasi secara daring. Sifat *web 2.0* adalah *Read-Write*. Yang menjadi kunci perbedaan dalam *Web 2.0* dan *Web 1.0* adalah keterbatasan pada *Web 1.0* yang mengharuskan pengguna internet untuk datang ke dalam *website* tersebut dan melihat satu persatu konten di dalamnya. Sedangkan *Web 2.0* memungkinkan pengguna internet dapat melihat konten suatu *website* tanpa harus berkunjung ke alamat situs yang bersangkutan. Kemampuan *web 2.0* dalam melakukan aktivitas *drag and drop*, *auto complete*, *chat*, *voice* dan dapat dilakukan layaknya aplikasi desktop.

### 3. *Web 3.0 / Web Semantic*

*Web 3.0* adalah sekumpulan teknologi yang menawarkan cara baru yang efisien dalam membantu komputer mengorganisasi dan menarik kesimpulan dari data daring. *Web 3.0*

menawarkan metode yang efisien dalam membantu komputer mengorganisasi dan menarik kesimpulan dari data daring. Konsep ini dapat diandaikan sebuah *website* sebagai sebuah intelektualitas buatan (*Artificial Intelligence*).

## 2.5 Komponen Dasar Website

Di dalam sebuah website terdapat 3 komponen utama yang menjadi dasar terbentuknya sebuah website tersebut, diantaranya adalah HTML, CSS, dan Javascript [3].

### 2.5.1 HTML

HTML singkatan dari *Hyper Text Markup Language*, yaitu skrip yang berupa tag-tag untuk membuat dan mengatur struktur *website* [1]. Disebut *hypertext* karena didalam HTML sebuah *text* biasa dapat berfungsi lain, kita dapat membuatnya menjadi *link* yang dapat berpindah dari satu halaman ke halaman lainnya hanya dengan meng-klik *text* tersebut. Kemampuan *text* inilah yang dinamakan *Hyper Text*, walaupun pada implementasinya nanti tidak hanya *text* yang dapat dijadikan *link*. Elemen HTML termasuk *heading*, paragraf, *hyperlink*, daftar, gambar, dan banyak lagi. Sintaks elemen HTML adalah sebagai berikut: [3]

1. Elemen HTML dimulai dengan *tag* pembuka / *tag* pembuka.
2. Elemen HTML diakhiri dengan *tag- tag* / penutup akhir.
3. Konten elemen adalah semua yang disisipkan di antara *tag* awal dan akhir.
4. Beberapa konten HTML berisi konten kosong (misalkan: `<br />` atau `<hr />`).
5. Elemen kosong ditutup di *tag* awal (gunakan spasi- / untuk menutup seperti pada `<br>`).
6. Sebagian besar elemen HTML dapat memiliki atribut.

Disebut *Markup Language* karena bahasa HTML menggunakan tanda (*mark*), untuk menandai bagian-bagian dari *text*. Misalnya, *text* yang berada di antara tanda tertentu akan menjadi tebal, dan jika berada di antara tanda lainnya akan tampak besar. Tanda ini kenal sebagai HTML tag [2]. Beberapa tugas utama HTML dalam membangun *website*, diantaranya sebagai berikut [1]:

1. Menentukan *layout website*.
2. Memformat teks dasar, seperti pengaturan paragraf dan format font.
3. Membuat *list*.
4. Membuat tabel.

5. Menyisipkan gambar, video, dan audio.
6. Membuat *link*.
7. Membuat formulir.

## 2.5.2 CSS

CSS singkatan dari *Cascading Style Sheets*, yaitu skrip yang digunakan untuk mengatur desain *website*. Walaupun HTML mempunyai kemampuan untuk mengatur tampilan *website*, namun kemampuannya sangat terbatas [1]. CSS memungkinkan untuk menentukan gaya untuk berbagai elemen halaman web. Gaya adalah aturan yang menentukan tampilan elemen halaman web. *Style sheet* adalah serangkaian aturan yang mendefinisikan gaya untuk halaman web atau seluruh situs web dengan *style sheet*, dapat mengubah tampilan halaman web atau halaman dengan mengubah karakteristik seperti *font family*, ukuran *font*, *margin*, dan spesifikasi tautan, serta elemen visual seperti warna dan batas. CSS tidak digunakan untuk menambahkan konten apapun ke situs web anda; itu hanya membuat konten terlihat lebih bergaya. [3]

