

# Jurnal Informatika Terpadu

Vol. 6 No. 1 Tahun 2020



# Jurnal Informatika Terpadu

**Jurnal Informatika Terpadu** jurnal ilmiah yang berasal dari mahasiswa/i S-1 pada prodi Teknik Informatika dan Sistem Informasi di STT Terpadu Nurul Fikri sebagai salah satu syarat untuk meraih gelar Sarjana (S-1).

## **Ketua Penyunting (*Editor-in-chief*)**

Sirojul Munir, S.Si., M.Kom.  
Teknik Informatika  
STT Terpadu Nurul Fikri

## **Dewan Penyunting (*Editorial Board Member*)**

Hilmy Abidzar Tawakal, S.T., M.Kom.  
Teknik Informatika  
STT Terpadu Nurul Fikri

Tiffany Nabarian, S.Kom., M.T.I.  
Teknik Informatika  
STT Terpadu Nurul Fikri

Suhendi, S.T., M.MSI.  
Sistem Informasi  
STT Terpadu Nurul Fikri

Zaki Imaduddin, S.T., M.Kom.  
Teknik Informatika  
STT Terpadu Nurul Fikri

Ahmad Rio Ardiansyah, S.Si., M.Si.  
Teknik Informatika  
STT Terpadu Nurul Fikri

## **Mitra Bestari (*Reviewer*)**

Drs. Rusmanto, M.M.  
Sistem Informasi  
STT Terpadu Nurul Fikri

Yekti Wirani, S.T., M.T.I.  
Sistem Informasi  
STT Terpadu Nurul Fikri

Henry Saptono, S.Si, M.Kom.  
Teknik Informatika  
STT Terpadu Nurul Fikri

Dr. Lukman Rosyidi, S.T., M.M., M.T.  
Teknik Informatika  
STT Terpadu Nurul Fikri

Amalia Rahmah, S.T., MT.  
Sistem Informasi  
STT Terpadu Nurul Fikri

Misna Asqia, M.Kom.  
Sistem Informasi  
STT Terpadu Nurul Fikri

Nurul Janah, S.IIP, M.Hum.  
Sistem Informasi  
STT Terpadu Nurul Fikri

## **Penyunting Pelaksana (*Assistant Editors*)**

Muh Syaiful Romadhon, S.Kom.  
Sistem Informasi  
STT Terpadu Nurul Fikri

Fasyikhatun Maidah, S.Kom.  
Sistem Informasi  
STT Terpadu Nurul Fikri

Jurnal Informatika Terpadu (*Journal of Integrated Informatics*) telah terindeks oleh Google Scholar. Tanggungjawab isi artikel berada di penulis bukan pada penerbit atau editor.

## **Diterbitkan oleh:**

LPPM STT Terpadu Nurul Fikri

## **Alamat Redaksi dan Distribusi:**

Kampus B STT Terpadu Nurul Fikri lantai 3  
Jl. Lenteng Agung Raya 20, Jakarta Selatan, DKI Jakarta, 12640  
Telp. 021 – 786 3191

Email: [lppm@nurulfikri.ac.id](mailto:lppm@nurulfikri.ac.id)

Website: <https://journal.nurulfikri.ac.id/index.php/jit/> dan [lppm.nurulfikri.ac.id](http://lppm.nurulfikri.ac.id)

Daftar Isi

<b>Analisis dan Pengembangan Sistem Desain Asset untuk Keramik Lantai berbasis Aplikasi Website</b>	<b>01</b>
Alvino Gartner, Sirojul Munir	
<b>Visualisasi Implementasi Autentifikasi <i>Digital Signature</i> menggunakan Fungsi Hash pada File Digital</b>	<b>07</b>
Rufaidah Taqiyyah, Ahmad Rio Adriansyah	
<b>Analisis dan Evaluasi Algoritma <i>MapReduce WordCount</i> pada Cluster Hadoop menggunakan Indikator Kecepatan</b>	<b>14</b>
Robiyatul Adawiyah, Sirojul Munir	
<b>Uji Performa dan Perbandingan RDBMS MySQL dan <i>Hive-Hadoop</i></b>	<b>20</b>
Nurul Azizah, Henry Saptono	
<b>Rancang Bangun Sistem Informasi Pengelolaan <i>Talent Film</i> berbasis Aplikasi Web</b>	<b>29</b>
Atikah Permata Sari, Suhendi	
<b>Analisis dan Penerapan <i>Enterprise Resource Planning (ERP)</i> Odoo 10 guna membuat Laporan Keuangan pada Yayasan Darul Jannah</b>	<b>38</b>
Rosy Annisa Firlana, Suhendi	
<b>Analisis dan Pengembangan Sistem Reservasi Online untuk Wisata berbasis Website Studi Kasus Pulau Harapan Kepulauan Seribu</b>	<b>45</b>
Rian Hardiansyah, Sirojul Munir	



## ANALISIS DAN PENGEMBANGAN SISTEM DESAIN ASSET UNTUK KERAMIK LANTAI BERBASISKAN APLIKASI WEBSITE

Alvino Gartner<sup>1</sup>, Sirojul Munir<sup>2</sup>

<sup>1,2</sup>Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri  
Jakarta Selatan, DKI Jakarta, Indonesia 12640

[alvino.gartner@gmail.com](mailto:alvino.gartner@gmail.com) , [rojulman@nurulfikri.ac.id](mailto:rojulman@nurulfikri.ac.id)

### Abstract

*The need for shelter is a basic need for humans after food and shelter. Each individual will prioritize to meet basic needs rather than secondary needs. A person's way of determining their occupancy tends to be very modern. That is because of the lifestyle that people do now, which makes social status so important, especially with the high mobility of modern society and a lifestyle that makes technology the primary support in everyday life. Likewise, about housing, the more lifestyle becomes the primary goal, the modern society will tend to want to own a dwelling of their own accord. Therefore, the researcher took the initiative to raise one of the primary materials in the dwelling, namely floor tiles, which will be processing in a customized manner by providing a website-based platform using a framework to determine the desired floor design.*

**Keywords:** Web Framework, Laravel, Ceramic Online Application

### Abstrak

Kebutuhan akan hunian merupakan kebutuhan dasar bagi manusia setelah pangan dan sadang. Setiap individu akan lebih mengutamakan untuk memenuhi kebutuhan dasar dari pada kebutuhan sekunder. Cara seseorang untuk menentukan hunian mereka cenderung dengan aktualitas yang sangat modern. Hal itu dikarenakan dari gaya hidup yang dilakukan masyarakat sekarang, dimana menjadikan status sosial itu begitu penting. Terlebih dengan mobilitas masyarakat modern yang cukup tinggi serta gaya hidup yang menjadikan teknologi sebagai penunjang utama dalam keseharian. Begitu juga terkait dengan hunian, semakin gaya hidup menjadi tujuan utama, maka masyarakat modern akan cenderung ingin memiliki hunian dengan kemauannya sendiri. Oleh karena itu peneliti berinisiatif untuk mengangkat salah satu material utama dalam hunian yaitu keramik lantai, yang akan diolah secara customize dengan menyediakan platform berbasis website menggunakan framework untuk menentukan desain lantai yang diinginkan.

**Kata kunci:** Web Framework, Laravel, Aplikasi Online Keramik

### 1. PENDAHULUAN

Kebutuhan akan hunian merupakan kebutuhan dasar bagi manusia setelah pangan dan sadang. Setiap individu akan lebih mengutamakan untuk memenuhi kebutuhan dasar dari pada kebutuhan sekunder begitu pula akan kebutuhan hunian yang mencapai 1,46 juta unit pertahunnya. Sedangkan kebutuhan yang terrealisasi dari data terakhir pertahunnya berkisar 400 unit [1]. Di sisi lain, cara seseorang untuk menentukan hunian mereka cenderung dengan aktualisasi yang sangat modern. Hal itu dikarenakan dari gaya hidup yang dilakukan masyarakat sekarang, dimana menjadikan status sosial itu begitu penting. Terlebih dengan mobilitas masyarakat modern yang cukup tinggi serta gaya hidup yang menjadikan teknologi sebagai penunjang utama dalam keseharian [3]. Akan tetapi, dengan aktualisasi yang tinggi terhadap kepuasan diri, ternyata mampu

menghasilkan dampak positif dimana akan memunculkan inovasi untuk memperbaharui sesuatu. Begitu juga terkait dengan hunian. Semakin gaya hidup menjadi tujuan utama, maka masyarakat modern akan cenderung ingin memiliki hunian dengan kemauannya sendiri.

Peraih *The Best Outstanding Corporate Innovator (OCI) Award* dan *Top 25 Indonesia Most Creative Companies 2016* dari Majalah SWA, yaitu *Paramount Land* telah mengusung tema hunian secara *customize* kepada pelanggan dan mendapatkan apresiasi yang sangat luar biasa [2]. Hal itu membuktikan bahwa gaya hidup adalah salah satu landasan para masyarakat dalam menentukan hunian.

Besarnya apresiasi akan hunian secara *customize* harus di ikuti dengan *tools-tools* pendukung, dari sekian banyak

*tools* pendukung yang telah di sediakan seperti AutoCAD *Architecture*, ArchiCAD, *Chief Architect* dan lain-lainnya masih terdapat kekurangan untuk memenuhi kebutuhan dalam melakukan *customize* hunian rumah dan salah satu kekurangannya adalah belum adanya *tools* pendukung untuk melakukan *customize* pada media keramik lantai. Oleh karena itu peneliti berinisiatif untuk mengangkat salah satu material utama dalam hunian yaitu keramik lantai, yang akan diolah secara *customize* dengan menyediakan *platform* berbasis *website* menggunakan *framework* untuk menentukan desain lantai yang diinginkan.

### 1.1 Rumusan Masalah

Masalah yang diangkat dalam tugas akhir ini yaitu “Bagaimana cara membuat website kramik lantai yang dapat di pesan secara *customize*?” Sesuai dengan latar belakang yang dijelaskan, maka diuraikan pokok - pokok masalah adalah sebagai berikut:

1. Bagaimana membangun sistem berbasis *web* untuk mempertemukan *vendor* dan *costumer* dengan *value customize* terhadap keramik lantai?
2. Apa saja fitur yang ada dalam aplikasi *customize* keramik lantai?

### 1.2 Tujuan dan Manfaat Penelitian

Tujuan yang hendak dicapai dalam penelitian dan penyusunan tugas akhir ini adalah:

1. Mampu membangun sistem terintegrasi berbasis *website* untuk mempertemukan *vendor* dan *costumer* dengan *value customize* terhadap keramik lantai.
2. Membangun aplikasi *customize* keramik lantai yang dapat membantu proses pemesanan keramik lantai dengan desain sendiri.

### 1.3 Batasan Masalah

Batasan masalah dalam penulisan tugas akhir ini adalah:

1. Aplikasi ini hanya dapat melakukan useran melalui desain yang telah di tentukan.
2. Aplikasi ini hanya untuk meng-*costum* desain pada media keramik lantai.

## 2. LANDASAN TEORI

### 2.1 Gaya Hidup

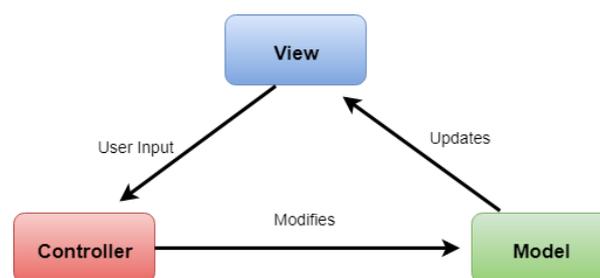
Gaya hidup bisa dikatakan sebagai identitas dari suatu kelompok. Mowen [4] menyebutkan bahwa gaya hidup seseorang dipengaruhi oleh perilaku yang akhirnya menentukan pola konsumsi orang tersebut. Dari hal tersebut diketahui bahwa gaya hidup sangat mempengaruhi seseorang untuk menentukan kemauannya untuk memilih.

### 2.2 Modernisasi

Modernisasi berasal dari kata *modern* yang diartikan sebagai suatu hal terbaru, mutakhir, atau sikap dan cara berpikir yang sesuai dengan tuntutan zaman. Modernisasi juga bisa diartikan sebagai proses perubahan sikap dalam menjalani hidup sesuai dengan tuntutan masa kini [5].

### 2.3 MVC

MVC (*Model View Controller*) merupakan pola atau teknik untuk mengembangkan aplikasi yang bertujuan untuk menghasilkan program yang disiplin dengan cara membaginya menjadi 3 bagian *model*, *view*, dan *controller* [7], seperti Gambar 1.



Gambar 1. Arsitektur MVC

Pada MVC terdapat bagian yang pertama yang disebut dengan *models*, *models* ini berfungsi untuk memanipulasi data (*insert*, *update*, *delete*, *search*) berserta aturan bisnisnya (*buisness processs*) seperti proses validasi dan relasi[7]. *Models* juga merupakan bagian dari aplikasi yang mengimplementasi logika untuk data aplikasi, Sehingga dengan adanya *models* maka aplikasi yang dihasilkan akan terstruktur dan data yang tersipan akan lebih aman karena harus melalui validasi data terlebih dahulu.

Bagian kedua yang disebut dengan *views*, di bagian ini merupakan komponen yang yang berisikan keseluruhan detail dari tampilan antar muka (*user interface*) untuk pengguna aplikasi seperti *admin*, *members* dan *guest*. Dengan *views* semua proses internal aplikasi akan akan di tampilkan kepada *user* dan menuntun alur interaksi *user* terhadap aplikasi. *Views* saling terhubung dengan *models* dan me-render isinya ke dalam permukaan layar, apabila terjadi perubahan pada *models* maka secara otomatis *views* akan menggambar ulang bagian layar yang terkena perubahan untuk menunjukkan perubahan tersebut [9].

Bagian terakhir yang di sebut dengan *controller*, pada bagian ini terjadi proses inputan dari pengguna dan meng-instruksikan *models* dan *views* untuk melakukan aksi berdasarkan masukan tersebut. Sehingga, *controller* bertanggung jawab atas penampungan *event* yang di lakukan oleh *user* [9].

Konsep *Model View Controller* (MVC) bertujuan agar sebuah aplikasi dapat mudah dielihara oleh orang-orang di dalam tim pengembangan yang berbeda spesifikasi pekerjaan, misalnya *database administrator* (DBA) untuk

mengurusi masalah basis data, *blok controller* untuk *programmer*, dan *blok view* untuk desainer antarmuka [8].

## 2.4 Framework

*Framework* merupakan kerangka kerja yang bertujuan untuk memudahkan dalam membuat sebuah aplikasi agar dapat dilakukan perubahan dengan cepat dan dapat digunakan kembali dengan aplikasi lainnya yang sejenis [10].

## 2.5 Laravel

Laravel adalah sebuah MVC *web development framework* untuk sebuah pengembangan aplikasi yang di harapkan dapat meningkatkan kualitas aplikasi yang di hasilkan, dengan mengurangi biaya pengembangan dan perbaikan serta menghasilkan *source code* yang rapih dan fungsional yang dapat meng-efesiansikan untuk implementasinya [12]. Laravel merupakan *framework* PHP yang menekankan pada kesederhanaan dan fleksibilitas pada desainnya [11]. Laravel merupakan *framework* PHP yang selalu *up-to-date* karena Laravel bersifat *open-source* dan dikembangkan secara bersama.

## 2.6 Systems Development Life Cycle (SDLC)

*Systems Development Life Cycle (SDLC)* merupakan kerangka kerja yang menyediakan urutan pekerjaan dalam membuat perangkat lunak. Tahapan pada proses SDLC pada umumnya adalah persyaratan fungsional perangkat lunak (*user requirement*), perancangan, pengujian, implementasi dan perawatan [13]. Dengan SDLC membuat pengembangan aplikasi menjadi terstruktur dan metodis karena harus dilakukan secara berurutan. Salah satu tipe SDLC yang paling banyak digunakan adalah *Waterfall*.

## 2.7 UML (Unified Modeling Language)

UML adalah sebuah bahasa pemodelan sebagai alat bantu untuk memvisualisasi, memspesifikasi, membangun dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis OO (Object Oriented) [14]. Di sisi lain UML sebagai bahasa pemodelan juga memberikan standar penulisan sebuah cetak biru (*blue print*) sebagai konsep bisnis proses, penulisan kelas-kelas dalam bahasa pemrograman yang spesifik, skema database dan komponen-komponen yang diperlukan dalam sistem *software*.

