

SISTEM MANAGEMENT KOST DPAVILLON
MENGGGUNAKAN LARAVEL

PROPOSAL TUGAS AKHIR



DISUSUN OLEH:
FERNANDO PRATAMA BASTIAN
G.231. 17.0019

PROGRAM STUDI S1-TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI
UNIVERSITAS SEMARANG
2021

PERSETUJUAN
PROPOSAL TUGAS AKHIR

DENGAN JUDUL

SISTEM MANAGEMENT KOST DPAVILLON
MENGGUNAKAN LARAVEL

NAMA : FERNANDO PRATAMA BASTIAN
NIM : G.231. 17.0019

TELAH DISETUJUI

TANGGAL XXX

GENAP Thn AKADEMIK 2021/2022

KOORDINATOR TUGAS AKHIR
S-1 TEKNIK INFORMATIKA

PEMBIMBING TUGAS AKHIR

Whisnumurti Adhiwibowo, M.Kom
NIS. 06557003102137

April Firman Daru, S.Kom., M.Kom
NIS. 06557003102133

KETUA PROGRAM STUDI
S-1 TEKNIK INFORMATIKA

Khoirudin, S.Kom, M.Eng
NIS. 06557003102173

ABSTRAK

Rumah kost adalah bisnis di bidang property yang menyediakan jasa/tempat untuk menyewa kamar/tempat tinggal untuk masyarakat yang membutuhkannya. Rumah Kost Dpavillion yang berada di Jl. Medoho Permai No.22a, Pandean Lamper, Kec. Gayamsari, Kota Semarang, Jawa Tengah 50166. Pengolahan data pada persewaan kamar kost sebagian masih dilakukan dengan sistem manual. Sistem manual ini mempunyai beberapa kendala, diantaranya pengecekan kamar kosong, pencatatan data penyewa baru, pencatatan dan pelaporan keuangan relatif lebih lama, lebih sulit, dan kurang akurat. Karena itu proses pendatan terkadang terhambat karena faktor manual tersebut dan juga untuk pembayaran terkadang masih banyak kendala. Maka dari itu penulis mengusulkan suatu rancangan pengembangan sistem management rumah kost, pembangunan sistem management ini dibuat menggunakan bahasa pemrograman *PHP* dan *database MySQL* pada *framework laravel*. Metode pengembangan system ini dibangun menggunakan model RAD (*Rapid Application Development*). Diharapkan dari pembuatan sistem management ini untuk dapat lebih cepat, akurat dan mudah digunakan dengan berbasis *website*.

Keyword : *Sistem management, Rumah kost, Framework Laravel*

PEMBIMBING TUGAS AKHIR

April Firman Daru, S.Kom., M.Kom

NIS. 06557003102133

A. Latar Belakang

Kost adalah salah satu tempat penyedia jasa sewa yang terdiri dari beberapa kamar dan setiap kamar memiliki beberapa fasilitas yang ditawarkan atau diberikan dan juga memiliki harga yang telah ditentukan oleh pemilik kost. Teknologi komunikasi sekarang menghadapi perkembangan yang sangat cepat. Sebagian besar kehidupan manusia telah terbantu oleh kehadiran komputer. Salah satu manfaat komputer dalam kehidupan manusia adalah untuk mengelola, mengakses, dan menyebarkan data atau informasi.

Rumah Kost Dpavillion ini sendiri berdiri pada tahun 2012 berada di Jl. Medoho Permai No.22a, Pandean Lamper, Kec. Gayamsari, Kota Semarang, Jawa Tengah 50166. Pengolahan data pada persewaan kamar kost sebagian masih dilakukan dengan sistem manual. Sistem manual ini mempunyai beberapa kendala, diantaranya pengecekan kamar kosong, pencatatan data penyewa baru, pencatatan dan pelaporan keuangan relatif lebih lama, lebih sulit, dan kurang akurat. Karena itu proses pendatan terkadang terhambat karena faktor manual tersebut dan juga untuk pembayaran terkadang masih banyak kendala.

Masalah tidak efisiennya sistem manual, mungkin bisa terselesaikan dengan adanya sistem manajemen berbasis web yang menjadi pilihan untuk dirancang dan dibuat dalam menyelesaikan masalah admin dan pemilik kost.