Fungsi utama dari css ialah mendesain, merancang, merubah, dan juga membentuk halaman pada *website* ataupun *blog* dan isi dari halaman *website* itu ialah tag-tag html, logikanya css itu bisa mengubah tag-tag html yang sederhana sehingga bisa menjadi lebih fungsional dan juga menarik [4]. CSS dapat menentukan gaya untuk elemen dalam satu halaman web atau di seluruh situs web secara keseluruhan, misalnya, jika anda ingin semua paragraf teks pada halaman web harus diindentasi oleh lima spasi, dapat menggunakan lembar gaya untuk menangani indentasi, daripada mengkodekan setiap paragraf dengan sebuah indentasi. Jika memutuskan ingin mengubah indentasi menjadi tiga spasi, akan mengubah hanya satu garis gaya *sheet* daripada mengubah pengkodean untuk setiap paragraph, sehingga dapat melihat bahwa menggunakan CSS menghemat banyak waktu dan membuatnya lebih mudah untuk membuat perubahan gaya. [3]

CSS bukan HTML; ini adalah bahasa terpisah yang digunakan untuk meningkatkan kemampuan tampilan HTML. *World Wide Web* konsorsium, organisasi yang sama yang mendefinisikan standar HTML, mendefinisikan spesifikasi untuk CSS. [3]

### 2.5.3 Javascript

JavaScript merupakan bahasa *script* populer yang dipakai untuk menciptakan halaman web yang dapat berinteraksi dengan pengguna dan dapat merespon *event* yang terjadi pada halaman. JavaScript merupakan perekat yang menyatukan halaman-halaman web. JavaScript, awalnya dikenal sebagai *LiveScript*, dikembangkan oleh Brendan Eich di Netscape Navigator 2.0. JavaScript merupakan bahasa *script* yang menghidupkan halaman-halaman HTML. JavaScript dapat dijalankan di hampir semua *platform*. JavaScript merupakan bahasa sisi-klien yang didesain pada *browser computer*, bukan pada *server*. Ia dibangun secara langsung ke dalam *browser*, Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, Opera, hampir semua browser. [5]

Berbeda dengan php yang diproses di sisi *server*, javascript diproses pada komputer klien. Karena pemrosesannya dilakukan di komputer klien, membuat javascript lebih interaktif dibanding php. Peran javascript dalam membuat *website* adalah memberikan efek animasi yang menarik dan interaktivitas dalam penanganan *event* yang dilakukan oleh pengguna *website* [1].

Program JavaScript dipakai untuk mendeteksi dan beraksi terhadap *event-event* yang disebabkan oleh pengguna. JavaScript dapat digunakan untuk mengendalikan tampilan halaman. JavaScript dipakai untuk memvalidasi apa yang diketikkan pengguna ke dalam sebuah *form* sebelum pengiriman *form* ke *server* dilakukan. JavaScript berperan sebagai bahasa pemrograman, yang memiliki konstruksi-konstruksi dasar seperti variabel dan tipe data, *loop* kontrol, statemen *if else*, statemen *switch*, fungsi, dan objek. JavaScript dapat dipakai untuk perhitungan aritmatika, manipulasi tanggal dan waktu, pemodifikasian *array*, *string* dan objek. Ia juga dapat menangani *event* yang diinisiasi pengguna, dan menetapkan pewaktu. Kombinasi dari HTML, CSS, dan JavaScript dapat menghasilkan halaman web yang terstruktur, bergaya, dan interaktif. Para perancang web sepakat bahwa ketiganya merupakan bagian penting dari sebuah halaman web. [5]