## 2.8 Black Box Testing

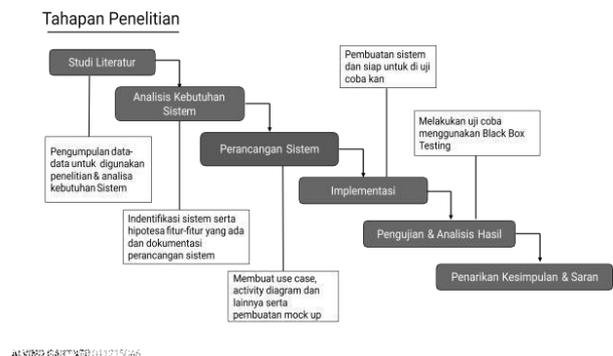
*Black Box Testing* atau uji fungsional sistem, berfokus pada pengujian atas program atau fungsional sistem yang di kembangkan. Dalam *black box* penguji/*tester* membutuhkan informasi terkait data *input* dan *output* yang diamati, tetapi *tester* tidak tahu bagaimana program atau sistem itu berkerja, karena *black box* mengabaikan *control structure* melainkan berfokus pada informasi apa yang

dihasilkan dari input yang diberikan apakah sesuai yang di harapkan oleh *user* [15].

## 3. METODE PENELITIAN

Metode yang digunakan dalam penelitian ini yaitu metode *Waterfall*. Model SDLC *waterfall* sering juga disebut model *sequential linear* atau alur hidup klasik. Model *Waterfall* menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai analisis, desain, pengkodean, pengujian, dan tahap pendukung.

Berikut adalah gambar model *Waterfall*:



Gambar 2. Waterfall Implementation

### 3.1 Requirements Definition

Pada tahap ini dilakukan pengumpulan data melalui observasi dari aplikasi terkait dan wawancara kepada pengguna dalam hal ini *user* keramik lantai dengan hasil yang diharapkan dari tahapan ini adalah daftar fitur-fitur yang dibutuhkan dalam aplikasi *web*.

### 3.2 System and Design

Pada tahap ini mengolah hasil data yang di dapat dari analisa kebutuhan dengan hasil yang di harapkan berupa *use case diagram*, *collaboration diagram*, ERD diagram dan *mockup* aplikasi.

### 3.3 Implementation and Unit Testing

Pada tahap ini peneliti akan menggunakan salah satu *framework* yaitu laravel untuk perancangan sistem pada bagian *back end* dan pada bagian *front end* menggunakan *Bootstrap* dengan database MySQL.

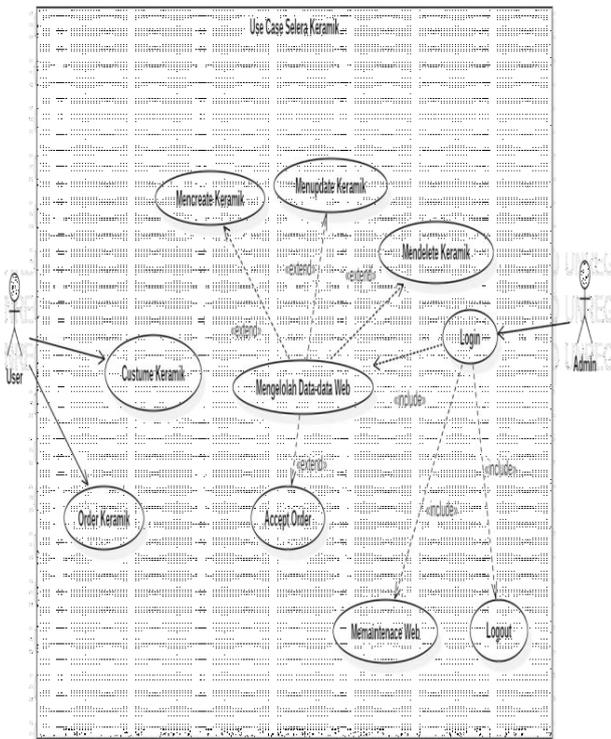
### 3.4 Integration and Testing

Pada tahap ini dilakukan uji *fungsionalitas* sistem menggunakan *black box testing* dengan hasil yang di harapkan adalah mengetahui apakah semua fungsi dan fitur sesuai dengan kebutuhan.

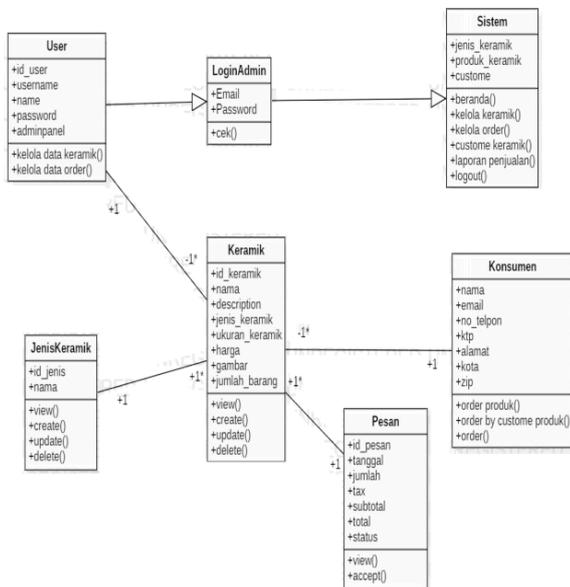
### 3.5 Operation and Maintenance

Pada tahap ini dilakukan uji *fungsionalitas* sistem menggunakan *black box testing* dengan hasil yang di

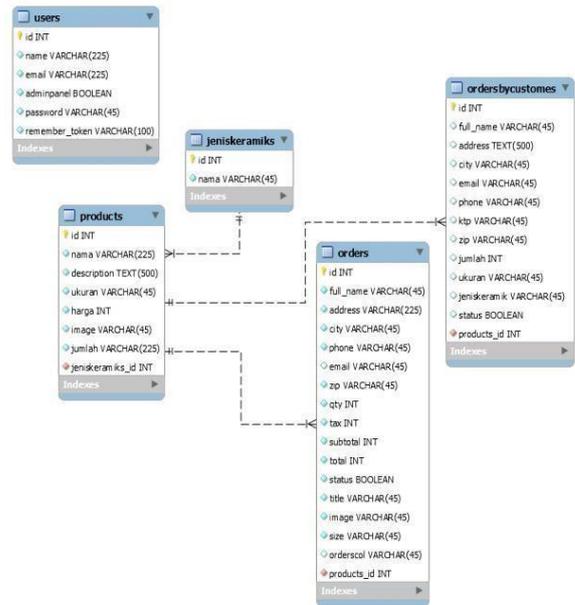
harapkan adalah mengetahui apakah semua fungsi dan fitur sesuai dengan kebutuhan.



Gambar 3. Diagram Use Case



Gambar 4. Class Diagram

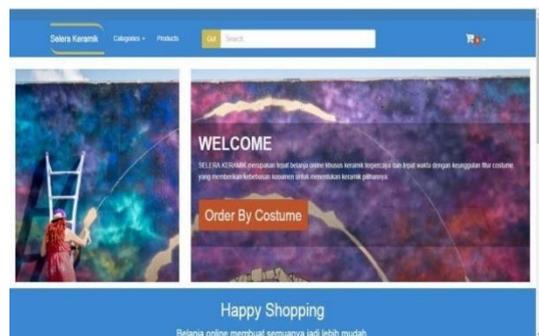


Gambar 5. Domain Model

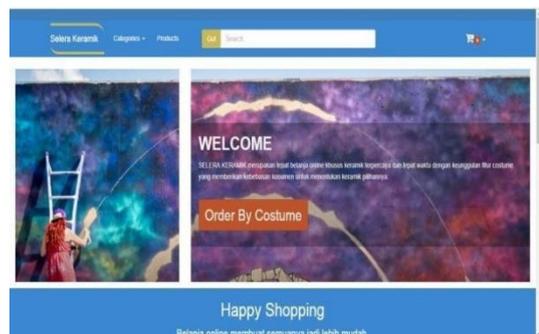
#### 4. IMPLEMENTASI DAN EVALUASI

Proses implementasi dan evaluasi yang di bahas mengenai alur kerja dari proses testing dengan menggunakan metode *black box*. Sistem Informasi pemesanan lantai keramik berbasis *web* dengan fitur *costume* ini merupakan aplikasi berbasis *web* yang dikembangkan dengan *framework* Laravel 5.5. Aplikasi yang dikembangkan masih berjalan pada *server local* dengan *hostname localhost*. Berikut adalah tampilan-tampilan dari implementasi.

##### 4.1 Halaman Publik atau Konsumen

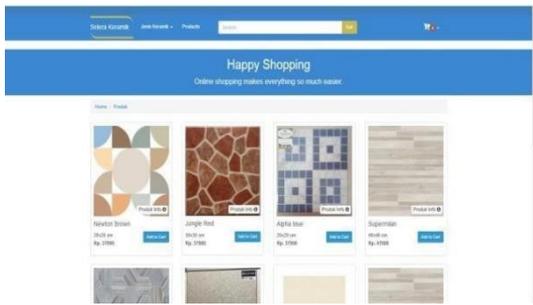


Gambar 6. Halaman Utama

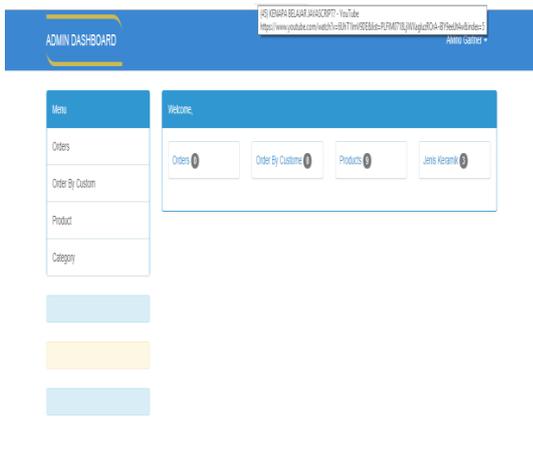


Gambar 7. Halaman All Product

4.2 Halaman Admin



Gambar 8. Halaman Fitur Utama



Gambar 9. Halaman All Orders

Berikut ini hasil pengujian fungsional terhadap prototype aplikasi bank sampah dengan menggunakan blackbox testing.

Tabel 1. Hasil Pengujian Fungsional

No	Skenario Uji	Hasil Uji	Keterangan
1	Register user admin	Berhasil	Data yang diinputkan berhasil masuk ke dalam database.
2	Login admin	Berhasil	User berhasil masuk ke dalam aplikasi.
3	Mengakses Selera Keramik Homepage	Sesuai	Melihat data produk yang dipasarkan secara detail, melakukan order dan order by custome.
4	Melakukan pencarian produk	Berhasil	Konsumen dapat melakukan pencarian produk dan melakukan
5	Melakukan pengurutan produk berdasarkan jenis keramik.	Berhasil	Konsumen dapat melakukan pengurutan produk dan melakukan order berdasarkan jenis keramik.

No	Skenario Uji	Hasil Uji	Keterangan
6	Melakukan order	Berhasil	Konsumen mengorder produk keramik lalu mengisikan jumlah barang dan ukurannya.
7	Melakukan pembayaran melalui bank	Berhasil	Konsumen mengisikan data diri yang nantinya akan diberikan notification melalui email atau nomer telepon untuk melakukan pembayaran.
8	Melakukan Order by custome	Berhasil	Konsumen mengisikan data diri dandesain keramik yang nantinya akan diberikan notification melalui email atau nomer telepon untuk melakukan pembayaran.
9	Admin mengakses data produk, jenis keramik, order dan Order by custome	Sesuai	Admin melihat banyak produk, jenis keramik, order dan order by custome.

Dari hasil pengujian fungsional pada aplikasi selera keramik berbasis web dari 13 butir uji dalam black box testing didapat 12 butir uji berjalan dengan baik atau 92,3% fitur berjalan dengan baik dan hanya satu fitur berjalan dengan catatan.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil penelitian Analisis dan Pengembangan Sistem Desain Asset untuk Keramik Lantai berbasis Aplikasi Website dihasilkan sistem informasi bernama selera keramik. Berikut ini kesimpulan dan saran yang penulis dapatkan.

1. Perancangan sistem informasi selera keramik dilakukan dengan tahapan-tahapan: pengumpulan data dan rujukan, studi pustaka untuk mendapatkan referensi aplikasi sejenis, desain aplikasi, pengembangan aplikasi, pengujian aplikasi dan implementasi.
2. Fitur aplikasi dari sistem informasi Selera Keramik diantaranya adalah: landing page, pesanan keramik, pesanan kustom keramik oleh pelanggan, pencarian produk keramik, pengurutan produk keramik, pembayaran melalui metode transfer. Dari hasil pengujian User Acceptance Test (UAT) yang yang

dilakukan oleh user admin dinyatakan 100% fitur berfungsi, dan 3 *user* konsumen didapat 5 fitur berjalan dengan baik (83,3%) dan 1 fitur berjalan dengan catatan upload bukti transfer belum berhasil.

## 5.2 Saran

Penulis menyadari bahwa sistem informasi Selera Keramik ini masih memiliki kekurangan, untuk itu apabila penelitian ini ingin dilanjutkan penulis akan memberikan beberapa saran mengenai bagian-bagian yang sebaiknya ditingkatkan yaitu:

1. Penelitian selanjutnya dapat menambahkan fitur *custome* keramik secara online di dalam aplikasi Selera Keramik dengan hasil gambar berbentuk file .img yang bisa di-*download* melalui aplikasi.
2. Penelitian selanjutnya dapat menambahkan bukti transfer bank dan menambahkan metode pembayaran melalui kartu kredit.
3. Penelitian selanjutnya dapat mengembangkan dari sisi tampilan aplikasi Selera Keramik agar lebih baik.

## DAFTAR PUSTAKA

- [1] T. Y. Kurniawan, "Kebutuhan Rumah Per Tahun Capai 1,4 Juta Unit," *Warta Ekonomi.co.id*, 21 November 2017, [Online]. Available: <https://www.wartaekonomi.co.id/read161810/kebutuhan-rumah-per-tahun-capai-14-juta-unit.html> [diakses 12 Maret 2018]
- [2] A. Erawan, "*Custom Homes Paramount Land* Menangi Dua Penghargaan," *Rumah.com*, 21 April 2016, [Online]. Available: <https://www.rumah.com/berita-properti/2016/4/123299/custom-homes-paramount-land-menangi-dua-penghargaan> [diakses 13 Maret 2018]
- [3] A. B. Susanto, "Potret-potret Gaya Hidup Modern," Penerbit Buku Kompas, 2001.
- [4] J. C. Mowen and M. Minor, "Akuntansi Manajemen," Jilid I, Edisi Kelima, Jakarta: Penerbit Erlangga, 2001.
- [5] Depdikbud RI, "Kamus Besar Bahasa Indonesia," hal. 589, Jakarta: Balai Pustaka, 1989.
- [6] S. Sabti, "Eksplorasi *Yii Framework* Sebagai Pendukung Pembuatan *Software* berbasis *Web* (Studi Kasus Aplikasi Forum)," *JBPTUNPASPP*, vol. 1, no. 1, pp. 5-1, 2012.
- [7] A. B. Warsito, M. Yusup, Yulianto, "Kajian *Yii Framework* dalam Pengembangan *Website* Perguruan Tinggi," *CCIT Journal*, Vol.7, No.3, 2014.
- [8] B. Sidik, "*Framework Codeigniter*," Bandung: Informatika, 2012.
- [9] A. Hidayat and B. Surarso, "Penerapan Arsitektur *Model View Controller* (MVC) dalam Rancang Bangun Sistem Kuis *Online* Adaptif," *SENTIKA*, hal. 58 – 64, Yogyakarta, 2012.
- [10] R. A. Sukamto and M. Shalahuddin, "Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)," Bandung: Modula, 2011.
- [11] D. Naista, "Bikin *Framework* PHP Sendiri dengan Teknik OOP dan MVC," Jakarta: Lokomedia, 2016.
- [12] B. P. Widodo and H. D. Purnomo, "Perancangan Aplikasi Pencarian Layanan Kesehatan berbasis HTML 5 *Geolocation*," *Jurnal Sistem Komputer*, 2016.
- [13] M. Tuteja, and D. Gaurav, "A Research Study on importance of Testing and Quality Assurance in *Software Development Life Cycle (SDLC) Models*," *International Journal of Soft Computing and Engineering (IJSCE)*, 2012.
- [14] H. Bangun, "Implementasi Sistem Informasi sebagai Media Promosi Katalog *Online* Pada Toko Buku dan Penyewaan Komik AI READ," *jbptunikompp*, Vol. 1, No. 1, hal.49-50, 2009.
- [15] W. E. Lewis, "*Software Testing and Continuous Quality Improvement*," 3rd Edition, U.S.A: Taylor & Francis Group, 2016.



## VISUALISASI IMPLEMENTASI AUTENTIFIKASI *DIGITAL SIGNATURE* MENGGUNAKAN FUNGSI HASH PADA FILE DIGITAL

Rufaidah Taqiyyah<sup>1</sup>, Ahmad Rio Adriansyah<sup>2</sup>

<sup>1,2</sup>Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri  
Jakarta Selatan, DKI Jakarta, Indonesia 12640  
rufaidah.chan@gmail.com , ahmad.rio.adriansyah@gmail.com

### Abstract

Information security is fundamental to maintain the integrity and authenticity of a document. Cryptography is the study of mathematical techniques related to aspects of information security. Cryptographic data integrity algorithms use to protect data blocks, such as protecting documents from changes, one of which is the digital signature algorithm. With a digital signature, we can verify the authenticity of the document and can find out whether the contents of the document have changed or not. However, no system is one hundred percent safe; one way to minimize losing security is to understand the security system. In this study, the author will create a system that visualizes the process of the digital signature algorithm to make it easier for someone to learn it.