Dibentuklah “SISTEM MANAGEMENT KOST DPAVILLION MENGGUNAKAN LARAVEL “ adalah sistem yang akan dibangun dengan fungsi customer kost dapat melihat detail kost dan booking terlebih dahulu dengan metode pembayaran bisa dilakukan via transfer bank verifikasi manual dan pembayaran saat checkin di lokasi. Dengan tujuan bisa memudahkan customer dalam pemilihan dan booking kost serta pembayaran. Sistem manajemen kost ini juga mempermudah admin dalam mengelola data laporan agar lebih efisien waktu.

B. Perumusan Masalah

Berdasarkan latar belakang diatas, maka penulis merumuskan permasalahan, yaitu bagaimana merancang dan membangun sistem management kost Dpavillion berbasis *website* yang dapat membantu pengelolaan laporan agar dapat meningkatkan efektivitas, efisiensi, dan produktivitas kerja.

C. Batasan Masalah

Agar tidak menyimpang dari permasalahan dan dapat mencapai tujuan yang diharapkan, maka peneliti membatasi permasalahan pada :

1. Membuat Sistem Management Kost Dpavillion berisi tentang informasi kebutuhan :
 - a. *Customer* : a) Cari kost
b) Booking
c) Pendaftaran akun baru
d) Pembayaran kost
 - b. *Admin* : a) Mengelola data kamar, booking dan pengeluaran
b) Mencetak laporan keuangan
 - c. *Pimpinan* : a) Kelola data admin
b) Mencetak laporan keuangan
2. Sistem yang di bangun berbasis *website* menggunakan *framework* laravel.
3. Menggunakan metode dengan model RAD (*Rapid Application Development*).

D. Tujuan Tugas Akhir

Tujuan dari pembangunan Sistem Management Kost Dpavillion berbasis *website* menggunakan *framework* laravel ini adalah sebagai berikut:

1. Menganalisis kebutuhan yang diperlukan untuk Sistem Management Kost Dpavillion berbasis *website*
2. Merancang dan membangun Sistem Management Kost Dpavillion berbasis *website* yang mampu mempermudah proses pengelolaan kost secara efektif dan akurat.
3. Melakukan pengujian dan mengimplementasikan Sistem Management Kost Dpavillion berbasis *website* pada admin dan pemilik kost.

E. Manfaat Tugas Akhir

1. Bagi Penulis
 - a. Penulis dapat menerapkan dan mengembangkan apa yang telah penulis terima tentang teknologi informasi selama belajar di Universitas Semarang.
 - b. Penulis menjadi mampu untuk membuat suatu sistem yang interaktif.
2. Bagi Tempat Kost
 - a. Membantu dalam proses pengolahan data yang lebih cepat, dan efisien waktu.
 - b. Sebagai masukan dan solusi yang dapat digunakan dalam pembangunan sistem di kemudian hari.
3. Bagi Akademik
 - a. Sebagai tolak ukur kemampuan mahasiswa dalam menguasai materi yang diajarkan selama perkuliahan.
 - b. Dapat digunakan sebagai kajian dan referensi pada perpustakaan serta dapat digunakan oleh pihak-pihak tertentu.

F. Tinjauan Pustaka

1. Penelitian Sebelumnya

- a. Menurut (Handayani, 2021) di dalam jurnal "*Sistem Informasi Manajemen Aplikasi Rumah Kost Dengan Menggunakan Rapid*

Application Development Berbasis Android Dan Sms Gateway” di Rumah Kost yang berada di wilayah Kota Bekasi belum memiliki sistem informasi manajemen mengenai ketersediaan, lokasi, fasilitas serta belum adanya sistem pembayaran secara online. Untuk mengatasi permasalahan tersebut penulis membangun sistem informasi manajemen berbasis android dan sms gateway dengan menggunakan metode Rapid Application Development, aplikasi dibangun menggunakan bahasa pemrograman *Java*, *Android Studio*, *Sublime* dan *database MySql*.

