

**PENGEMBANGAN SISTEM PENGELOLAAN INFORMASI KEUANGAN  
MENGUNAKAN TEKNOLOGI RESTFUL API DAN METODE MVC  
PADA PESANTREN ROUDHOTUL JANNAH MAKASSAR**

**SKRIPSI**



**NUR AGHNI FAUZI ABDULLAH**

**H13116018**

**PROGRAM STUDI SISTEM INFORMASI**

**DEPERTEMEN MATEMATIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS HASANUDDIN**

**MAKASSAR**

**2023**

**PENGEMBANGAN SISTEM PENGELOLAAN INFORMASI  
KEUANGAN MENGGUNAKAN TEKNOLOGI RESTFUL API  
DAN METODE MVC PADA PESANTREN ROUDHOTUL  
JANNAH MAKASSAR**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Sains  
Pada Program Studi Sistem Informasi Departemen Matematika Dan Ilmu  
Pengetahuan Alam Universitas Hasanuddin**

**NUR AGHNI FAUZI ABDULLAH**

**H13116018**

**PROGRAM STUDI SISTEM INFORMASI**

**DEPERTEMEN MATEMATIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS HASANUDDIN**

**MAKASSAR**

**2023**

## LEMBAR PERNYATAAN KEOTENTIKAN

Yang bertanda tangan dibawah ini

Nama : Nur Aghni FAuzi Abdullah

NIM : H13116018

Program Studi : Sistem Informasi

Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul :

**Pengembangan Sistem Pengelolaan Informasi Keuangan  
Menggunakan Teknologi Restful API dan Metode MVC Pada  
Pesantren Roudhotul Jannah Makassar**

Adalah karya tulisan saya sendiri, bukan merupakan pengambil alihan tulisan oranglain dan bahwa skripsi yang saya tulis ini benar – benar merupakan hasil karya saya sendiri.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka sayabersedia menerima sanksi atas perbuatan tersebut.

Makassar, 24 Juli 2023



  
**NUR AGHNI FAUZI ABDULLAH**

H13116018

**PENGEMBANGAN SISTEM PENGELOLAAN INFORMASI  
KEUANGAN MENGGUNAKAN TEKNOLOGI RESTFUL API  
DAN METODE MVC PADA PESANTREN ROUDHOTUL  
JANNAH MAKASSAR**

Disusun dan diajukan oleh :

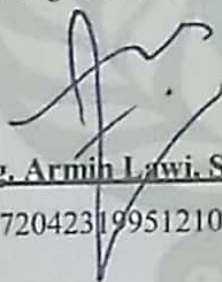
**NUR AGHNI FAUZI ABDULLAH**

**H13116018**

Telah dipertahankan dihadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana pada Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin dan dinyatakan telah memenuhi syarat kelulusan

**Menyetujui**

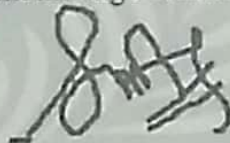
Pembimbing Utama,



**Dr. Eng. Armin Lawi, S.Si., M.Eng.**

NIP. 197204231995121001

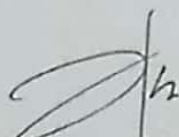
Pembimbing Pertama



**Supri Bin Hi Amir, S.Si., M.Eng**

NIP. 198805042019031012

**Ketua Program Studi,**



**Dr. Hendra, S.Si., M.Kom**

NIP. 197601022002121001



## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Nur Aghni Fauzi Abdullah

NIM : H13116018

Program Studi : Sistem Informasi

Judul Skripsi : Pengembangan Sistem Pengelolaan Informasi Keuangan Menggunakan Teknologi Restful API dan Metode MVC Pada Pesantren Roudhotul Jannah Makassar

Telah berhasil mempertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

### DEWAN PENGUJI

		Tanda Tangan
Ketua	: Dr. Eng. Armin Lawi, S.Si., M.Eng	(.....)
Sekretaris	: Supri Bin Hj Amir, S.Si., M.Eng	(.....)
Anggota	: Dr. Hendra, S.Si., M.Kom	(.....)
Anggota	: Edy Saputra, S.Si., M.Si	(.....)

Ditetapkan di : Makassar  
Tanggal : 24 Juli 2023



## KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT, atas segala limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul **“Pengembangan Sistem Pengelolaan Informasi Keuangan Menggunakan Teknologi Restful API dan Metode MVC Pada Pesantren Roudhotul Jannah Makassar”**. Pembuatan skripsi ini merupakan salah satu syarat untuk penyelesaian studi penulis pada jenjang pendidikan Strata Satu Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Hasanuddin.

Selama proses penyelesaian tugas akhir ini penulis menerima begitu banyak bantuan dari berbagai pihak baik berupa materi maupun non materi, untuk itu pada kesempatan kali ini penulis mengucapkan segala wujud bakti dan kasih sayang kupersembahkan skripsi ini terkhusus kepada kedua orang tua tercinta, Ibunda **Dra. Muryanti** dan Ayahanda **Drs. Abdullah Sammana**, terima kasih atas segala pengorbanan, kesabaran, doa, dan dukungan yang tak ternilai hingga penulis dapat menyelesaikan studi. Terima kasih yang sebesar- besarnya kepada:

1. Rektor Universitas Hasanuddin Makassar **Prof. Dr. Ir. Jamaluddin Jompa, M.Sc.** dan seluruh Wakil Rektor dalam Lingkungan Universitas Hasanuddin.
2. Bapak Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam **Dr. Eng Amiruddin** dan para Wakil Dekan serta seluruh staff yang telah memberikan bantuan selama penulis mengikuti pendidikan di FMIPA Universitas Hasanuddin
3. Bapak **Prof. Dr. Nurdin, S.Si., M.Si** selaku Ketua Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam.
4. Bapak **Dr. Hendra, S.Si., M.Kom.** sebagai Ketua Program Studi Sistem Informasi Universitas Hasanuddin dan selaku penguji atas masukan pada penelitian yang telah dilakukan oleh penulis.
5. Bapak **Dr. Eng. Armin Lawi, S.Si., M.Eng.** selaku dosen pembimbing utama atas segala ilmu, bantuan, saran, nasehat, dan motivasi yang telah diberikan selama proses menjalani pendidikan serta kesabaran dalam membimbing penulis dalam proses penyusunan skripsi ini.