### 2.6 Laravel

Laravel adalah sebuah PHP *Framework* yang dirancang untuk pengembang yang membutuhkan sebuah *toolkit* yang sederhana dan elegan untuk menciptakan aplikasi *web* berfitur lengkap. Laravel diciptakan oleh Taylor Otwell pada tahun 2011 [6]. Laravel dirilis dibawah lisensi MIT dengan kode sumber yang sudah disediakan oleh Github, sama seperti *framework-*

*framework* yang lain, Laravel dibangun dengan konsep MVC (*Model-View-Controller*), kemudian Laravel dilengkapi juga *command line tool* yang bernama "Artisan" yang bisa digunakan untuk *packaging bundle* dan instalasi *bundle* melalui *command prompt* [7].

Laravel merupakan *software open source* yang memiliki fitur-fitur untuk membantu pengembangan *web*. Selain itu, *website* yang dibangun dengan menggunakan Laravel ini juga aman dari berbagai serangan yang terjadi pada situs *web*. Lalu, *script code* yang dibuat pada Laravel ini juga sangat rapih, karena setiap *script code* yang dibuat akan dipisah sesuai dengan fungsinya masing-masing, seperti *script code* untuk *front end* akan ditempatkan pada *blade engine template* dan untuk *back end* akan ditempatkan pada *controller* [6]. Beberapa fitur yang dimiliki oleh *framework* Laravel adalah sebagai berikut: [7]

- a. *Bundles* yaitu sebuah fitur dengan sistem pengemasan modular dan berbagai *bundle* telah tersedia untuk digunakan dalam aplikasi Anda.
- b. *Eloquent ORM* merupakan penerapan PHP lanjutan dari pola "*active record*" menyediakan metode internal untuk mengatasi kendala hubungan antara objek *database*. Pembangunan *query* Laravel Fluent didukung *Eloquent*.
- c. *Application Logic* merupakan bagian dari aplikasi yang dikembangkan baik menggunakan *Controllers* maupun sebagai bagian dari deklarasi *Route*. Sintaks yang digunakan untuk mendefinisikannya mirip dengan yang digunakan oleh *framework* Sinatra.
- d. *Reverse Routing*, mendefinisikan hubungan antara *Link* dan *Route*, sehingga jika suatu saat ada perubahan pada *route* secara otomatis akan tersambung dengan *link* yang relevan. Ketika *Link* yang dibuat dengan menggunakan nama-nama dari *Route* yang ada, secara otomatis Laravel akan membuat URI yang sesuai.
- e. *Restful Controllers*, memberikan sebuah *option* (pilihan) untuk memisahkan logika dalam melayani HTTP GET dan permintaan POST.
- f. *Class Auto Loading*, menyediakan otomatis *loading* untuk *class-class* PHP, tanpa membutuhkan pemeriksaan manual terhadap jalur masuknya. Fitur ini mencegah *loading* yang tidak perlu.
- g. *View Composers* adalah kode unit *logical* yang dapat dijalankan ketika sebuah *View* di *load*.
- h. *IoC Container* memungkinkan untuk objek baru yang dihasilkan dengan mengikuti prinsip control pembalik, dengan pilihan contoh dan referensi dari objek baru sebagai Singletons.

- i. *Migrations* menyediakan versi sistem *control* untuk skema *database*, sehingga memungkinkan untuk menghubungkan perubahan pada basis kode aplikasi dan keperluan yang dibutuhkan dalam merubah tata letak *database*, mempermudah dalam penempatan dan memperbarui aplikasi.
- j. Unit Testing, mempunyai peran penting dalam *framework* Laravel, dimana unit testing ini mempunyai banyak tes untuk mendeteksi dan mencegah regresi. Unit testing dapat dijalankan melalui fitur "*artisan command-line*".
- k. *Automatic pagination* menyederhanakan tugas dari penerapan halaman, menggantikan penerapan yang manual dengan metode otomatis yang terintegrasi ke Laravel.