- b. Menurut (Rivaldi, 2021) pada jurnal “*Sistem Manajemen Penyewaan Kamar Kos Berbasis Web Web-Based Boarding House Rental Management System*” pada persewaan kamar kos sebagian masih dilakukan dengan sistem manual. Sistem manual ini mempunyai beberapa kendala, diantaranya pengecekan kamar kosong, pencatatan data penyewa baru, pencatatan dan pelaporan keuangan relatif lebih lama, lebih sulit, dan kurang akurat. Untuk mengatasi permasalahan tersebut penulis membangun sistem dalam penelitian ini dengan Analisis dan perancangan sistem dilakukan menggunakan UML, sistem dibangun menggunakan bahasa pemrograman PHP dan MySQL sebagai server database, serta mengintegrasikan google maps API sebagai media pemetaan lokasi.
- c. Menurut (Al Ghiffari et al., 2020) di dalam jurnal “*Perancangan Sistem Informasi Manajemen Pengelolaan Hunian Sewa Berbasis Website Pada Kos-kosan Pondok Salma*” Sistem manajemen yang diterapkan pada hunian sewa ini masih dengan sistem konvensional, seperti pembayaran dan laporan keluhan masih melalui penjaga hunian sewa tersebut. Sistem yang diterapkan sekarang dianggap kurang efektif untuk mengolah segala kegiatan hubungan antara penghuni hunian sewa dan penjaga hunian sewa ataupun pemilik dari hunian sewa tersebut. Untuk menanggapi permasalahan tersebut, dilakukan perancangan sistem informasi manajemen untuk

pengelolaan hunian sewa berbasis website menggunakan metode Agile Development. Perancangan sistem juga dilakukan dengan menggunakan Unified Modelling Language.

2. Pengertian Sistem

Secara etimologi, Sistem berasal dari bahasa Latin (*systema*) dan bahasa Yunani (*systema*) yang berarti suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai tujuan (A. S. Nugroho, 2017).

(Jogiyanto, 2018) mendefinisikan sistem sebagai gabungan dari berbagai elemen yang berhubungan dan berinteraksi untuk mencapai tujuan tertentu. system ini menggambarkan kejadian-kejadian dan kesatuan objek nyata, misalnya tempat, benda, dan orang-orang yang benar-benar ada dan nyata. Sehingga secara umum sistem diartikan sebagai suatu kumpulan objek atau unsur-unsur atau bagian-bagian yang memiliki arti berbeda-beda yang saling memiliki hubungan, saling berkerjasama dan saling mempengaruhi satu sama lain serta memiliki keterikatan pada rencana yang sama dalam mencapai suatu tujuan tertentu pada lingkungan yang kompleks.

3. Pengertian Manajemen

Managemen disebut sebagai seni untuk merealisasikan pekerjaan melalui orang lain dalam kaitannya dengan organisasi bahwa manager dalam mencapai tujuan organisasi dengan cara pengaturan orang lain (pegawai). Manajemen mencakup fungsi perencanaan, pengorganisasian, penyusunan personalia, pengarahan dan pengawasan (I Heryanto, 2016).

Menurut George R. Terry (Ahmad, 2018) manajemen merupakan proses yang khas yang terdiri dari tindakan-tindakan

perencanaan, pengorganisasian, penggerakkan, dan pengawasan untuk mencapai tujuan yang telah ditetapkan melalui pemanfaatan sumber daya manusia dan sumber daya lainnya. Definisi lain Stoner AF (Ahmad, 2018) menyatakan bahwa manajemen merupakan proses perencanaan, pengorganisasian dan pengawasan antar anggota organisasi dengan menggunakan seluruh sumber.

4. Pengertian Sistem Manajemen

Definisi Sistem Manajemen menurut (Ahmad, 2018) sistem manajemen adalah jaringan prosedur pengolahan data yang dikembangkan dalam organisasi dan disatukan apabila di pandang perlu, dengan maksud memberikan data kepada manajemen setiap waktu diperlukan, baik data yang bersifat intern maupun yang bersifat ekstern, untuk dasar pengambilan keputusan dalam rangka mencapai tujuan organisasi.

5. Pengertian Kost

Indekos atau kos/kost adalah sebuah jasa yang menawarkan sebuah kamar atau tempat untuk ditinggali dengan sejumlah pembayaran tertentu untuk setiap periode tertentu (umumnya pembayaran per bulan). Kata ini diserap dari frasa bahasa Belanda "in de kost". Definisi "in de kost" sebenarnya adalah "makan di dalam", tetapi dapat pula berarti "tinggal dan ikut makan" di dalam rumah tempat menumpang tinggal. wikipedia.org. (2021, 06 Desember)

6. Framework Laravel

a. Framework

Menurut (Supono, 2016), framework adalah sekumpulan dari fungsi-fungsi atau prosedur dan class-class untuk bertujuan tertentu

yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang programmer, tanpa harus membuat fungsi atau class dari awal.