**Keywords:** Information System Security, Cryptography, Digital Signature Algorithm

### Abstrak

Keamanan informasi sangatlah penting untuk menjaga integritas dan keaslian dari suatu dokumen. Kriptografi adalah studi teknik matematika yang berkaitan dengan aspek keamanan informasi. Algoritma integritas data kriptografi digunakan untuk melindungi blok data seperti melindungi dokumen dari perubahan, salah satunya adalah algoritma *digital signature*. Dengan *digital signature* kita dapat memverifikasi keaslian dari dokumen dan dapat mengetahui isi dokumen tersebut sudah diubah atau belum. Namun tidaklah ada sistem yang seratus persen aman, salah satu cara untuk meminimalisir resiko dari kehilangan keamanan adalah dengan memahami sistem keamanan tersebut. Pada penelitian kali ini, penulis akan membuat sistem yang memvisualisasikan proses dari algoritma *digital signature* untuk memudahkan seseorang dalam mempelajarinya.

**Kata kunci:** Keamanan Sistem Informasi, Kriptografi, Algoritma Digital Signature

### 1. PENDAHULUAN

Keamanan sistem informasi sangatlah penting untuk menjaga integritas dan keaslian dari suatu dokumen. Salah satu cara untuk menjaga integritas dan keaslian dari suatu dokumen adalah dengan menggunakan *digital signature*. Dengan *digital signature* kita dapat memverifikasi keaslian dari dokumen dan dapat mengetahui isi dokumen tersebut sudah diubah atau belum.

Namun tidaklah ada sistem yang seratus persen aman, selalu ada seseorang yang mencari celah dari suatu sistem, entah untuk membuat sistem tersebut lebih baik ataupun untuk menghancurkan sistem tersebut. Ketika seseorang merasa nyaman dengan suatu sistem, maka kewaspadaan terhadap keamanannya akan melemah. Banyak resiko yang didapatkan ketika keamanan itu hilang, mulai dari yang kecil sampai besar.

Pemahaman terhadap keamanan sangatlah penting sebagai salah satu cara untuk meminimalisir resiko dari kehilangan keamanan. Oleh sebab itu penulis akan memvisualisasikan proses dari algoritma *digital signature* untuk memudahkan seseorang dalam mempelajari salah satu sistem keamanan dokumen.

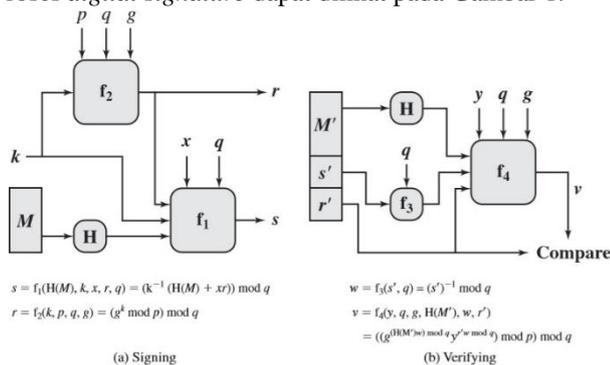
### 2. KONSEP KEAMANAN KOMPUTER, ALGORITMA INTEGRITAS DATA KRIPTOGRAFI, *DIGITAL SIGNATURE ALGORITHM*

Menurut *National Institute of Standards and Technology (NIST) Computer Security Handbook*, keamanan komputer adalah perlindungan yang diberikan pada sistem informasi otomatis untuk mencapai tujuan yang berlaku untuk menjaga integritas, ketersediaan, dan kerahasiaan sumber daya sistem informasi (termasuk *hardware, software*,

firmware, informasi atau data, dan telekomunikasi). Dari pengertian tersebut dapat diambil tiga konsep untuk memenuhi tujuan utama dari keamanan komputer, yaitu integritas, ketersediaan, dan kerahasiaan yang membentuk apa yang sering disebut sebagai *CIA triad*. Namun di beberapa bidang keamanan menyatakan perlu adanya konsep tambahan, diantaranya yang sering disebutkan adalah keaslian dan akuntabilitas [1]. Pada penelitian ini akan difokuskan kepada dua konsep keamanan yaitu integritas untuk menjamin keaslian dari dokumen dan keaslian untuk memastikan dokumen bersifat asli, dapat diverifikasi, dan dapat dipercaya.

Kriptografi adalah studi teknik matematika yang berkaitan dengan aspek keamanan informasi. Layanan dasar yang disediakan kriptografi adalah kemampuan mengirim informasi antar pengguna tanpa ada orang lain yang dapat membacanya [2]. Algoritma integritas data digunakan untuk melindungi blok data, seperti melindungi suatu dokumen dari perubahan. *Digital signature* adalah salah satu algoritma untuk menjaga integritas suatu dokumen.

Teknologi tanda tangan digital memanfaatkan sepasang kunci privat-publik yang dibuat untuk keperluan seseorang. Kunci privat disimpan oleh pemiliknya dan digunakan untuk membuat tanda tangan digital, sedangkan kunci publik dapat digunakan oleh siapa saja yang ingin memeriksa tanda tangan digital yang bersangkutan pada suatu dokumen [3]. DSA adalah salah satu mekanisme otentikasi, karna algoritma ini memungkinkan pembuat pesan untuk melampirkan kode yang bertindak sebagai tanda tangan digital. Dengan algoritma ini, pembuat dan penerima pesan dapat terlindungi satu sama lain yang memungkinkan terjadi perselisihan antar keduanya [5]. Proses *digital signature* dapat dilihat pada Gambar 1.

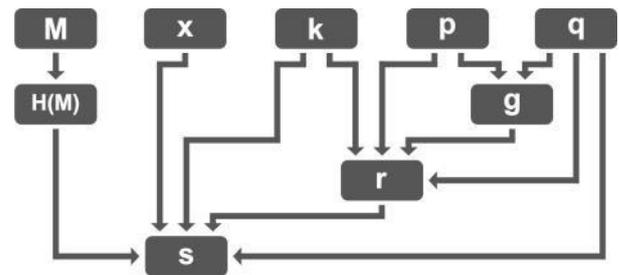


Gambar 1. Proses Digital Signature [4]

### 3. ANALISIS ALGORITMA DIGITAL SIGNATURE

Analisis ini dilakukan berdasarkan studi pustaka yang sudah dilakukan sebelumnya. Berdasarkan hasil analisis tersebut, penulis membuat daftar apa saja yang dibutuhkan dalam pembuatan algoritma sistem keamanan yaitu *digital signature* yang terdiri dari dua proses yaitu *signing* dan *verifying*.

#### 3.1 Proses Signing

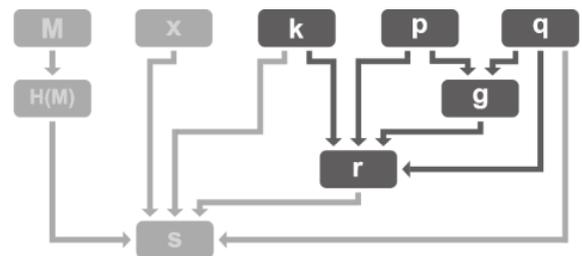


Gambar 2. Kebutuhan Proses Signing

Proses *signing* adalah proses untuk membuat *digital signature* terhadap dokumen yang menandakan bahwa dokumen tersebut adalah benar milik dari seseorang yang membuat file tersebut. *Output* dari proses *signing* adalah nilai *s* dan nilai *r* yang nantinya akan dipakai untuk proses *verifying*. Keseluruhan *item* yang dibutuhkan dalam proses *signing* dapat dilihat pada Gambar 2.

##### 1. Nilai r

Nilai *r* adalah nilai yang didapatkan dari komputasi nilai rahasia per-dokumen (*k*) dan komponen *Global public-key* yaitu bilangan prima (*p*), pembagi utama (*q*), dan nilai *g*.



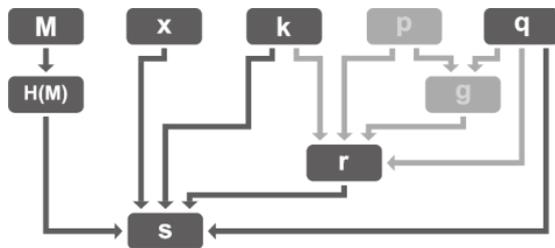
Gambar 3. Proses Mendapatkan Nilai r

Tabel 1. Komputasi untuk Mendapatkan Nilai r

No	Yang Dibutuhkan	Komputasi
1	Nilai rahasia per-dokumen ( <i>k</i> )	Random atau pseudorandom integer $0 < k < q$
2	Nilai <i>p</i> (komponen <i>Global public-key</i> )	Bilangan prima $2^{(L-1)} < p < 2^L$ <i>L</i> = kelipatan $64512 \leq L \leq 1024$ $(g^k \bmod p) \bmod q$
3	Nilai <i>q</i> (komponen <i>Global public-key</i> )	Pembagi utamadari ( <i>p</i> -1) $2^{159} < q < 2^{160}$
4	Nilai <i>g</i> (komponen <i>Global public-key</i> )	$h^{(p-1)/q} \bmod p$ $1 < h < (p-1)$ $g > 1$

2. Nilai s

Nilai s adalah nilai yang didapatkan dari komputasi nilai hash (H(M)), *private-key* (x), nilai rahasia per-dokumen (k), nilai r, dan pembagi utama (q).



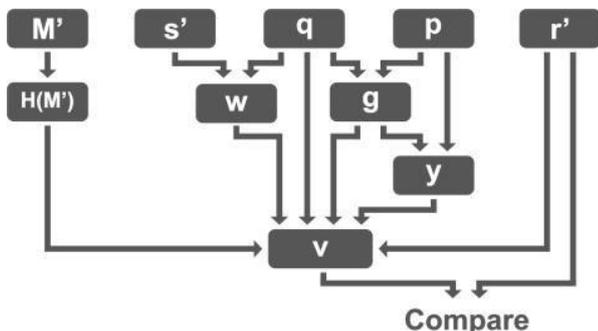
Gambar 4. Proses Mendapatkan Nilai s

Tabel 2. Komputasi untuk Mendapatkan Nilai s

No	Yang Dibutuhkan	Komputasi
1	Nilai hash (H(M))	$M \rightarrow f(H) \rightarrow H(M)$
2	<i>Private-key</i> (x)	Random atau pseudorandom integer $0 < x < q$
3	Nilai rahasia per-dokumen (k)	Random atau pseudorandom integer $0 < k < q$ $(k^{-1}(H(M) + xr)) \bmod q$
4	Nilai r	$(g^k \bmod p) \bmod q$
5	Nilai q (komponen <i>Global public-key</i> )	Pembagi utamadari (p-1) $2^{159} < q < 2^{160}$

3.2 Proses Verifying

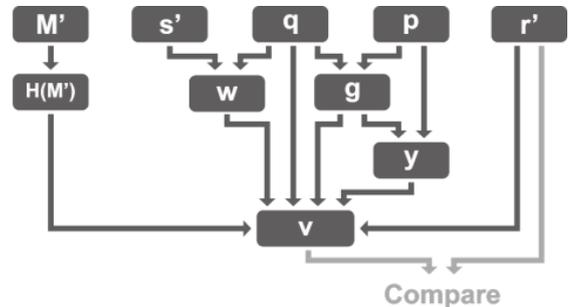
Proses *verifying* adalah proses untuk membuktikan bahwa dokumen tersebut adalah benar milik dari seseorang yang membuat file tersebut atau sudah diubah oleh orang lain. *Output* dari *verifying* adalah nilai v yang nantinya akan dibandingkan dengan nilai r. Keseluruhan *item* yang dibutuhkan dalam proses *verifying* dapat dilihat pada Gambar 5.



Gambar 5. Kebutuhan Proses Verifying

1. Nilai v

Nilai v adalah nilai yang didapatkan dari komputasi nilai hash (H(M)); nilai hasil *signing* yaitu nilai s dan nilai r; komponen *Global public-key* yaitu bilangan prima (p), pembagi utama (q), dan nilai g; *public-key* (y); dan nilai w.



Gambar 6. Proses Mendapatkan Nilai v

Tabel 3. Komputasi untuk Mendapatkan Nilai v

No	Yang Dibutuhkan	Komputasi
1	Nilai hash (H(M))	$M \rightarrow f(H) \rightarrow H(M)$
2	Nilai r (hasil <i>signing</i> )	$(g^k \bmod p) \bmod q$
3	Nilai s (hasil <i>signing</i> )	$(k^{-1}(H(M) + xr)) \bmod q$
4	Nilai p (komponen <i>Global public-key</i> )	Bilangan prima $2^{(L-1)} < p < 2^L$ L = kelipatan 64 $512 \leq L \leq 1024$
5	Nilai q (komponen <i>Global public-key</i> )	Pembagi utama dari (p-1) $2^{159} < q < 2^{160}$
6	Nilai g (komponen <i>Global public-key</i> )	$h^{(p-1)/q} \bmod p$ $1 < h < (p-1)$ $g > 1$
7	<i>Public-key</i> (y)	$g^x \bmod p$
8	Nilai w	$(s^{-1}) \bmod q$

2. Compare Nilai v dan Nilai r

Proses ini adalah proses terakhir dari *verifying*. Pada proses ini nilai v akan dibandingkan dengan nilai r. Jika nilai v sama dengan nilai r, maka dokumen tersebut adalah benar milik dari seseorang yang membuat file tersebut. Namun jika nilai v tidak sama dengan nilai r, berarti menunjukkan bahwa dokumen tersebut telah diubah oleh orang lain.

Tabel 4. Komputasi dari Nilai v dan Nilai r

No	Yang Dibutuhkan	Komputasi
1	Nilai r	$(g^k \bmod p) \bmod q$

No	Yang Dibutuhkan	Komputasi
2	Nilai v	$((g^{u1} y^{u2}) \bmod p) \bmod q$ $u1 = (H(M') w) \bmod q$ $u2 = (r') w \bmod q$

#### 4. VISUALISASI ALGORITMA DIGITAL SIGNATURE

Dari analisis algoritma yang sudah dilakukan, penulis akan memvisualisasikan algoritma tersebut agar nantinya mudah di pelajari oleh pengguna.

##### 4.1 Proses Signing

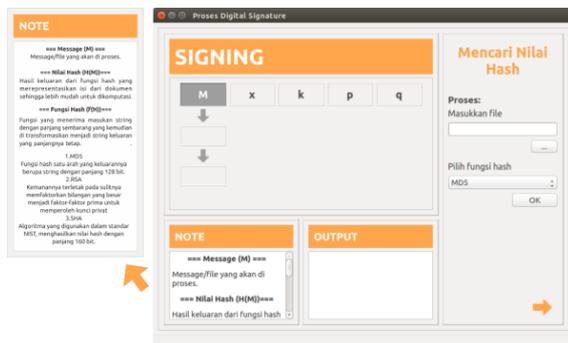
Pada proses *signing* ini akan divisualisasikan tahap demi tahap dari memasukkan file yang akan di-*signing* sampai mendapatkan hasil *signing* yaitu nilai r dan nilai s. Tahapan dari proses *signing* adalah sebagai berikut:

1. Mencari Nilai Hash.
2. Mencari Bilangan Prima.
3. Mencari Pembagi Utama.
4. Mencari Random atau Pseudorandom Integer.
5. Mencari Nilai g.
6. Mencari Nilai r.
7. Mencari Nilai s.

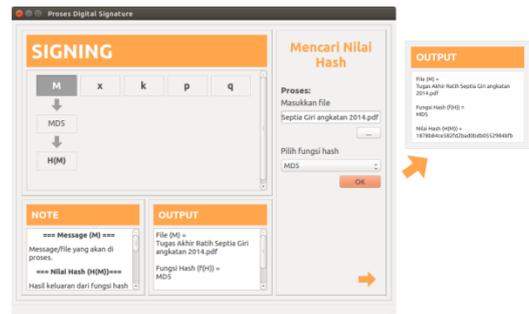
Dari tahapan-tahapan di atas, akan dijelaskan secara singkat penjelasan beserta visualisasi dari proses *signing* yang telah dibuat oleh penulis.

##### 1. Mencari Nilai Hash

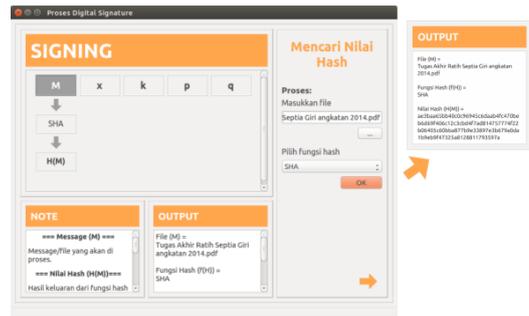
Pada tahapan ini pengguna dapat memasukkan file berformat pdf dan dapat memilih fungsi hash yang diinginkan untuk memproses file tersebut. Ketika tombol "OK" ditekan, maka file tersebut akan di proses menjadi nilai hash dengan panjang tetap sesuai dengan fungsi hash yang pengguna pilih. Jika fungsi hash yang dipilih adalah MD5, maka nilai hash yang dihasilkan berupa string dengan panjang 128 bit. Jika fungsi hash yang dipilih adalah SHA, maka nilai hash yang dihasilkan berupa string dengan panjang 160 bit.



