

BAB II LANDASAN TEORI

2.1. Penelitian Terdahulu

Penelitian yang dilakukan oleh penulis tidak terlepas dari penelitian terdahulu yang pernah dilakukan, pada objek penelitian Cantigi Camp sendiri khususnya pada proses pemesanan belum ada penelitian atau perancangan sistem yang diusulkan sebelumnya sehingga membuat penulis tertarik untuk melakukan penelitian ini. Berikut beberapa Penelitian terdahulu yang berhubungan dengan topik peneliti dijadikan sebagai bahan referensi.

Berdasarkan penelitian yang sudah dilakukan Mia Fitriawati, S.Kom, M.Kom dan Damaiyanti Ema Novelina Sihombing dengan judul Sistem Informasi Pemesanan Wisata dan ekspedisi berbasis Web pada Kampung Batu Malakasari. Penelitian ini bertujuan untuk memberikan usulan kepada tempat wisata Kampung Batu Malakasari berupa prancang bangun sitem informasi pemesanan wisata berbasis web. Metode pengembangan sistem yang digunakan yaitu metode terstruktur dengan alat bantu flowmap, DFD. Perancangan dan pembangunan sistem informasi membantu memberikan manfaat yaitu menarik perhatian yang mampu memberikan keuntungan bagi perusahaan.. Persamaan dengan penelitian yang akan dilakukan yaitu menganalisis dan merancang sistem informasi pemesanan wisata. Namun yang menjadi perbedaannya, penelitian sebelumnya adalah fitur penjadwalan dan reschedule

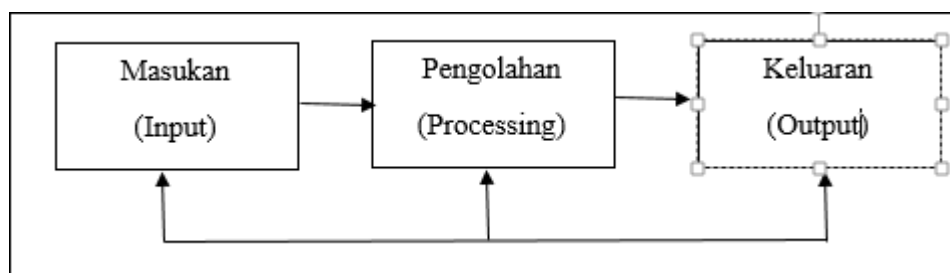
yang dapat dilakukan mandiri oleh pengunjung. Penambahan fitur ini yang menjadikan perbedaan antara penelitian ini dengan penelitian yang akan diusulkan. [1,p.2]

Penelitian lainnya yang sebanding dengan penelitian yang akan diusulkan adalah dari Panji Lesmana berjudul Sistem Informasi Pemesanan lapangan Futsal berbasis Web di Comet Futsal. Penelitian ini bertujuan untuk membangun sebuah sistem informasi baru sebagai media pemesanan online untuk menyewa lapangan futsal. Sistem ini menunjang efektifitas, produktifitas, dan efisiensi pada tempat penyewaan lapangan futsal tersebut. Model pengembangan sistem pada penelitian ini yang dikembangkan menggunakan model *prototype*. Hasil penelitian disimpulkan bahwa Web dapat digunakan sebagai media informasi yang efektif dan efisien bagi para calon pelanggan untuk mengetahui ketersediaan lapangan untuk disewa. Penelitian yang telah dilaksanakan memiliki persamaan yaitu merancang sistem informasi untuk media informasi pemesanan atau penyewaan. Namun yang menjadi perbedaan pada penelitian sebelumnya berfokus pemesanan online saja. Sedangkan penelitian yang akan dibuat bukan hanya pemesanan juga penjadwalan, pembayaran, dan pelaporan .[2,p.3]

2.2. Definisi Sistem

Sistem dapat diartikan suatu keadaan antara satu unsur dengan unsur lainnya atau variabel-variabel didalam sebuah kumpulan atau himpunan dimana adanya saling interaksi, saling ketergantungan, dan saling terorganisasinya antara satu dengan unsur lainnya. Sistem menurut pandangan para ahli merupakan, sebagai

berikut: Menurut Murdick dan Ross (1993) mengartikan sistem sebagai penggabungan suatu perangkat elemen untuk mencapai tujuan bersama; menurut Scott menjelaskan sistem yang didalamnya terdapat unsur masukan(*input*), pengolahan (*processing*), serta keluaran(*output*); menurut Gaspert sistem memiliki ciri pokok yaitu sistem ini dioperasikan didalam suatu lingkungan, sistem ini terdiri dari unsur-unsur, sistem di tandai dengan adanya keterhubungan, dan sistem ini memiliki satu fungsi dan atau tujuan utama [3,p.3]. Untuk menggambarkan sistem yang dimaksud yaitu model sebuah sistem ditunjukkan dalam gambar berikut:



Gambar 2. 1 Model Sistem
[3,p.4]

Dengan demikian seperti yang telah disebutkan mengenai sistem, dapat disimpulkan bahwa sistem dapat diartikan sebagai sebuah kumpulan yang terdiri dari berbagai unsur yang membentuk sebuah himpunan dimana himpunan ini dapat saling berinteraksi dalam sebuah proses aktivitas untuk mencapai sebuah tujuan tertentu.

2.3. Karakteristik Sistem

Pengembangan sebuah sistem dikatakan baik jika terdapat unsur-unsur (karakteristik) yang membentuknya, berikut merupakan karakteristi dari sebuah sistem adalah:

1. Komponen

Komponen sistem terdiri dari bagian- bagian atau subsistem yang saling berinteraksi untuk membetuk suatu kesatuan

2. Batasan Sistem(*Boundary*)

Batasan ini merupakan area dimana area ini merupakan pemisah antara satu sistem dengan sistem lainnya atau dengan lingkungan luarnya.

3. Lingkungan luar sistem (*environment*)

Lingkungan ini merupakan batasan luar sistem yang dapat mempengaruhi kinerja operasi sistem. Lingkungan ini dapat memberikan keuntungan ataupun kerugian terhadap sistem, maka dari itu haruslah ada pengendalian terhadap lingkungan luar ini.

4. Penghubung sistem (*Interface*)

Adanya alat penghubung yang dapat menghubungkan subsistem satu dengan yang lainnya, penghubung ini sekaligus sebagai media yang mengalirkan sumberdaya dimana output yang dikeluarkan dari suatu sistem bisa menjadi input dari subsistem laiinya.