Menurut (Rahmawati, 2017), Framework merupakan struktur konseptual dasar yang digunakan untuk memecahkan sebuah permasalahan atau isu kompleks.

- 1) Isi dari framework berupa fungsi, plugin dan konsep untuk membentuk suatu sistem tertentu.
- 2) Dengan menggunakan framework, aplikasi dapat tersusun dan terstruktur dengan rapi.
- 3) Dalam framework tersedia struktur aplikasi yang baik, standard coding, best practice, design pattern dan common function.

Maksud dari standard coding adalah sebuah standar yang harus diikuti programmer untuk menulis kode. Kemudian untuk best practice yaitu kumpulan action yang telah teruji oleh para expert. Sedangkan maksud dari design pattern adalah teknik - teknik yang menjadi best practice. Lalu untuk common function berarti fungsi - fungsi atau library yang telah umum digunakan dalam pengembangan sistem.

b. Laravel

Laravel adalah framework bahasa pemrograman Hypertext Preprocessor (PHP) yang diciptakan oleh Taylor Otwell yang dirilis pada 9 Juni 2011. Laravel adalah sebuah framework PHP yang dirilis di bawah lisensi perusahaan MIT. Dibangun dengan konsep MVC (Model View Controller). Laravel adalah pengembangan website berbasis MVP yang ditulis dengan bahasa pemrograman PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dan berfungsi untuk meningkatkan

pengalaman bekerja dengan aplikasi yang menyediakan sintaks yang ekspresif, jelas dan menghemat waktu (Hermanto, 2019).

7. Pengertian *PHP*

Menurut Supono (2016), *PHP* (*PHP : Hypertext preprocessor*) adalah suatu bahasa pemrograman yang digunakan untuk menterjemahkan baris kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang bersifat *server-side* yang dapat ditambah ke dalam *HTML*.

Menurut (Tim, 2016), *PHP* memiliki beberapa kelebihan jika dibandingkan dengan bahasa pemrograman web lainnya, antara lain yaitu :

- a. Bahasa pemrograman *PHP* adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
- b. Banyak web server yang mendukung *PHP*. Mulai dari Apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relatif mudah. Ada juga paket web server yang memudahkan melakukan instalasi sekali klik, seperti XAMPP yang tersedia untuk berbagai sistem operasi.
- c. *PHP* lebih mudah dalam sisi pengembangan karena banyaknya milis, group, facebook, dan developer yang siap membantu dalam pengembangan.
- d. Dalam sisi pemahaman, *PHP* adalah bahasa scripting yang paling mudah dipahami karena memiliki referensi yang banyak.
- e. *PHP* adalah bahasa pemrograman open source yang dapat digunakan diberbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara runtime melalui console, serta dapat menjalankan perintah - perintah sistem.

Pada awalnya *PHP* merupakan kependekan dari *Personal Home Page*. *PHP* pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu *PHP* masih bernama *Form Interpreted* (FI), yang wujudnya masih berupa sekumpulan skrip yang digunakan untuk

mengolah data formulir dari web. Selanjutnya Rasmus merilis kode sumber untuk umum dan menamakannya PHP/FI. Dengan perilsan kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan PHP. PHP memiliki 8 (delapan) tipe data, yaitu (Abdulloh, 2018) :

1. *Integer*

Tipe ini meliputi semua bilangan bulat dengan range - 2,147,483,648 sampai +2,147,483,647 pada *platform* 32bit. PHP juga akan mengkonversi secara otomatis bila suatu bilangan berada diluar *range* tersebut ke dalam tipe data *floating point*. Tipe ini juga dapat dinyatakan dalam bentuk oktal (berbasis 8), desimal (berbasis 10), heksadesimal (berbasis 16).

2. *Floating Point*

Tipe ini biasa digunakan dalam bilangan pecahan namun bisa juga bilangan desimal. Tipe ini memiliki range 1.7E-308 sampai 1.7E+308. Dapat dinyatakan dalam bentuk bilangan desimal atau dalam bentuk pangkat.