6. Bapak **Supri Bin Hj Amir, S.Si, M.Eng.** sebagai dosen pembimbing pertama yang telah memberi bantuan, saran, motivasi, serta segala ilmunya dalam membimbing penulis dalam proses penyusunan skripsi ini.
7. Bapak **Edy Saputra, S.Si., M.Si.** dan sebagai penguji atas masukan pada penelitian yang telah dilakukan oleh penulis.
8. Keluarga besar om **Dhomdhom Hudi Akbar** yang telah banyak membantu dari awal sampai akhir pengerjaan skripsi dan memberikan masukan dalam penulisan skripsi
9. Keluarga besar **Sistem Informasi Unhas 2016** yang selalu mendukung dan membantu penulis selama menjalani pendidikan.
10. Semua pihak yang telah memberikan bantuan kepada penulis baik berupa materi dan non materi yang tidak dapat penulis sebutkan satu per satu, terima kasih untuk bantuan dan dukungannya.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan karena keterbatasan penulis. Oleh karena itu, saran dan kritik demi penyempurnaan skripsi ini sangat penulis harapkan. Akhir kata, semoga apa yang disajikan dalam skripsi ini dapat bermanfaat bagi setiap yang membacanya.

Makassar 24 Juli 2023



Penulis

**PERNYATAAN PERSETUJUAN REPUBLIK TUGAS AKHIR  
UNTUK KEPENTINGAN AKADEMIS**

---

---

Sebagai civitas akademik Universitas Hasanuddin, saya yang bertanda tangan di bawah ini:

Nama : Nur Aghni Fauzi Abdullah  
NIM : H13116018  
Program Studi ; Sistem Informasi  
Departemen : Matematika  
Fakultas : Matematika dan Ilmu Pengetahuan Alam  
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul:

**“Pengembangan Sistem Pengelolaan Informasi Keuangan Menggunakan Teknologi Restful API dan Metode MVC Pada Pesantren Roudhotul Jannah Makassar”.**

beserta perangkat yang ada (jika diperlukan). Terkait dengan hal di atas, maka pihak universitas berhak menyimpan, mengalih-media/format-kan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di Makassar pada tanggal 24 Juli 2023

Yang Menyatakan



( Nur Aghni Fauzi Abdullah )



## ABSTRAK

Pondok Pesantren Roudhotul Jannah Makassar merupakan sekolah swasta yang bergerak dijenjang pendidikan SMP, SMA dan Non-Pendidikan. Dalam operasional pondok pesantren agar dapat berjalan dengan baik salah satu hal yang harus dapat dikelola dengan baik adalah informasi administrasi keuangan sekolah. System pengolahan keuangan yang berjalan saat ini masih pada proses pencatatan secara *konvensional* dalam hal transaksi pembayaran SPP.

Berdasarkan permasalahan tersebut maka dibutuhkan aplikasi informasi administrasi keuangan. Metode yang digunakan untuk membangun aplikasi informasi administrasi keuangan adalah metode RESTFull API, MVC (*Model View Controller*) dengan menggunakan UML (*Unified Modelling Language*) untuk menggambarkan model sistem yang dapat melakukan proses administrasi pembayaran sekolah menjadi lebih mudah, menyajikan informasi terkait rekapitulasi pembayaran SPP siswa, memudahkan siswa dalam melihat status pembayaran kapanpun dan dimanapun, dan meminimalisir adanya kesalahan dalam mengisi data yang dilakukan secara manual

Aplikasi informasi keuangan Pondok Pesantren Roudhotul Jannah Makassar mampu melakukan proses pencatan , pengolahan data, pembuatan laporan keuangan secara otomatis dan cepat.

**Kata Kunci :** Aplikasi, system informasi, RESTFull API, MVC, UML

## ABSTRACT

Pondok Pesantren Roudhotul Jannah Makassar is a private school engaged in junior high school, high school and non-educational education. In the operation of Islamic boarding schools so that they can run well, one of the things that must be managed properly is information on school financial administration. The current financial processing system is still in the conventional recording process in terms of SPP payment transactions.

Based on these problems, the application of financial administration information is needed. The method used to build financial administration information applications is the RESTFull API method, MVC (Model View Controller) using UML (Unified Modeling Language) to describe a system model that can make school payment administration processes easier, present information related to student tuition payment recapitulation, make it easier for students to see payment status anytime and anywhere, and minimize errors in filling in data that is done manually.

The application of financial information at the Roudhotul Jannah Makassar Islamic Boarding School is capable of carrying out the recording process, data processing, making financial reports automatically and quickly.

