

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Tinjauan pustaka dilakukan berdasarkan kajian penelitian-penelitian terdahulu yang dapat menjadi bahan acuan dan kajian dalam melakukan penelitian saat ini. Adapun topik yang dijadikan pembahasan dalam penelitian ini, yaitu mengenai implementasi sistem informasi yang digunakan sebagai media pendukung dalam operasi dan manajemen notulen rapat. Namun dengan topik implementasi sistem informasi telah berbagai macam penelitian yang dilakukan dan telah berbagai macam metode yang digunakan dalam pelaksanaannya. Beberapa jurnal penelitian ini yang dapat dijadikan acuan dan kajian dalam penyusunan tugas akhir ini adalah sebagai berikut.

Menurut Nuraeni (2010) melakukan penelitian dengan judul “Perancangan Sistem Informasi Penjaminan Mutu Perguruan Tinggi Bidang Sumber Daya Manusia” menyebutkan bahwa SDM yang berada di lingkungan perguruan tinggi yang terdiri dari dosen dan tenaga penunjang mempunyai posisi yang strategis dan efektivitas kinerja yang dapat mempengaruhi standarisasi mutu sebuah perguruan tinggi sehingga dapat meningkatkan penjaminan mutu SDM. Salah satu syarat yang harus dipenuhi untuk memenuhi peningkatan penjaminan mutu SDM dengan adanya ketersediaan basis data yang akurat yang dapat digunakan sebagai pengambil keputusan. Oleh karena itu perlu adanya sistem informasi penjaminan mutu SDM yang mempunyai fungsi memudahkan bagi perguruan tinggi dalam menjalankan mutu, sehingga prosesnya dapat dijalankan secara bertahap, dapat didukung oleh pangkalan basis data, dan sistem yang terintegrasi.

Santoso dkk. (2012) dalam penelitiannya mengenai perancangan dan pembuatan sistem informasi manajemen fakultas teknologi industri bertujuan untuk mempermudah melakukan perubahan halaman website yang telah ada dan untuk menyimpan hasil rapat yang dapat diakses secara online dengan menggunakan *database SQL Server 2005*, *Apache 2.2 web server*, dan *PHP 5.26*. Penelitian tersebut menggunakan *content management system (CMS)* karena memudahkan pengguna untuk mengganti halaman website tanpa harus mengganti *script* pemrograman yang telah ada. Selain itu CMS juga tidak menyatukan antara isi dan

desain sehingga tiap bagian bisa memiliki tampilan yang berbeda-beda. Hasil dari penelitian ini juga adalah dapat menyimpan data notulen dan menyediakan fasilitas mengunduh untuk administrasi persuratan tertentu.

Menurut penelitian yang dilakukan oleh Pratiwi, dkk (2018) dengan judul “*Traceability* Kebutuhan dan Perancangan Konseptual Manajemen Rapat Terintegrasi” menyatakan bahwa rapat dapat dipandang menjadi alat sebagai pertukaran informasi dan pengambilan keputusan. Sehingga pengaruh kehadiran peserta karna kurangnya informasi tentang pelaksanaan rapat, diskusi rapat yang belum terfokus ke agenda rapat, dan notulen rapat yang belum lengkap akan menjadi permasalahan dalam pengambilan keputusan dan tindak lanjut hasil rapat itu sendiri. Dengan berbagai permasalahan tersebut muncul usulan untuk melakukan perencanaan perbaikan manajemen rapat dengan diskripsi kebutuhan antarmuka, fungsional dan non-fungsional. Adapun hasil akhir dari penelitian ini yaitu *mapping* kebutuhan terhadap diagram statis dan dinamis yang berhasil dilakukan melalui *traceability matrix* dan *mock-up* aplikasi yang dirancang dengan Sembilan fitur utama.

Rismayani dan Nur (2016) dengan judul “Sistem Informasi Agenda Rapat Berbasis *Web* Menggunakan *SMS Gateway* (Studi Kasus Politeknik TEDC Bandung)” menyatakan bahwa pada Politeknik TEDC Bandung mempunyai agenda rapat yang masih dilakukan secara manual yaitu dengan membagikan surat atau mendatangi langsung ke karyawan. Pada kasus ini peneliti memberikan solusi untuk mengatasi masalah tersebut dengan membuat aplikasi yang membantu untuk membuat agenda rapat dan memberikan kemudahan kepada staf bagian karyawan dalam menyampaikan informasi agenda rapat. Dalam pembuatan aplikasi tersebut peneliti memanfaatkan teknologi *web* dan *SMS gateway* dimana dapat mempermudah staf dalam menyampaikan informasi jadwal agenda rapat via SMS kepada dosen atau karyawan di Politeknik TEDC Bandung. Penulis juga menggunakan Codeigniter yang merupakan salah satu *framework* PHP dalam membangun aplikasi web, serta menjadikan *MySQL* sebagai media penyimpanan data aplikasi web ini.