3. *String*

Tipe data string dinyatakan dengan mengapitnya menggunakan tanda petik tunggal (' ') atau tanda petik ganda (" "). Perbedaan dari penggunaan keduanya adalah dengan tanda petik tunggal kita tidak dapat menggunakan *variable* dan *escape sequence handling* bersama dalam suatu kalimat.

4. *Boolean*

Tipe data boolean digunakan untuk menyimpan nilai *true* atau *false*. Biasanya tipe data ini mayoritas digunakan untuk melakukan pengecekan kondisi pada php.

5. Null

Tipe data yang tidak memuat apapun. Setiap Variabel yang diset menjadi Tipe Data NULL ini akan menjadikan Variabel tersebut kosong.

6. Array

Tipe ini dapat mengandung satu atau lebih data juga dapat diindeks berdasarkan numerik atau string. Ia juga mendukung multidimensi dan membolehkan semua datanya berbeda tipe data.

7. Object

Tipe data object dapat berupa bilangan, variable atau fungsi. Object dibuat dengan tujuan agar para programmer terbiasa dengan OOP, meski fasilitas ini masih minim.

8. Resource

Tipe Data Spesial yang satu ini di khususkan untuk menyimpan resource, sumber atau alamat. Variabel tersebut hanya dapat diciptakan oleh suatu fungsi khusus yang mengembalikan nilai berupa resource seperti penggunaan fungsi fopen, opendir, mysql_connect, mysql_query dan sebagainya.

Beberapa kelebihan PHP dari bahasa pemrograman *web*, antara lain :

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. *Web* Server yang mendukung PHP dapat ditemukan dimana-mana, mulai dari 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 di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan

secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

8. *MySQL*

MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersil. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. MySQL memiliki beberapa keistimewaan, antara lain (B. Nugroho, 2019):

1. *Portabilitas.*

MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Max Os X Server, Solaris, Amiga, dan masih banyak lagi.

2. Perangkat lunak sumber terbuka.

MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.

3. *Multi-user.*

MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah.

4. *Performance-tuning.*

MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL persatuan waktu.

5. Ragam tipe data.

MySQL memiliki ragam tipe data yang sangat kaya, seperti signed/unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.

6. Perintah dan fungsi.

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *select* dan *where* dalam perintah (*query*).

7. Keamanan.

MySQL memiliki beberapa lapisan keamanan seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenskripsi.

8. Skalabilitas dan pembatasan.

MySQL mampu menangani basis data secara besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. Konektivitas.

MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, UNIX soket (UNIX), atau Named Pipes (NT).

10. Lokalisasi.

MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, Bahasa Indonesia belum termasuk di dalamnya.

11. Antar muka.

MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

12. Klien dan peralatan

MySQL dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.

13. Struktur tabel

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan dengan basis data lainnya semacam PostgreSQL ataupun Oracle.

9. XAMPP

Menurut (Riyanto, 2014), *XAMPP* adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis *PHP* dan menggunakan pengolah data *MySQL* di komputer secara local. *XAMPP* berperan sebagai *website server* pada komputer. *XAMPP* juga dapat disebut sebuah *CPanel server virtual*, yang dapat membantu anda melakukan preview sehingga dapat memodifikasi *website* tanpa harus online atau ter akses dengan internet.

Software XAMPP bersifat *open sources* yang dapat diperoleh secara gratis dari situs www.apachefriends.org. *XAMPP* adalah perangkat lunak yang mendukung banyak sistem operasi dan merupakan komplikasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri dan terdiri atas *Apache*, *MySQL*, dan bahasa pemrograman *PHP*.

Menurut (Purbadian, 2016), berpendapat bahwa *XAMPP* merupakan suatu *software* yang bersifat *open source* yang merupakan pengembangan dari *LAMP* (*Linux*, *Apache*, *MySQL*, *PHP* dan *Perl*).

Berdasarkan pengertian diatas dapat disimpulkan bahwa *Xampp* merupakan tool pembantu pengembangan paket perangkat lunak berbasis *open source* yang menggabungkan *Apache web server*, *MySQL*, *PHP* dan beberapa modul lainnya didalam satu paket aplikasi.