## 2.7 Apache dan Phpmyadmin

Apache berfungsi sebagai *web server*, yaitu tempat menyimpan file-file php dan file-file lainnya yang diperlukan *website*, sedangkan Phpmyadmin merupakan aplikasi berbasis *web* yang digunakan untuk membuat *database* MySQL sebagai tempat untuk menyimpan data-data *website*. Keduanya biasanya sudah disediakan dalam satu paket aplikasi seperti Appserv atau Xampp.

Xampp adalah salah satu paket installer yang berisi Apache yang merupakan *web server* tempat menyimpan file-file yang diperlukan *website*, dan Phpmyadmin sebagai aplikasi yang digunakan untuk perancangan *database* MySQL [8].

## 2.8 Emas

Emas dalam bahasa Inggris disebut "gold". Kata ini merupakan serapan dari bahasa Jerman Kuno, yaitu "ghel" yang berarti bersinar atau 10 kuning. Secara kimiawi, emas adalah unsur kimia dengan simbol Au, dari bahasa latin "Aurum" yang artinya cahaya fajar. Emas mempunyai nomor atom 79. Kode ISO-nya adalah XAU. Emas telah digali dan dicari untuk dijadikan koin, perhiasan, dan benda seni lain sejak awal sejarah manusia. Emas murni memiliki warna kuning cerah dan berkilau sehingga dianggap sangat menarik. [9]

Semakin tinggi kadar emas, semakin mahal pula harga perhiasan tersebut, begitu juga sebaliknya. Sebagai logam mulia yang sejak dulu dianggap sangat bernilai karena jumlahnya yang terbatas dan memiliki tampilan unik berkilau. Semakin banyak kandungan emas, semakin berkilau perhiasan tersebut. Tak berlebihan jika emas ialah jenis logam mulia yang paling

populer di masyarakat. Bahkan sering kali, istilah logam mulia diidentikkan dengan emas. Waktu telah membuktikan emas merupakan jenis logam mulia yang sudah sangat lama dimanfaatkan baik sebagai bahan baku perhiasan hingga alat tukar dan investasi. Selain itu, emas juga dimanfaatkan untuk bahan kosmetik, bahan makanan hingga bahan baku merakit komponen elektronik. Di antara banyak penggunaannya, emas perhiasan dan emas batangan atau juga kerap disebut logam mulia, merupakan yang paling dikenal masyarakat. Emas juga dianggap salah satu aset *save haven* karena mampu menangkal inflasi. [9]

Emas merupakan salah satu bentuk investasi yang cenderung bebas risiko. Dikutip dari website [www.orori.com](http://www.orori.com) terdapat beberapa kelebihan emas adalah: [10]

1. Keamanan

Investasi emas aman dari biaya administrasi, pajak, dan lain-lain. Jika menyimpan uang di bank, maka akan dikenai biaya administrasi. Sementara itu, jika menyimpan di lembaga investasi lain, kita harus membayar biaya perantara (broker), biaya administrasi, pajak, dan sebagainya.

2. Perlindungan

Investasi emas dapat melindungi aset dari inflasi dan deflasi. Sementara itu, jika menyimpan aset dalam bentuk investasi lain, kemungkinan terkena dampak inflasi sangat besar.

3. Mudah dicairkan (likuid)

Saat membutuhkan uang, investor pasti akan mencairkan sebagian investasi yang dimiliki. Namun, tidak semua jenis investasi dapat dicairkan dengan mudah. Diperlukan waktu lebih lama untuk mencairkan investasi dalam bentuk properti, kendaraan, barang seni, dan sebagainya. Berbeda dengan emas yang dapat segera dijual kapan saja dengan mudah dan cepat.

4. Tahan Lama

Berbagai jenis investasi seperti surat berharga, kendaraan, properti, dan karya seni akan hilang nilainya apabila terbakar, terendam air, atau terkena bencana alam. Sementara itu, walaupun terbakar, terendam air, atau terkena bencana alam, nilai emas masih tetap seperti semula. Karena itu, emas disebut-sebut sebagai logam yang tahan segala cuaca.