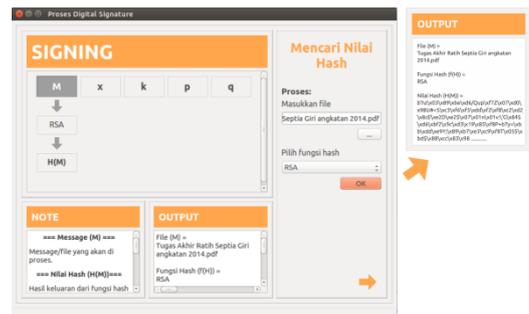
Gambar 8. Tampilan Pertama Sebelum Diproses



Gambar 9. Tampilan Setelah Diproses menggunakan MDS



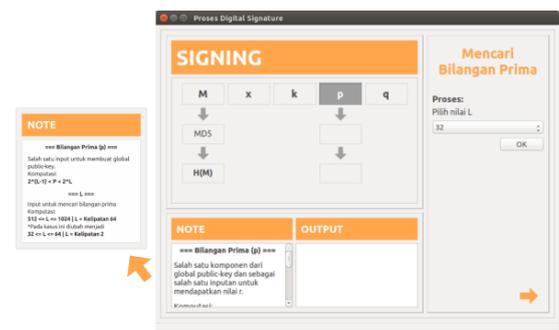
Gambar 10. Tampilan Setelah Diproses menggunakan SHA



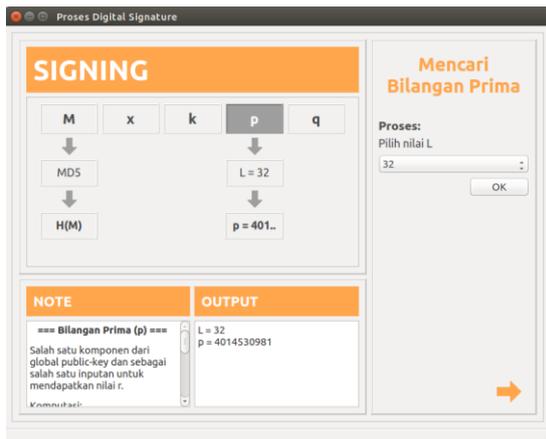
Gambar 11. Tampilan Setelah Diproses menggunakan SHA

##### 2. Mencari Bilangan Prima

Pada tahap ini akan mencari salah satu komponen *global public-key* yaitu nilai p. Nilai p akan dipilih secara *random*, dimana nilai p adalah bilangan prima dan  $2(L-1) < p < 2L$ . Pada algoritma yang semestinya, nilai L adalah kelipatan 64 dimana  $512 \leq L \leq 1024$ . Namun, pada sistem ini nilai L diperkecil agar komputasi dapat dilakukan secara cepat. Pengguna dapat memilih nilai L dengan 32 ataupun 64.



Gambar 12. Tampilan Kedua Sebelum Diproses



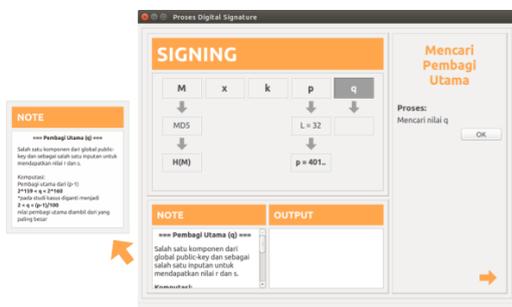
Gambar 13. Tampilan Setelah Diproses dengan L = 32



Gambar 14. Tampilan Setelah Diproses dengan L = 64

3. Mencari Pembagi Utama

Pada tahap ini akan mencari salah satu komponen *global public-key* yaitu nilai q. Nilai q adalah pembagi utama dari (p-1) dimana nilai q adalah bilangan prima dan  $2159 < q < 2160$ . Namun pada sistem ini karna nilai L diperkecil maka nilai q tidak akan mencapai 2159 sehingga batas tersebut dihapuskan, pada sistem ini nilai q akan diambil dari pembagi yang paling besar.



Gambar 15. Tampilan Ketiga Sebelum Diproses



Gambar 16. Tampilan Ketiga Setelah Diproses

4. Mencari Random atau Pseudorandom Integer
- Pada tahap ini akan mencari nilai x sebagai private-key dan nilai k sebagai nilai rahasia perdokumen, dimana nilai x dan nilai k adalah *random* atau *pseudorandominteger* antara 0 dan nilai q.



Gambar 17. Tampilan Keempat Sebelum Diproses



Gambar 18. Tampilan Keempat Setelah Diproses

5. Mencari Nilai g
- Pada tahap ini akan mencari salah satu komponen *global public-key* yaitu nilai g. Nilai g didapatkan dari  $h(p-1)/q \text{ mod } p$  dimana  $1 < h < (p-1)$ .



Gambar 19. Tampilan Kelima Sebelum Diproses

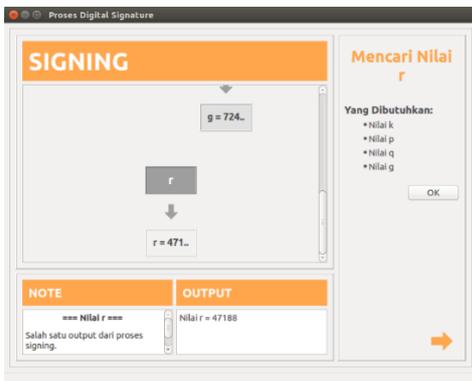


Gambar 20. Tampilan Kelima Setelah Diproses

6. Mencari Nilai r  
 Pada tahap ini akan mencari salah satu dari *output* proses *signing* yaitu nilai r, dimana nilai r didapatkan dari  $(gk \text{ mod } p) \text{ mod } q$ .



Gambar 21. Tampilan Keenam Sebelum Diproses



Gambar 22. Tampilan Keenam Setelah Diproses

7. Mencari Nilai s  
 Pada tahap ini akan mencari salah satu dari *output* proses *signing* yaitu nilai s, dimana nilai s didapatkan dari  $(k-1)(H(M) + xr) \text{ mod } q$ .



Gambar 23. Tampilan Ketujuh Sebelum Diproses



Gambar 24. Tampilan Ketujuh Setelah Diproses

8. Hasil Proses *Signing*  
 Pada tahap ini akan ditampilkan input file yang di proses dan hasil dari *signing* file tersebut yaitu nilai r dan nilai s.

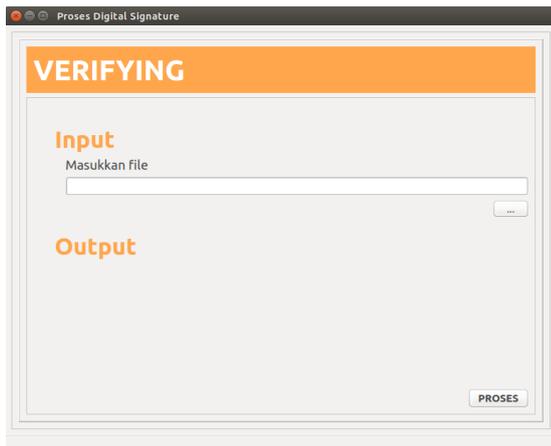


Gambar 25. Hasil dari Proses *Signing*

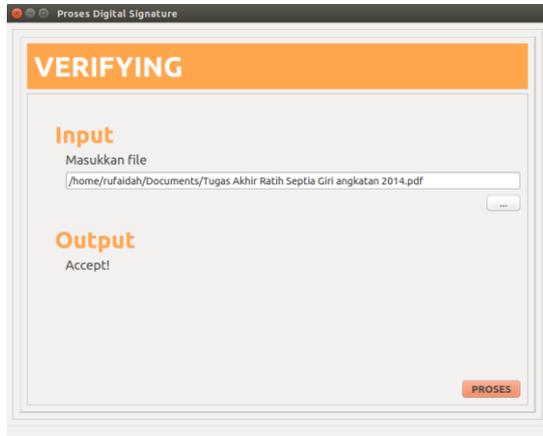
#### 4.2 Proses *Verifying*

Pada proses *verifying* ini tidak divisualisasikan secara detail seperti proses *signing*. Pada tahap ini pengguna

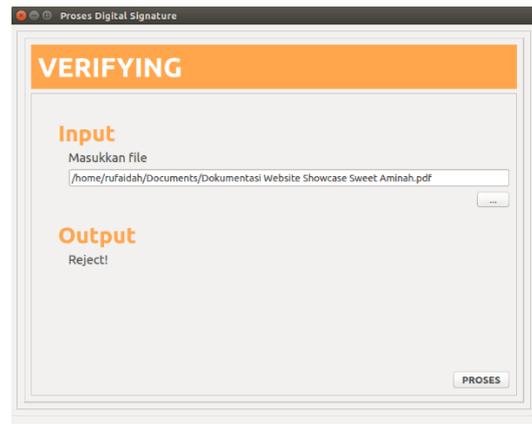
hanya dapat memasukkan file yang akan di *verify*, kemudian file tersebut akan diproses sehingga mendapatkan nilai  $v$ . Nilai  $v$  didapatkan dari  $((gu_1 yu_2) \bmod p) \bmod q$  dimana  $u_1 = (H(M') w) \bmod q$  dan  $u_2 = (r' w) \bmod q$ . Kemudian nilai  $v$  akan dibandingkan dengan nilai  $r$ , jika nilai  $v$  sama dengan nilai  $r$  maka akan tampil tulisan “Accept!” yang berarti file tersebut adalah benar file yang sudah di-*signing* dan tidak diubah isinya oleh orang lain, dan jika nilai  $v$  tidak sama dengan nilai  $r$  maka akan tampil tulisan “Reject!” yang berarti file tersebut berbeda dari file yang sudah di *signing* ataupun isi file tersebut sudah diubah oleh orang lain.



Gambar 26. Tampilan Awal Proses *Verifying*



Gambar 27. Tampilan Setelah Dimasukkan File yang Sama



Gambar 28. Tampilan Setelah Dimasukkan File yang Sama

## 5. KESIMPULAN

Sistem visualisasi dari algoritma *digital signature* ini telah dicoba oleh tiga narasumber dan kemudian narasumber menyatakan pendapatnya mengenai sistem ini. Mereka menyatakan bahwasanya alur sudah jelas namun tampilan terlalu kaku, dan juga kurang informatif.

Oleh karena itu dapat disimpulkan bahwasanya sistem ini tidak cocok untuk digunakan oleh orang awam, seseorang harus mempunyai ilmu mengenai *digital signature* terlebih dahulu untuk bisa memahami sistem ini.

Untuk penelitian kedepannya diharapkan dapat menambahkan informasi lebih banyak sehingga lebih banyak ilmu yang dapat di ambil oleh pengguna.

## DAFTAR PUSTAKA

- [1] W. Stallings, “*Cryptography and Network Security Principles and Practice*,” pp. 9-11, United States of America, 2011.
- [2] M. K. Islam, M. Hossain, M. Nashiry, “*Security of Cryptographic Algorithm SHA and MD5*,” p. 28, 2012
- [3] N. Herawati, R. R. Isnanto, A. Fatchurrohman, “Perancangan dan Implementasi DSA (*Digital Signature Algorithm*) menggunakan Bahasa Pemrograman Java,” pp. 1-2, 2008
- [4] W. Stallings, “*Cryptography and Network Security Principles and Practice*,” p. 406, United States of America, 2011.
- [5] W. Stallings, “*Cryptography and Network Security Principles and Practice*,” pp. 396-398, United States of America, 2011.



## ANALISIS DAN EVALUASI ALGORITMA *MAPREDUCE WORDCOUNT* PADA CLUSTER *HADOOP* MENGGUNAKAN INDIKATOR KECEPATAN

Robiyatul Adawiyah<sup>1</sup>, Sirojul Munir<sup>2</sup>

<sup>1,2</sup>Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri  
Jakarta Selatan, DKI Jakarta, Indonesia 12640  
[arobiyatul8@gmail.com](mailto:arobiyatul8@gmail.com), [rojulman@gmail.com](mailto:rojulman@gmail.com)

### Abstract

*This research proposes to analyze the speed of the MapReduce algorithm on the Hadoop cluster and find out the time it takes to process GDELTA data on Hadoop. This study uses qualitative analysis methods. The data analysis concludes that the Word Count algorithm applied to the GDELTA data set can run on the Hadoop cluster. The speed of the WordCount algorithm on MapReduce, which applies to the GDELTA data set on Hadoop, affects if the nodes using add, which in the study used as many as two physical machine nodes. Hadoop can process data that is large and a lot because Hadoop processes data in a distributed manner. Hadoop speed can adjust by adding nodes as well as other settings such as block size.*

**Keywords:** *Hadoop, MapReduce, WordCount*

### Abstrak

Penelitian diajukan untuk menganalisis kecepatan algoritma *MapReduce* pada *cluster Hadoop* dan mengetahui waktu yang dibutuhkan dalam mengolah data GDELTA pada *Hadoop*. Penelitian ini menggunakan metode analisis kualitatif. Berdasarkan analisa data yang telah dilakukan, diperoleh kesimpulan bahwa algoritma *WordCount* yang diterapkan pada data set GDELTA dapat berjalan pada *cluster Hadoop*. Kecepatan algoritma *WordCount* pada *MapReduce* yang diterapkan untuk data set GDELTA pada *hadoop* berpengaruh apabila node yang digunakan ditambah, dimana dalam penelitian menggunakan sebanyak 2 *node physical machine*. *Hadoop* dapat mengolah data yang memiliki ukuran besardan banyak karena *Hadoop* mengolah data secara terdistribusi. Kecepatan *Hadoop* dapat diatur dengan menambahkan *node* dan juga pengaturan lainnya seperti halnya *block size*.

**Kata kunci:** *Hadoop, MapReduce, WordCount*

### 1. PENDAHULUAN

Saat ini, teknologi informasi berkembang pesat dengan ditandai melimpahnya data yang tersedia, mula dari data di media sosial, *E-Commerce*, kependudukan, industri, dan masih banyak lagi. Data yang semakin besar ukurannya tersebut dan sudah sangat sulit untuk dikoleksi, disimpan, dikelola maupun dianalisa dengan menggunakan sistem *database* biasa dikarenakan ukurannya yang terus bertambah disebut dengan *big data* [2]. *International Data Corporation* memperkirakan ukuran data semesta digital berada pada angka 0.18 zettabytes (1 zettabytes =  $10^{12}$  bytes), serta meramalkan akan menjadi 10 kali lipatnya setiap 5 tahun [2].

Salah satu data dengan kapasitas dan ukuran besar yaitu data dari GDELTA (*Global Dataset of Event, Language and Tone*) *Project* dimana data tersebut menampung data dari seluruh dunia. Proyek GDELTA adalah yang terbesar, paling

komprehensif, dan database terbuka dengan resolusi tinggi yang pernah dibuat. Pada data tahun 2015 saja mencatat hampir tiga perempat dari satu triliun snapshot emosional dan lebih dari 1,5 miliar referensi lokasi, sementara total arsipnya mencapai lebih dari 215 tahun, menjadikannya salah satu *dataset open-access spatio-temporal* terbesar yang ada dan mendorong batas-batas studi "*big data*" dari masyarakat manusia global. *Global Knowledge Graph*-nya menghubungkan orang-orang, organisasi, lokasi, tema, jumlah, gambar, dan emosi dunia ke dalam satu jaringan tunggal di seluruh dunia [5]. Adanya suatu sistem dalam pengolahan data memungkinkan pengguna untuk mempermudah dalam mengolah suatu data, ada beberapa sistem atau *framework* yang menggunakan teknik dan algoritma dalam mengolah suatu *big data* seperti *clustering* dan klasifikasi dari suatu *dataset*. *Clustering* adalah salah satu teknik yang saat ini banyak digunakan dalam mengolah

suatu *dataset*. Tujuan *clustering* itu sendiri adalah untuk mengumpulkan data dan mengelompokkannya menjadi sekumpulan data yang sangat penting dan mempunyai nilai ke dalam sebuah kluster untuk selanjutnya dapat digunakan.

Dalam menggunakan sistem pengolahan data ada beberapa teknik dan algoritma *clustering* yang dapat digunakan, namun penggunaan algoritma untuk mengolah dan memproses data yang terlalu banyak membutuhkan waktu yang sangat lama dalam proses komputasinya jika hanya menggunakan algoritma yang sekuensial. *Hadoop* merupakan salah satu *framework* yang sering digunakan dalam hal pengolahan data dengan jumlah yang besar dimana *Hadoop* memiliki sistem khusus untuk mendukung kinerja komputasi yang dirancang khusus untuk menyimpan dan menganalisis data dalam jumlah yang sangat besar dan tidak terstruktur.

