

## BAB II LANDASAN TEORI

### 2.1. Konsep Dasar Sistem Informasi

Sistem Informasi dibangun dari dua unsur yaitu sistem dan informasi. Untuk lebih jelas mengenai Sistem Informasi, maka akan diuraikan ke dalam definisi sistem, informasi dan sistem informasi.

#### 2.1.1. Definisi Sistem

Berdasarkan Wikipedia, sistem berasal dari bahasa Latin (*systema*) dan bahasa Yunani (*systema*) adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan. Istilah ini sering dipergunakan untuk menggambarkan suatu set entitas yang berinteraksi, di mana suatu model matematika seringkali bisa dibuat. Sistem juga merupakan kesatuan bagian-bagian yang saling berhubungan yang berada dalam suatu wilayah serta memiliki item-item penggerak, contoh umum misalnya seperti negara. Negara merupakan suatu kumpulan dari beberapa elemen kesatuan lain seperti provinsi yang saling berhubungan sehingga membentuk suatu negara di mana yang berperan sebagai penggerak yaitu rakyat yang berada dinegara tersebut.

Adapun pengertian sistem menurut Jeperson Hutahean yang menyatakan sistem sebagai suatu jaringan kerja dari

prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran yang tertentu[1]. Sementara menurut Maniah dan Dini Hamidini mengatakan bahwa sistem adalah kumpulan dari elemen-elemen berupa data, jaringan kerja dari prosedur-prosedur yang saling berhubungan, sumber daya manusia, teknologi baik *hardware* maupun *software* yang saling berinteraksi sebagai satu kesatuan untuk mencapai tujuan/sasaran tertentu yang sama[2]. Dari beberapa pengertian diatas penulis menyederhanakan pengertian sistem menjadi suatu kumpulan dari komponen-komponen yang berhubungan satu dengan yang lainnya untuk mencapai suatu tujuan bersama.

Menurut Jeperson Hutahaean[1] agar sistem dapat dikatakan sistem yang baik, maka sistem harus memiliki :

1. Komponen Sistem

Memiliki beberapa komponen yang saling berinteraksi, yang memiliki arti saling bekerja sama untuk membentuk satu kesatuan. Komponen sistem sendiri terdiri dari komponen yang berupa subsistem atau bagian-bagian lain dari sistem.

2. Batasan Sistem (*Boundary*)

Batasan sistem adalah batasan antara sistem satu dengan sistem lain. Batasan sistem ini dapat

memungkinkan sistem dipandang sebagai satu kesatuan. Delapan batasan dari suatu sistem juga menunjukkan ruang lingkup (*Scope*) dari sistem tersebut.

### 3. Lingkungan Luar Sistem (*Environment*)

Lingkungan luar sistem adalah kondisi yang ada di luar sistem, namun dapat mempengaruhi jalan atau beropersinya sistem. Lingkungan sendiri dapat bersifat menguntungkan dan merugikan. Sifat menguntungkan ini harus tetap di jaga, sedangkan sifat merugikan harus dijaga dan dikendalikan agar tidak merusak kelangsungan hidup sistem.

### 4. Penghubung Sistem (*Interface*)

Penghubung sistem adalah media bantu untuk menghubungkan antara suatu subsistem dengan subsistem lain. Keluaran dari subsistem akan menjadi masukan untuk subsistem lain melalui penghubung.

### 5. Masukkan Sistem (*Input*)

Masukkan adalah energi yang dimasukkan ke dalam sistem, yang berupa perawatan dan masukkan sinyal. Perawatan *input* adalah energi yang dimasukkan agar sistem dapat beroperasi, sedangkan sinyal *input*

adalah energi yang diproses untuk mendapatkan keluaran.

#### 6. Keluaran Sistem (*Output*)

Keluaran sistem adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang memiliki nilai guna dan sisa pembuangan.

#### 7. Pengolah Sistem

Pengolah Sistem adalah bagian yang akan merubah masukkan menjadi keluaran.