5. Mudah Dipindahkan (portabel)

Emas merupakan benda bernilai tinggi yang mudah dibawa dan dipindahkan. Emas 1kg dapat dibawa dengan memasukkannya ke dalam saku jaket sebab besarnya hanya seukuran bungkus rokok sehingga tidak mengundang perhatian orang yang berniat jahat.

## 2.9 Metode *Prototype*

*Prototype* adalah suatu proses yang memungkinkan *developer* membuat sebuah model *software*, metode ini baik digunakan apabila *client* tidak bisa memberikan informasi yang maksimal mengenai kebutuhan yang diinginkan. Secara ideal *prototype* berfungsi sebagai sebuah mekanisme untuk mengidentifikasi kebutuhan *software*, bila *prototype* yang sedang bekerja dibangun pengembangannya harus menggunakan fragmen-fragmen program yang ada atau mengaplikasikan alat-alat bantu (contohnya: *report generator*, *window manager* dll) dimana memungkinkan program yang bekerja untuk dimunculkan secara tepat. *Prototyping* model sendiri mempunyai tujuan yaitu mengembangkan model awal *software* menjadi sebuah sistem yang final. *Prototype* dibedakan menjadi 2, yaitu: [11]

1. *Paper Prototype*, menggambarkan interaksi manusia dan mesin dalam sebuah bentuk yang memungkinkan *user* mengerti bagaimana interaksi itu terjadi.
2. *Working Prototype*, adalah *prototype* yang mengimplementasikan beberapa bagian dari fungsi *software* yang diinginkan seperti pada pendekatan pengembangan *software* yang diinginkan seperti pada pendekatan pengembangan *software*. Model ini dimulai dengan:
  - a. Pengumpulan kebutuhan *developer* dan *customer*
  - b. Menentukan semua tujuan *software*

Berikut ini adalah tahapan-tahapan dari *prototype*: [11]

1. Pengumpulan Kebutuhan  
*Developer* dan klien akan bertemu terlebih dahulu dan kemudian menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya.
2. Perancangan  
Perancangan dilakukan dengan cepat dan rancangan tersebut mewakili semua aspek *software* yang diketahui, dan rancangan ini menjadi dasar pembuatan *prototype*.
3. Menentukan Apakah *Prototype* Dapat Diterima  
Pengembang mendemonstrasikan prototipe kepada pengguna untuk mengetahui apakah telah memberikan hasil yang memuaskan. Jika ya, langkah 4 akan diambil. Jika tidak, prototipe

direvisi dengan mengulang kembali langkah 1, 2, dan 3 dengan pemahaman yang lebih baik mengenai kebutuhan pengguna.

#### 4. Mengkodekan Sistem

Dalam tahap ini *prototyping* yang sudah di sepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.

#### 5. Evaluasi Sistem

Pelanggan mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan. Jika ya, sistem dapat digunakan dan jika tidak, ulangi langkah 2 dan 3.

### 2.9.1 Jenis-Jenis Prototipe

Kata prototipe banyak digunakan untuk berbagai macam hal. Daripada berupaya menyatukan semua kegunaan ini kedalam satu definisi atau berusaha memerintahkan satu pendekatan yang benar terhadap suatu topik *prototyping* yang kontroversial, penulis menggambarkan bagaimana masing-masing dari beberapa konsepsi *prototyping* sangat berguna bila diterapkandalam suatu situasi tertentu [12].

#### 1. Prototipe *Patched-up*

Jenis *prototyping* yang pertama ini berkaitan dengan penyusunan sistem yang bekerja namun *patch* atau *patch*-bersama-sama. Contohnya dalam sistem informasi adalah model kerja yang memiliki semua fitur yang diperlukan namun tidak efisien. Dalam hal *prototyping* ini, pengguna dapat berinteraksi dengan sistem sesuai dengan antarmukanya serta jenis keluaran yang tersedia. Pencarian dan penyimpanan informasi bisa menjadi sangat tidak efisien, karena program-programnya ditulis secara cepat dengan tujuan agar bisa berfungsi, atau lebih dari sekedar efisien. Contoh lain untuk prototipe *patch-up* adalah suatu sistem informasi yang memiliki semua prototipe yang diajukan, tetapi menjadi model dasar yang sebenarnya akan ditingkatkan.