Sari dan Istikoma (2018) melakukan penelitian dengan judul “Analisis dan Perancangan Sistem Informasi Rapat Online FMIPA UNTAN menggunakan

UML” menyatakan bahwa pelaksanaan rapat pada Fakultas Matematika dan Ilmu Pengetahuan Alam (IPA) masih dilakukan secara konvensional sehingga menimbulkan beberapa kendala antara lain penyebaran undangan rapat yang tidak tepat waktu sampai kepada peserta yang diundang, perubahan jadwal yang berubah secara tiba-tiba sehingga tidak sesuai dengan yang tertera diundangan, adapun kendala yang lain dokumentasi daftar hadir rapat dan notulensi rapat yang tidak tersusun dengan rapih. Dengan kendala tersebut terbentuklah sebuah perancangan aplikasi rapat *online* berbasis website menggunakan UML (*Unified Modeling Language*).

Berdasarkan penelitian-penelitian terdahulu kemudian akan dilakukan penelitian yang berjudul Implementasi Pembuatan *Web* Manajemen Notulensi Rapat Berbasis Sistem Informasi dengan *Framework* Laravel (Studi Kasus: Fakultas Teknik Universitas Muhammadiyah Yogyakarta). Penelitian ini bertujuan untuk mempermudah notulen dalam mengelola data agenda rapat dengan baik. Sistem ini akan dirancang dengan metode MVC (*Model-View-Controller*). *Model-View-Controller* atau MVC adalah sebuah metode untuk membuat aplikasi dengan memisahkan data (*Model*) dari tampilan (*View*) dan cara bagaimana memprosesnya (*Controller*). Website ini akan dibangun dengan menggunakan *framework* Laravel yang mendukung metode MVC. Kelebihan menggunakan metode MVC adalah kemudahan dalam proses pengeditan web terutama bagian tampilan (*view*). Dan untuk kelebihan Laravel adalah memberikan kemudahan dan kecepatan dalam hal mendesain dan *coding* halaman web dikarenakan memiliki *script* yang sederhana, tidak harus mengerjakan dari nol, dapat digunakan dengan mudah dan untuk keamanannya pun dibanding *framework* lain masih jauh lebih baik.

2.2 Landasan Teori

2.2.1 Sistem Informasi

Menurut Laudon (2012:16) menyatakan bahwa sistem informasi merupakan kumpulan kompone-komponen yang saling berkaitan untuk bekerja sama dalam mengumpulkana, mengolah, menyimpan, dan menampilkan informasi sehingga dapat mendukung dalam mengambil keputusan, koordinasi, pengaturan, Analisa, dan visualisasi pada sebuah organisasi atau sistem.

2.2.2 Sistem Informasi Manajemen

Menurut McLeod dan Schell (2008:40) yang menyatakan bahwa sistem berbasis komputer yang memberikan informasi kepada pengguna sesuai kebutuhan pengguna disebut sistem informasi manajemen. Adapun bentuk dari informasi yang diterima dapat berupa laporan periodik, laporan khusus dan *output* dari simulasi informasi yang ada.

2.2.3 Unified Modeling Language (UML)

Unified Modeling Language merupakan sebuah pemodelan yang diterapkan ke sistem atau perangkat lunak dengan konsep berorientasi objek. Dengan adanya pemodelan ini yang dapat menyederhanakan masalah yang kompleks atau rumit sehingga dapat lebih mudah dipahami dan dipelajari (Nugroho, 2010:6). Pada UML terdapat beberapa model yang dapat dipakai dalam merancang sebuah perangkat lunak atau sistem antara lain *Use Case Diagram*, *Class Diagram* dan *Activity Diagram*. Berikut adalah penjelasannya:

a. Use Case Diagram

Use Case Diagram menggambarkan aktivitas-aktivitas dan interaksi yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. Dengan kata lain, menggambarkan akan siapa saja yang menggunakan sistem dan dengan cara apa pengguna untuk berinteraksi dengan sistem (Whitten dkk, 2004:257). Simbol-simbol yang terdapat pada *Use Case Diagram* seperti tabel 2.1.