Keywords: Application, information system, RESTFull API, MVC, UML

**DAFTAR ISI**

HALAMAN JUDUL..... i

LEMBAR PERNYATAAN KEOTENTIKA..... iii

LEMBAR PERSETUJUAN..... iv

HALAMAN PENGESAHAN..... v

KATA PENGANTAR..... vi

PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK  
KEPENTINGAN AKADEMIS..... viii

ABSTRAK.....ix

ABSTRACT..... x

DAFTAR ISI..... xi

DAFTAR GAMBAR..... xiv

DAFTAR TABEL..... xvii

DAFTAR LAMPIRAN..... xviii

BAB I PENDAHULUAN ..... 1

    1.1 Latar Belakang..... 1

    1.2 Rumusan Masalah..... 3

    1.3 Tujuan Penelitian..... 3

    1.4 Batasan Masalah..... 3

    1.5 Manfaat Penelitian..... 3

BAB II TINJAUAN PUSTAKA..... 4

    2.1 Sistem Informasi Pengelolaan Keuangan..... 4

    2.2 *Unified Modelling Language (UML)*..... 5

        2.2.1 *Class Diagram*..... 6

        2.2.2 *Use Case Diagram*..... 8

2.2.3	<i>Activity Diagram</i> .....	9
2.2.4	<i>Sequence Diagram</i> .....	10
2.3	<i>Entity Relationship Diagram (ERD)</i> .....	11
2.4	RESTFUL API.....	12
2.5	Konsep MVC ( <i>Model-View-Controller</i> ).....	14
2.6	<i>Framework Laravel</i> .....	15
2.7	<i>Hypertext Preprocessor (PHP)</i> .....	15
2.8	<i>Apache</i> .....	16
2.9	MariaDB.....	16
2.10	<i>Black Box Testing</i> .....	16
BAB III METODE PENELITIAN.....		17
3.1	Waktu dan Lokasi Penelitian.....	17
3.2	Alat Pendukung Penelitian.....	17
3.3	Metode Pengumpulan Data.....	17
3.3.1	Observasi.....	18
3.3.2	Wawancara.....	18
3.3.3	Internet.....	18
3.3.4	Study Literatur.....	18
3.4	Metode Pengembangan Sistem.....	18
3.5	Prosedur Pengembangan Sistem.....	20
3.5.1	Analisis Kebutuhan.....	20
3.5.2	Perancangan Sistem.....	20
3.5.3	Implementasi.....	21
3.5.4	Pengujian Sistem.....	21
3.5.5	Pemeliharaan Sistem.....	21
BAB IV HASIL DAN PEMBAHASAN.....		22
4.1	Hasil Pengembangan Sistem.....	22
4.2	Analisis Kebutuhan Sistem.....	22

4.3	Perancangan Sistem.....	25
4.3.1	Perancangan <i>Use Case Diagram</i> .....	25
4.3.2	Perancangan <i>Activity Diagram</i> .....	29
4.3.3	Perancangan <i>Entity Relational Diagram</i> (ERD).....	38
4.3.4	Perancangan Struktur Tabel.....	39
4.4	Rancangan Antarmuka/ <i>Interface</i> .....	46
4.5	Pengujian Sistem.....	66
4.5.1	Pengujian Aplikasi Halaman <i>Login</i> .....	67
4.5.2	Pengujian Aplikasi Admin.....	67
4.5.3	Pengujian Aplikasi Halaman Siswa.....	69
4.6	Konsep MVC dan Restful API.....	70
4.6.1	Konsep MVC (Model-View-Controller).....	71
4.6.2	Teknologi Restful API.....	73
BAB V SIMPULAN DAN SARAN.....		77
5.1	Simpulan.....	77
5.2	Saran.....	77
DAFTAR PUSTAKA.....		78
LAMPIRAN.....		80

**DAFTAR GAMBAR**

Gambar 2.1 Arsitektur RESTFUL API..... 13

Gambar 2.2 Konsep MVC..... 15

Gambar 3.1 *Waterfall Model* (S.Pressman, 2014)..... 19

Gambar 4.1 *Use Case Diagram Admin*..... 26

Gambar 4.2 *Use Case Diagram Petugas Sekolah*..... 27

Gambar 4.3 *Use Case Diagram Siswa*..... 28

Gambar 4.4 *Activity Diagram Mengelolah Admin*..... 29

Gambar 4.5 *Activity Diagram Mengelola Tahun Ajaran*..... 30

Gambar 4.6 *Activity Diagram Mengelola Data Role*..... 31

Gambar 4.7 *Activity Diagram Mengelola Data Kelas*..... 32

Gambar 4.8 *Activity Diagram Mengelola Data Siswa*..... 33

Gambar 4.9 *Activity Diagram Memproses Pembayaran*..... 34

Gambar 4.10 *Activity Diagram Melihat Status Pembayaran*..... 35

Gambar 4.11 *Activity Diagram Melihat Rekapitulasi*..... 35

Gambar 4.12 *Activity Diagram User Permission*..... 36

Gambar 4.13 *Activity Diagram Mengelola Role Permissin*..... 36

Gambar 4.14 *Activity Diagram Mengelola data SPP*..... 37

Gambar 4.15 Perancangan *Entity Relational Diagram*..... 38

Gambar 4.16 Tabel Users..... 39

Gambar 4.17 Tabel Petugas..... 40

Gambar 4.18 Tabel <i>Permission</i> .....	41
Gambar 4.19 Tabel Siswa.....	41
Gambar 4.20 Tabel Kelas.....	42
Gambar 4.21 Tabel Pembayaran.....	43
Gambar 4.22 Tabel SPP.....	43
Gambar 4.23 Tabel Role.....	44
Gambar 4.24 Tabel Model <i>Has Rules</i> .....	44
Gambar 4.25 Tabel Model <i>Has Permission</i> .....	45
Gambar 4.26 Tabel <i>Role Has Permission</i> .....	45
Gambar 4.27 Tabel <i>Migration</i> .....	46
Gambar 4.28 Rancangan Antarmuka Beranda.....	47
Gambar 4.29 Rancangan Antar Muka Login.....	48
Gambar 4.30 Rancangan Antarmuka Dashboard.....	50
Gambar 4.31 Rancangan Antar Muka Pembayaran.....	51
Gambar 4.32 Rancangan Antarmuka Rekapitulasi.....	53
Gambar 4.33 Rancangan Antarmuka Keuangan SPP.....	54
Gambar 4.34 Rancangan Antarmuka Tahun Pembayaran.....	56
Gambar 4.35 Rancangan Antarmuka Data Pengguna.....	57
Gambar 4.36 Rancangan Antarmuka SPP Tahun Ajaran.....	59
Gambar 4.37 Rancangan Antarmuka Data Petugas.....	60
Gambar 4.38 Rancangan Antarmuka Data Kelas.....	62
Gambar 4.39 Rancangan Antarmuka Data Siswa.....	63