#### 2. Prototipe Non-operasional

Konsepsi *prototype* kedua adalah prototipe dari model skala nganggur yang disusun untuk menguji beberapa rancangan tertentu. Satu model skala nganggur dari suatu sistem informasi bisa diproduksi bila pengkodean yang diperlukan oleh aplikasi terlalu mahal untuk prototipe kecuali bila ada gagasan yang dianggap sangat bermanfaat dari sistem bisa diperoleh melalui *prototyping* masukan dan keluaran saja. Mengenai pemrosesan ini, karena

masalah biaya dan waktu, tidak akan diprototipekan. Beberapa keputusan mengenai kegunaan sistem masih dapat dibuat berdasarkan masukan dan keluaran yang diprototipekan.

### 3. Prototipe *First-of-Series*

Konsepsi *prototyping* ketiga melibatkan penciptaan suatu model skala lengkap pertama dari sistem, yang disebut pilot. Jenis *prototyping* semacam ini digunakan ketika merencanakan banyak versi dari sistem informasi yang sama. Model kerja skala lengkap ini memungkinkan pengguna bisa mendapat pengalaman interaksi yang realistis dengan sistem baru, yang sebelumnya meminimalkan biaya untuk menanggulangi masalah-masalah yang ada. Contohnya ditemukan dalam instalasi perbankan untuk pentransferan dana secara elektronik. Pertama-tama, dipasang sebuah prototipe berskala lengkap di salah satu dari dua lokasi yang ada, dan apabila berhasil, dipasang duplikat-duplikatnya di semua lokasi berdasarkan pola-pola yang digunakan konsumen serta faktor-faktor penting lainnya.

### 4. Prototipe Fitur-fitur Terpilih

Konsepsi *prototyping* keempat berkaitan dengan pembangunan suatu model operasional yang mencakup beberapa, tapi tidak semua, fitur-fitur yang dimiliki sistem final. Analoginya, ibarat sebuah pusat perbelanjaan retail baru yang akan dibuka sebelum pembangunan seluruh bagian toko selesai. Pada pusat retail yang baru dibangun tersebut, fungsi utamanya ialah agar orang bisa membeli beberapa barang tertentu. Bila melakukan *prototyping* sistem informasi dengan cara ini, meski tidak semua, fitur-fitur intinya harus dimasukkan. Bila *prototyping* jenis ini dilakukan, sistem akan bisa tercapai sesuai modul sehingga bila fitur-fitur yang dibuat prototipenya dianggap berhasil, maka akan dipadukan ke dalam sistem final yang besar tanpa melakukan pekerjaan yang besar dalam antarmuka.

## 2.9.2 Kelebihan dan Kelemahan *Prototype*

Adapun kelebihan dari *prototype* adalah sebagai berikut: [12]

1. Komunikasi akan terjalin baik antara pengembang dan pelanggan.
2. Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan setiap pelanggannya.
3. Pelanggan berperan aktif dalam proses pengembangan sistem.
4. Lebih menghemat waktu dalam pengembangan sistem.

5. Penerapan menjadi lebih mudah karena pemakai mengetahui apa yang diharapkannya.

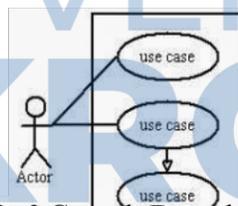
Adapun kelemahan dari *prototype* adalah sebagai berikut: [13]

1. Pelanggan kadang tidak melihat atau menyadari bahwa perangkat lunak yang ada belum mencantumkan kualitas perangkat lunak secara keseluruhan dan juga belum memikirkan kemampuan pemeliharaan untuk jangka waktu lama.
2. Pengembang biasanya ingin cepat menyelesaikan proyek sehingga menggunakan algoritma dan bahasa pemrograman yang sederhana untuk membuat *prototyping* lebih cepat selesai tanpa memikirkan lebih lanjut bahwa program tersebut hanya merupakan sebuah kerangka kerja (*blueprint*) dari sistem.
3. Hubungan pelanggan dengan computer yang disediakan mungkin tidak mencerminkan teknik perancangan yang baik dan benar.