5. Masukkan sistem(input)

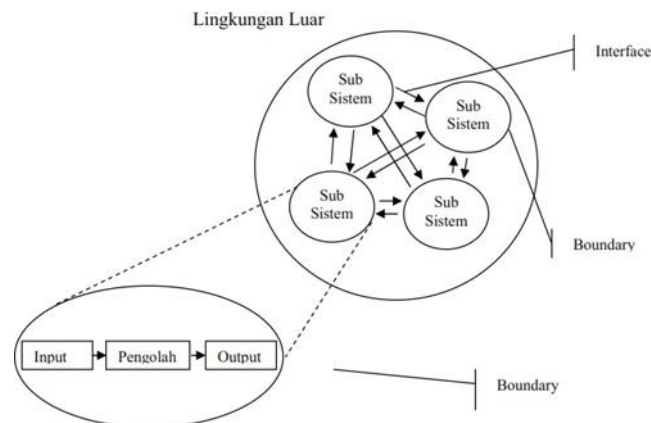
Adanya sebuah energi yang diamsukkan kedalam sistem computer misalnya berupa perawatan(*maintenance*) yang dapat digunakan sistem untuk melakukan operasi dan masukkan sinyal(*signal input*)energi yang di dapat dari hasil proses untuk mendapatkan keluaran.

6. Keluaran Sistem (Output)

Hasil dari proses pengolahan dan menghasilkan keluaran.

7. Pengolahan Sistem

Sistem ini merupakan sistem pengolahan yang dapat merubah masukan menjadi keluaran. Misalnya sistem produksi yang menghasilkan bahan jadi dari proses kelola bahann mentah, dan Sistem akuntansi yang digunakan untuk mengahsilkan informasi berupa laporan-laporan yang berasal dari proses pengolahan data-data akuntansi[4,p.3]. Berikut merupakan gambar karakteristik sistem:



Gambar 2. 2 Karakteristik Sistem Informasi [4,p.5]

2.4. Definisi Informasi

Informasi adalah data yang diproses (diolah) menjadi sesuatu yang dapat berguna dalam pemakaiannya.[5,p.36] Data disebut juga bahan mentah dari informasi yang melalui suatu proses transformasi data tersebut diubah menjadi lebih bermakna. Informasi menurut beberapa ahli yaitu McFadden,dkk(1999) Informasi berupa data yang sudah diproses sedemikian rupa, dalam penggunaannya dapat meningkatkan pengetahuan seseorang. Lalu Davis (1999) mengartikan informasi merupakan data yang sudah diolah untuk membentuk sebuah bentuk yang lebih berarti bagi penerimanya dan digunakan untuk pengambilan keputusan saat ini atau untuk dimasa mendatang.[6,p.45]

2.5. Karakteristik Informasi

Sebuah informasi dibutuhkan untuk mendukung proses pengambilan keputusan , dalam tingkatan organisasi atau individu untuk memenuhi kebutuhan tersebut diperlukan karakteristik informasi yang berbeda pula, berikut merupakan karakteristik dari sebuah informasi adalah sebagai berikut:

1. Kepadatan informasi

Kepadatan informasi terdiri dari informasi yang dihasilkan seperti apa berupa informasi yang terperinci(detail) dan kurang padat atau kah informasi yang tersaring dan lebih ringkas(padat),itu tergantung kebutuhannya.

2. Luas Informasi

Luas informasi yang dimiliki ini apakah informasi tersebut memiliki karakteristik

yang terfokus pada suatu masalah tertentu ataukah informasi yang dibutuhkan tersebut memiliki karakteristik informasi yang semakin luas dikarenakan kebutuhan yang berbeda.

3. Frekuensi Informasi

Frekuensi Informasi dapat dibedakan menjadi 2 kategori yaitu dibutuhkan dalam jangka rutin ataukah tidak, tergantung kepada jenis kebutuhan yang diperlukan untuk memenuhi kebutuhan kerja yang terstruktur dan memiliki pola waktu yang jelas.

4. Skedul Informasi

Informasi dapat diterima berdasarkan jadwal yang jelas dan tersusun karena kebutuhan informasi untuk memenuhi tugas yang terstruktur.

5. Waktu Informasi

Informasi yang diperlukan adalah informasi historis dimana informasi ini dibutuhkan untuk memenuhi kebutuhan kerja dalam penengendalian operasi yang digunakan untuk memeriksa tugas-tugas rutin yang sudah terjadi. Ataukah informasi yang diberikan adalah informasi yang yang memiliki waktu informasinya ke masa depan yaitu informasi tentang suatu prediksi.[5,p.70]

2.6 Pilar Informasi

Pilar Informasi adalah dasar yang membentuk sebuah informasi yang baik, untuk digunakan bagi penerima informasi untuk memenuhi kebutuhan. Adapun pilar informasi dibagi menjadi tiga pilar informasi yaitu: sesuai dengan fakta kenyataan yang ada atau bisa dikatakan *relevan*, tepat waktu(*timeliness*), serta

informasi yang diberikan memiliki nilai yang akurat yang dapat dipercaya (*accurate*). Berikut merupakan gambar pilar-pilar informasi:



Gambar 2. 3 Pilar-Pilar Informasi yang berguna
[5,p.38]

2.7 Definisi Sistem Informasi

Sistem informasi adalah suatu sistem yang melibatkan beberapa komponen dalam organisasi mengolah data menjadi informasi yang lebih berguna untuk mencapai suatu tujuan. Berikut ini beberapa definisi sistem informasi menurut para ahli :

1. Alter (1992) mengatakan definisi dari Sistem Informasi adalah kombinasi antara prosedur kerja, informasi, orang, dan teknologi informasi yang saling diorganisasikan untuk mencapai tujuan dalam suatu organisasi.
2. Bodnar dan Hopwood (1993), Sistem informasi adalah kumpulan dari perangkat keras dan perangkat lunak yang dirancang untuk mengubah data ke dalam bentuk informasi yang lebih berguna.
3. Gelinan, Oram, dan Wiggins (1990), Sistem informasi adalah sebuah yang

dibuat manusia yang secara umum terdiri atas komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan, dan mengelola data serta menyediakan informasi keluaran kepada para pemakainya.