Tabel 2.1 Simbol-simbol Use Case Diagram

No.	Gambar	Nama Gambar	Keterangan
1.		<i>Use case</i>	Merupakan fungsionalitas yang disediakan sistem sebagai unit yang bertukar pesan dengan aktor.

Tabel 2.1 Simbol-simbol Use Case Diagram (Lanjutan)

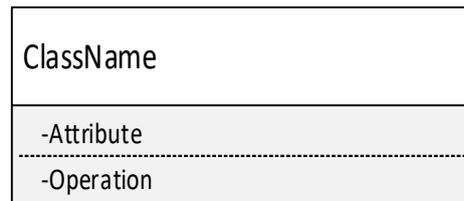
No.	Gambar	Nama Gambar	Keterangan
2.		<i>Actor</i>	Merupakan penggambaran abstrak dari orang yang mengaktifkan fungsi dari target sistem dan merupakan orang yang berinteraksi dengan <i>use case</i> .
3.		<i>Association</i>	Digambarkan dengan garis tanpa panah mengindikasikan siapa yang berinteraksi secara langsung dengan sistem.
4.		<i>Include</i>	Mengidentifikasi hubungan antar dua use case dimana satu use case memanggil use case yang lain.
5.		<i>Extend</i>	Merupakan perluasan dari use case jika kondisi atau syarat terpenuhi.

b. Class Diagram

Menurut Permana (2015:38) *class* diagram merupakan sebuah spesifikasi yang jika di dekskripsikan akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class Diagram* memberikan pandangan secara luas dari suatu sistem dengan menunjukkan kelas-kelasnya dan

hubungan antar kelas tersebut. *Class Diagram* seperti pada gambar 2.2 memiliki 3 komponen penting antara lain:

1. Nama, merupakan nama dari sebuah kelas.
2. Atribut, merupakan properti dari sebuah kelas.
3. Operasi, merupakan hal yang bisa dilakukan oleh sebuah kelas terhadap kelas lain.



Gambar 2.2 *Class Diagram*

Dalam suatu *class diagram* atribut dan *method* dapat memilih salah satu sifat sebagai berikut:

1. *Private*, tidak dapat dipanggil dari luar kelas yang bersangkutan.
2. *Protected*, hanya bisa dipanggil oleh kelas yang bersangkutan dan anak kelas yang mewarisinya.
3. *Public*, dapat dipanggil oleh siapa saja.

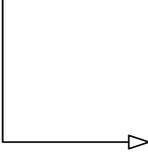
Class diagram juga memiliki hubungan antar *class* yang dapat menghubungkan satu *class* dengan *class* lain. Berikut hubungan antar *class*:

1. Asosiasi (*Association*), yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah query antar *class*.
2. Agregasi (*Aggregation*), yaitu hubungan yang menyatakan bagian (“terdiri atas..”). Dapat dijelaskan dengan kasus contoh sebuah mobil, dimana mobil mempunyai kaca spion, mobil tetap berfungsi tanpa adanya kaca spion, sehingga mobil dapat dipisahkan dari kaca spion.
3. Pewarisan (*Inheritance*), yaitu hubungan hirarki antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metode *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya.
4. Generalisasi (*Generalization*) merupakan kebalikan dari pewarisan.

5. Komposisi (*Composition*), yaitu hubungan yang menyatakan hubungan (“bagian dari..”). dapat dijelaskan dengan kasus contoh mobil, dimana mobil memiliki mesin. Maka dapat dikatakan mesin merupakan bagian dari mobil. Jika mesin tidak ada sebuah mobil tidak akan berjalan atau berfungsi sebagai mobil.

Simbol-simbol yang terdapat pada Class Diagram dapat dilihat pada tabel 2.2.