### 1.1 Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah “Bagaimana algoritma *WordCount* diterapkan pada *dataset Global Dataset of Events, Language and Tone (GDEL T)*” dengan pertanyaan penelitian ini adalah:

1. Apakah algoritma *WordCount* yang diterapkan pada *dataset GDEL T* dapat berjalan pada cluster *Hadoop*?
2. Apakah kecepatan algoritma *WordCount* pada *dataset GDEL T* dengan bertambahnya *node* akan lebih cepat?

### 1.2 Tujuan dan Manfaat Penelitian

1. Menganalisis kecepatan algoritma *MapReduce* pada cluster *Hadoop*.
2. Mengetahui waktu yang dibutuhkan dalam mengolah data *GDEL T* pada *Hadoop*.

### 1.3 Batasan Masalah

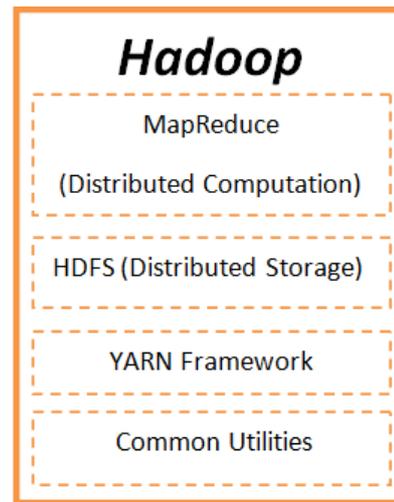
Batasan masalah dalam penelitian ini adalah:

1. Menggunakan sistem operasi Ubuntu.
2. Menggunakan *Hadoop*.
3. Menggunakan dua atau tiga *node*.

## 2. METODE PENELITIAN

### 2.1 *Hadoop*

*Hadoop* adalah *framework* Apache berjenis *open source* yang ditulis dalam Java yang memungkinkan pemrosesan terdistribusi dari kumpulan data besar kelompok komputer menggunakan model pemrograman sederhana. *Hadoop* bekerja di lingkungan yang menyediakan penyimpanan dan komputasi terdistribusi di seluruh kluster komputer. *Hadoop* dirancang untuk meningkatkan dari satu *server* hingga ribuan mesin, masing-masing menawarkan komputasi dan penyimpanan lokal [3].



Gambar 1. Arsitektur *Hadoop*

*Hadoop Framework* mencakup empat modul berikut :

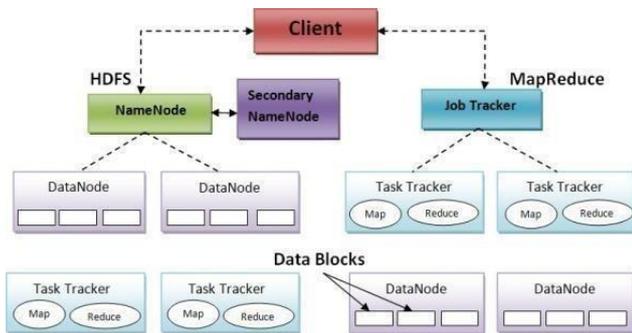
1. *Hadoop Common*  
*Hadoop Common* adalah pustaka dan utilitas Java yang diperlukan oleh modul *Hadoop* lain. Pustaka-pustaka ini menyediakan file sistem dan abstraksi tingkat OS dan berisi file Java yang diperlukan dan skrip yang diperlukan untuk memulai *Hadoop*.
2. *Hadoop YARN*  
*YARN* adalah kerangka kerja untuk penjadwalan pekerjaan dan pengelolaan sumber daya kluster.
3. *HDFS*  
Sistem file terdistribusi yang menyediakan akses *throughput* tinggi ke data aplikasi.
4. *MapReduce*  
*MapReduce* adalah sistem berbasis-*YARN* untuk pemrosesan paralel dari kumpulan data besar.

### 2.2 *HDFS*

*Hadoop Distributed File System (HDFS)* didasarkan pada *Google File System (GFS)* dan menyediakan sistem file terdistribusi yang dirancang untuk dijalankan pada kelompok besar mesin komputer kecil dengan cara yang dapat diandalkan dan toleran. *HDFS* menggunakan metode ini ketika mereplikasi data untuk redundansi data di beberapa rak. Pendekatan ini mengurangi dampak pemadaman listrik rak atau kegagalan sakelar; jika salah satu kegagalan perangkat keras ini terjadi, data akan tetap tersedia [3].

*HDFS* menggunakan arsitektur *master/slave* dimana *master* terdiri dari *NameNode* tunggal yang mengelola metadata sistem file dan satu atau lebih *slave DataNodes* yang menyimpan data aktual. Sebuah file dalam namespace *HDFS* dibagi menjadi beberapa blok dan blok tersebut disimpan dalam satu set *DataNodes*. *NameNode* menentukan pemetaan blok ke *DataNodes*. *DataNodes*

menangani operasi baca dan tulis dengan sistem file. Mereka juga mengurus pembuatan blok, penghapusan dan replikasi berdasarkan instruksi yang diberikan oleh *NameNode*.



Gambar 2. HDFS

*Hadoop* didukung oleh platform GNU/Linux. Kerangka *Hadoop* dapat dikonfigurasi dalam tiga mode berikut: *Mode Standalone*, *Pseudo-Distributed Mode*, dan *Mode Terdistribusi Penuh*. Secara default, *Hadoop* dikonfigurasi untuk berjalan dalam mode tidak terdistribusi, sebagai satu proses Java. Ini berguna untuk *debugging*. Ini juga dapat dijalankan pada *node* tunggal dalam mode *pseudo-distributed* dimana setiap *daemon* berjalan dalam proses Java yang terpisah. Mode *pseudo-distributed* juga disebut sebagai *nodenode* tunggal.

### 2.3 MapReduce

*MapReduce* sebuah *framework* untuk aplikasi dan *programming* yang dikenalkan oleh *Google* dan digunakan untuk melakukan suatu pekerjaan dari komputasi terdistribusi yang dijalankan pada sebuah *cluster* [3]. *MapReduce* ini terdiri dari konsep fungsi *map* dan *reduce* yang biasa digunakan pada *functional programming* [4].

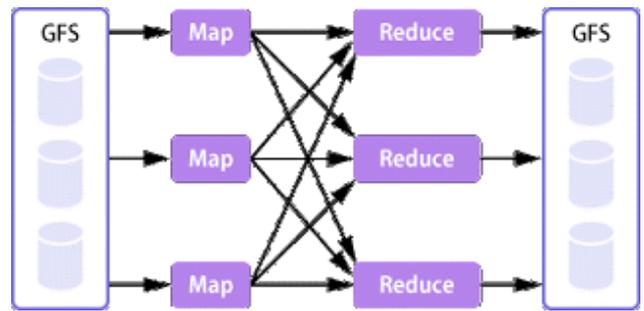
Salah satu program yang menggunakan konsep *MapReduce* yang telah disediakan oleh *Hadoop* adalah *WordCount*. *WordCount* merupakan program yang bertujuan untuk menghitung kata pada *file plaintext*. Proses *MapReduce* pada *WordCount* ini dibagi menjadi 2 tahap yaitu proses *mapping* dan *reducing*.

### 2.4 Tahapan Proses MapReduce

*Hadoop* menyediakan dua jenis *slot* untuk melakukan *MapReduce* yaitu *slot map* dan *slot reduce*. Secara default *Hadoop* telah menentukan jumlah *slot map* dan *slot reduce* untuk setiap *node* yaitu dua *slot map* dan satu *slot reduce*. Pada saat memproses data, *Hadoop* terlebih dahulu melakukan proses *mapping* pada *task* yang terdapat pada *slot map* sampai selesai kemudian dilanjutkan dengan proses *reduce* pada *slot reduce*.

*MapReduce* terdiri atas tiga tahap, yaitu tahap *map*, tahap *shuffle*, dan terakhir tahap *reduce*. Untuk tahapan *shuffle* dan

*reduce* digabungkan ke dalam satu tahap besarnya yaitu tahap *reduce*.



Gambar 3. MapReduce

1. Tahap *map*, memproses data inputan yang umumnya berupa file yang tersimpan dalam HDFS, inputan tersebut kemudian diubah menjadi *tuple* yaitu pasangan antara *key* dan *value*-nya.
2. Tahap *reduce*, memproses data inputan dari hasil proses *map*, yang kemudian dilakukan tahap *shuffle* dan *reduce* yang hasil *dataset* barunya disimpan di HDFS kembali.

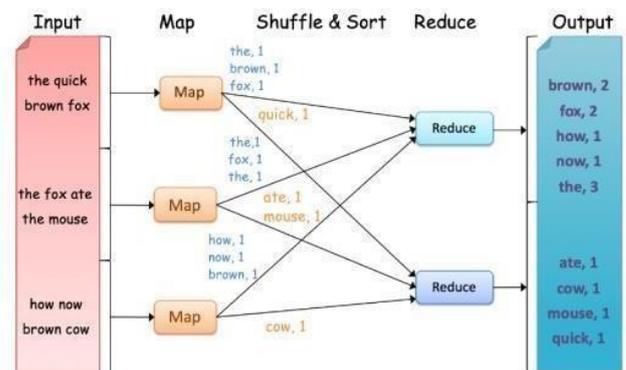
## 3. PERANCANGAN SISTEM

### 3.1 Algoritma Wordcount

*Wordcount* adalah algoritma yang sederhana dan mudah dipahami yang dapat dengan mudah diimplementasikan sebagai aplikasi *mapreduce*. Diberikan seperangkat dokumen teks, program menghitung jumlah kemunculan setiap kata. Algoritma ini terdiri dari tiga bagian utama [7].

1. Main Program
2. Mapper
3. Reducer

Berikut contoh dari *Wordcount*:



Gambar 4. Contoh Wordcount

Fungsi *map* mengeluarkan setiap kata ditambah dengan jumlah perhitungan yang terkait. Dokumen yang dimasukkan *tokenized*, dimana *key* adalah nama dokumen dan *value* adalah isi dokumen.

*Reducer*: Fungsi *reduce* menjumlahkan semua jumlah yang dikeluarkan untuk kata tertentu.

*Mapper Class*: *Wordcount Mapper Class* dibuat dengan meng-*extend Mapper Class* dan fungsi *map* diimplementasikan dengan meng-*override method* pada *Mapper Class*. Fungsi *mapper* mengambil *key-value* sebagai *input* dan *output key-value* sebagai *output*.

*Reducer Class*: *Wordcount Reducer Class* dibuat dengan meng-*extend org.apache.hadoop.mapreduce.Reducer class* dan *reduce method* diimplementasikan dengan meng-*override reduce method* dari *Reducer Class*. Fungsi *reduce* mengumpulkan semua *key-value* (word,1) digenerasi dengan beberapa fungsi *map*.

Kode Akhir:

```
import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class WordCount {

    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "word count");
        job.setJarByClass(WordCount.class);
        job.setMapperClass(WordCountMapper.class);
        job.setCombinerClass(WordCountReducer.class);
        job.setReducerClass(WordCountReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }

    public static class WordCountReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
        private IntWritable result = new IntWritable();

        public void reduce(Text key, Iterable<IntWritable> values, Context context)
            throws IOException, InterruptedException {
            int sum = 0;
            for (IntWritable val : values) {
                sum += val.get();
            }
            result.set(sum);
            context.write(key, result);
        }
    }

    public static class WordCountMapper extends Mapper<Object, Text, Text, IntWritable> {
        private final static IntWritable one = new IntWritable(1);
        private Text word = new Text();

        public void map(Object key, Text value, Context context) throws IOException,
            InterruptedException {
            StringTokenizer itr = new StringTokenizer(value.toString());
            while (itr.hasMoreTokens()) {
                word.set(itr.nextToken());
                context.write(word, one);
            }
        }
    }

    public static class WordCountReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
        private IntWritable result = new IntWritable();

        public void reduce(Text key, Iterable<IntWritable> values, Context context)
            throws IOException, InterruptedException {
            int sum = 0;
            for (IntWritable val : values) {
                sum += val.get();
            }
            result.set(sum);
            context.write(key, result);
        }
    }
}
```

Gambar 5. Kode Akhir

### 3.2 Kebutuhan Hardware

Tabel 1. Spesifikasi Komputer

Hardware	PC 1	PC 2	PC 3
CPU	Intel Core i7-7700	Intel Core i7-7700	Intel Core i7-7700
RAM	8.00 GB	8.00 GB	8.00 GB
Sistem Operasi	Ubuntu 18.04	Ubuntu 18.04	Ubuntu 18.04

## 4. IMPLEMENTASI DAN PENGUJIAN

### 4.1 Instalasi dan Konfigurasi

Instalasi dan konfigurasi konfigurasi *Hadoop* terdapat dalam halaman lampiran.

### 4.2 Menjalankan WordCounter

1. Membuat direktori didalam file system dengan perintah:  
\$ `hadoop fs -mkdir -p /tes1/hduser/input`
2. Masukkan data yang akan diolah dengan perintah:  
\$ `hadoop fs -put data3.csv /tes1/hduser/input`
3. Mengecek data dengan perintah:  
\$ `hadoop fs -ls /tes1/hduser/input/`

Apabila data telah tersedia di dalam *file system* akan muncul keterangan seperti berikut:

```
Found 1 items -rw-r--r-- 1 hduser supergroup
1001914352      2019-08-19   13:44
/tes1/hduser/input/data3.csv
```

4. Menjalankan program *WordCount*  
\$ `cd /usr/local/hadoop`

```
$ hadoop jar wordcount.jar wordcount
/tes1/hduser/input /tes1/hduser/output
```

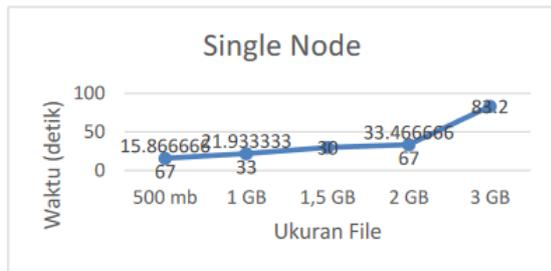
### 4.3 Pengujian Skenario Pertama

Skenario pertama bertujuan untuk mengetahui pengaruh *physical machine* sebagai *node* terhadap kecepatan *MapReduce* pada *Hadoop*. Pada skenario pertama, ukuran file yang digunakan pada saat *single node* yaitu 500 MB, 1 GB, 1.5 GB, 2 GB. Percobaan pada skenario pertama dilakukan sebanyak 15 kali percobaan.

1. Hasil Pengujian  
Hasil kecepatan rata-rata *MapReduce* pada skenario pertama yang dijalankan secara *single node* menggunakan *physical machine*.

Tabel 2. Pengujian Pertama

Ukuran File	Rata-rata Waktu (detik)
500 mb	15,8667
1 GB	21,9333
1,5 GB	30
2 GB	33,4667
3 GB	83,2



Gambar 6. Grafik Pengujian Pertama

2. Analisis

Berdasarkan hasil pengujian yang dilakukan pada skenario pertama dapat disimpulkan bahwa semakin besar ukuran file yang diuji maka kecepatan *MapReduce* akan semakin menurun.

4.4 Pengujian Skenario Kedua

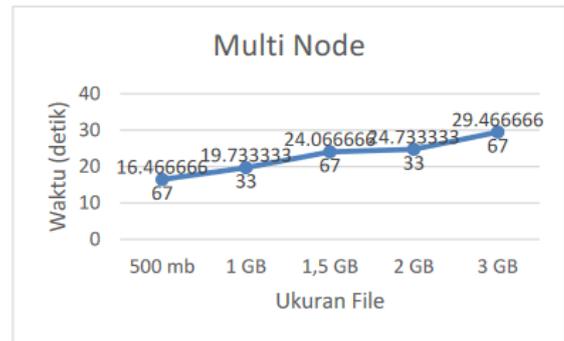
Skenario Kedua bertujuan untuk mengetahui pengaruh *physical machine* sebagai *node* terhadap kecepatan *MapReduce* pada *Hadoop*. Pada skenario kedua, ukuran *file* yang digunakan pada saat *multinode* yaitu 500 MB, 1 GB, 1.5 GB, 2 GB dan 3 GB. Percobaan pada skenario pertama dilakukan sebanyak 15 kali percobaan.

1. Hasil Pengujian

Hasil kecepatan rata-rata *MapReduce* pada skenario kedua yang dijalankan secara *multimode* menggunakan *physical machine*.