4. Hall (2001), Sistem informasi adalah rangkaian prosedur formal dimana data dikelompokkan, lalu diproses menjadi informasi, dan didistribukan Kepada para pemakainya.
5. Turban, McLean, dan Wetherbe (1999), Sebuah sistem informasi mengumpulkan, memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk tujuan yang spesifik.
6. Wilkinson (1992), Sistem informasi adalah kerangka kerja yang mengkoordinasikan sumber daya (manusia, komputer) untuk mengubah masukan (*input*) menjadi keluaran (informasi), guna mencapai sasaran perusahaan. [6,p.9]

2.8 Komponen Sistem Informasi

Sistem informasi tersusun dari beberapa komponen atau elemen. Komponen sistem informasi tersebut adalah sebagai berikut:

- a. Perangkat keras atau yang disebut hardware , perangkat ini mencakup piranti-piranti fisik misalnya computer, printer, keyboard, dan lain-lain.
- b. Perangkat Lunak atau yang disebut software (program) merupakan sekumpulan instruksi-instruksi yang memungkinkan perangkat keras dalam melakukan proses.

- c. Prosedur, merupakan sekumpulan aturan yang digunakan dalam mewujudkan pemrosesan data dan menghasilkan sebuah keluaran yang dikehendaki.
- d. Orang, meruoakan semua pihak yang memiliki tanggung jawab di dalam pengembangan sebuah sistem, pemrosesan, dan penggunaan keluaran sistem informasi.
- e. Basis Data atau yang disebut *database* merupakan tabel, hubungan, dan lain-lain yang berhubungan dengan penyimpanan data.
- f. Jaringan computer dan komunikasi data merupakan sistem penghubung yang memungkinkan sumber dipakai secara bersama atau diakses oleh sejumlah pemakai. [6,p.71]

2.9 Basis Data

Basis Data (*Database*) adalah pengorganisasian sekumpulan data yang saling berkaitan satu sama lain sehingga mempermudah dalam aktivitas untuk memperoleh sebuah informasi. Basis data dibuat bertujuan untuk mengatasi problem pada sistem yang memakai pendekatan berbasis berkas. [6,p.218]

Basis data merupakan suatu sistem yang memiliki karakteristik seperti berikut:

1. Kumpulan-kumpulan interaksi data yang disimpan bersama-sama dan tidak saling mengganggu satu sama yang lainnya atau membentuk duplikat data.

2. Kumpulan-kumpulan data tersebut dapat digunakan oleh sebuah program secara optimal.
3. Penambahan data baru, modifikasi dan pengambilan kembali dari data dapat dilakukan dengan mudah dan terorganisasi.

Dalam sebuah arsitektur *database* terdapat tiga tingkatan yang ketiganya tersebut saling mendukung. Berikut ini adalah penjelasannya:

1. Level Internal yaitu level dimana berhubungan langsung dengan basis data dan menjelaskan bagaimana data disimpan dalam basis data. Level ini berurusan dengan hal yang antarlain sebagai berikut: alokasi ruang penyimpanan data dan indeks, deskripsi rekaman dalam penyimpanan, kompresi data dan teknik enkripsi data.
2. Level Eksternal merupakan level yang berhubungan langsung dengan pengguna. Pada level ini, pengguna cukup mengenal struktur data yang sederhana dalam basis data supaya dapat mengakses basis data.
3. Level konseptual menjelaskan data apa saja yang tersimpan dalam basis data dan juga menjabarkan hubungan- hubunganantar data. Level ini biasa dipakai oleh administrator data. [6,p.224]

Operasi yang dilakukan pada sebuah *database* berdasarkan atas tabel-tabel dan hubungannya. Beberapa istilah dalam model relasional antara lain *query* ,*table*, *record*, *indeks*, *field* yang akan dijelaskan lebih rinci sebagai berikut:

1. *Entity* atau tabel dalam model relasional digunakan untuk mendukung antar muka komunikasi antara pemakai dengan professional komputer.

2. *Record* atau baris atau dalam istilah model relasional yang formal disebut tuple adalah kumpulan data yang terdiri dari satu atau lebih.
3. *Field* atau kolom, dalam model relasional formal disebut *attribute* adalah sekumpulan data-data yang mempunyai atau menyimpan sebuah fakta yang sama atau sejenis untuk setiap baris yang terdapat pada tabel.
4. Indeks adalah salah satu jenis tipe dari suatu tabel tertentu yang berisi sebuah nilai-nilai *field* kunci.
5. *Query* adalah sekumpulan perintah *Structure Query Language* (SQL) yang dibentuk untuk memanggil kelompok *record* tertentu dari satu tabel atau tabel lainnya untuk melakukan operasi pada tabel.

2.10 Sistem Informasi Pemesanan (Sistem Pemrosesan Transaksi)

Sistem pemrosesan transaksi dipakai untuk menghimpun, menyimpan, dan memproses data transaksi serta kadangkala menegndalikan keputusan yang merupakan bagian dari transaksi.[6,p.102]

Sistem Informasi Pemesanan adalah bagian dari [sistem informasi](#) yang merupakan sebuah sistem yang menjalankan dan mencatat transaksi rutin harian yang diperlukan untuk menjalankan [bisnis](#). Contohnya adalah seperti memasukkan pesanan penjualan, pemesanan hotel, penggajian , pencatatan karyawan dan pengiriman.

Tujuan utama dari sistem pada tingkat ini adalah untuk menjawab pertanyaan rutin dan melacak arus transaksi yang melalui organisasi. Pada tingkat operasional,

tugas, sumber daya, dan tujuan ditentukan sebelumnya dan sangat terstruktur. Keputusan untuk memberikan kredit kepada pelanggan, contohnya, dilakukan oleh pengawas tingkat yang lebih rendah sesuai dengan kriteria yang telah ditetapkan sebelumnya. Yang harus ditentukan adalah apakah pelanggan memenuhi kriteria.

Manajer butuh sistem untuk memonitor status operasional internal dan hubungan perusahaan dengan lingkungan eksternal. Sistem Pemrosesan Transaksi juga merupakan pembuat utama informasi bagi jenis sistem lainnya. Sistem Pemrosesan transaksi seringkali sangat penting bagi bisnis sehingga kegagalan sistem selama beberapa jam dapat mengakibatkan kejatuhan perusahaan dan mungkin perusahaan lain yang berhubungan dengannya. [7.p.1]

2.11. Pemodelan : *Object Oriented Programming*

Object-Oriented Programming (OOP) adalah sebuah pendekatan untuk pengembangan suatu software dimana dalam struktur software tersebut didasarkan kepada interaksi objek dalam penyelesaian suatu proses / tugas. Interaksi tersebut mengambil form dari berbagai pesan dan mengirimkannya kembali melalui objek tersebut. Objek tersebut memproses pesan tadi menjadi sebuah tindakan / metode.