#### 8. Sasaran Sistem

Sasaran sistem adalah tujuan (*goal*) atau sasaran (*objective*) dari sebuah sistem. Sasaran dari sistem sangat menentukan masukkan yang dibutuhkan sistem dan keluaran yang akan dihasilkan oleh sistem.

### **2.1.2. Definisi Informasi**

Menurut Wikipedia, informasi adalah pesan (ucapan atau ekspresi) atau kumpulan pesan yang terdiri dari order sekuens dari simbol, atau makna yang dapat ditafsirkan dari pesan atau kumpulan pesan. Informasi dapat direkam atau ditransmisikan. Hal ini dapat dicatat sebagai tanda-tanda, atau sebagai sinyal berdasarkan gelombang. Informasi adalah jenis acara yang mempengaruhi suatu negara dari sistem dinamis. Para konsep memiliki banyak arti lain dalam konteks yang berbeda.

Informasi bisa dikatakan sebagai pengetahuan yang didapatkan dari pembelajaran, pengalaman, atau instruksi. Namun, istilah ini memiliki banyak arti bergantung pada konteksnya, dan secara umum berhubungan erat dengan konsep seperti arti, pengetahuan, *negentropy*, persepsi, *stimulus*, komunikasi, kebenaran, representasi, dan rangsangan mental.

Menurut Jogiyanto HM. informasi adalah hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian – kejadian (*event*) yang nyata (*fact*) yang digunakan untuk pengambilan keputusan[3]. Sedangkan menurut Gordon B. Davis, informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat bagi pengambilan keputusan saat ini atau mendatang[4]. Ada juga menurut Lani Sidharta yang menyatakan informasi sebagai data yang disajikan dalam bentuk yang berguna untuk membuat keputusan[5]. Dari beberapa pengertian yang dikemukakan sebelumnya, bisa disimpulkan bahwa informasi adalah data yang menunjukkan fakta yang telah diolah dan dibentuk sedemikian rupa sehingga bisa digunakan untuk mengambil kesimpulan.

### **2.1.3. Definisi Sistem Informasi**

Menurut O'Brien, sistem informasi adalah suatu kombinasi teratur apapun dari *people* (orang), *hardware*

(perangkat keras), *software* (piranti lunak), *computer networks and data communications* (jaringan komunikasi), dan *database* (basis data) yang mengumpulkan, mengubah dan menyebarkan informasi di dalam suatu bentuk organisasi[6]. Sementara menurut Lani Sidharta sistem informasi adalah sistem buatan manusia yang berisi himpunan terintegrasi dari komponen-komponen manual dan komponen-komponen terkomputerisasi yang bertujuan untuk mengumpulkan data, memproses data, dan menghasilkan informasi untuk pemakai[5].

Selain yang telah disebutkan diatas, ada juga pengertian sistem informasi menurut Tata Sutabri yang menyatakan sistem informasi merupakan suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi organisasi yang bersifat manajerial dalam kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan[7]. Bisa disimpulkan bahwa sistem informasi adalah sebuah sistem yang diterapkan dalam suatu organisasi yang terdiri dari komponen-komponen berupa manusia dan komputer yang berfungsi untuk mengumpulkan, mengubah dan menyebarkan informasi.

#### 2.1.4. Definisi Aplikasi

Secara bahasa, kata aplikasi berasal dari Bahasa Inggris “*Application*” yang berarti penggunaan atau penerapan. Menurut penulis, aplikasi berarti sebuah *software* (perangkat lunak) yang digunakan pada komputer yang memiliki fungsi-fungsi tertentu. Aplikasi dirancang dan memiliki tujuan untuk membantu menyelesaikan pekerjaan tertentu. Aplikasi terdiri dari berbagai jenis, yaitu aplikasi *desktop*, *mobile* atau *website*. Biasanya aplikasi dibuat menggunakan setidaknya satu bahasa pemrograman seperti Java, Python, dan PHP.