Gambar 4.40 Rancangan Antarmuka Status Pembayaran.....	65
Gambar 4.41 Antarmuka User Permission.....	65



**DAFTAR TABEL**

Tabel 2.1 Simbol <i>Class Diagram</i> .....	7
Tabel 2.2 Simbol <i>Use Case Diagram</i> .....	8
Tabel 2.3 Simbol <i>Activity Diagram</i> .....	9
Tabel 2.4 Simbol <i>Sequence Diagram</i> .....	10
Table 4.1 Pengujian Aplikasi Login.....	67
Table 4.2 Pengujian Aplikasi Admin.....	67
Table 4.3 Pengujian Aplikasi Halaman Siswa.....	69

**DAFTAR LAMPIRAN**

Lampiran 1 Source Code Program Model.....	81
Lampiran 2 Source Code Program Controller.....	88
Lampiran 3 Source Code Program View.....	105
Lampiran 4 Tampilan Aplikasi.....	113

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Perkembangan dalam bidang ilmu pengetahuan dan teknologi berkembang dengan tujuan untuk memberikan kemudahan bagi manusia dalam melaksanakan tugas dan jadi kepentingannya. Begitu banyak bagian ilmu pengetahuan dan teknologi yang mengalami perkembangan yang begitu pesatnya, hal itu didukung oleh sumber daya alam dan sumber daya manusia yang memadai dan berkualitas. Pemanfaatan Teknologi Informasi dan Komunikasi (TIK) dapat diaplikasikan ke dalam berbagai aspek guna memudahkan pekerjaan manusia yang semula dikerjakan dalam bentuk manual berubah menjadi sistem yang telah terkomputerisasi. Salah satu contohnya adalah pada bidang pelayanan administrasi di berbagai instansi, sekolah adalah salah satu contoh instansi yang masih menggunakan sistem administrasi secara manual. Teknologi informasi berbasis *web* telah memberikan banyak kemudahan kepada setiap pengguna dalam mengaksesnya kapan dan dimana saja untuk mendapatkan suatu informasi. Penggunaan media internet di Indonesia semakin bertumbuh pesat, keadaan ini memaksa sebagian besar masyarakat untuk menguasai penggunaan media internet tersebut, semakin mudahnya akses masyarakat mendapatkan informasi. Hampir disetiap bidang kehidupan manusia didominasi oleh media internet, baik bidang kesehatan, keuangan, akademik semuanya hampir pasti menggunakan media internet untuk mengakses data. Diberikannya kemudahan tersebut sangat membantu setiap orang dalam proses pengadministrasian karena semua hal tersebut bisa langsung tersistem, sehingga laporan-laporan bisa diberikan dengan sangat cepat, jikapun ada kesalahan bisa dengan segera untuk diupdate diperbaharui tanpa jeda waktu.

Pengelolaan kelangkah pengelolaan keuangan sekolah yang dimulai dari penerimaan hingga bagaimana keuangan yang digunakan dipertanggung jawabkan secara obyektif dan sistematis. Sistem administrasi keuangan siswa di sekolah seperti pembayaran SPP merupakan salah satu contoh sistem administrasi sekolah yang penulis melihat secara langsung sebagian besar penulisan datanya

masih menggunakan sistem manual. Penulisan data dengan menggunakan sistem manual terkadang terjadi kesalahan (*human error*), hal ini disebabkan karena penulisan pencatatan dilakukan seadanya dan kurang ketelitian saat penginputan data.

Menjelaskan bahwa untuk kegiatan keuangan di sekolah yang meliputi proses pengolahan data keuangan, penyimpanan data keuangan, dilakukan dengan cara mencatat keseluruhan pembayaran siswa ke dalam sebuah buku kas dan buku besar. Sehingga dalam penyimpanan data laporan keuangan tersebut masih kurang efektif dan sering terjadi kesalahan serta dokumen-dokumen laporan keuangan tersebut sering hilang apabila tahun ajaran telah berganti. Untuk itu, mencapai dan mempercepat proses input data dan pelaporan maka diperlukan suatu sistem yang terintegrasi dengan pemakaian teknologi komputer (Wulansari, 2013).

Administrasi pendidikan merupakan keseluruhan proses kerjasama dengan memanfaatkan dan memberdayakan segala sumber yang tersedia melalui aktivitas perencanaan, pengorganisasian, penggerakan, pemotivasian, pengendalian, pengawasan dan supervise, serta penilaian untuk mewujudkan sistem pendidikan yang efektif, efisien dan berkualitas. Perkembangan dan pemanfaatan teknologi informasi terutama dalam pemanfaatan teknologi komputer juga menjadi perhatian dari kalangan pendidikan, khususnya instansi sekolah. Berbagai program sekolah bisa dikembangkan menjadi sistem yang kinerjanya menggunakan perangkat komputer, dan salah satu program tersebut adalah sistem informasi administrasi pembayaran sekolah (Setiawati, 2012).