Object-Oriented Programming (OOP) adalah suatu metode pemrograman yang berorientasi kepada objek. Tujuan dari OOP tersebut diciptakan untuk memudahkan pengembangan program dengan cara mengikuti model yang ada sebelumnya dalam kehidupan sehari-hari. Singkatnya, setiap bagian dari suatu permasalahan adalah objek, dan objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil lagi. OOP tidak hanya sebuah konsep

pemrograman, namun merupakan cara berpikir mengenai aplikasi yang mempelajari bahwa aplikasi tidak hanya terdiri dari beberapa prosedur melainkan objek dan real-entity.[8]

2.12. Unified Modeling Language (UML)

Menurut Nugroho, Unified Modeling Language (UML) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berorientasi objek. Pemodelan sesungguhnya digunakan untuk menyederhanakan permasalahan-permasalahan yang rumit sehingga dapat lebih mudah dipahami dan dimengerti.[9] Sedangkan menurut Martin Fowler, Unified Modeling Language keluarga notasi grafis yang didukung oleh meta-model tunggal yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek.. [10] dan menurut Ditha Febrianisa, UML Diagram bisa diibaratkan seperti cetakan biru untuk membangun sebuah rumah. Satu set cetakan biru biasanya membantu pembangunnya dengan gambaran yang jelas untuk saluran air, listrik, pemanas, dan sejenisnya, setiap UML Diagram membantu tim developer program dengan gambaran yang jelas untuk sistem tersebut.

UML Diagram dapat disimpulkan sebagai sebuah rancangan untuk membangun sebuah perangkat lunak yang didalamnya berisi alur dari proses yang sudah ada untuk kemudian diterjemahkan kedalam bahasa pemrograman.

1. Use Case Diagram

Use case adalah kegiatan atau urutan interaksi yang saling berkaitan antara sistem dan aktor. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. *Use case* juga digunakan untuk membentuk perilaku (behaviour) sistem yang akan dibuat. Sebuah *use case* menggambarkan sebuah interaksi antara pengguna (aktor) dengan sistem yang ada.

Tujuan *Use Case* adalah Memetakan kebutuhan sistem ,Merepresentasikan interaksi pengguna terhadap sistem ,Untuk mengetahui kebutuhan diluar sistem . Adapun Komponen *use case diagram* adalah sebagai berikut:

a. Aktor

Aktor merupakan hal terpenting dari *use case diagram*, akan tetapi aktor bukanlah bagian dari *use case diagram*, namun untuk dapat terciptanya suatu *use case diagram* diperlukan beberapa aktor. Itulah pengertian Actor pada komponen *use case diagram* secara singkat .

b. Use Case

Use case merupakan sebuah gambaran fungsionalitas dari suatu sistem, sehingga pengguna sistem tersebut paham secara jelas mengenai kegunaan sistem atau aplikasi yang akan dibangun. Itulah pengertian *use case* pada komponen *use case diagram* secara singkat.

c. Relasi

Use case diagram memiliki relasi sebagai berikut:

- 1) *Association* : Menghubungkan link antar element.
- 2) *Generalization* : Sebuah elemen dapat merupakan spesialis dari elemen lainnya bisa disebut dengan pewarisan sifat (*inheritance*).
- 3) *Dependency* : Sebuah elemen bergantung dalam beberapa cara ke element lain.
- 4) *Aggregation* : Bentuk association dimana sebuah elemen berisi elemen lainnya.

Tipe Relasi/ *Streotype* yang mungkin terjadi pada use case diagram :

- 1) <<include>>, kelakuan yang harus terpenuhi agar sebuah event dapat terjadi.
- 2) <<extends>>, kelakuan yang hanya berjalan pada kondisi tertentu.
- 3) <<communicates>>, mungkin ditambahkan untuk asosiasi yang menunjukkan asosiasinya adalah communicates association.

2. Skenario *Use Case*

Skenario *Use Case* digunakan untuk memudahkan dalam menganalisa skenario yang akan kita gunakan pada fase-fase selanjutnya dengan melakukan penilaian terhadap skenario tersebut.

3. *Activity Diagram*

Activity Diagram adalah diagram yang menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan adalah

bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Komponen-komponen yang ada pada activity diagram adalah sebagai berikut:

- a. *Nodes* (initial dan final) adalah simbol untuk memulai (initial) dan mengakhiri (final) suatu activity diagram.
- b. *Activity* (aktifitas) adalah proses komputasi yang bisa berupa kata kerja atau ekspresi dan bersifat atomik atau tidak dapat didekomposisi
- c. *Flow* adalah awal dari proses yang paralel dan mampu menggambarkan aktivitas yang mungkin terjadi secara concurrent.
- d. *Join* adalah akhir dari suatu proses paralel.
- e. *Decision* adalah pilihan untuk mengambil keputusan.
- f. *Partition* digunakan untuk menjelaskan siapa yang melakukan aktivitas dalam activity diagram. Untuk melakukan partisi dapat dilakukan dengan menggunakan Swim Lane.
- g. *Signal* adalah tanda untuk memulai sebuah aktivitas.

4. *Sequence Diagram*

Sequence diagram adalah diagram yang menggambarkan kelakuan objek pada usecase dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram

sekuen maka harus diketahui objek-objek yang terlibat dalam use case beserta metode-metode yang dimiliki kelas yang diinisialisasi dalam objek itu. Jumlah dari diagram sekuen yang harus digambarkan sebanyak *usecase* yang memiliki proses sendiri atau yang penting semua usecase yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak usecase yang ada maka diagram sekuen yang dibuat semakin banyak.

Komponen Sequence Diagram terdiri dari :

- a. **Aktor** : Menggambarkan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem.
- b. **Boundary** : Menggambarkan interaksi antara satu atau lebih aktor dengan sistem, memodelkan bagian dari sistem yang bergantung pada pihak lain disekitarnya dan merupakan pembatas sistem dengan dunia luar.
- c. **Control** : Menggambarkan, mengkoordinasikan perilaku sistem dan dinamika dari suatu sistem, menangani tugas utama dan mengontrol alur kerja suatu sistem.
- d. **Entity** : Menggambarkan informasi yang harus disimpan oleh sistem (struktur data dari sebuah sistem).
- e. **Object Message** : Menggambarkan pesan/hubungan antar obyek yang menunjukkan urutan kejadian yang terjadi.
- f. **Message to Self** : Menggambarkan pesan/hubungan obyek itu sendiri, yang menunjukkan urutan kejadian yang terjadi.

- g. *Return Message* : Menggambarkan pesan/hubungan antar obyek, yang menunjukkan urutan kejadian yang terjadi
- h. *Lifeline* : Eksekusi obyek selama sequence (message dikirim atau diterima dan aktifasinya).