## 2.10 Teknik Pengembangan Sistem

### 2.10.1 Use Case

*Use case* diagram adalah diagram yang membantu dalam menyusun requirement sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua fitur yang ada pada sistem. *Use case* diagram menggunakan fungsionalitas yang diharapkan dari sebuah sistem. *Use case* diagram menekankan apa yang dibuat sistem, dan bukan bagaimana. Sebuah *use case* merepresentasikan sebuah interaksi antara *actor* dengan sistem [13].

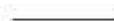


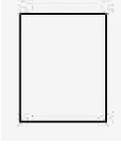
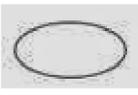
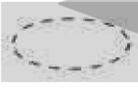
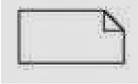
Gambar 2. 2 Contoh Bentuk Diagram Usecase

Berikut adalah notasi-notasi yang ada pada *use case* diagram: [13]

Tabel 2. 1 Simbol dan Notasi Use Case

NO	GAMBAR	NAMA	KETERANGAN

1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
3		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use cases</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.

			
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Untuk setiap *use case*, deskripsi *use case* dikembangkan dalam bentuk tabel. Deskripsi kasus penggunaan mendokumentasikan *name of use case*, *actor*, *description*, *successful completion*, *alternative*, *precondition*, *postcondition*, *assumptions* [16].

<b>Name of Use Case:</b>	Credit card validation process
<b>Actor:</b>	Customer
<b>Description:</b>	Describes the credit card validation process
<b>Successful Completion:</b>	<ol style="list-style-type: none"> <li>1. Customer clicks the input selector and enters credit card number and expiration date</li> <li>2. System verifies card</li> <li>3. System sends authorization message</li> </ol>
<b>Alternative:</b>	<ol style="list-style-type: none"> <li>1. Customer clicks the input selector and enters credit card number and expiration date</li> <li>2. System rejects card</li> <li>3. System sends rejection message</li> </ol>
<b>Precondition:</b>	Customer has selected at least one item and has proceeded to checkout area
<b>Postcondition:</b>	Credit card information has been validated Customer can continue with order
<b>Assumptions:</b>	None

Gambar 2. 3 Contoh Deskripsi Use Case

## 2.10.2 Basis Data

Sebuah basis data adalah sebuah kumpulan data yang saling berhubungan secara logis, dan merupakan sebuah penjelasan dari data tersebut, yang di desain untuk menemukan data yang diutuhkan oleh sebuah organisasi. Di dalam basis data, semua data diintegrasikan dengan menghindari duplikasi data. Basis data dapat digunakan oleh banyak department dan pemakai. Basis data tidak hanya memegang data operasional organisasi, tetapi juga penjelasan mengenai data tersebut. Karena alasan tersebut basis data juga dideskripsikan sebagai kumpulan data yang saling terintegrasi. Basis data juga merupakan sekumpulan elemen data terintegrasi yang secara logika saling berhubungan. Basis data mengonsolidasikan berbagai catatan yang terlebih dahulu disimpan dalam file-file terpisah ke dalam satu gabungan umum elemen data yang menyediakan data untuk banyak aplikasi. Elemen data mendeskripsikan entitas-entitas dan hubungan antara entitas-entitas tersebut. [14]

Desain basis data adalah proses membuat desain yang akan mendukung operasional dan tujuan perusahaan. Tujuan desain basis data adalah: [14]

1. Menggambarkan relasi data antara data yang dibutuhkan oleh aplikasi dan *user view*.
2. Menyediakan model data yang mendukung seluruh transaksi yang diperlukan.
3. Menspesifikasikan desain dengan struktur yang sesuai dengan kebutuhan sistem.