Sistem pembayaran saat ini masih manual dengan mencatat pada kartu pembayaran kemudian data pembayaran direkap secara manual pada buku. Hal ini menyebabkan proses memakan waktu lama dan tingkat kesalahan yang besar, selain itu akses data untuk membuat laporan akan sulit bila dibutuhkan seketika. Melalui pemanfaatan teknologi informasi yang ditunjang dengan kemajuan kesisteman, perangkat keras dan perangkat lunak, masalah pelaksanaan administrasi pembayaran sekolah yang dikemukakan di atas bisa diatasi dengan menggunakan sistem informasi administrasi keuangan.

## 1.2 Rumusan Masalah

Rumusan masalah yang dihadapi dari penelitian ini adalah bagaimana mengembangkan sebuah sistem informasi administrasi keuangan sekolah berbasis web.

## 1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah mengembangkan sistem administrasi keuangan SMA dan SMP berbasis web dengan *Framework Laravel*.

## 1.4 Batasan Masalah

1. Sistem administrasi dikembangkan berbasis *web*.
2. Sistem informasi menggunakan bahasa pemrograman PHP dengan *Framework Laravel*
3. Sistem informasi ini mencakup memasukkan siswa baru, pembayaran biaya pendidikan, biaya asrama, biaya makan, biaya admin pondok dan infaq pembangunan

## 1.5 Manfaat Penelitian

Manfaat yang didapat dari penelitian ini adalah:

1. Proses administrasi pembayaran sekolah menjadi lebih mudah.
2. Menyajikan informasi terkait rekapitulasi pembayaran SPP siswa.
3. Memudahkan siswa dalam melihat status pembayaran kapanpun dan dimanapun.
4. Meminimalisir adanya kesalahan dalam mengisi data yang dilakukan secara manual.

## BAB II TINJAUAN PUSTAKA

### 2.1 Sistem Informasi Pengelolaan Keuangan

Sistem informasi adalah suatu komponen yang terdiri dari manusia, teknologi informasi, dan prosedur kerja yang memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk mencapai suatu tujuan. Informasi yang dimasukan atau ditampilkan dapat berupa teks, gambar, audio dan video (Agus Mulyanto, 2009). Prosedur sistem informasi yaitu mengumpulkan, memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk tujuan tertentu. Sistem ini tidak harus terkomputerisasi, walaupun kebanyakan memang terkomputerisasi. Sistem informasi berbasis komputer (*Computer Based Information System*) adalah sistem informasi yang menggunakan teknologi komputer untuk melakukan beberapa atau seluruh pekerjaan yang diberikan (Zaimatus Sholikhah, 2014).

Sistem informasi keuangan merupakan suatu kerangka pengkoordinasian sumber daya (*data, materials, equipment, suppliers, personal, and funds*) untuk mengkonversi input berupa data ekonomik menjadi keluaran berupa informasi keuangan yang digunakan untuk melaksanakan kegiatan suatu entitas dan menyediakan informasi akuntansi bagi pihak-pihak yang berkepentingan (Wilkinson, 2000). Sistem informasi keuangan adalah sistem informasi yang memberikan informasi kepada orang atau kelompok baik di dalam perusahaan maupun di luar perusahaan mengenai masalah keuangan dan menyediakan informasi mengenai arus uang bagi para pemakai diseluruh perusahaan. sistem informasi keuangan diharapkan dapat mempermudah dan mempercepat kinerja bagi biro keuangan untuk melakukan proses pengelolaan keuangan (Mohamad Nurul Misbah, 2014).

Tujuan dari sistem informasi keuangan menurut Mulyadi (2001) antara lain:

1. Untuk menyediakan informasi bagi pengelola kegiatan.
2. Untuk memperbaiki informasi yang dihasilkan oleh sistem yang sudah ada, baik mengenai mutu, ketepatan penyajian maupun struktur informasi.

3. Untuk memperbaiki pengendalian akuntansi dan pengecekan intern, yaitu untuk memperbaiki tingkat keandalan (*realibility*) informasi akuntansi dan untuk menyediakan catatan lengkap mengenai pertanggung jawaban dan perlindungan kekayaan perusahaan.
4. Untuk mengurangi biaya klerikal dalam penyelenggaraan catatan akuntansi. Proses pencatatan, penjurnalan sampai ke pelaporan di mana proses ini bersifat matematis (hubungan buku besar dapat ditunjukkan dalam persamaan akuntansi), sistematis (urutan mengerjakannya jelas) dan logis (unsur pertimbangan atau *judgement* tidak terlibat lagi) akan dituangkan dalam proses terkomputerisasi dalam sistem informasi keuangan. Perkembangan teknologi informasi sangat membantu dalam pengembangan sistem informasi keuangan.

Sistem informasi keuangan berbasis web adalah suatu sistem perangkat keuangan yang menggunakan jaringan internet sebagai tempat untuk penyediaan atau penerimaan data yang diperlukan oleh user (Woro M.D.T Utami, 2011). Sistem informasi keuangan berbasis web yang dapat diakses oleh banyak user yang ditergrasikan untuk mempermudah user untuk mengambil data yang ada. Sistem informasi berbasis web memiliki kelebihan proses pengelolaan keuangan dan pengajuan dana akan lebih cepat karena tidak memerlukan proses yang panjang. Dari segi ekonomi sistem ini sudah terstruktur dan tidak perlu mencetak dokumen berulang-ulang sedangkan dari segi efesiensi sistem ini tidak membutuhkan banyak tenaga untuk proses pengelolaan keuangan. dari sedi pelayana sistem informasi berbasis web memberikan informasi yang akurat (Mohamad Nurul Misbah, 2014).

Berdasarkan pengertian diatas penulis mengambil kesimpulan bahwa sistem informasi adalah kumpulan data yang terintegrasi dan saling melengkapi dengan menghasilkan *output* yang baik guna untuk memecahkan masalah dan pengambilan keputusan.