5. *Class Diagram*

Class Diagram adalah diagram yang menunjukkan class-class yang ada dari sebuah sistem dan hubungannya secara logika. Class diagram menggambarkan struktur statis dari sebuah sistem. Karena itu class diagram merupakan tulang punggung atau kekuatan dasar dari hampir setiap metode berorientasi objek termasuk UML. [11]

a. *Class* :

Blok – blok pembangun pada pemrograman berorientasi objek. Sebuah class digambarkan sebagai sebuah kotak yang terdiri dari 3 bagian. Bagian atas adalah bagian nama dari class, Bagian tengah mendefinisikan property/atribut class. Bagian akhir mendefinisikan method-method dari sebuah class.

b. *Association* :

Sebuah asosiasi merupakan sebuah relationship paling umum antara 2 *class* dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 *class*. Garis ini bisa melambangkan tipe-tipe *relationship* dan juga menampilkan hukum-hukum multiplisitas pada sebuah relationship (Contoh : *one-to-one*, *one-to-many*, *many-to-many*)

c. *Composition* :

Jika sebuah class tidak bisa berdiri sendiri dan harus merupakan bagian dari class yang lain, maka class tersebut memiliki relasi composition terhadap class tempat dia bergantung tersebut. Sebuah *relationship composition* digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi / solid.

d. *Dependency* :

Kadangkala sebuah class menggunakan class yang lain. Hal ini disebut dependency. Umumnya penggunaan dependency digunakan untuk menunjukkan operasi pada suatu class yang menggunakan class yang lain.

e. *Aggregation* :

Mengindikasikan keseluruhan bagian *relationship* dan biasanya disebut sebagai relasi.

6. *Object Diagram*

Diagram objek menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem. dalam diagram objek harus di pastikan bahwa semua kelas yang sudah didefinisikan pada diagram kelas harus dipakai objeknya, karena jika tidak pendefinisian kelas itu tidak dapat dipertanggung jawabkan.

Terdapat dua *stereotype* terhadap *dependency* di antara objek, yaitu:

- a. *Become*, menspesifikasikan target adalah objek yang sama dengan sumber, tapi pada titik waktu berikutnya dan dapat mempunyai nilai, *state* dan peran yang berbeda.
- b. *Call*, menspesifikasikan operasi sumber memanggil operasi target.

- c. *Copy*, menspesifikasikan objek target adalah nyata, tapi independen, mengkopi sumber.

7. *Deployment Diagram*

Deployment diagram merupakan gambaran proses-proses berbeda pada suatu sistem yang berjalan dan bagaimana relasi di dalamnya. Hal inilah yang mempermudah user dalam pemakaian sistem yang telah dibuat dan diagram tersebut merupakan diagram yang statis. Misalnya untuk mendeskripsikan sebuah situs web, *deployment diagram* menunjukkan komponen perangkat keras ("node") apa yang digunakan (misalnya, web server, server aplikasi, dan *database server*), komponen perangkat lunak ("artefak") apa yang berjalan pada setiap node.(misalnya, aplikasi web, *database*), dan bagaimana bagian-bagian yang berbeda terhubung (misalnya JDBC, REST, RMI).

Node digambarkan sebagai kotak, dan artefak yang dialokasikan ke setiap node digambarkan sebagai persegi panjang di dalam kotak. Node mungkin memiliki subnodes, yang digambarkan sebagai kotak *nested*. Sebuah node tunggal secara konseptual dapat mewakili banyak node fisik, seperti sekelompok database server.

Komponen *Deployment Diagram* :

- a. *Component*: Pada *deployment diagram*, komponen-komponen yang ada diletakkan didalam node untuk memastikan keberadaan posisi mereka.
- b. *Node* : *Node* menggambarkan bagian-bagian *hardware* dalam sebuah sistem. Notasi untuk node digambarkan sebagai sebuah kubus 3 dimensi.

- c. *Association* : Sebuah *association* digambarkan sebagai sebuah garis yang menghubungkan dua node yang mengindikasikan jalur komunikasi antara komponen - komponen *hardware*.
- d. *Dependency* : Merupakan relasi yang menunjukkan bahwa perubahan pada salah satu elemen memberi pengaruh pada elemen lain.
- e. *Generalization* : Menunjukkan hubungan antara elemen yang lebih umum ke elemen yang lebih spesifik. Dengan *generalization*, *class* yang lebih spesifik (*subclass*) akan menurunkan atribut dan operasi dari *class* yang lebih umum (*superclass*).
- f. *Note* : Menunjukkan catatan untuk komentar dari suatu pesan antar elemen, sehingga bisa langsung terlampir dalam model.
- g. *Class : Interface* merupakan kumpulan operasi tanpa implementasi dari suatu class. Implementasi operasi dalam *interface* dijabarkan oleh operasi didalam *class*.

8. *Component Diagram*

Component diagram dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada didalam sistem. diagram komponen juga dapat digunakan untuk memodelkan hal-hal berikut :

- a. *Source code* program perangkat lunak.
- b. Komponen *executable* yang di lepas ke *user*.
- c. Basis data secara fisik.

d. Sistem yang harus beradaptasi dengan sistem lain

e. *Framework* sistem,

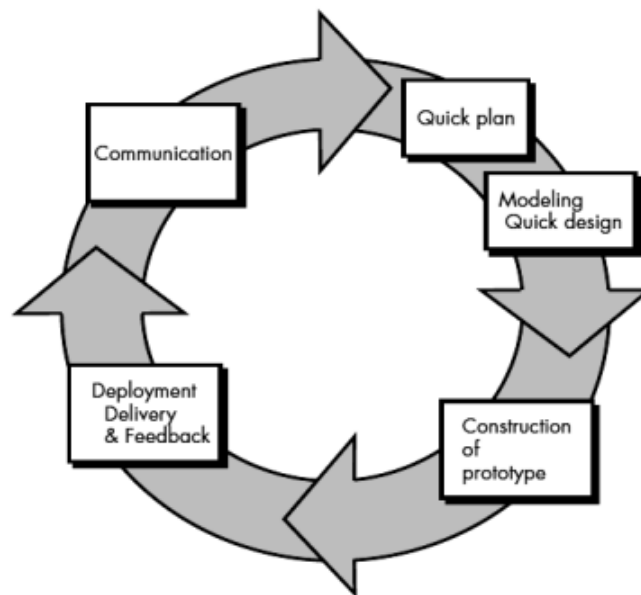
Framework pada perangkat lunak merupakan kerangka kerja yang dibuat untuk memudahkan pengembangan dan pemeliharaan aplikasi, contohnya seperti struts dari apache yang menggunakan prinsip desain *Model-View-Controller* (MVC) dimana *source code* program dikelompokkan berdasarkan fungsinya. dimana *controller* berisi *source code* yang menangani request dan validasi, model berisi *source code* yang menangani manipulasi data dan *bussiness logic*, dan *view* berisi *source code* yang menangani tampilan. Komponen dasar yang biasanya ada dalam suatu sistem adalah sebagai berikut :

- a. Komponen *user interface* yang menangani tampilan.
- b. Komponen *bussiness process* yang menangani fungsi-fungsi proses bisnis.
- c. Komponen data yang menangani manipulasi data.
- d. Komponen *security* yang menangani keamanan sistem.