Tabel 3. Pengujian Kedua

Ukuran File	Rata-rata Waktu (detik)
500 mb	16,4667
1 GB	19,7333
1,5 GB	24,0667
2 GB	24,7333
3 GB	29,4667

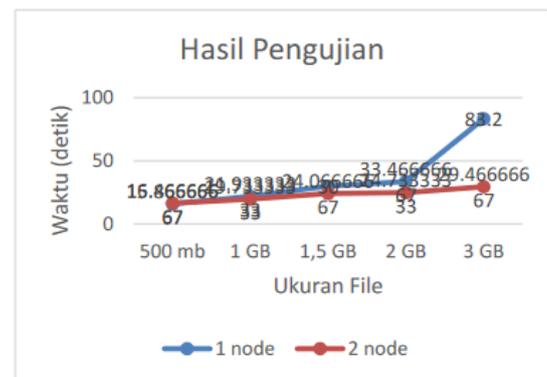


Gambar 7. Grafik Pengujian Kedua

2. Analisis

Berdasarkan hasil pengujian yang dilakukan pada skenario kedua dapat disimpulkan bahwa jumlah *physical machine* yang digunakan sebagai *node* dapat mempengaruhi kecepatan proses *MapReduce* pada *Hadoop*. Penambahan jumlah *physical machine* dengan spesifikasi yang sama sesuai perancangan dapat mempercepat kecepatan rata-rata *MapReduce* pada setiap file yang diuji. Hal ini terjadi karena setiap *node* akan mengambil *resource* dari komputer yang berbeda dimana setiap *node* akan mengambil *resource* dari masing-masing komputer, sehingga dengan menambah jumlah *node* akan memperringan proses kerja dari *cluster* yang digunakan.

4.5 Grafik Hasil Pengujian



Gambar 8. Grafik Hasil Pengujian

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

- Algoritma Word Count* yang diterapkan pada data set GDELT dapat berjalan pada *cluster Hadoop*.
- Kecepatan *Algoritma WordCount* pada *MapReduce* yang diterapkan untuk *dataset* GDELT pada *Hadoop* berpengaruh apabila *node* yang digunakan ditambah, dimana dalam penelitian menggunakan sebanyak 2 *node physical machine*.
- Hadoop* dapat mengolah data yang memiliki ukuran besar dan banyak karena *Hadoop* mengolah data

secara terdistribusi. Kecepatan *Hadoop* dapat diatur dengan menambahkan *node* dan juga pengaturan lainnya seperti halnya *blocksize*.

## 5.2 Saran

- a. Disarankan untuk menganalisa kecepatan *MapReduce* dengan menggunakan algoritma yang berbeda dan lebih kompleks seperti algoritma *Agglomerative* atau algoritma *K-Means Clustering*.
- b. Disarankan bagi penelitian selanjutnya untuk meneliti pengaruh *blocksize* pada kecepatan algoritma *MapReduce WordCount*.

## DAFTAR PUSTAKA

- [1] I. N. Aziz, Fitriyani, K. R. S. Wiharja, "Analisis Pengolahan *Text File* pada *Hadoop Cluster* dengan memperhatikan Kapasitas *Random*," Bandung: Universitas Telkom, 2015.
- [2] C. Lam, "*Hadoop in Action*," Stamford: Manning Publications Co., 2011.
- [3] M. Industri, "Definisi *Cloud Computing*," Meruvian.org Cloud Computing, 2013.
- [4] "Apache *Hadoop*," Hadoop Apache, [Online]. Available: <http://hadoop.apache.org/> [diakses 08 Juni 2018]
- [5] K. Letaru, "*The GDELT Project*," GDELTProject, [Online]. Available: [www.gdeltproject.org](http://www.gdeltproject.org) [diakses 08 Juni 2018]
- [6] J. D. Ghenawat, "*MapReduce: Simplified Data Processing on Large Clusters*," 2004.
- [7] M. G. Noll, "*Running Hadoop On Ubuntu Linux (Multi-Node Cluster)*," 2004-2019, [Online]. Available: <https://www.michael-noll.com/tutorials/running-hadoop-on-ubuntu-linux-multi-node-cluster/> [diakses Juli 2019]



## UJI PERFORMA DAN PERBANDINGAN RDBMS MYSQL DAN HIVE-HADOOP

Nurul Azizah<sup>1</sup>, Henry Saptono<sup>2</sup>

<sup>1,2</sup>Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri  
Jakarta Selatan, DKI Jakarta, Indonesia 12640  
[nurul.azizah195@gmail.com](mailto:nurul.azizah195@gmail.com), [henry@nurulfikri.co.id](mailto:henry@nurulfikri.co.id)

### Abstract

*The amount of data that is proliferating today requires fast storage as well, and this is because they need for data is also critical and accessing it also requires fast time. Therefore we need to know a system that supports big data processing or big data promptly. Fast enough. We need to know how the performance of a relational database system and a system that processes data is not relational. The existence of MySQL and Hive-Hadoop can make us find which system we should use to store data and obtain the desired data quickly. This analysis and comparison research wants to know how the performance of the two systems, namely between MySQL and Hive-Hadoop, can access structured data. This research will be conducted by using experimental methods and manipulating the experimental environment with computer virtualization. The conclusion that Hive-Hadoop can be faster to process extensive data compared to MySQL because Hive-Hadoop uses HDFS, where files will distribute in blocks of files to connected computers.*

**Keywords:** *Hive-Hadoop, MySQL, Big Data, HDFS*

### Abstrak

Banyaknya data yang berkembang pesat saat ini membutuhkan penyimpanan yang cepat pula, hal ini karena kebutuhan akan data juga sangat penting dan dalam mengakses pun membutuhkan waktu yang cepat pula, oleh karena itu kita perlu mengetahui sistem yang mendukung pemrosesan data besar atau *big data* dengan waktu yang cukup cepat. Kita perlu tau bagaimana performa dari sebuah sistem *database* yang relasional dan sistem yang memproses data secara tidak relasional. Adanya MySQL dan *Hive-Hadoop* bisa membuat kita menemukan sistem mana yang harus kita pakai untuk selanjutnya dalam menyimpan data dan memperoleh data yang diinginkan secara cepat. Penelitian analisa dan perbandingan ini ingin mengetahui bagaimana performa dari kedua sistem yakni antara MySQL dan *Hive-Hadoop* yang keduanya bisa mengakses data secara terstruktur. Penelitian ini akan dilakukan dengan metode percobaan dan memanipulasi lingkungan percobaan dengan virtualisasi komputer. Dengan kesimpulan bahwa *Hive-Hadoop* bisa lebih cepat untuk memproses data besar dibandingkan dengan MySQL, karena *Hive-Hadoop* menggunakan HDFS dimana file akan didistribusikan secara blok-blok file ke dalam komputer yang terhubung.

**Kata kunci:** *Hive-Hadoop, MySQL, Big Data, HDFS*

### 1. PENDAHULUAN

Sejak beberapa tahun terakhir ini, perkembangan IT sedang berkembang dengan sangat cepat dan pesat. Dengan perkembangan tersebut, tidak dapat dipungkiri bahwa perkembangan banyaknya data juga sangat berpengaruh dimulai dari memproses data yang besardi suatu kegiatan teknologi. Banyaknya data yang semakin besar dan semakin sulit untuk ditampung pada sistem *database* manapun sehingga sulit untuk dianalisadan diproses dengan cepat. Oleh karena itu, data yang semakin banyak dan besar ini mulai dikenal dengan nama *big data*. Dengan *big data* ini,

proses pencarian, penyimpanan data, dan analisa pada data akan membutuhkan waktu yang cukup lama.

Awalnya banyak sekali organisasi atau perusahaan yang menggunakan RDBMS untuk memproses atau menyimpan data-data yang dimiliki. Namun, seiring perkembangan teknologi informasi yang bertujuan untuk mempermudah kinerja dalam pencarian, penyimpanan data dan lainnya tidak membutuhkan waktu lama lagi untuk melakukan tugasnya. Ada teknologi baru bernama *Hadoop* yang menggunakan teknologi HDFS dan *MapReduce*. Adanya teknologi tersebut maka *Hadoop* memproses data secara

terdistribusi dalam beberapa komputer bahkan ribuan, dengan HDFS dan *MapReduce* maka kinerja *Hadoop* sangat terbantu. Jika makin banyak data yang tersimpan di satu server, maka penyimpanan *storage server* akan semakin penuh. Bila *Hadoop* hanya punya satu *server*, maka kemungkinan *server* tersebut bisa mengalami masalah dan mengakibatkan *storage* tidak dapat diakses kembali.

Kedua teknologi ini sama-sama untuk bisa untuk memproses data namun berbeda cakupan. RDBMS digunakan untuk data yang terstruktur sedangkan *Hadoop* bisa digunakan untuk data yang tidak terstruktur. Dan juga *Hadoop* bukanlah sebuah *database* melainkan sebuah framework yang memungkinkan untuk memproses data berukuran besar. Oleh karena itu, untuk memudahkan dalam pengetahuan dan penggunaan sistem ini, maka penulis membuat analisa performa untuk keduanya sehingga tahu bagaimana performa diantara keduanya dalam memproses data yang besar. Serta tepat dalam menggunakan *software* dan *hardware* yang dipilih dan digunakan.

### 1.1 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan diatas, maka ada dua masalah utama dalam pembuatan penelitian ini:

1. Apa sajakah indikator atau parameter yang digunakan untuk menganalisa performa dari teknologi MySQL dan *Hadoop*?
2. Bagaimanakah performa dari kedua teknologi yang dianalisa?

### 1.2 Batasan Masalah

Adapun untuk batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Analisa performa dari kedua teknologi yang diteliti tidak termasuk keamanan.
2. Penelitian ini dilakukan dalam lingkup percobaan saja bukan di sebuah instansi atau perusahaan.
3. Lingkungan percobaan pada virtualisasi.

### 1.3 Tujuan dan Manfaat

Secara umum tujuan serta manfaat dari penelitian ini adalah untuk mengetahui bagaimana performa dari *Hadoop* dan MySQL dalam memproses data dalam jumlah besar. Secara khusus, penelitian ini akan menjadi ilmu pengetahuan tambahan bagi peneliti terkait *Hadoop* dan MySQL dan sebagai karya tulis yang bisa digunakan pada lingkungan kampus atau rujukan dalam mengimplementasi.

## 2. LANDASAN TEORI

### 2.1 Database

*Database* adalah kumpulan data (elementer) yang secara logika berkaitan dalam mempresentasikan fenomena/fakta

secara terstruktur dalam domain tertentu untuk mendukung aplikasi dalam sistem tertentu [1].

*Database* menurut Connolly adalah kumpulan data yang secara logis terkait satu sama lain dan digunakan bersama oleh banyak pemakai, serta dirancang untuk memenuhi kebutuhan informasi banyak pemakai dalam suatu organisasi. Artinya *database* merupakan penyimpanan data yang tunggal dan besar yang dapat digunakan secara simultan oleh banyak bagian departemen dan user [2].

### 2.2 RDBMS

RDBMS (*Relational Database Management System*) merupakan sekumpulan program komputer yang mengendalikan pembuatan, perawatan, dan penggunaan sebuah organisasi *database* dengan menggunakan komputer sebagai *platform*. DBMS memungkinkan pengguna program aplikasi yang berbeda untuk dengan mudah mengakses *database* yang sama. DBMS menggunakan berbagai macam model *database*, seperti *network model* atau *relational model* [3].

### 2.3 MySQL

MySQL adalah perangkat lunak sistem manajemen basis data yang digunakan untuk menyimpan dan mengolah data [2]. MySQL adalah sebuah server *database SQL multiuser* dan *multi-threaded*. *Database* ini dibuat untuk keperluan sistem *database* yang cukup baik, MySQL untuk sistem operasi Unix bersifat *freeware*, dan terdapat versi *shareware* untuk sistem operasi Windows [3].

### 2.4 Big Data

*Big data* adalah data yang melebihi proses kapasitas dari konversi sistem *database* yang ada. Data terlalu besar dan terlalu cepat atau tidak sesuai dengan struktur arsitektur *database* yang ada. Untuk mendapatkan nilai dari data, maka harus memilih jalan alternatif untuk memprosesnya [4].

### 2.5 HADOOP

*Hadoop* adalah *platform* perangkat lunak *open source* yang dikelola oleh *Apache Software Foundation*. Ini adalah *platform* yang paling dikenal untuk menyimpan dan mengelola sejumlah besar data secara efisien dan hemat biaya. *Hadoop* merupakan *software open-source* yang dibuat menggunakan Java untuk penyimpanan terdistribusi dan pemrosesan terdistribusi untuk data yang sangat besar yang berada dibawah *Apache Software Foundation*.

Inti dari *Hadoop* ada 2 yakni HDFS dan *MapReduce*. HDFS merupakan file terdistribusi yang tinggi toleransi kegagalannya, maksud dari file terdistribusi sendiri adalah data yang tersimpan tidak berada hanya di satu *hardisk* tetapi disimpan dalam beberapa *hardisk* komputer sedangkan *MapReduce* adalah model pemrograman standar

fungsional dari *Hadoop*, yang digunakan untuk memproses data dalam jumlah besar.

## 2.6 HIVE

*Hive* adalah perangkat lunak *data warehouse* berbasis *open source* yang disediakan oleh Apache yang digunakan untuk Query dan mengelola kumpulan data yang didistribusikan secara SQL. *Hive* membantu untuk membuat konsep data dengan cara terstruktur dan menyediakan bahasa *query* seperti SQL yang dikenal sebagai *HiveQL* [5]. *Hive* memiliki *metastore* yang mana sebagai tempat menyimpan data. layanan *metastore* berjalan di JVM yang sama dengan layanan *Hive* yang menggunakan *database* tertanam yakni *derby* yang disimpan di sistem file lokal dalam mode ini.

## 3. METODE PENELITIAN

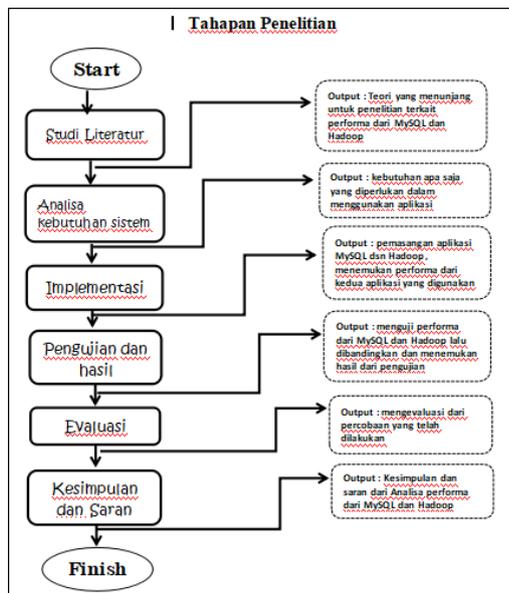
### 3.1 Jenis Penelitian

Jenis penelitian yang digunakan pada penelitian ini adalah jenis **penelitian percobaan** yang berupaya mengisolasi serta kontrol di masing-masing situasi yang sesuai dengan situasi yang hendak diteliti lalu mengamati pada efek maupun pengaruh saat *conditional sentence* memanipulasi kondisi tersebut.

Peneliti akan melakukan manipulasi atau eksperimen terhadap MySQL dan *Hadoop* sehingga peneliti mengetahui performa keduanya.

### 3.2 Tahapan penelitian

Berikut adalah alur tahapan penelitian yang akan dilakukan:



Gambar 1. Tahapan Penelitian

Tahapan penelitian yang pertama adalah Studi Literatur mencari teori-teori yang mendukung untuk penelitian, kedua adalah Analisa kebutuhan sistem yakni mencari tahu kebutuhan dari sistem yang akan coba diteliti dan rancangan pengujian sistem yang akan diuji, ketiga adalah

Implementasi yakni melakukan pemasangan aplikasi yang akan digunakan untuk penelitian, keempat adalah pengujian dan hasil dimana pengujian dilakukan dan akan didapati hasil dari rancangan pengujian yang telah dibuat untuk penelitian. Kelima adalah Evaluasi dari percobaan yang sudah dilakukan dalam penelitian, dan yang terakhir adalah kesimpulan dari penelitian yang sudah dilakukan dan saran untuk para peneliti selanjutnya yang akan meneliti sistem yang sama atau sejenisnya.

### 3.3 Rancangan Penelitian

#### a. Metode Pengumpulan Data

Digunakan untuk memperoleh informasi dan data sebagai acuan rancangan untuk menyelesaikan masalah yang peneliti ambil. Dalam penelitian ini terdapat beberapa metode yang dilakukan dalam pengumpulan data, yaitu:

##### 1. Studi Pustaka

Langkah pertama yang dilakukan dalam pengumpulan data adalah studi pustaka. Studi pustaka ini akan mengarahkan peneliti untuk mencari data serta informasi melalui dokumen-dokumen berupa dokumen tertulis, foto-foto, gambar maupun dokumen elektronik yang bisa mendukung dalam proses penulisan dan penelitian.

##### 2. Eksperimen

Melakukan eksperimen atau percobaan yang memiliki tujuan untuk menganalisis faktor-faktor yang merupakan performa dari MySQL dan *Hadoop* yang akan digunakan untuk bahan laporan dalam penulisan tugas akhir ini. Melakukan eksperimen atau percobaan yang memiliki tujuan untuk menganalisis faktor-faktor yang merupakan performa dari MySQL dan *Hadoop* yang akan digunakan untuk bahan laporan dalam penulisan tugas akhir ini.