Ada beberapa pendekatan yang dapat digunakan dalam mendesain basis data, yaitu: [14]

1. *Top-down*

Diawali dengan membuat data model. Pendekatan *Top-down* dapat diilustrasikan menggunakan *entity-relationship* (ER) model yang *high level*, lalu mengidentifikasi *entity*, dan *relationship* antar-*entity* organisasi. Pendekatan ini sesuai bagi basis data yang kompleks.

2. *Bottom-up*

Dimulai dari level dasar *attribute* (*property entity* dan *relationship*), menganalisa hubungan antar-*attribute*, mengelompokkannya dalam suatu relasi yang menggambarkan tipe *entity* dan relasi antara *entity*. Pendekatan ini sesuai bagi basis data dengan jumlah *attribute* yang sedikit.

3. *Inside-out*

Mirip seperti pendekatan *Bottom-up*, perbedaannya adalah tahap awal mengidentifikasi *major entity* lalu menguraikannya menjadi *entity* relasi-relasi dan *attribute-attribute* yang berhubungan dengan *major entity*.

4. *Mixed*

Menggunakan pendekatan *Bottom-up* dan *Top-down*.

### 2.10.3 MySQL

MySQL merupakan sebuah *server database* SQL atau DBMS yang *multiuser*, *multithread* yang bersifat *open source*. Sifatnya yang *open source* membuat MySQL dapat dipergunakan dan didistribusikan untuk kepentingan individu maupun *corporate* tanpa memerlukan lisensi dari pembuatnya.

MySQL dapat dijalankan dalam berbagai platform system operasi antara lain windows, Linux, Unix, Sun OS, dan lain-lainnya. MySQL dikembangkan oleh Michael Widenius yang disebut Monthy dengan berlandaskan pada aplikasi Unired dan Rutim Isam yang dibuatnya sendiri. MySQL sangat populer, utamanya untuk aplikasi web. MySQL merupakan komponen database "M" yang ada pada LAMP, WAMP, MAMP, XAMP, BAMP, dan digunakan ditool *bugtracking* seperti bugzilla: [8]

MySQL banyak dipakai untuk kepentingan penanganan *database* karena selain handal juga bersifat *open source*. Konsekuensi dari *open source*, perangkat lunak ini dapat dipakai oleh siapa

saja tanpa membayar dan *source code*-nya bisa diunduh oleh siapa saja. *Database* menyatakan suatu wadah untuk mengelola data. *Database* mencakup sejumlah tabel dan berbagai objek yang terkait dengan pengelolaan data. [15]

## 2.11 PHP

PHP singkatan dari *Hypertext Preprocessor* yang merupakan *server-side programming*, yaitu bahasa pemrograman yang diproses di sisi *server*. Fungsi utama PHP dalam membangun *website* adalah untuk melakukan pengolahan data pada *database*. Data *website* akan dimasukkan ke *database*, diedit, dihapus dan ditampilkan pada *website* yang diatur oleh PHP [1].

Dibandingkan bahasa pemrograman lainnya PHP memiliki kelebihan lainnya. Hal itu terbukti dari adanya standar yang menjadikan PHP menjadi dasar bagi programmer *web* diseluruh dunia. Dimana *market share* penggunaan bahasa pemrograman *server side* untuk mayoritas *website* diseluruh dunia dipegang PHP dibandingkan bahasa pemrograman lainnya.

Salah satu kelebihan PHP adalah tidak melakukan kompilasi dalam penggunaannya. Selain itu PHP juga dapat mudah ditemukan mulai dari apache, IIS, Lighttpd hingga Xitami dengan konfigurasi yang relatif mudah. Kelebihan lainnya dari PHP adalah pengembangan yang lebih mudah. Hal itu dikarenakan terdapat banyak milis-milis dan *developer* yang membantu dalam proses pengembangan. Selanjutnya, PHP sebagai bahasa *open source* dapat digunakan dalam berbagai mesin baik Linux, Windows, Unix/ Macintosh serta dapat dijalankan secara *runtime* melalui *console* [16].

UNIVERSITAS  
MIKROSKIL