2.13. Metode Pengembangan Perangkat Lunak : *Prototype*

Menurut Roger S. Pressman, dalam melakukan perancangan sistem yang akan dikembangkan dapat menggunakan metode *prototype*. Metode ini cocok digunakan untuk mengembangkan sebuah sistem. Metode ini dimulai dengan pengumpulan kebutuhan pengguna, kemudian membuat sebuah rancangan kilat yang selanjutnya akan dievaluasi kembali sebelum diproduksi secara benar. [12]

Prototype bukanlah merupakan sesuatu yang lengkap, tetapi sesuatu yang harus dievaluasi dan dimodifikasi kembali. Segala perubahan dapat terjadi pada saat *prototype* dibuat untuk memenuhi kebutuhan pengguna dan pada saat yang sama memungkinkan pengembang untuk lebih memahami kebutuhan pengguna secara lebih baik.



Gambar 2. 4 Paradigma Pembuatan Prototype

(Sumber : Roger, S. Pressman, Ph.D. *Rekayasa Perangkat Lunak (Pendekatan Praktisi)* Edisi 7 : Buku 1, 2012)

Gambar model *prototype* menurut Roger S. Pressman [12]:

1. *Communication* (Komunikasi)

Pada tahap ini dilakukan komunikasi dengan *stakeholder* pengumpulan kebutuhan dari sistem dengan cara mendengar kebutuhan dari pelanggan. Untuk membuat suatu sistem yang sesuai kebutuhan, maka harus diketahui terlebih dahulu

bagaimana sistem yang sedang berjalan untuk kemudian mengetahui masalah yang terjadi.

2. *Quick Plan* (Perencanaan Cepat)

Pada tahap ini, dilakukan perencanaan cepat untuk menentukan kebutuhan apa saja yang akan dilibatkan dalam tahap selanjutnya dan dalam tahapan ini juga tim akan melakukan pemodelan secara kasar mengenai sistem yang akan dibangun.

3. *Modeling Quick Design* (Pemodelan Rancangan Cepat)

Pada tahap ini, dilakukan pemodelan perencanaan dari tahap sebelumnya dengan menggunakan alat bantu seperti DFD, ERD dan Flowchart untuk pendekatan terstruktur. Sedangkan untuk pendekan objek, maka dapat menggunakan UML.

4. *Construction of Prototype* (Pembuatan Prototype)

Pada tahap ini, rancangan yang sebelumnya sudah dibuat akan mulai dikonversikan menjadi suatu program yang akan dibangun.

5. *Deployment Delivery & Feedback* (Penyerahan dan Memberikan Umpan Balik Terhadap Pengembangan Prototipe)

Pada tahap ini, prototype yang sudah dibangun akan diserahkan kepada pelanggan untuk dinilai dan diberikan umpan balik mengenai kinerja prototipe tersebut. Apabila dalam prototipe tersebut masih ada kekurangan, maka akan kembali ke langkah ke 1 untuk memperbaiki kekurangan tersebut.

2.14. Perangkat Lunak Pendukung

Penelitian ini menggunakan beberapa perangkat lunak untuk membangun Sistem Informasi Pelayanan Masyarakat pada Kantor Desa Tanimulya Kabupaten Bandung Barat. Berikut adalah urutan perangkat lunak yang digunakan dalam membangun sistem informasi tersebut :

2.14.1. *Perl Hypertext Preprocessor (PHP)*

Menurut Arief, *PHP (Perl Hypertext Preprocessor)* adalah bahasa server-side-scripting yang menyatu dengan HTML untuk membuat halaman web yang dinamis. [13] Dengan menggunakan program PHP, sebuah *website* akan lebih interaktif dan dinamis. Adapun kelebihan-kelebihan dari PHP yaitu:

1. PHP merupakan sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya. Tidak seperti halnya bahasa pemrograman aplikasi yang lainnya.
2. PHP dapat berjalan pada *web server* yang dirilis oleh Microsoft, seperti IIS atau PWS juga pada apache yang bersifat *open source*.
3. Karena sifatnya yang *open source*, maka perubahan dan perkembangan interpreter pada PHP lebih cepat dan mudah, karena banyak milis-milis dan *developer* yang siap membantu pengembangannya.
4. Jika dilihat dari segi pemahaman, PHP memiliki referensi yang begitu banyak sehingga sangat mudah untuk dipahami.
5. PHP dapat berjalan pada 3 operating sistem, yaitu: Linux, unix, dan windows, dan juga dapat dijalankan secara *runtime* pada suatu *console*.

2.14.2. Bootstrap

Bootstrap adalah sebuah *framework* CSS yang menyediakan kumpulan komponen-komponen antarmuka dasar pada web yang telah dirancang sedemikian rupa untuk digunakan bersama-sama. Selain komponen antarmuka, *Bootstrap* juga menyediakan sarana untuk membangun *layout* halaman dengan mudah dan rapi, serta modifikasi pada tampilan dasar HTML untuk membuat seluruh halaman web yang dikembangkan senada dengan komponen-komponen lainnya. Bootstrap dibuat untuk memberikan sekumpulan perangkat yang dapat digunakan untuk membangun *website* sederhana dengan mudah.

2.14.3. Cascading Style Sheet (CSS)

CSS merupakan salah satu kode pemrograman yang bertujuan untuk menghias dan mengatur gaya tampilan/layout halaman web supaya lebih elegan dan menarik. CSS adalah sebuah teknologi internet yang direkomendasikan oleh World Wide Web Consortium atau W3C pada tahun 1996. Awalnya, CSS dikembangkan di SGML pada tahun 1970, dan terus dikembangkan hingga saat ini. CSS telah mendukung banyak bahasa markup seperti HTML, XHTML, XML, SVG (Scalable Vector Graphics) dan Mozilla XUL (XML User Interface Language).