#### b. Lingkungan Percobaan

Penelitian ini akan dilakukan dalam lingkungan percobaan dengan teknologi virtualisasi.

#### c. Alat dan Bahan

Untuk melakukan penelitian ini, peneliti menggunakan perangkat:

##### 1. Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini adalah:

- Model : Toshiba DT01ACA100 (MS20A810)
- Processor : Intel® Core™ i7-7700 CPU @ 3.60GHz × 8
- RAM : 8GB
- Hardisk : 1,0TB

##### 2. Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian ini adalah:

- Sistem Operasi Xubuntu 16.04 LTS
- MySQL
- Hadoop
- Hive
- Derby
- JDK 1.8
- SSH

#### 4. ANALISA DAN PERANCANGAN

##### 4.1 Analisa Kebutuhan Software

Software yang akan digunakan pada penelitian ini adalah diantaranya :

1. VirtualBox, karena lingkungan percobaannya adalah virtual, maka diperlukan salah satu aplikasi untuk virtualisasi yakni Virtual Box.
2. OS Linux Xubuntu 16.04, merupakan sistem operasi yang akan digunakan untuk penelitian.
3. MySQL Versi 5.7 , *software database* yang akan dijalankan untuk query *big data*.
4. Hadoop Versi 2.7.2, platform berbasis java untuk mendukung aplikasi yang berjalan pada *bigdata*.
5. Hive Versi 1.2.2, perangkat lunak *datawarehouse* yang digunakan untuk query dan mengolah data yang didistribusikan secara SQL dan Hadoop.
6. Derby Versi 10.13.1.1, *open source* sebagai *metastore* yang tertanam Hive.
7. MapReduce, model komputasi berbasis java pada system terdistribusi dalam rangka mendukung aplikasi *Big Data*.
8. JDK 1.8.0\_221, perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode Java ke *bytecode*.
9. SSH, protokol jaringan kriptografi untuk komunikasi data yang aman.

##### 4.2 Analisa Kebutuhan Software

Berdasarkan kebutuhan software yang diperlukan, maka kebutuhan hardware yang digunakan pada penelitian ini adalah sebagai berikut :

- 1) Spesifikasi minimum untuk menggunakan MySQL
  - Processor : Intel Core or Xeon 3GHz (or Dual Core 2GHz) or equal AMD CPU.
  - Core : Single (Dual/Quad Core is recommended)
  - RAM : 4 GB (6 GB recommended)
- 2) Spesifikasi minimum untuk menggunakan Hadoop
  - Processor : Intel i3 or above
  - RAM : 4 GB
  - Free Disk Space : 25 GB
- 3) Spesifikasi fisik yang digunakan
  - Processor : Intel® Core™ i7-7700 CPU

- RAM : 8 GB
- Hardisk : 1,0 TB

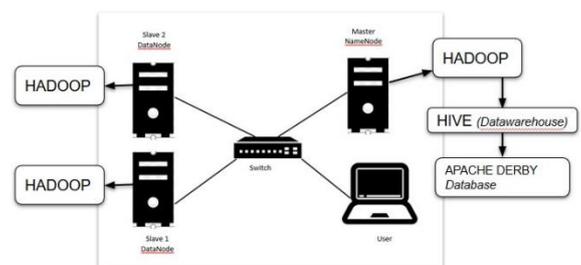
Tabel 1. Spesifikasi Hardware pada Mesin Virtual yang akan digunakan sebagai Penelitian

Node	Processor	RAM	OS
H A D O O P	Master	Intel® Core™ i7-7700 CPU @ 3.60GHz x 8	2 GB Xubuntu 16.04
	Slave1	Intel® Core™ i7-7700 CPU @ 3.60GHz x 8	2 GB Xubuntu 16.04
	Slave2	Intel® Core™ i7-7700 CPU @ 3.60GHz x 8	2 GB Xubuntu 16.04
MySQL Server	Intel® Core™ i7-7700 CPU @ 3.60GHz x 8	2 GB	Xubuntu 16.04

##### 4.3 Rancangan Sistem

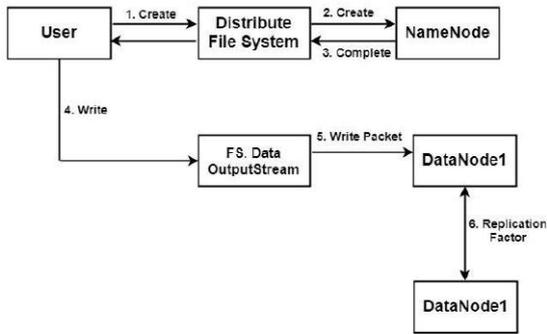
###### 1. Rancangan Arsitektur Sistem Hadoop

Rancangan arsitektur yang akan digunakan untuk sistem Hadoop adalah peneliti akan menggunakan 3 server yang diantaranya 1 master dan 2 slave yang mana pada masing-masing server akan dipasang sebuah Hadoop dan di dalam server master akan dipasang juga sebuah Hive dan Derby sebagai pendukung Hadoop yang akan digunakan untuk memproses data secara SQL.



Gambar 2. Rancangan Arsitektur dari Sistem Hadoop

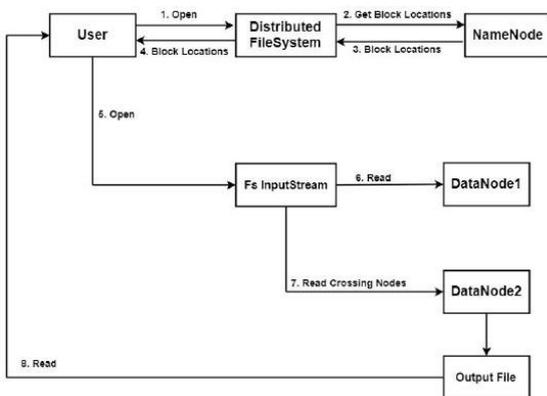
Untuk proses write yang terjadi pada Hadoop akan dijalankan di HDFS dengan skema berikut:



Gambar 3. Proses Write yang Terjadi di HDFS

Pada gambar 3, bisa dijelaskan bahwa jika ada perintah menyimpan oleh *user*, maka komputer *user* akan berkomunikasi dengan *namenode* untuk memberitahu bahwa ada file yang akan disimpan di HDFS serta menanyakan lokasi *datanode* yang bisa diakses untuk menyimpan data yang akan disimpan, setelah mendapat daftar nama dan alamat *datanode* yang tersedia maka komputer *user* akan langsung mentransfer data ke *datanode* yang ada. Data yang ditransfer sudah otomatis dibelah menjadi *block* atau kepingan data yang tersimpan dalam *datanode*. Setelah kepingan data diterima maka *datanode* akan mengirimkan laporan ke *namenode* bahwa data telah diterima dan disimpan.

Sedangkan untuk proses *read* pada HDFS adalah sebagai berikut:

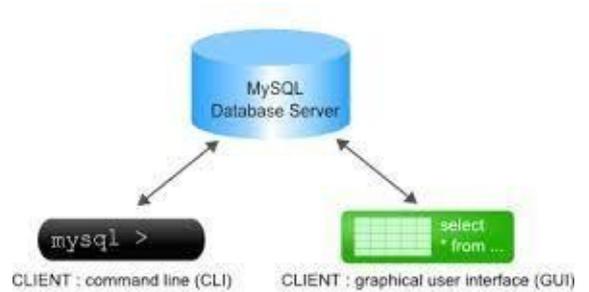


Gambar 4. Proses Read yang terjadi di HDFS

Pada gambar 4, ditunjukkan alur dari proses *read* data atau file di HDFS. Saat *user* memerintahkan eksekusi perintah *read* di komputer *user*, maka akan ada komunikasi dengan *namenode* untuk menanyakan nama dan alamat *datanode* yang harus diakses untuk mendapatkan data yang diinginkan. Maka setelah *user* mendapatkan nama dan alamat *datanode* akan secara langsung mengakses *datanode* yang bersangkutan dan kemudian data akan ditampilkan sesuai perintah dari *user*.

2. Rancangan Arsitektur RDBMS MySQL

Peneliti akan melakukan perancangan arsitektur MySQL dengan menggunakan arsitektur MySQL *Server* yang biasa digunakan oleh para pengguna pada umumnya, yakni peneliti akan menggunakan satu *client* dan satu *server*. Dimana satu *server* ini akan menjadi *node* utama dimana data disimpan dan *client* yang melakukan *read* dan *write*. Seperti yang terlihat pada gambar berikut:



Gambar 5. Rancangan Arsitektur MySQL

Untuk alur kerja dari MySQL adalah akan terjadinya dialog antara *client* dan *server* melalui *daemon* pada MySQL, dimana *daemon* ini adalah sebutan untuk program yang berjalan pada *background* (berjalan tanpa campur tangan manusia). *Daemon* tidak akan aktif bekerja, kecuali jika ada permintaan kepada *daemon* untuk melakukan pekerjaan. Setelah *daemon* MySQL dijalankan, *daemon* tersebut akan memberikan proses dialog dengan *client*. *Client* kemudian diminta untuk melakukan autentifikasi untuk mengolah *database* MySQL.

4.4 Rancangan Pengujian

Pengujian pada *Hadoop* akan menggunakan *Hive* yang merupakan *data warehouse* yang ada di *Hadoop* itu sendiri dan *Derby* sebagai *database* eksternalnya dan untuk MySQL akan diuji secara langsung di MySQL, dimana dua perangkat tersebut memungkinkan untuk melakukan *create* dan *select*, sehingga bisa dibandingkan kinerja keduanya berdasarkan percobaan pada 2 perangkat tersebut. *Query* yang akan digunakan adalah *query* “*select*” dan “*insert*” untuk diujikan pada kedua perangkat.

1. Pengujian *Hadoop*

Perangkat yang akan digunakan dalam pengujian *Hadoop* ini yaitu *Hive* dengan *database* eksternalnya yakni *Derby*. Berikut akan dijabarkan langkah-langkah yang akan dilakukan untuk proses pengukuran pada *Hadoop*.

a. Pengujian Lama Waktu Load Data

Untuk melakukan pengujian lama waktu *load data* adalah sebagai berikut:

1. Memastikan topologi sistem yang diukur.
2. Menentukan *data warehouse* dan *database* yang akan digunakan yaitu *Hive* dan *Derby*.
3. Menentukan data yang akan digunakan untuk

- pengujian.
4. Menentukan *query* yang akan digunakan yaitu *insert*.
  5. Menentukan berapa kali pengujian akan dilakukan.
  6. Menjalankan *Hive* yang telah terinstal di komputer untuk menguji kinerja *Hadoop*.
  7. Melihat *output* yang dihasilkan dari alat ukur yaitu waktu *load data* pada *Hadoop* yang telah ditentukan datanya.
  8. Ulangi langkah-langkah di atas untuk penambahan data 2GB-5GB. Dan selanjutnya dicatat waktu pengukuran saat pengujian.
- b. Pengujian Lama Waktu *Select Data*  
 Untuk melakukan pengujian lama waktu *select data* adalah sebagai berikut:
1. Memastikan topologi sistem yang diukur.
  2. Menentukan *data warehouse* dan *database* yang akan digunakan yaitu *Hive* dan *Derby*.
  3. Menentukan data yang akan digunakan untuk pengujian.
  4. Menentukan *query* yang akan digunakan yaitu *select*.
  5. Menentukan berapa kali pengujian akan dilakukan.
  6. Menjalankan *Hive* yang telah terinstal di komputer untuk menguji kinerja *Hadoop*.
  7. Melihat *output* yang dihasilkan dari alat ukur yaitu waktu *query data* pada *Hadoop* yang telah ditentukan datanya.
  8. Ulangi langkah-langkah diatas untuk penambahan data 2GB-5GB. Dan selanjutnya dicatat waktu pengukuran saat pengujian.

**Tabel 2.** Contoh Tabel Hasil Eksekusi *Load* dan *Select Data*

Ukuran data	1	2	3	4	5
	GB	GB	GB	GB	GB
Pengujian ke	(s)	(s)	(s)	(s)	(s)
Pengujian 1					
Pengujian 2					
Pengujian 3					

2. Pengujian MySQL

Untuk pengujian MySQL rancangan pengujian yang akan digunakan dalam proses pengukuran adalah sebagai berikut:

- a. Pengujian Lama Waktu *Load Data*
1. Memastikan topologi sistem yang akan diukur.
  2. Menentukan *database* yang akan digunakan yaitu MySQL.
  3. Menentukan macam-macam *query* yang akan digunakan, diantaranya *Insert*.
  4. Melakukan pengujian beberapa kali dengan penambahan data untuk menemukan

- perbedaan kecepatan waktu *load data* seiring banyaknya data yang di *insert*.
5. Melihat *output* yang dihasilkan dari alat ukur yaitu kecepatan *load data*.

b. Pengujian Lama Waktu *Select Data*

1. Memastikan topologi sisten yang akan diukur.
2. Menentukan *database* yang akan digunakan yaitu MySQL.
3. Menentukan macam-macam *query* yang akan digunakan, diantaranya *select*.
4. Melakukan pengujian beberapa kali dengan penambahan data untuk menemukan perbedaan kecepatan waktu *query* eksekusi seiring banyaknya data yang di *select*.
5. Melihat *output* yang dihasilkan dari alat ukur yaitu kecepatan *query eksekusi*.
6. Setelah itu dibandingkan hasil masing-masing percobaan untuk dianalisa.

**Tabel 3.** Contoh Tabel Hasil *Load* dan *Select Data* di MySQL

Ukuran data	1	2	3	4	5
	GB	GB	GB	GB	GB
Pengujian ke	(s)	(s)	(s)	(s)	(s)
Pengujian 1					
Pengujian 2					
Pengujian 3					

5. IMPLEMENTASI DAN ANALISA

5.1 Data Penelitian

Data yang akan digunakan dalam penelitian ini adalah data dengan format CSV yang berukuran 1GB sampai 5GB. Data didapat dari *website* <http://data.gdeltproject.org/events/index.html>. Peneliti menggunakan data yang di dapat dari *website* tersebut dengan ukuran 1GB lalu *dicopy-paste* agar bisa mendapatkan data sampai 5GB. Data ini merupakan data yang terstruktur dalam 1 tabel sehingga menggunakan format CSV agar memudahkan dimasukkan ke dalam format *database* terstruktur. Untuk memudahkan memasukkan data ke dalam tabel sistem SQL, maka data dibuat dengan format delimiter (pembatas) koma, sehingga perintah atau format yang dijalankan bisa disesuaikan dengan format SQL.

5.2 Pengujian

Pada pengujian ini, peneliti sudah memastikan bahwa parameter yang akan digunakan sebagai bahan uji dalam penelitian ini adalah diantaranya

- a. Besarnya data yakni antara 1GB sampai dengan 5GB.
- b. *Query* eksekusi pada *Hive* dan MySQL yakni *Load data* dan *select data*.

Paramater yang digunakan diatas adalah uji coba yang dilakukan oleh peneliti sehingga bisa mengetahui bagaimanakah performa dari MySQL dan Hadoop yang digunakan untuk bahan penelitian.

1. Pengujian Hadoop

Langkah-langkah yang dilakukan untuk menguji Hadoop:

- 1) Setelah memastikan Hadoop, Hive dan Derby terpasang (ada dalam lampiran). Masuk ke sistem hive  
**\$ bin/hive**
- 2) Cek database  
**hive>show databases;**
- 3) Membuat database dan menggunakan database tersebut  
**hive>create database tugas;**  
**hive>use tugas;**
- 4) Create table data  
**hive>create table data1gb(kode1 int, kode2 int, kode3 char(20), kode4 char(20));**
- 5) Melakukan load data ke dalam tabel  
**hive>load data local inpath '/home/document/data1gb.csv' overwrite into table data1gb;**
- 6) Mencatat waktu yang diperoleh dari load data
- 7) Melakukan query select dan mencatat waktu hasil query  
**hive> select \* from data1gb;**
- 8) Melakukan langkah 5-7 untuk menguji data 1GB sampai 3 kali sebagai perbandingan.
- 9) Melakukan langkah 4 sampai 8 untuk data 2GB sampai 5GB.