#### 2.1.5. Definisi Pemeriksaan Gigi

Menurut drg. Melia dari HelloSehat ([hellosehat.com](http://hellosehat.com)), pemeriksaan gigi oleh dokter gigi bisa mendeteksi dan mengobati berbagai gangguan yang umum menyerang area mulut. Seperti gigi berlubang (karies) dan penyakit gusi. Selain itu, dokter gigi juga dapat memeriksa tanda-tanda dan gejala-gejala penyakit gigi dan gusi lainnya seperti kanker mulut.

Saat pemeriksaan gigi rutin, dokter akan mengecek kondisi gigi pasien, apakah ada lubang gigi, patah, retak, tambalan gigi yang rusak, atau ada plak dan karang gigi. Dokter gigi juga akan menilai seberapa tinggi risiko pasien terhadap masalah gigi

berlubang. Dokter gigi juga akan memeriksa posisi gigi bungsu yang tumbuh miring atau lajur susunan gigi yang berantakan. Setelah itu dokter gigi akan memberikan masukan dan saran jika perlu tindakan lebih lanjut.

Selain memeriksa gigi, dokter juga akan mengecek kondisi gusi dan jaringan pendukung gigi lainnya yang sangat berperan dalam kesehatan gigi. Mulai dari lidah, langit-langit, hingga persendian rahang. Begitu juga dengan masalah-masalah lain yang muncul pada jaringan penyangga gigi, seperti gusi berdarah, gusi bengkak, gusi turun, dan gigi goyang yang disebabkan karena gusi rusak, semuanya akan diperiksa. Nantinya dokter gigi akan memeriksa dan memberikan penanganan yang tepat.

Diluar hal-hal yang disebutkan diatas, dokter gigi juga akan mengecek seberapa baik pasien dalam merawat kesehatan mulut dan giginya. Tak lupa, dokter akan menanyakan kebiasaan harian pasien, seperti pola makan, merokok, parafungsi (kebiasaan sukagigit pensil, kuku, menggertakkan rahang, bruxism atau menggemeretak gigi) yang bisa berdampak buruk pada gigi. Setelah itu biasanya dokter gigi akan memberikan saran untuk mengubah kebiasaan pasien jika diperlukan[8].

### **2.1.6. Definisi Aplikasi Pemeriksaan Gigi**

Menurut penulis, aplikasi pemeriksaan gigi adalah aplikasi yang berguna untuk membantu dokter gigi dalam melakukan pemeriksaan gigi pasien. Aplikasi pemeriksaan gigi seharusnya bisa memudahkan dokter dalam membuat hasil pemeriksaan yang mendetail tanpa perlu menuliskan keterangan terlalu banyak ke dalam kertas. Ini juga berarti semua data hasil pemeriksaan harus bisa dimasukkan ke dalam aplikasi dan juga bisa diakses oleh dokter gigi yang lain. Karena aplikasi pemeriksaan gigi ditargetkan untuk dokter gigi, maka prioritas utama dari bahasa atau istilah yang digunakan begitu juga dengan pengilustrasian gigi harus disesuaikan dengan kebutuhan dan pemahaman dokter gigi.

### **2.1.7. Definisi Klinik Gigi**

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

setiap orang yang mengabdikan diri dalam bidang kesehatan serta memiliki pengetahuan dan/ atau keterampilan melalui pendidikan di bidang kesehatan yang untuk jenis tertentu memerlukan kewenangan untuk melakukan upaya kesehatan.

Menurut Peraturan Menteri Kesehatan Republik Indonesia Nomor 9 Tahun 2014 klinik adalah fasilitas pelayanan kesehatan yang menyelenggarakan pelayanan kesehatan perorangan yang menyediakan pelayanan medis dasar dan/ atau spesialisik. Klinik dapat mengkhususkan pelayanan pada satu bidang tertentu berdasarkan cabang/disiplin ilmu atau sistem organ. Klinik gigi menurut peraturan menteri nomor 920/Menkes/Per/XIII/1986 merupakan sarana pelayanan kesehatan gigi dan mulut yang diberikan kepada masyarakat[9].