10. Unified Modeling Language (UML)

Pada perkembangan teknik pemograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik

pemrograman berorientasi objek yaitu *Unified Modelling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung (Muslihudin, 2016).

UML hanya berfungsi sebagai untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek.

Beberapa jenis *UML* yang digunakan untuk merancang sistem diantaranya, yaitu :


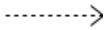

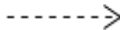


a. *Use case Diagram*

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 aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu (Muslihudin, 2016).

Use case diagram dapat sangat membantu bila sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-



include dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain. Berikut adalah simbol - simbol *Use Case Diagram*.

Tabel 1 Simbol *Use case Diagram*
(Muslihudin, 2016)

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

Lanjutan Tabel 1 Simbol *Use case Diagram*

(Muslihudin, 2016)

SIMBOL	NAMA	KETERANGAN
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

b. *Activity Diagram*


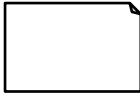
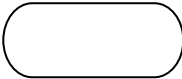


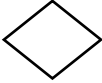
Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi (Muslihudin, 2016).

Activity diagram merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan



proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. Berikut adalah simbol - simbol *Activity Diagram*.

Tabel 2 Simbol *Activity Diagram*
(Muslihudin, 2016)

SIMBOL	NAMA	KETERANGAN
	State	Kondisi yang mungkin dialami oleh suatu obyek.
	Note	Note digunakan untuk memberikan keterangan atau komentar
	Aktivitas	Perilaku obyek yang dilakukan saat obyek berada dalam <i>state</i> tertentu.
	Start State	<i>Start state</i> digunakan untuk memulai diagram <i>statechart</i> .
	End State	End state digunakan untuk mengakhiri diagram.
	Decision	<i>Decision</i> digunakan sebagai pilihan untuk pengambilan keputusan.

Lanjutan Tabel 2 Simbol *Activity Diagram*
(Muslihudin, 2016)

SIMBOL	NAMA	KETERANGAN
	Penggabungan / Join	Digunakan untuk split dan join. pada saat diagram akan dibagi 2, bar ini akan ditambahkan. dan sebelum diagram digabung menjadi satu, sebagai join.
	Asosiasi (<i>association</i>)	Apa yang menghubungkan antara objek satu dengan objek yang lainnya.

c. *Class Diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain (Muslihudin, 2016). *Class* memiliki tiga area pokok :

1. Nama (dan stereotype)
2. Atribut
3. Metoda

Class dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*. Hubungan antar *class* yaitu:

1. Asosiasi

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

Hubungan yang menyatakan bagian (“terdiri atas..”).



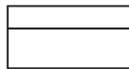

3. Pewarisan

Hubungan hirarkis 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.

4. Hubungan Dinamis

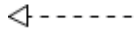
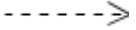

Rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain.

Tabel 3. Simbol *Class Diagram*
(Muslihudin, 2016)

SIMBOL	NAMA	KETERANGAN
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan > 2 objek.
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor

Lanjutan Tabel 3. Simbol *Class Diagram*

(Muslihudin, 2016)

SIMBOL	NAMA	KETERANGAN
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

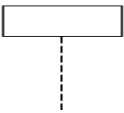


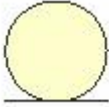
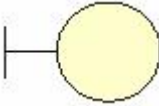

d. *Sequence Diagram*

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 vertikal (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. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan (Muslihudin, 2016).

Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*. Berikut adalah simbol - simbol *Sequence Diagram*.

Tabel 4. Simbol *Sequence Diagram*

(Muslihudin, 2016)

SIMBOL	NAMA	KETERANGAN
	<i>LifeLine</i>	Menggambarkan objek <i>entity</i> , antarmuka yang saling berinteraksi.
	<i>Message</i>	Menggambarkan pengiriman pesan
	<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem
	<i>Entity Class</i>	Menggambarkan hubungan kegiatan yang akan dilakukan
	<i>Boundary Class</i>	Menggambarkan sebuah penggambaran dari form
	<i>Control Class</i>	Menggambarkan penghubung antara <i>boundary</i> dengan tabel