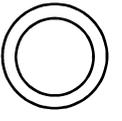
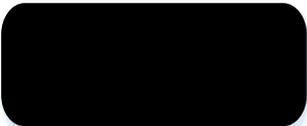
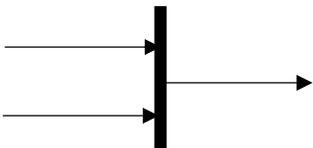
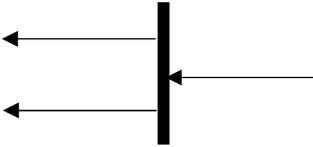
Tabel 2.2 Simbol-Simbol Hubungan Antar Class Diagram

No.	Gambar	Nama Gambar	Keterangan
1.		<i>Association</i>	Garis yang menghubungkan class satu dengan class lainnya dengan makna umum.
2.		<i>Aggregation</i>	Menghubungkan antar class dengan makna untuk semua bagian.
3		<i>Inheritance / Generalization</i>	Menunjukkan operasi pewarisan atau generalisasi pada suatu class
4.		<i>Composition</i>	Menunjukkan hubungan antar <i>class</i> dimana agresi yang menjadi bagian diciptakan setelah <i>class</i> menjadi <i>whole</i> dibuat.

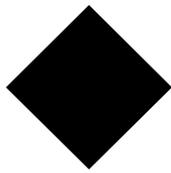
c. Activity Diagram

Activity Diagram merupakan bentuk alur kerja yang menggambarkan aktifitas sistem atau sebuah aplikasi yang mirip dengan *flowchart* (Permana, 2015:42). Dalam UML (*Unified Modeling Language*) biasanya digunakan untuk menjelaskan aktifitas komputer atau aktifitas organisasi. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel 2.3.

Tabel 2.3 Simbol-simbol Activity Diagram

No.	Gambar	Nama Gambar	Keterangan
1.		<i>Start Point</i>	Bagaimana objek dibentuk dari awal.
2		<i>End Point</i>	Bagaimana objek dibentuk dan diakhiri
3.		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
4.		<i>Join</i> (Penggabungan)	Digunakan untuk membuat penggabungan action dalam kondisi tertentu.
5.		<i>Fork</i> (Percabangan)	Digunakan untuk membuat percabangan action dalam kondisi tertentu.

Tabel 2.3 Simbol-simbol Activity Diagram (Lanjutan)

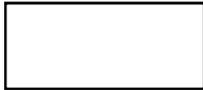
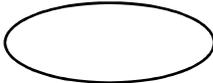
No.	Gambar	Nama Gambar	Keterangan
6.		<i>Decision</i>	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu

2.2.4 Entity Relationship Diagram (ERD)

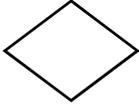
Entity relationship diagram merupakan model data yang menggunakan beberapa notasi untuk menggambarkan data dalam konteks entitas dan hubungan yang didekskripsikan oleh data tersebut (Whitten dkk, 2004:281).

ERD biasanya digunakan untuk memodelkan data yang nantinya akan dikembangkan menjadi *database*. Dengan model data ini juga dapat membantu saat melakukan analisis dan perancangan *database* karena dapat menunjukkan bermacam-macam data yang dihubungkan dan hubungan antar data. Komponen-komponen dalam ER-Diagram dapat dilihat pada tabel 2.4.

Tabel 2.4 Simbol-simbol Entity Relationship Diagram

No.	Gambar	Nama Gambar	Keterangan
1.		<i>Entity</i>	Objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain.
2.		<i>Attribute</i>	Atribut yang dimiliki entitas, yang mendeskripsikan karakteristik dari entitas tersebut.

Tabel 2.4 Simbol-simbol Entity Relationship Diagram (Lanjutan)

No.	Gambar	Nama Gambar	Keterangan
3.		<i>Relationship</i>	Hubungan yang terjadi antar entitas yang berbeda.

2.2.5 Normalisasi

Normalisasi adalah teknik analisis data yang mengatur atribut data dalam kelompok untuk membentuk entitas yang nonredundan, stabil, fleksibel dan mudah beradaptasi (Whitten dkk, 2004:307). Tahapan proses sistematis yang dilakukan mulai dari bentuk tidak normal menjadi bentuk normal memiliki suatu syarat yang harus dipenuhi pada saat menuju yang lebih baik, adapun syarat-syaratnya:

1. Semakin berkurang bentuk tidak normal.
2. Semakin banyak jumlah tabel.
3. Semakin tidak terdapat *anomaly insert, update* dan *delete*.