#### **2.1.8. Definisi Sistem Informasi Terintegrasi**

Menurut Andreas Dan (andreasdan.com), sistem informasi terintegrasi merupakan sebuah sistem yang memungkinkan berbagi data untuk seluruh organisasi. Sistem terintegrasi juga bisa disebut sebagai sistem ERP (*Enterprise Resource Planning*) karena semua data operasional terletak pada *database* pusat dan dapat diakses oleh pengguna disuatu organisasi. Dengan sebuah sistem terintegrasi, perusahaan dapat menyimpan informasi secara elektronik. Hal ini akan menggantikan tumpukan kertas dan *file* yang membuat kesulitan dalam mengambil suatu

keputusan. informasi dengan sistem terintegrasi dapat menyebabkan proses bisnis yang lebih efisien yang harganya lebih rendah dari sistem yang tidak terintegrasi[10].

### **2.1.9. Definisi Sistem Informasi Klinik Gigi Terintegrasi**

Menurut penulis, sistem informasi klinik gigi terintegrasi adalah sebuah sistem informasi yang diterapkan di banyak klinik gigi sekaligus dan berpusat pada satu sumber data utama. Sistem ini membolehkan klinik gigi untuk menggunakan data yang sama dengan banyak klinik gigi lainnya. Sistem informasi klinik gigi terintegrasi akan mencakup semua transaksi dalam klinik dan juga menambahkan beberapa hal baru, seperti ketersediaan *customer service*. Hal ini pastinya akan memberikan pengalaman baru tidak hanya untuk dokter gigi tapi juga pasien dan petugas klinik lainnya, dengan menyuguhkan aplikasi-aplikasi sesuai dengan perannya.

## **2.2. Metode Pendekatan Berorientasi Objek**

Pendekatan berorientasi objek merupakan suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem (sistem perangkat lunak, sistem informasi atau sistem lainnya). Metode ini memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata. Ada banyak cara untuk mengabstraksikan dan memodelkan objek-objek tersebut, mulai

dari abstraksi objek, kelas, hubungan antar kelas sampai abstraksi sistem. Saat mengabstraksikan dan memodelkan objek-objek, data dan proses-proses yang dimiliki oleh objek tersebut akan dienkapsulasi (dibungkus) menjadi satu kesatuan.

Dalam rekayasa perangkat lunak, konsep pendekatan berorientasi objek dapat diterapkan pada tahap analisis, perancangan, pemrograman, dan pengujian perangkat lunak. Ada berbagai teknik yang dapat digunakan pada masing-masing tahap tersebut, dengan aturan dan alat bantu pemodelan tertentu. Sistem yang dihasilkan dengan pendekatan berorientasi objek adalah sebuah sistem yang komponennya dibungkus (dienkapsulasi) menjadi kelompok data dan fungsi. Setiap komponen dalam sistem tersebut dapat mewarisi atribut dan sifat dan komponen lainnya. dan dapat berinteraksi satu sama lain.

Karakteristik atau sifat-sifat yang dimiliki sebuah sistem berorientasi objek adalah sebagai berikut:

1. Abstraksi, prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan
2. Enkapsulasi, Pembungkusan atribut data dan layanan (operasi-operasi) yang dimiliki objek. untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya

3. Pewarisan (*inheritance*), mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dan dirinya
4. Generalisasi dan Spesialisasi, menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus
5. Komunikasi Antar Objek, komunikasi antar objek dilakukan lewat pesan (*message*) yang dikirim dan satu objek ke objek lainnya
6. *Reusability*, pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut
7. *Polymorphism*, kemampuan suatu objek untuk digunakan di banyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.