G. Metodologi Penelitian

Untuk melaksanakan penelitian diperlukan metode penelitian yang tepat. Adapun metodologi yang diterapkan memiliki beberapa tahap berikut:

1. Metode Pengumpulan Data

a. Pengamatan (Observasi)

Pengamatan langsung terhadap alur pengelolaan kost yang dilakukan admin dan pemilik kost Dpavillon untuk memperoleh informasi yang nantinya akan diolah kedalam sistem management kost.

b. Wawancara (*Interview*)

Penulis mengumpulkan data dengan cara melakukan wawancara dengan pemilik kost Dpavillon.

c. Studi Pustaka

Mengumpulkan data dan informasi dengan cara mencari, menganalisa dan menyimpulkan dari berbagai buku bacaan lain yang berhubungan dengan objek penelitian.

2. Sumber Data

Sumber data yang digunakan antara lain :

1. Data Primer

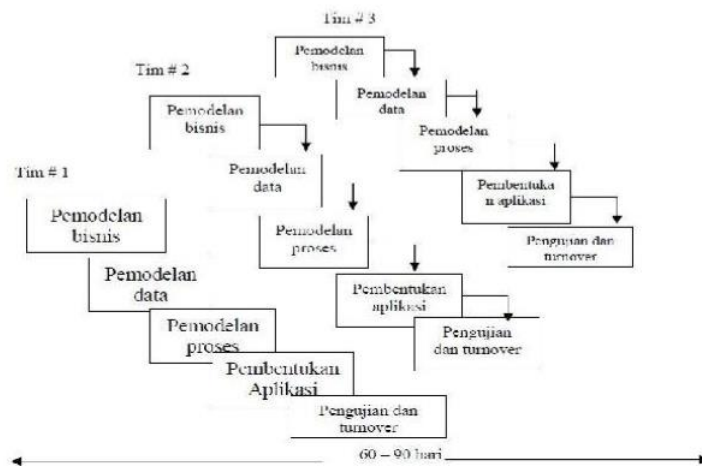
Data Primer adalah data-data yang didapat secara langsung dari sumber yang menjadi objek penelitian. Data Primer diperoleh dengan cara observasi dan wawancara langsung terhadap pemilik kost. contohnya banyaknya jumlah kamar kost, jumlah penghuni yang ada, sejarah berdiri hingga kepengurusan.

2. Data Sekunder

Data Sekunder adalah data yang diperoleh secara tidak langsung dari sumber yang dijadikan objek penelitian dan digunakan sebagai pendukung data primer.

3. Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan adalah metode RAD (*Rapid Application Development*).



Gambar 1 Model RAD (Rosa A.S, & Shalahudin, 2018)

Berdasarkan Gambar 1 di atas, berikut penjelasan dari fase dan tahapan dari metode RAD , terdapat lima fase dalam RAD yang melibatkan penganalisis dan pengguna dalam tahap penilaian, perancangan, dan penerapan. Adapun kelima fase tersebut adalah Pemodelan bisnis, Pemodelan data, Pemodelan proses, Pembuatan aplikasi, Pengujian dan pergantian. Sesuai dengan metodologi RAD, berikut ini adalah tahap-tahap pengembangan aplikasi dari tiap-tiap fase pengembangan aplikasi.

1. Pemodelan Bisnis

Pemodelan yang di lakukan untuk memodelkan fungsi bisnis untuk mengetahui informasi apa yang terkait dengan proses bisnis, informasi apa yang harus dibuat, siapa yang harus membuat informasi itu, bagaimana alur informasi itu, proses apa saja yang terkait informasi itu.

Pada tahap ini, penulis melakukan wawancara kepada Pemilik Kost Dpavillon mengenai apa saja data yang di perlukan dalam pembuatan Sistem Management Kost Dpavillon Menggunakan Laravel untuk menunjang kegiatan admin kost.

2. Pemodelan Data

Memodelkan data apa saja yang dibutuhkan berdasarkan pemodelan bisnis dan mendefinisikan atribut-atributnya beserta relasinya dengan data-data yang lain. Pada tahap ini, penulis mengumpulkan data apa saja yang di butuhkan yaitu data kamar, booking dan pengeluaran keuangan. Untuk menentukan atribut dari setiap data dan relasi antar data yang telah di kumpulkan dengan MySQL.