Pada desember 1996, W3C memperkenalkan Level 1 spesifikasi CSS atau juga dikenal CSS1 yang mendukung format, warna font teks, dan lain-lain. Kemudian, Mei 1998, W3C menerbitkan CSS2 yang di dalamnya diatur fungsi peletakan elemen. Dan sekarang, W3C telah memperbaiki dan meningkatkan Kemampuan CSS2 ke CSS3. CSS digunakan oleh web programmer dan juga blogger untuk menentukan warna, tata letak font, dan semua aspek lain dari presentasi dokumen di situs mereka. Saat ini, hampir tidak ada situs web yang dibangun tanpa kode CSS.

2.14.4. Database Management System (DBMS)

DBMS adalah singkatan dari “*Database Management System*” yaitu sistem pengorganisasian dan sistem pengolahan *database* pada komputer. DBMS atau *database management system* ini merupakan perangkat lunak (*software*) yang dipakai untuk membangun basis data yang berbasis komputerisasi.

DBMS (*Database Management system*) ini juga dapat membantu dalam memelihara serta pengolahan data dalam jumlah yang besar, dengan menggunakan DBMS bertujuan agar tidak dapat menimbulkan kekacauan dan dapat dipakai oleh *user* sesuai dengan kebutuhan.

DBMS ialah perantara untuk user dengan basis data, untuk dapat berinteraksi dengan DBMS dapat memakai bahasa basis data yang sudah ditentukan oleh perusahaan DBMS. Bahasa basis data umumnya terdiri dari berbagai

macam instruksi yang diformulasikan sehingga instruksi tersebut dapat di proses oleh DBMS.

Perintah atau instruksi tersebut umumnya ditentukan oleh *user*, adapun bahasa yang digunakan dibagi kedalam 2 (dua) macam diantaranya sebagaimana di bawah ini:

1. DDL (*Data Definition Language*)

Yang pertama adalah bahasa DDL atau kepanjangannya *Data Definition Language*, yaitu dipakai untuk menggambarkan desain dari basis data secara menyeluruh. DDL (*Data Definition Language*) dapat dipakai untuk membuat tabel baru, memuat indeks, maupun mengubah tabel. Hasil dari kompilasi DDL akan disimpan di kamus data. Itulah definisi dari DDL.

2. DML (*Data Manipulation Language*)

DML atau kepanjangannya *Data Manipulation Language*, yaitu dipakai untuk memanipulasi dan pengambilan data pada suatu basis data, misalnya seperti penambahan data yang baru ke dalam suatu basis data, menghapus data pada suatu basis data dan mengubah data pada suatu basis data. Itulah definisi dari DML.

2.14.5. *MySQL*

MySQL merupakan DBMS yang bersifat *open source* dimana pengguna dapat melakukan modifikasi pada kodenya yang berorientasi pada performa dan merupakan DBMS yang mendukung beberapa tipe *table (storage engine)*. Tipe *table* yang digunakan secara *default* adalah *MyISAM* yang memiliki kelebihan akses data yang cepat tetapi tidak mendukung fitur *foreign key*. Jika pengguna ingin

ada dukungan foreign key pada *table*, pengguna dapat menggunakan tipe *table* InnoDB.

Kelebihannya:

1. *Free/gratis*.
2. Selalu stabil dan cukup tangguh.
3. Keamanan yang cukup baik.
4. Sangat mendukung transaksi, dan dukungan dari banyak komunitas.
5. Sangat fleksibel dengan berbagai macam program.
6. Perkembangan yang cepat.

Kekurangannya:

1. Kurang mendukung koneksi bahasa pemrograman misalnya seperti Visual Basic (VB), Foxpro, Delphi sebab koneksi ini dapat menyebabkan *field* yang dibaca harus sesuai dengan koneksi bahasa pemrograman visual tersebut.
2. Data yang dapat ditangani belum besar dan belum mendukung *windowing Function*.

2.14.6. XAMPP

XAMPP ialah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan campuran dari beberapa program. Yang mempunyai fungsi sebagai server yang berdiri sendiri (localhost), yang terdiri dari program MySQL database, Apache HTTP Server, dan penerjemah ditulis dalam bahasa pemrograman PHP dan Perl.

Nama XAMPP merupakan singkatan dari X (empat sistem operasi), Apache, MySQL, PHP dan Perl. Program ini tersedia di bawah GNU General Public License dan bebas, adalah mudah untuk menggunakan web server yang dapat melayani tampilan halaman web yang dinamis. Jika ingin mendapatkan XAMPP dapat mendownload langsung dari situs resminya.

Bagian-bagian XAMPP

1. **htdocs** adalah folder di mana Anda meletakkan file yang akan dijalankan, seperti file PHP, HTML dan script lainnya.
2. **phpMyAdmin** adalah bagian untuk mengelola database MySQL yang dikomputer. Untuk membukanya, membuka browser dan ketik alamat `http://localhost/phpMyAdmin`, halaman phpMyAdmin akan muncul.
3. **Control Panel** yang berfungsi untuk mengelola layanan (service) XAMPP. Seperti stop service (berhenti), atau mulai (mulai).

Komponen XAMPP

1. Apache 2.4.4
2. MySQL 6.5.11
3. PHP 5.5.0
4. phpMyAdmin 4.0.4
5. FileZilla FTP Server 0.9.41

6. Tomcat 7.0.41 (with mod_proxy_ajp as connector)
7. Strawberry Perl 5.16.3.1 Portabel
8. XAMPP Control Panel 3.2.1 (dari hackattack142)

Singkatan pada XAMPP :

X : Program ini dapat dijalankan di banyak sistem operasi, seperti Windows, Linux, Mac OS, dan Solaris.

A : Apache, server aplikasi Web. Apache tugas utama adalah untuk menghasilkan halaman web yang benar kepada pengguna terhadap kode PHP yang sudah dituliskan oleh pembuat halaman web. jika perlu kode PHP juga berdasarkan yang tertulis, dapat database diakses dulu (misalnya MySQL) untuk mendukung halaman web yang dihasilkan.

M : MySQL, server aplikasi database. Pertumbuhannya disebut SQL singkatan dari Structured Query Language. SQL merupakan bahasa terstruktur yang difungsikan untuk mengolah database. MySQL dapat digunakan untuk membuat dan mengelola database dan isinya. Bisa juga memanfaatkan MySQL guna untuk menambahkan, mengubah, dan menghapus data dalam database.