Didalam pemrograman atau pendekatan berorientasi objek terdapat elemen-elemen yang dikenal dengan nama-nama berikut:

1. *Class & object*,
2. Enkapsulasi
3. Atribut
4. Operasi atau Metode (Method)
5. *Package*
6. Antarmuka (*Interface*)[11]

### 2.2.1. Unified Modeling Language

Menurut Aditya Rahmatullah Pratama dari Codepolitan (codepolitan.com), UML (*Unified Modeling Language*) merupakan pengganti dari metode analisis berorientasi *object* dan desain berorientasi objek (OOAD&D / *object oriented analysis and design*) yang dimunculkan sekitar akhir tahun 80-an dan awal tahun 90-an. UML merupakan gabungan dari metode Booch, Rumbaugh (OMT) dan Jacobson. Tetapi UML mencakup lebih luas daripada OOAD. Pada pertengahan saat pengembangan UML, dilakukan standarisasi proses dengan OMG (*Object Management Group*) dengan harapan UML bakal menjadi bahasa standar pemodelan pada masa yang akan datang (yang sekarang sudah banyak dipakai oleh berbagai kalangan).

UML adalah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan *artifacts* (bagian dari informasi yang digunakan untuk dihasilkan oleh proses pembuatan perangkat lunak, *artifact* tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya. Selain itu UML adalah bahasa pemodelan yang menggunakan konsep orientasi *object*. UML dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson di bawah bendera *Rational Software Corps*. UML menyediakan notasi-notasi yang membantu memodelkan sistem dari

berbagai prespektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.

UML digambarkan dengan bantuan diagram-diagram. Diagram-diagram tersebut menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem yang dirancang. Adapun jenis diagram yang digunakan untuk menggambar UML antara lain :

1. *Use Case Diagram*, menggambarkan sejumlah *external actors* dan hubungannya ke *use case* yang diberikan oleh sistem. *Use case* adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari *use case symbol* namun dapat juga dilakukan dalam *activity diagrams*. *Use case* digambarkan hanya yang dilihat dari luar oleh *actor* (keadaan lingkungan sistem yang dilihat *user*) dan bukan bagaimana fungsi yang ada di dalam sistem.
2. *Class Diagram*, menggambarkan struktur statis class di dalam sistem. *Class* merepresentasikan sesuatu yang ditangani oleh sistem. *Class* dapat berhubungan dengan yang lain melalui berbagai cara: *associated* (terhubung satu sama lain), *dependent* (satu class tergantung/menggunakan class yang lain), *specialized* (satu class merupakan

spesialisasi dari class lainnya), atau *package* (grup bersama sebagai satu unit). Sebuah sistem biasanya mempunyai beberapa *class* diagram.

3. *State Diagram*, menggambarkan semua *state* (kondisi) yang dimiliki oleh suatu *object* dari suatu *class* dan keadaan yang menyebabkan *state* berubah. Kejadian dapat berupa *object* lain yang mengirim pesan. *State class* tidak digambarkan untuk semua *class*, hanya yang mempunyai sejumlah *state* yang terdefinisi dengan baik dan kondisi *class* berubah oleh *state* yang berbeda.
4. *Sequence Diagram*, menggambarkan kolaborasi dinamis antara sejumlah *object*. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.
5. *Collaboration Diagram*, menggambarkan kolaborasi dinamis seperti *sequence diagrams*. Dalam menunjukkan pertukaran pesan, *collaboration diagrams* menggambarkan *object* dan hubungannya (mengacu ke konteks). Jika penekannya pada waktu atau urutan gunakan *sequence diagrams*, tapi jika penekanannya pada konteks gunakan *collaboration diagram*.

6. *Activity Diagram*, menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.
7. *Component Diagram*, menggambarkan struktur fisik kode dari komponen. Komponen dapat berupa *source code*, komponen biner, atau *executable component*. Sebuah komponen berisi informasi tentang *logic class* atau *class* yang diimplementasikan sehingga membuat pemetaan dari *logical view* ke *component view*.
8. *Deployment Diagram*, menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak sistem, menunjukkan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya. Di dalam *nodes*, *executable component* dan *object* yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh *node* tertentu dan ketergantungan komponen[12].