2.2.6 System Development Life Cycle (SDLC)

System development life cycle biasanya juga disebut *system life cycle* merupakan metodologi yang implementasikan ke dalam suatu sistem atau perangkat lunak dengan meliputi tahap perancangan, pemeliharaan dan pengembangan serta penggunaan (Kadir, 2014:344). SDLC mempunyai beberapa model yang dapat diterapkan, seperti model RAD (*Rapid Application Development*), model iteratif, model *prototype*, model spiral, model *waterfall* dan lainnya.

Model *waterfall* menurut Rosa dan Shalahuddin (2011:26-29) menyebutkan bahwa model *waterfall* merupakan model pengembangan sebuah sistem yang memiliki tahapan sekuensial atau terurut. Adapun tahapan-tahapan yang terdapat dalam model *waterfall* diantaranya *planning, analysis, design, coding, testing* dan *implementation*. Setiap tahapan akan diselesaikan secara berurutan dimana tahapan akan berjalan jika tahapan sebelumnya sudah dijalankan. Keuntungan yang didapatkan dalam menggunakan model *waterfall* dalam membangun sistem ini yaitu memiliki proses yang berurut, dokumentasi yang jelas dari setiap tahapan dan setiap tahapan tidak saling tumpang tindih.

2.2.7 Web

Web merupakan aplikasi yang sangat pesat di zaman sekarang ini. Aplikasi Web pertama kali dibuat dengan menggunakan bahasa pemrograman yang disebut HTML (*HyperText Markup Language*) dan dengan memakai protokol yang bernama HTTP (*HyperText Transfer Protocol*) (Kadir, 2005:3). Menurut Kadir (2004:2) banyak contoh aplikasi web yang telah diterapkan seperti berikut ini:

- 1) Mesin pencari atau *search engine* (altavista, google, dan lain-lain).
- 2) Toko online (bookpool, amazon, dan lain-lain).
- 3) Lelang online (eBay).
- 4) Situs-situs berita (detik.com, cnn, dan lain-lain).
- 5) Layanan akademis perguruan tinggi.

2.2.8 Web Editor

Menurut Arief (2011:19) dalam membangun sebuah web diperlukannya web editor yang merupakan program aplikasi yang mempunyai fungsi untuk mengetikkan perintah-perintah dokumen web baik *client-side scripting* maupun *server-side scripting*. Pada penulisan tugas akhir ini program aplikasi web editor yang akan digunakan dalam membangun web ini yaitu Visual Studio Code.

2.2.9 Web Browser

Menurut Arief (2011:19) web browser adalah sebuah program yang berfungsi untuk menampilkan dokumen-dokumen web dalam format HTML. Berikut contoh web browser yang beredar saat ini beserta web engine yang digunakan:

- 1) Web Engine Webkit: Safari, Google Chrome.
- 2) Web Engine Trident: Microsoft Internet Explorer, Maxthon.
- 3) Web Engine Gecko: Mozilla Firefox.
- 4) Web Engine Presto: Opera.

2.2.10 Web Server

Web server menurut Kadir (2005:4) adalah server yang melayani permintaan klien terhadap halaman web. Adapun beberapa contoh perangkat lunak web server yaitu, Apache, IIS (*Internet Information Server*) dan Xitami. Dalam penelitian ini web server yang akan digunakan adalah Apache yang mendukung bahasa pemrograman PHP.

2.2.11 Database Server

Meneurt Arief (2011:20) *database* server adalah program yang digunakan untuk menyimpan data yang akan diolah di halaman web. *Database* biasanya disebut *Database Management System* (DBMS) merupakan sebuah aplikasi yang menjembatani user dan data-data di dalam *database*. Jenis DBMS yang digunakan adalah berarsitektur client-server. Aplikasi web merupakan aplikasi web client-server yang dapat diakses oleh banyak user dan berbagai macam tempat dalam mengakses web tersebut sepanjang user terkoneksi dengan internet, oleh karena itu jenis *database* server yang harus digunakan dalam membuat aplikasi web yaitu *Database Management System* (DBMS). Ada beberapa hal yang harus diperhatikan dalam menentukan *database* yang akan digunakan dalam menyimpan data-data di halaman web:

- 1) Jenis DBMS yang digunakan adalah yang berbasis *client-server* / *database* server.
- 2) Sistem operasi yang digunakan yang dapat mendukung banyak sistem operasi.
- 3) Open source atau Closed source.
- 4) File instalasi yang dipilih, apakah yang versi install (EXE) atau yang versi terkompres (ZIP atau RAR).
- 5) Platform prosesor, apakah berbasis 32-bit atau 64-bit.