## **2.2 Unified Modelling Language (UML)**

*Unified Modelling Language* (UML) adalah salah satu alat bantu dalam dunia pengembangan sistem berorientasi obyek karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat

cetak biru atas visi mereka ke bentuk baku, mudah dipahami lengkap dengan mekanisme efektif untuk berbagi dan mengomunikasikan rancangan mereka dengan yang lain.

UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT (*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*). Metode dari ketiganya digabungkan dengan membuang elemen yang tidak praktis ditambah dengan elemen-elemen dari metode lain yang lebih efektif dan elemen-elemen baru yang belum ada pada metode terdahulu sehingga UML lebih ekspresif dan seragam dari pada metode lainnya. Pendekatan-pendekatan yang dipakai dalam menganalisis berorientasi obyek antara lain:

- a. Pendekatan *top down* yaitu memecahkan masalah ke dalam bagian-bagian terkecil atau per *level* sehingga mudah untuk diselesaikan.
- b. Pendekatan modul yaitu membagi sistem ke dalam modul – modul yang dapat beroperasi tanpa ketergantungan.
- c. Penggunaan alat-alat bantu dalam bentuk grafik dan teks sehingga mudah untuk mengerti serta dikoreksi apabila terjadi perubahan.
- d. Pendekatan dalam menganalisis berorientasi obyek dilengkapi dengan alat – alat dan teknik - teknik yang dibutuhkan dalam pengembangan sistem, sehingga hasil akhir dari sistem yang dikembangkan akan didapatkan sistem yang dapat terdefinisi dengan baik dan jelas. Maka menganalisis berorientasi obyek akan dilengkapi dengan alat dan teknik di dalam mengembangkan sistem.

### **2.2.1 Class Diagram**

*Class* adalah deskripsi kelompok obyek-obyek dengan properti, perilaku dan relasi yang sama. *Class diagram* bisa memberikan pandangan global atas sebuah sistem. Hal tersebut tercermin dari *class-class* yang ada dan relasi satu dengan lainnya (Munawar, 2015),.

*Class* adalah sebuah spesifikasi yang akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (*attribute/property*) suatu sistem sekaligus menawarkan



layanan untuk memanipulasi keadaan tersebut (*method/function*). *Class Diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain (Triandini, 2012).

*Class* memiliki tiga area pokok:

- a. Nama (*Stereotype*)
- b. Attribute
- c. Metoda (*Operation*)

Adapun *Attribute* dan *operation* dapat memiliki salah satu sifat berikut:



- a. *Private*, tidak dapat dipanggil dari luar class yang bersangkutan.
- b. *Protected*, hanyadapat dipanggil oleh class yang bersangkutan dan anak - anak yang mewarisinya.
- c. *Public*, dapat dipanggil oleh siapa saja.





Hubungan Antar *Class*

- a. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan class yang memiliki atribut berupa *class* lain, atau class yang harus mengetahui eksistensi *class* lain. Panah navigasi menunjukkan arah *query* antar *class*.
- b. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas.”).
- c. Pewarisan, yaitu hubungan hierarki antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.

Hubungan dinamis, yaitu rangkaian pesan (*message*) yang dihubungkan dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.

**Tabel 2.1 Simbol *Class Diagram***







No	Gambar	Nama	Keterangan
1		<i>Generalization</i>	Hubungan dimana objek anak berbagi perilaku
2		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut

3		<i>Collaboration</i>	Deskripsi dari urutan aksi – aksi yang ditampilkan.
4		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
5		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen
6		<i>Association</i>	Menghubungkan antara objek satu dengan objek lainnya

**2.2.2 Use Case Diagram**

Menurut Triandini (2012), Diagram *use case* menunjukkan tiga aspek dari *system* yaitu *actor*, *use case* dan *system* atau *sub system boundary*. *Actor* mewakili peran orang, *system* yang lain atau alat ketika berkomunikasi dengan *use case*.

**Tabel 2.2 Simbol Use Case Diagram**

No	Simbol	Keterangan
1		Mewakili <i>User</i> .
2		Abstraksi dari interaksi antara sistem dan aktor.
3		Abstraksi dari penghubung antara aktor dan <i>usecase</i> .
4		Menunjukkan spesialisasi <i>actor</i> untuk dapat berpartisipasi dalam <i>use case</i> .
5		Menunjukkan <i>use case</i> merupakan fungsionalitas dari <i>use case</i> lainnya.
6		Menunjukkan <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi.

*Use case diagram* terdiri dari:

- a. *Actor* : *Actor* adalah *abstraction* dari orang dan *system* yang lain yang mengaktifkan fungsi dari target *system*. Orang atau *system* bisa muncul dalam beberapa pesan.
- b. *Use case* : *Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi



antara *actor* dengan sistem. Setiap *use case* harus diberi nama yang menyatakan apa hal yang dicapai dari hasil interaksinya dengan *actor*. Nama *use case* boleh terdiri dari beberapa kata dan tidak boleh adadua *use case* yang memiliki nama yang sama. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.





### 2.2.3 Activity Diagram

*Activity diagram* memodelkan *workflow* proses bisnis dan urutan aktivitas dalam sebuah proses. Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status. *activity diagram* dibuat pada awal pemodelan proses untuk membantu memahami keseluruhan proses. *Activity diagram* juga bermanfaat untuk menggambarkan interaksi antara beberapa *use case* ( Triandini, 2012). Adapun definisi *activity diagram* :

- a. Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses.
- b. Dipakai pada *business modeling* untuk memperlihatkan urutan aktivitas proses bisnis.
- c. Struktur diagram ini mirip *flowchart* atau Data Flow Diagram pada perancangan terstruktur.
- d. Sangat bermanfaat apabila membuat diagram ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan.
- e. *Activity diagram* dibuat berdasarkan sebuah atau beberapa *use case* pada *use case diagram*.