### 2.3. Perangkat Lunak Pendukung

Pembuatan aplikasi yang akan dibahas menggunakan sejumlah perangkat lunak pendukung, seperti penggunaan Angular sebagai bahasa pemrograman utama untuk aplikasi yang dibuat, dan penggunaan PHP sebagai bahasa pemrograman yang digunakan untuk pembuatan

*Application Programming Interface* (API) yang dibutuhkan. Selain itu ada juga *framework* yang digunakan yaitu Laravel dan Ionic. MySQL digunakan sebagai *Database Management System* (DBMS) utama untuk menyimpan data. Ada juga CentOS yang digunakan sebagai sistem operasi di *server* dan penggunaan Git untuk manajemen proyek dengan anggota tim lainnya.

### 2.3.1. PHP

Menurut Andre Pratama dari duniaikom.com, PHP adalah bahasa pemrograman script *server-side* yang didesain untuk pengembangan web. Selain itu, PHP juga bisa digunakan sebagai bahasa pemrograman umum (wikipedia). PHP dikembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan sekarang dikelola oleh *The PHP Group*.

PHP adalah bahasa pemrograman yang hanya dijalankan di server atau biasa disebut sebagai *server-side*. Bahasa pemrograman *server-side* harus dijalankan dengan aplikasi khusus seperti Apache dan memerlukan *server* untuk dijalankan, lain halnya dengan yang berjalan di *client-side* yang bisa diakses langsung melalui *browser* atau *shortcut* aplikasi.

Pada awalnya PHP merupakan singkatan dari *Personal Home Page*. Sesuai dengan namanya, PHP digunakan untuk membuat *website* pribadi. Dalam beberapa tahun perkembangannya, PHP

menjelma menjadi bahasa pemrograman *web* yang powerful dan tidak hanya digunakan untuk membuat halaman *web* sederhana, tetapi juga *website* populer yang digunakan oleh jutaan orang seperti wikipedia, wordpress, joomla, dll. Kini PHP adalah singkatan dari *PHP: Hypertext Preprocessor*, sebuah kepanjangan rekursif, yakni permainan kata dimana kepanjangannya terdiri dari singkatan itu sendiri: *PHP: Hypertext Preprocessor*.

Kelebihan PHP dari bahasa pemrograman lain antara lain :

1. Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak dapat melakukan sebuah kompilasi dalam penggunaannya.
2. *Web Server* yang mendukung PHP dapat ditemukan dimana-mana dari mulai apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan *developer* yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan diberbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara runtime melalui *console* serta juga dapat menjalankan perintah-perintah sistem[13].

### 2.3.2. Angular

Menurut Willem Chua dari Binus *University* (socs.binus.ac.id), Angular adalah sebuah *platform* untuk mengembangkan *front-end* dari sebuah aplikasi *web* yang berbasis *open source*. *Platform* ini dikembangkan oleh tim Google dan komunitas *open source* Angular. *Platform* ini menggunakan bahasa Typescript, sebuah pengembangan dari bahasa Javascript yang merupakan bahasa utama yang digunakan untuk *web development*. Angular sendiri merupakan *platform* yang dibangun untuk meningkatkan performa dari AngularJS. *Platform* ini sering disebut juga dengan Angular 2 atau Angular 2+.

*Platform* ini seringkali digunakan untuk membangun *single page application* (SPA). Penggunaan dari SPA ini berarti bahwa segala komponen yang akan digunakan dalam aplikasi *web* tersebut akan di-load dalam *browser* terlebih dahulu agar *user* tidak perlu melakukan *page load* ketika membuka *page* baru. Seringkali SPA ini berupa *bundle* dari *code* Javascript yang dibuka oleh browser secara bersamaan.

Terdapat beberapa fitur dari Angular yang membuatnya menjadi salah satu *platform* yang paling banyak digunakan dalam pengembangan aplikasi *web*, antara lain sebagai berikut:

1. Modularitas

*Framework* Angular menyediakan konsep bahwa komponen-komponen yang dibangun dalam Angular dapat dengan mudah digunakan kembali dalam komponen lainnya tanpa perlu melakukan *coding* dari awal. Modularitas berarti, komponen-komponen umum seperti *card*, *toolbar*, *navbar*, dan *container* dapat dibangun sekali dan digunakan berkali-kali dalam komponen tersebut.