3. Pemodelan Proses

Mengimplementasikan fungsi bisnis yang sudah didefinisikan terkait dengan pendefinisian data. Pada tahap ini, penulis melakukan gambaran sebuah rancangan sistem serta desain sistem guna mendapatkan persetujuan user untuk menilai apakah masih ada kekurangan yang masih belum sesuai kebutuhan perusahaan dengan bantuan Diagram UML.

4. Pembuatan Aplikasi

Mengimplementasikan pemodelan proses dan data menjadi program. Pada tahap ini, penulis melakukan pembuatan Sistem Management Kost Dpavillon dengan menggunakan bahasa pemrograman Laravel dan database MySQL.

5. Pengujian dan Pergantian

Menguji komponen-komponen yang dibuat. Jika sudah teruji maka tim pengembang komponen dapat mengembangkan komponen berikutnya. Pada tahap ini, setelah penulis melakukan coding dan sistem telah selesai dibuat maka di lakukan pengujian. Pengujian sistem yang di lakukan adalah *black box* dan *white box*.

DAFTAR PUSTAKA

- Abdulloh, R. (2018). *7 in 1 Pemrograman Web Untuk Pemula*. Elex Media Komputindo.
- Ahmad, L. (2018). *Sistem Informasi Manajemen: Buku Referensi: Sistem Informasi Manajemen* (Vol. 1). KITA Publisher.
- Al Ghiffari, M. A., Praptono, B., & Sagita, B. H. (2020). Perancangan Sistem Informasi Manajemen Pengelolaan Hunian Sewa Berbasis Website Pada Kost-kostan Pondok Salma. *EProceedings of Engineering*, 7(2).
- Handayani, D. dan H. L. (2021). *SISTEM INFORMASI MANAJEMEN APLIKASI RUMAH KOST DENGAN MENGGUNAKAN RAPID APPLICATION DEVELOPMENT BERBASIS ANDROID DAN SMS GATEWAY*.
<http://repository.ubharajaya.ac.id/id/eprint/6820%0A>
- Hermanto, B. (2019). Sistem Informasi Manajemen Keuangan pada PT. Hulu Balang Mandiri Menggunakan Framework Laravel. *Jurnal Komputasi*, 7(1), 17–26.
- I Heryanto, T. T. (2016). *Manajemen Proyek Berbasis Teknologi Informasi*. Informatika Bandung.
- id.wikipedia.org. “Indekost”, <https://id.wikipedia.org/wiki/Indekost>, diakses pada 06 Desember 2021 pukul 15.15.
- Jogiyanto, H. . (2018). *Sistem Informasi Manajemen (Edisi 2)*. Universitas Terbuka.
- Muslihudin, M. (2016). *Analisis Dan Perancangan Sistem Informasi Menggunakan Model Terstruktur Dan UML*. Penerbit Andi.
- Nugroho, A. S. (2017). *Analisis dan Perancangan Sistem Informasi*. Trans Tekno.
<http://103.255.15.77/detail-opac?id=300171>
- Nugroho, B. (2019). *Aplikasi Pemrograman WEB Dinamis dengan PHP dan MySQL (Studi Kasus Membuat Sistem Informasi Pengola*. Gava Media.
<https://www.gavamedia.net/produk-462-aplikasi-pemrograman-web-dinamis-dengan-php-dan-mysql--studi-kasus-membuat-sistem-informasi-pengola.html>

- Rivaldi, A. dkk. (2021). *SISTEM MANAJEMEN PENYEWAAN KAMAR KOST BERBASIS WEB WEB-BASED BOARDING HOUSE RENTAL MANAGEMENT SYSTEM*. Vol. 9 No. 1 (2021): Vol 9 No. 1 Juni 2021.
<https://doi.org/https://doi.org/10.34151/script.v9i1.3639>
- Riyanto. (2014). *Membuat Aplikasi Mini Market Integrasi Barcode Reader dengan PHP & MySQL*. Gava Media.
- Rosa A.S, & Shalahudin, M. (2018). *Rekayasa Perangkat Lunak (Terstruktur dan Beorientasi Objek)*. Informatika Bandung.
- Supono, V. P. (2016). *Pemrograman Web dengan menggunakan PHP dan FRAMEWORK CODEIGNITER*. Yogyakarta: Deepublish.