**Tabel 2.3 Simbol Activity Diagram**

No.	Gambar	Nama	Keterangan
1		<i>Swimlane</i>	Menunjukkan yang bertanggung jawab dalam melakukan aktivitas suatu diagram.
2		<i>Action</i>	Langkah-langkah dalam activity.


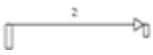

3		<i>Initial State</i>	Menunjukkan dimana aliran kerja dimulai
4		<i>Activity Final Node</i>	Menunjukkan dimana aliran kerja diakhiri
5		<i>Decision Node</i>	Menunjukkan suatu keputusan yang mempunyai satu atau lebih transisi sesuai dengan suatu kondisi.
6		<i>Control Flow</i>	Menunjukkan bagaimana kendali suatu aktivitas terjadi pada aliran kerja dalam tindakan tertentu.

#### 2.2.4 Sequence Diagram

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem termasuk pengguna, *display* dan sebagainya berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi *vertical* (waktu) dan dimensi *horizontal* (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. *Sequence diagram* diawali dari apa yang menggerakkan aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Masing-masing objek, termasuk *actor*, memiliki *lifeline vertikal*. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi atau metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message* (Triandini, 2012).

*Sequence Diagram* digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh obyek dan *message* (pesan) yang diletakkan di antara obyek-obyek ini di dalam *use case*. Diagram menunjukkan interaksi yang terjadi antar objek. Diagram ini menekankan pada basis keberurutan waktu dari pesan-pesan yang terjadi (Munawar, 2005).

Tabel 2.4 Simbol *Sequence Diagram*

No	Gambar	Nama	Keterangan
1		<i>Lifeline</i>	Obyek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang berisi informasi tentang aktivitas yang terjadi.
3		<i>Message</i>	Spesifikasi dari komunikasi antara objek yang memuat informasi-informasi tentang aktivitas yang terjadi.

### 2.3 *Entity Relational Diagram (ERD)*

*Diagram Entity Relationship (ERD)* adalah representasi visual yang digunakan untuk menggambarkan struktur logis dari sistem basis data. Ini merupakan alat konseptual yang membantu memodelkan hubungan antara entitas (atau objek) yang berbeda dalam basis data. ERD secara luas digunakan dalam perancangan basis data untuk memastikan pemahaman yang jelas tentang model data dan memfasilitasi komunikasi antara para pemangku kepentingan yang terlibat dalam proses pengembangan.

*Entity Relationship Diagram (ERD)* adalah sebuah model data yang digunakan untuk menggambarkan hubungan antara entitas (objek atau konsep) dalam suatu sistem informasi atau basis data. ERD merupakan alat visual yang efektif untuk memperlihatkan bagaimana data terorganisasi dan saling berhubungan dalam sebuah sistem.

Tujuan utama ERD adalah untuk menyediakan pandangan yang jelas dan terstruktur tentang struktur data, serta membantu para pengembang dan analis sistem memahami bagaimana entitas-entitas dalam sistem saling berinteraksi dan berhubungan satu sama lain.

Berikut adalah komponen-komponen utama dari *Diagram Entity-Relationship*:

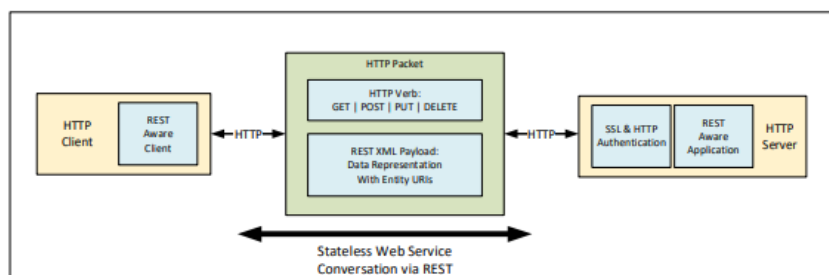
- a. Entitas: Entitas merepresentasikan objek atau konsep dunia nyata, seperti orang, tempat, barang, peristiwa, atau transaksi. Dalam diagram, entitas digambarkan sebagai persegi panjang dengan nama di dalamnya.
- b. Atribut: Atribut adalah informasi khusus yang terkait dengan entitas tertentu yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Gambar atribut diwakili oleh simbol *elips*.
- c. Relasi (hubungan): Hubungan mewakili koneksi antara dua entitas atau lebih. Hubungan ini mencerminkan cara entitas berinteraksi atau terkait satu sama lain dalam basis data. Jenis hubungan yang umum termasuk "*one-to-one*, *one-to-many*, *many-to-one*, dan *many-to-many*".
- d. Kardinalitas: Kardinalitas adalah atribut dari hubungan yang menggambarkan jumlah entitas dalam setiap sisi hubungan. Kardinalitas dapat menjadi "*one*" (1) atau "*Many*" (N). Misalnya, hubungan "*one-to-many*" antara entitas.
- e. Atribut Kunci: Setiap entitas memiliki satu atau lebih atribut yang berfungsi sebagai identifikasi unik untuk entitas tersebut. Atribut kunci ini disebut sebagai "*Primary Key*." *Primary Key* digunakan untuk mengidentifikasi setiap entitas secara unik dalam basis data.

## 2.4 RESTFUL API

RESTFUL API / REST API merupakan implementasi dari API (*Application Programming Interface*). REST (*Representational State Transfer*) adalah suatu arsitektur metode komunikasi yang menggunakan protokol HTTP untuk pertukaran data dan metode ini sering diterapkan dalam pengembangan aplikasi. Dimana tujuannya adalah untuk menjadikan sistem yang memiliki performa yang baik, cepat dan mudah untuk di kembangkan (*scale*) terutama dalam pertukaran dan komunikasi data. RESTFUL API memiliki 4 komponen penting di dalamnya diantaranya adalah :

- a. URL Design: RESTful API diakses menggunakan protokol HTTP. Penamaan dan struktur URL yang konsisten akan menghasilkan API yang baik dan mudah untuk dimengerti developer. URL API biasa disebut endpoint dalam pemanggilannya.