## 2. *Object Oriented Programming*

Penggunaan Typescript memberikan kemampuan bagi *developer* untuk menerapkan konsep *Object Oriented Programming* (OOP). Itu berarti, komponen dalam Angular dapat dikembangkan dengan menerapkan prinsip *class* dan *object*. Hal ini mempermudah *developer* untuk mengembangkan *code* dan lebih menyesuaikan dengan paradigma *programming* yang diterapkan pada bahasa pemrograman seperti Java atau C#.

## 3. *Reactive Programming*

Dalam web *programming*, terkadang hasil *response* dari server muncul secara *asynchronous*. Agar *experience user* tetap terjaga dengan baik, maka kita tidak dapat selalu menggunakan *code* yang dijalankan secara berurutan. Kita juga harus memiliki *code* yang memiliki kemampuan untuk berjalan secara *asynchronous*. Hal ini dapat dibuat dengan menggunakan *tools* bernama *reactive Javascript* (RxJS), yang merupakan salah satu fitur yang dapat digunakan dalam

Angular. Pada RxJS, *observable* digunakan untuk menampung perubahan-perubahan yang terjadi secara *asynchronous* agar *code* dapat berjalan walaupun aplikasi masih melakukan *fetch request* kepada *server*[14].

### 2.3.3. Laravel

Menurut IDCloudHost ([idcloudhost.com](http://idcloudhost.com)), Laravel adalah sebuah *framework* PHP yang dirilis dibawah lisensi MIT, dibangun dengan konsep MVC (*model view controller*). Laravel adalah pengembangan *website* berbasis MVP yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, jelas dan menghemat waktu.

MVC adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari presentasi. MVC memisahkan aplikasi berdasarkan komponen-komponen aplikasi, seperti manipulasi data, *controller*, dan *user interface*.

1. Model, Model mewakili struktur data. Biasanya model berisi fungsi-fungsi yang membantu seseorang dalam pengelolaan basis data seperti memasukkan data ke basis data, pembaruan data dan lain-lain.

2. *View, View* adalah bagian yang mengatur tampilan ke pengguna. Bisa dikatakan berupa halaman web.

3. *Controller, Controller* merupakan bagian yang menjembatani model dan *view*.

Beberapa fitur yang terdapat di Laravel :

1. *Bundles*, yaitu sebuah fitur dengan sistem pengemasan *modular* dan tersedia beragam di aplikasi.

2. *Eloquent ORM*, merupakan penerapan PHP lanjutan menyediakan metode internal dari pola "*active record*" yang mengatasi masalah pada hubungan objek database.

3. *Application Logic*, merupakan bagian dari aplikasi, menggunakan *controller* atau bagian *Route*.

4. *Reverse Routing*, mendefinisikan relasi atau hubungan antara *Link* dan *Route*.

5. *Restful controllers*, memisahkan logika dalam melayani *HTTP GET* dan *POST*.

6. *Class Auto Loading*, menyediakan *loading* otomatis untuk class PHP.

7. *View Composer*, adalah kode unit logikal yang dapat dieksekusi ketika *view* sedang *loading*.

8. *IoC Container*, memungkinkan obyek baru dihasilkan dengan pembalikan *controller*.

9. *Migration*, menyediakan sistem kontrol untuk skema *database*.

10. *Unit Testing*, banyak tes untuk mendeteksi dan mencegah regresi.
11. *Automatic Pagination*, menyederhanakan tugas dari penerapan halaman[15].

#### 2.3.4. Ionic

Menurut penulis, Ionic adalah *framework* yang dapat digunakan untuk mengembangkan aplikasi *mobile* secara *hybrid*, yang dimaksud sebagai *hybrid* adalah kemampuan sebuah program untuk di-*compile* dan dijalankan untuk beberapa *platform* sekaligus tanpa keharusan untuk melakukan pengkodean ulang untuk tiap target *platform*. Hal ini menjadikan aplikasi yang di buat menggunakan Ionic bisa dijalankan di *platform* Android, iOS atau bahkan Windows. Ionic memerlukan Angular atau React Native sebagai pemrograman utamanya, disertai dengan HTML & CSS untuk membangun tampilannya.