Sehingga dalam penelitian ini, peneliti mengambil keputusan untuk memakai program aplikasi MySQL dalam sebagai *database* server yang akan digunakan.

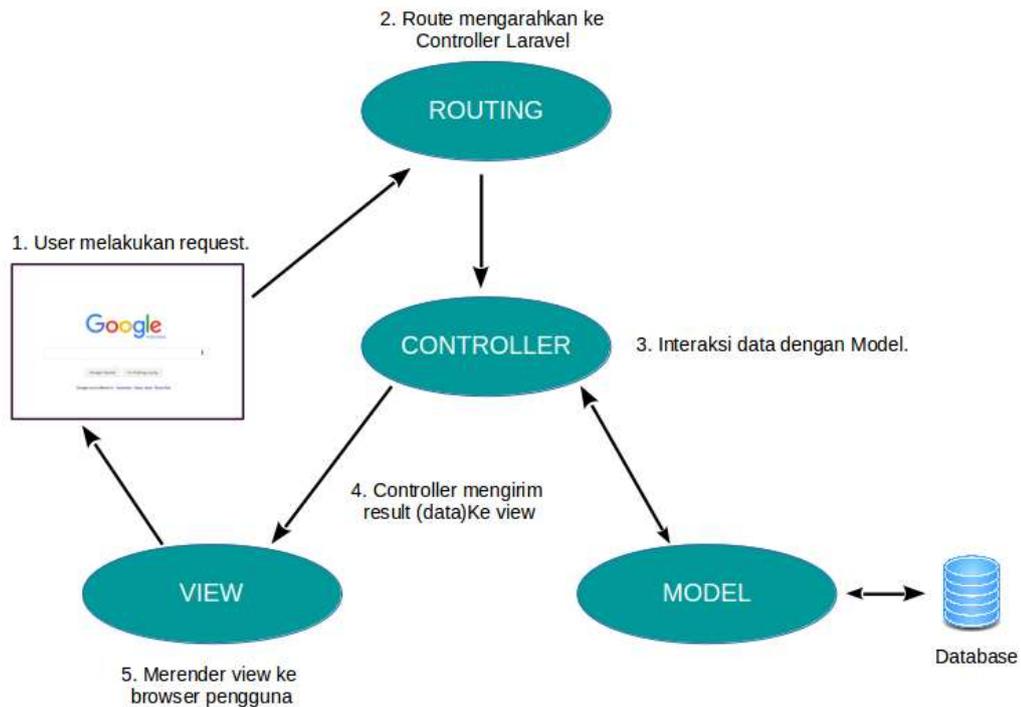
2.2.12 PHP

Menurut Saputra (2018:2) PHP mempunyai kepanjangan yaitu *Hypertext Preprocessors* dimana huruf “P” di kata PHP itu sendiri. PHP juga merupakan bahasa pemrograman *server-side script* yang digunakan dalam pengembangan aplikasi berbasis *web*.

Bahasa pemrograman PHP memiliki beberapa kelebihan yaitu bersifat *open source* dimana pengguna bebas menggunakan program PHP secara gratis tanpa memberikan *royalty*, memiliki komunitas yang besar dan mudah dipelajari. Dengan mempertimbangkan beberapa alasan, penulis menetapkan bahasa pemrograman PHP yang dipakai dalam membangun proyek sistem informasi ini.

2.2.13 Laravel

Laravel adalah sebuah *framework* PHP yang bersifat *opensource* (gratis) dan dikembangkan oleh Taylor Otwell dengan lisensi dibawah MIT *license*. Dibuatnya larvael untuk membantu para *developer* dalam kasus membuta *web* dengan bahasa pemrograman yang sederhana, elegan, ekspresif, dan menyenangkan (Nugraha, 2014). Berikut diagram kerja *framework* laravel.



Gambar 2.1 Diagram Kerja Framework Laravel

Gambar 2.1 merupakan diagram kerja *framework* laravel dimana *user* melakukan *request* kepada sistem kemudian *route* mengarahkan *request* dari *user* ke *controller* laravel. *Controller* berinteraksi dengan model yang digunakan untuk mengakses data dari *database* dan dikirimkan kembali ke *controller* untuk mengirim hasil ke *view*. *View* memberikan hasil berupa tampilan dan data kepada *user* atau pengguna.