- b. HTTP Verbs: Setiap request yang dilakukan terdapat metode yang dipakai agar server mengerti apa yang sedang di request client
- GET, GET adalah metode HTTP Request yang paling simpel, metode ini digunakan untuk membaca atau mendapatkan data dari sumber
  - POST, POST adalah metode HTTP *Request* yang digunakan untuk membuat data baru dengan menyisipkan data dalam *body* saat *request* dilakukan
  - PUT, PUT adalah metode HTTP Request yang biasanya digunakan untuk melakukan *update data resource*.
  - DELETE, DELETE adalah metode HTTP *Request* yang digunakan untuk menghapus suatu data pada *resource*.
- c. HTTP *Response Code*: HTTP *response code* adalah kode standarisasi dalam menginformasikan hasil request kepada client. Secara umum terdapat 3 kelompok yang biasa kita jumpai pada RESTful API yaitu :
- 2XX : adalah response code yang menampilkan bahwa request berhasil.
- 4XX : adalah response code yang menampilkan bahwa request mengalami kesalahan pada sisi client.
- 5XX : adalah response code yang menampilkan bahwa request mengalami kesalahan pada sisi server.
- d. *Format Response*: Setiap request yang dilakukan client akan menerima data response dari server, response tersebut biasanya berupa data XML ataupun JSON. Setelah mendapatkan data respon tersebut barulah klien bisa menggunakannya dengan cara memarsing data tersebut dan diolah sesuai kebutuhan.



**Gambar 2.1** Arsitektur RESTFUL API

## 2.5 Konsep MVC (*Model-View-Controller*)

MVC adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data (*Model*), menyajikan data (*View*), dan logika untuk menangani kejadian (*event*) dari data ke penyajian maupun sebaliknya (*Controller*). MVC utamanya digunakan ketika mengembangkan perangkat lunak dengan antarmuka grafis (*Graphical User Interface*, GUI). MVC merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi web. MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi, seperti manipulasi data, *user interface*, dan bagian yang menjadi pengontrol aplikasi (Naista David, 2016).

Sistem dari MVC cukup sederhana, membeda-bedakan setiap bagian sesuai dengan fungsinya. Jadi, halaman yang di *request* oleh *user* akan di kontrol oleh *Controller*, lalu *Controller* akan memberikan tampilan kepada *user* pada *View*, dan untuk bagian data akan di atur oleh Model (Naista David, 2016). Terdapat tiga jenis komponen yang membangun suatu pola MVC dalam suatu aplikasi yaitu:

### a. *Model*

*Model* merupakan struktur yang bertugas dalam membantu setiap proses yang akan berhubungan dengan *database* seperti memasukkan data, mengedit data dan menghapus data dalam sebuah *database*. (Rolly Yesputra dan Nasrun M., 2018).

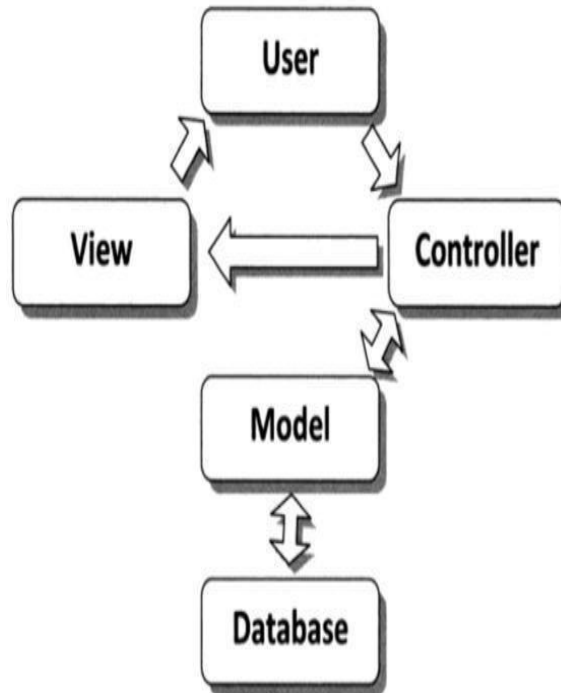
### b. *View*

*View* merupakan bagian yang menangani *presentation logic*. Berupa *template* html/xhtml atau php yang dapat berinteraksi langsung dengan pengguna dengan merepresentasikan bagaimana data ditampilkan.

### c. *Controller*

*Controller* Merupakan bagian yang mengatur atau mengontrol hubungan segala data yang diproses pada model untuk ditampilkan pada view atau sebagai penghubung antara model dan view. *Controller* berfungsi untuk menerima request dan data dari user, kemudian menentukan apa yang akan diproses oleh aplikasi.





Gambar 2.2 Konsep MVC

## 2.6 *Framework* Laravel

Laravel adalah sebuah *framework* PHP yang dirilis di bawah lisensi MIT, dibangun dengan konsep MVC (*model viewcontroller*). 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 (Pribadi, 2016).

## 2.7 *Hypertext Preprocessor* (PHP)

Menurut Andre (2014), PHP adalah bahasa pemrograman *script server-side* yang didesain untuk pengembangan web. Selain itu, PHP juga bisa digunakan sebagai bahasa pemrograman umum. PHP dikembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan sekarang dikelola oleh The PHP Group. PHP disebut bahasa pemrograman *server side* karena PHP diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman *client-side* seperti *Java Script* yang diproses pada *web browser* (*client*).