P : PHP, bahasa pemrograman web. Bahasa pemrograman PHP adalah bahasa pemrograman untuk membuat web yang server-side scripting. PHP digunakan

untuk membuat halaman web dinamis. Sistem manajemen database yang sering digunakan dengan PHP adalah MySQL. namun PHP juga mendukung Pengelolaan sistem database Oracle, Microsoft Access, Interbase, d-base, PostgreSQL, dan sebagainya.

P : Perl, bahasa pemrograman untuk semua tujuan, pertama kali dikembangkan oleh Larry Wall, mesin Unix. Perl dirilis pertama kali tanggal 18 Desember 1987 yang ditandai dengan keluarnya Perl 1. Pada versi-versi selanjutnya, Perl juga tersedia untuk berbagai sistem operasi Unix (SunOS, Linux, BSD, HP-UX), juga tersedia untuk sistem operasi seperti DOS, Windows, PowerPC, BeOS, VMS, EBCDIC, dan PocketPC.

2.15. Arsitektur Aplikasi

Arsitektur aplikasi berisi penjelasan tentang jaringan komputer, jenis-jenis jaringan komputer, topologi jaringan komputer, dan manfaat jaringan komputer.

2.16. Jaringan Komputer

Jaringan komputer adalah suatu sistem yang didalamnya terdiri dari dua atau lebih perangkat komputer serta perangkat - perangkat lainnya yang dibuat atau dirancang untuk dapat berkerja sama dengan tujuan agar dapat berkomunikasi, mengakses informasi, meminta serta memberikan layanan atau *service* antara komputer satu dengan yang lainnya.

Menurut Jafar Noor Yudianto, jaringan komputer adalah sebuah sistem yang terdiri atas komputer-komputer yang didesain untuk dapat berbagi sumber daya (*printer*,

CPU), berkomunikasi (surel, pesan instan), dan dapat mengakses informasi (peramban web). Tujuan dari jaringan komputer adalah agar dapat mencapai tujuannya, setiap bagian dari jaringan komputer dapat meminta dan memberikan layanan (*service*). Pihak yang meminta/menerima layanan disebut klien (*client*) dan yang memberikan/mengirim layanan disebut peladen (*server*). Desain ini disebut dengan sistem *client-server*, dan digunakan pada hampir seluruh aplikasi jaringan computer. [14]

Hosting atau sering juga disebut dengan *WebHosting* merupakan suatu infrastruktur yang digunakan untuk menyimpan data dari website atau aplikasi sehingga bisa diakses menggunakan *internet*. Singkatnya jika sebuah smartphone membutuhkan kartu memori untuk menyimpan data seperti gambar, *file*, dan lainnya, fungsi hostingan mempunyai peran yang sama seperti kartu memori yang digunakan pada *smartphone*.

Hosting sendiri mempunyai kapasitas yang bisa digunakan sesuai dengan kebutuhan suatu *website* atau aplikasi. Misalnya *hosting* digunakan oleh website universitas untuk menyimpan data para mahasiswa, atau bisa juga menyimpan informasi seperti gambar, artikel, file dokumen dan lainnya di database yang ada pada hostingan. Secara sederhananya hosting merupakan rumah dari suatu *website* yang dimana didalamnya tersimpan banyak sekali properti (data) yang ada pada rumah tersebut. Hosting memiliki bermacam-macam jenis dengan kebutuhannya sendiri, ada yang gratisa dan juga yang berbayar. Tentu masing-masing mempunyai keunggulannya tersendiri.

Domain adalah nama dari *website* yang akan dihosting ke dalam *server*. *Domain* adalah identitas dari sebuah *website* yang berisi *link* / url yang digunakan oleh orang-orang untuk mengakses sebuah *website*. *Domain* adalah nama unik dari sebuah *IP Address* karena sebuah *website* yang telah dihosting memiliki *IP Address* masing-masing. Fungsi dari *domain* ini selain identitas sebuah *website* adalah untuk menyamakan *IP Address* tersebut agar pengguna dapat dengan mudah mengakses *website* tanpa harus mengetikkan *IP Address website* tersebut.

2.17 Bahasa Pemrograman JavaScript

JavaScript merupakan sebuah bahasa pemrograman web yang bersifat *Client Side Programming Language* yang pemrosesannya dilakukan disisi *client*. Aplikasi *client* tersebut merujuk kepada aplikasi web browser seperti *Mozilla Firefox* dan *Google Chrome*.

Pada masa awal perkembangannya *JavaScript* memiliki fungsi untuk membuat interaksi antara *situs web* dengan *user* menjadi lebih cepat karena dalam prosesnya tidak harus menunggu proses yang terjadi di *web server*. .

2.18 CodeIgniter

CodeIgniter adalah sebuah **framework PHP** yang dapat mempercepat pengembangan atau pembuatan sebuah **website**. Dengan dilengkapi **library** yang banyak dan **helper** yang berguna didalamnya sehingga dapat mempermudah proses **development**. Banyak **framework PHP** yang telah ada saat ini. Salah satunya adalah **framework codeIgniter**. CodeIgniter termasuk

dalam aplikasi *open source* (gratis) berupa *framework* dengan model konsep MVC (*Model, View, Controller*) yang di memiliki fungsi untuk membuat sebuah halaman web dinamis yang berbasis PHP.

Konsep MVC (*Model, View, Controller*) merupakan salah suatu konsep yang populer dalam pembangunan sebuah aplikasi web, konsep ini berawal dari bahasa pemrograman *Small Talk*. MVC memisahkan pengembangan aplikasi menjadi beberapa komponen utama untuk membangun sebuah aplikasi web seperti manipulasi data (*model*), *user interface* (*view*), dan bagian yang menjadi kontrol aplikasi (*Controller*). Berikut ini penjelasan lebih rinci dari ketiga bagian tersebut :

1. *View*, bagian ini memiliki fungsi untuk menangani *presentation logic*. Pada sebuah aplikasi web bagian *view* ini biasanya berupa *file template* HTML, bagian ini diatur oleh sebuah *controller*. *View* berguna untuk menerima dan merepresentasikan sebuah data kepada *user*.
2. *Model*, bagian ini memiliki hubungan langsung dengan *database* untuk memanipulasi sebuah data (*creat, update, delete, read, search*), menangani validasi yang dikirim dari bagian *controller*, bagian ini tidak dapat berhubungan langsung dengan bagian *view*.
3. *Controller*, bagian ini merupakan bagian yang memiliki fungsi untuk mengatur hubungan antara bagian *view* dan bagian *model*, menerima permintaan data dari *user*, lalu bagian ini kemudian menentukan apa yang akan diproses pada aplikasi.