#### 2.3.5. MySQL

Menurut Yasin K. dari NiagaHoster (niagahoster.co.id), MySQL adalah sebuah *database management system* (manajemen basis data) menggunakan perintah dasar SQL (*Structured Query Language*) yang cukup terkenal. MySQL merupakan *software open source* dengan dua bentuk lisensi, yaitu *Free Software* (perangkat lunak

bebas) dan *Shareware* (perangkat lunak berpemilik yang penggunaannya terbatas). Jadi MySQL adalah *database server* yang gratis dengan lisensi GNU *General Public License* (GPL) sehingga dapat Anda pakai untuk keperluan pribadi atau komersil tanpa harus membayar lisensi yang ada.

Seperti yang sudah disinggung di atas, MySQL masuk ke dalam jenis RDBMS (*Relational Database Management System*). Maka dari itu, istilah semacam baris, kolom, tabel, dipakai pada MySQL. MySQL akan menyimpan data ke dalam tabel-tabel tertentu dan bisa diakses atau diolah menggunakan SQL. Jadi MySQL adalah *database management system* yang menggunakan bahasa SQL sebagai bahasa penghubung antara perangkat lunak aplikasi dengan *database server*[16].

#### **2.3.6. CentOS**

Menurut Ayoni Sulthon dari DomaiNesia (domainesia.com), CentOS (*Community Enterprise Operating System*) adalah distro Linux yang cocok dipakai dalam skala *Enterprise* atau skala perusahaan yang bebas biaya atau Gratis. CentOS sendiri di code dari *source code Red Hat Enterprise* (RHEL) yang dikembangkan dalam sebuah komunitas yang disebut *CentOS Project*. CentOS ini dirilis di bawah *General Public License* (GPL) dan nantinya dikembangkan oleh komunitas yang disebut *CentOS Project*[17].

### 2.3.7. Git

Menurut Ahmad Muhardian dari Petanikode (petanikode.com), Git adalah salah satu sistem pengontrol versi (*Version Control System*) pada proyek perangkat lunak yang diciptakan oleh Linus Torvalds. Pengontrol versi bertugas mencatat setiap perubahan pada *file* proyek yang dikerjakan oleh banyak orang maupun sendiri. Git dikenal juga dengan *distributed revision control* (VCS terdistribusi), artinya penyimpanan *database* Git tidak hanya berada dalam satu tempat saja.

Semua orang yang terlibat dalam pengkodean proyek akan menyimpan *database* Git, sehingga akan memudahkan dalam mengelola proyek baik *online* maupun *offline*. Dalam Git terdapat *merge* untuk menyebut aktifitas penggabungan kode. Sedangkan pada VCS (*Version Control System*) yang terpusat, *database* disimpan dalam satu tempat dan setiap perubahan disimpan di tempat yang sama[18].

## 2.4. Konsep Jaringan

Konsep jaringan yang digunakan untuk aplikasi yang dibuat adalah jaringan *client server*. Menurut penulis, jaringan *client server* adalah arsitektur jaringan dimana ada dua peran utama yaitu *client* dan *server*. Perangkat *client* akan melakukan permintaan atau request data dan akan dijawab atau diberikan *response* oleh *server*. Hasil *response* yang diberikan akan ditampilkan pada perangkat *client*, baik berupa *web*,

gambar atau data yang kemudian diolah oleh aplikasi pada perangkat *client*.

*Server* terbagi menjadi beberapa jenis berdasarkan tugas atau fungsinya. Seperti *web server* yang khusus untuk menangani *request* halaman *web*, atau *database server* yang menjadi penyimpanan data utama dari suatu *web* atau *API server*. Ini tidak berarti semua jenis *server* diharuskan ada untuk suatu jaringan *client server*, karena kenyataannya satu perangkat *server* bisa digunakan untuk banyak kebutuhan sekaligus. Walaupun idealnya satu perangkat *server* dibuat khusus untuk menangani satu kebutuhan *client*.