Tabel 5. Pengujian Load Data pada Hadoop

Ukuran data	1 GB	2 GB	3 GB	4 GB	5 GB
Pengujian ke	(s)	(s)	(s)	(s)	(s)
Pengujian 1	60,642	94,478	107,629	152,722	329,638
Pengujian 2	61,945	106,198	107,334	169,987	329,946
Pengujian 3	62,034	108,041	108,424	171,264	330,103
Rata-rata	61,54	102,9	107,79	257,656	385,229

Hasil keseluruhan pengujian dari select data dapat dilihat dalam tabel berikut:

Tabel 6. Pengujian Select Data pada Hadoop

Ukuran data	1 GB	2 GB	3 GB	4 GB	5 GB
Pengujian ke	(s)	(s)	(s)	(s)	(s)
Pengujian 1	0 sec, 8map	0 sec, 18 map	0 sec, 26 map	0 sec, 35 map	0 sec, 43 map
Pengujian 2	0 sec, 8 map	0 sec, 17 map	0 sec, 28 map	0 sec, 34 map	0 sec, 43 map
Pengujian 3	0 sec, 8map	0 sec, 17 map	0 sec, 26 map	0 sec, 34 map	0 sec, 44 map
Rata-rata	0 sec, 8 map	0 sec, 17 map	0 sec, 26 map	0 sec, 34 map	0 sec, 43 map

2. Pengujian MySQL

Langkah-langkah pengujian dalam Load data yang dilakukan pada MySQL:

- 1) Pastikan masuk ke dalam MySQL untuk memulaimelakukan pengujian  
**\$ mysql -u root -p**
- 2) Mengecek dan membuat database  
**mysql > show databases;**  
**mysql > create database tugas;**  
*(keterangan: nama database menggunakan nama sesuai yang diinginkan)*
- 3) Menggunakan database yang sudah dibuat  
**mysql > use tugas;**
- 4) Membuat tabel 1GB  
**mysql > create table data1gb(kode1 int, kode2 int, kode3 int, kode4 int);**
- 5) Melakukan load data ke dalam tabel  
**mysql > load data local infile '/home/document/data1gb.csv' into table data1gb fields terminated by ',' lines terminated by '\n';**  
*[keterangan : load data local infile(pada file penyimpanan local), „home/document/data1gb.csv“ (lokasi datanya), fields terminated by „,“ , lines terminated by „\n“ (baris baru menggunakan enter)]*
- 6) Mencatat waktu yang diperoleh dari load data
- 7) Melakukan query select dan mencatat waktu hasil query  
**mysql> select \* from data1;**
- 8) Melakukan langkah 5-7 untuk menguji data 1GB sampai 3 kali sebagai perbandingan.
- 9) Melakukan langkah 4 sampai 8 untuk data 2GB sampai 5GB
- 10) Menghitung rata-rata waktu yag didapatkan

Hasil dari pengujian secara keseluruhan pada saat load data dapat dilihat dalam tabel berikut:

Tabel 7. Pengujian *Load Data* pada MySQL

Ukuran data	1	2	3	4	5
Pengujian ke	GB	GB	GB	GB	GB
	(s)	(s)	(s)	(s)	(s)
Pengujian 1	2 hours 29 min 19,09 sec	2 hours 29 min 32,15 sec	2 hours 30 min 43,09 sec	2 hours 32 min 47,59 sec	2 hours 33 min 13,05 sec
Pengujian 2	2 hours 29 min 20,74 sec	2 hours 29 min 32,40 sec	2 hours 30 min 44,19 sec	2 hours 30 min 48,34 sec	2 hours 33 min 13,46 sec
Pengujian 3	2 hours 29 min 21,67 sec	2 hours 29 min 32,42 sec	2 hours 30 min 44,27 sec	2 hours 30 min 48,42 sec	2 hours 33 min 14,04 sec
Rata-rata	2 hours 29 min 20,5 sec	2 hours 29 min 32,32 sec	2 hours 30 min 43,85 sec	2 hours 30 min 48,11 sec	2 hours 33 min 13,51 sec

Hasil pengujian secara keseluruhan pada saat *select data* dapat dilihat dalam tabel berikut:

Tabel 8. Hasil *Select Data* di MySQL

Ukuran data	1	2	3	4	5
Pengujian ke	GB	GB	GB	GB	GB
	(s)	(s)	(s)	(s)	(s)
Pengujian 1	36,20 sec	40,15 sec	44,56 sec	48,51 sec	57,32 sec
Pengujian 2	40,13 sec	48,34 sec	62,39 sec	65,27 sec	71,44 sec
Pengujian 3	44,27 sec	62,41 sec	70,12 sec	73,26 sec	75,17 sec
Rata-rata	40,2 sec	50,3 sec	59,02 sec	62,34 sec	67,97 sec

- Analisa Pengujian *Load Data* MySQL vs *Hadoop*  
 Pada pengujian yang telah dilakukan pada *load data* di MySQL dan *Hadoop* ternyata semakin banyaknya data semakin banyak waktu yang diperlukan serta setelah dibandingkan waktu yang diperlukan lebih efektif pada saat *load data* di *Hadoop* dibandingkan dengan di MySQL. Berikut adalah tabel perbandingan waktu *load data* antara MySQL dan *Hadoop*:

Tabel 9. Rata-rata Waktu *Load Data* di MySQL dan *Hadoop*

Ukuran Data (GB)	Load data MySQL(s)	Load data Hadoop(s)
1 GB	2 hours 29 min 20,5 sec	61,54 sec
2 GB	2 hours 29 min 32,32 sec	102,9 sec
3 GB	2 hours 30 min 43,85sec	107,79 sec

Ukuran Data (GB)	Load data MySQL(s)	Load data Hadoop(s)
4 GB	2 hours 30 min 48,11 sec	257,656 sec
5 GB	2 hours 33 min 13,51sec	385,229 sec

Berdasarkan tabel 9, waktu yang diperlukan oleh MySQL lebih lama dibandingkan *Hadoop*, dalam hal ini karena MySQL butuh memproses data secara terstruktur dan dalam satu manajemen. Sedangkan untuk *Hadoop*, dalam memproses data dibantu proses dari HDFS yakni file yang dimasukkan akan didistribusikan ke dalam *node-node* lain yang terhubung satu dengan yang lainnya. Sehingga waktu yang diperlukan untuk *load data* jauh lebih cepat dibandingkan dengan MySQL.

- Analisa Pengujian *Select Data* MySQL vs *Hadoop*  
 Untuk pengujian *select data* di MySQL dan *Hadoop* didapatkan hasil berikut ini:

Tabel 10. Hasil Pengujian *Select Data* di MySQL dan *Hadoop*

Ukuran Data (GB)	Select data MySQL(s)	Select data Hadoop(s)
1 GB	40,2 sec	0 sec, 8 map
2 GB	50,3 sec	0 sec, 17 map
3 GB	59,02 sec	0 sec, 26 map
4 GB	62,34 sec	0 sec, 34 map
5 GB	67,97 sec	0 sec, 43 map

Dari tabel 10, kita bisa simpulkan semakin banyaknya data yang dimasukkan akan semakin lama juga waktu yang dibutuhkan untuk membaca data. MySQL memerlukan waktu untuk membaca satu per satu data yang telah dimasukkan lalu kemudian dipilih untuk dilihat secara keseluruhan. Untuk *Hadoop*, data yang telah dipetakan akan secara cepat didapatkan dan ditampilkan oleh sistem sehingga waktu yang diperlukan sangat cepat dibandingkan dengan MySQL.

## 6. KESIMPULAN DAN SARAN

### 6.1 Kesimpulan

Kesimpulan dari pengujian yang telah dilakukan oleh peneliti terkait performa MySQL dan *Hadoop* adalah sebagai berikut:

- Indikator atau parameter yang digunakan pada penelitian ini adalah besarnya data yakni mulai 1GB sampai 5GB, kriteria data tersebut diujikan pada sistem MySQL dan *Hadoop*, dan parameter lainnya adalah kueri eksekusi yang digunakan yakni *load data* dan *select data* karena pada penelitian ini, peneliti

ingin mengetahui performa dari masing-masing sistem dalam bentuk menulis dan membaca data sehingga kueri yang digunakan adalah *load data dan select data*.

2. Performa yang ditampilkan oleh kedua sistem ini sangat signifikan dilihat dari lamanya waktu. Berdasarkan dari parameter yang digunakan oleh peneliti, bahwa ketika *load data* ke dalam MySQL membutuhkan waktu yang sangat lama jika dibandingkan dengan *Hadoop*. Diantara keduanya memiliki karakteristik yang berbeda-beda sehingga jika dibandingkan berdasarkan waktu memang akan berbeda jauh, karena secara umum *Hadoop* merupakan file sistem untuk mendistribusikan data sedangkan MySQL merupakan *database* untuk penyimpanan data.

## 6.2 Saran

Saran dari peneliti untuk para peneliti yang ingin menguji performa *Hadoop* dan MySQL selanjutnya adalah sebagai berikut:

1. Karena penelitian ini membahas tentang *big data*, maka gunakanlah komputer dengan spesifikasi yang tinggi atau jika ingin melakukan percobaan maka harus dengan RAM minimal 4 GB, *hardisk* menggunakan SSD, *processor* minimal Intel i3.
2. Menggunakan komputer fisik dan bukan komputer virtual untuk uji coba *big data* melalui sistem atau aplikasi apapun, agar hasil yang didapatkan bisa lebih akurat.
3. Pastikan bahwa pada komputer yang akan digunakan hanya ada aplikasi *database* atau yang berkaitan untuk *database*.
4. Hanya gunakan OS yang ringan untuk mempermudah pemrosesan sistem dan data, sebaiknya gunakan OS Linux versi server serta gunakan aplikasi yang terbaru agar masih ada versi yang bisa terkoneksi dengan aplikasi lain.
5. Pastikan hanya menjalankan 1 *service* tanpa ada proses lain yang berjalan ketika *load data* atau *select data*, karena akan sangat mempengaruhi prosesor dalam bekerja.

## DAFTAR PUSTAKA

- [1] S. Minarni, "Sistem Informasi *Inventory* Obat pada Rumah Sakit Umum Daerah (RSUD) Padang," *Jurnal Momentum*, p. 105, 2014.
- [2] F. P. H. H. Susanto, "Perbandingan Unjuk Kerja MySQL *Cluster* Versus MySQL Konvensional," Yogyakarta, 2015.
- [3] W. S. D. S. Ida Bagus Teguh Teja Murti, Artist, "Analisa Perbandingan Kinerja DSMS dan RDBMS pada Studi Kasus Transaksi Online," [Art], Institut Teknologi Sepuluh November.
- [4] A.E. T., "Rancangan Aplikasi *Smart City* berbasis *Mobile* untuk meningkatkan Kualitas Layanan Publik Studi Kasus Pemkot Bogor," *Jurnal Teknik Komputer AMIK BSI*, Vol. 3, p.26, 2016.
- [5] K. S. Raste, "*Big Data Analytics-Hadoop Performance Analysis*," in *In Partial Fulfillment of the Requirements for Degree Master of Science in Computer Science*, San Diego, 2014.



## RANCANG BANGUN SISTEM INFORMASI PENGELOLAAN TALENT FILM BERBASIS APLIKASI WEB

Atikah Permata Sari<sup>1</sup>, Suhendi<sup>2</sup>

<sup>1, 2</sup>Sistem Informasi, Sekolah Tinggi Teknologi Terpadu Nurul Fikri  
Jakarta Selatan, DKI Jakarta, Indonesia 12640  
atikahps38@gmail.com, suhendi@nurulfikri.ac.id

### Abstract

*This research discusses the challenge of designing a talent data management application at Hartini Cinema Management using the CodeIgniter Framework. Hartini Cinema Management is long-standing management in the entertainment field for soap operas and the Indonesian big screen. Management promotes talents at PH Televisi Indonesia through direct meetings to come to the PH office or through social media owned by Hartini Cinema Management. To managing talent data, management is still taking talent data from Instagram accounts owned by talents which require a long time and money because management does not yet have a data management application. For this reason, in the discussion of this final project, the researchers designed a talent data management application from bio, photos to websites connected to the Instagram talent account. On this website, there is cost-efficiency. This data management website does not require a quota (internet fees) because this website is not online but localhost, while data collection on Instagram requires a quota (fee). This website is faster to enter data than Instagram, which requires indefinite loading times due to the network.*

**Keywords:** Data management, website, localhost, Codeigniter Framework

### Abstrak

Penelitian ini membahas tentang perancangan aplikasi pengelolaan data talent pada Hartini Cinema Management menggunakan CodeIgniter Framework, Hartini Cinema Management merupakan salah satu management yang sudah lama di dalam bidang *entertainment* untuk perfilman seperti sinetron dan layar lebar Indonesia. Management mempromosikan para talent pada PH Televisi Indonesia, melalui pertemuan langsung untuk datang ke kantor PH atau melalui sosial media yang dimiliki Hartini Cinema Management ini. Untuk mengelola data talent management masih mengambil data talent dari akun Instagram yang dimiliki talent yang memerlukan waktu yang lama dan biaya, karna Management belum memiliki aplikasi pengelolaan data. Untuk itu dalam pembahasan tugas akhir ini peneliti merancang aplikasi pengelolaan data talent dari biodata, foto, hingga website terhubung pada akun Instagram talent. Di website ini ada efisiensi biaya. Karena di website pengelolaan data ini tidak memerlukan kuota (biaya internet) sebab website ini tidak online tetapi localhost, sedangkan pendataan di instagram sangat memerlukan kuota (biaya). Website ini lebih cepat untuk memasukan pendataan dibandingkan dengan instagram, yang memerlukan waktu loading yang tidak tentu karena jaringan.

**Kata kunci:** Pengelolaan data, website, localhost, CodeIgniter Framework

### 1. PENDAHULUAN

Perkembangan teknologi informasi dalam era globalisasi sangat menjadi perhatian yang serius dari berbagai pihak baik itu badan pemerintahan dan parapengusaha, termasuk dalam dunia *entertainment* di bidang perfilman yang cepat mengikuti perkembangan teknologi untuk pendataan.

Tetapi perkembangan teknologi pendataan masih belum tersimpan efektif dan efisien pada Hartini Cinema Management, karna mereka menyimpan dan mengambil

data para talent mereka hanya di Instagram. Yang bisa saja sewaktu-waktu Instagram mereka terkena *hack* oleh pihak yang tidak bertanggung jawab.

Oleh sebab itu semakin meningkatnya teknologi dan kecepatan arus informasi yang berkompeten dan berkualitas untuk mengelola data sehingga memacu organisasi atau perusahaan membuat suatu website, untuk mempromosikan dan mengelola datanya di suatu website

dan dalam hal ini penulis ingin membuat suatu *website* untuk menyimpan dan mengelola data yang tertata.

Di samping itu, saat ini *internet* juga merupakan jaringan luas manusia dan informasi sehingga memungkinkan *internet* dapat digunakan sebagai media bisnis dalam menjalankan usahanya, yaitu menjual jasa dan produk secara *online*, serta memungkinkan pelanggan potensial, calon pelanggan dan mitra bisnis dapat mengakses informasi yang berkaitan dengan produk-produk dan paket layanan mereka sehingga akhirnya melakukan pembelian terhadap produk-produk serta layanan tersebut.

### 1.1 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas maka perumusan masalah yang muncul adalah bagaimana membuat pendataan yang efektif dan efisien dengan Aplikasi Pengelolaan data *Talent* menggunakan *CodeIgniter Framework* pada *Hartini Cinema Management*?

### 1.2 Tujuan dan Manfaat

Dilihat dari rumusan masalah yang sudah dipaparkan, maka tujuan yang didapat sebagai berikut:

Tujuan dari penelitian tugas akhir ini adalah membuat aplikasi pengelolaan data *talent* yang efektif dan efisien menggunakan *CodeIgniter Framework* pada *Hartini Cinema Management*.

Sedangkan manfaat dari penelitian ini adalah:

Manfaat dari membuat dan membangun pengelolaan data berbasis aplikasi pengelolaan data *talent* menggunakan *CodeIgniter Framework* pada *Hartini Cinema Management* adalah kecepatan dan keakuratan dalam pendataan *talent*

### 1.3 Batasan Masalah

Batasan masalah dalam pembuatan Aplikasi Pengelolaan data *Talent* Menggunakan *CodeIgniter Framework* pada *Hartini Cinema Management* hanya untuk penyimpanan biodata *talent* dari *Instagram* ke *website*.

## 2. KAJIAN LITERATUR

Kajian *literatur* mengenai definisi dan teori yang berkaitan dengan penelitian yang akan diteliti pada masalah yang terjadi di *Hartini Cinema Management* dan keterkaitan dengan peneliti-peneliti terdahulu. Kajian *literatur* ini akan melakukan kajian yang terkait pada pembuatan aplikasi pengelolaan data *talent* pada *Hartini Cinema Management* menggunakan *CodeIgniter Framework*. Berikut adalah teori-teori yang berlandaskan pada perancangan aplikasi pengelolaan data *talent* pada *Hartini Cinema Management* menggunakan *CodeIgniter Framework*.

### 2.1 Pengertian Website

*Website* dapat diartikan sebagai kumpulan halaman yang berisi informasi data digital baik berupa teks, gambar, animasi, suara dan video atau gabungan dari semuanya yang disediakan melalui jalur koneksi internet sehingga dapat diakses dan dilihat oleh semua orang di seluruh dunia. Halaman *website* dibuat menggunakan bahasa standar yaitu HTML. Skrip HTML ini akan diterjemahkan oleh *web browser* sehingga dapat ditampilkan dalam bentuk informasi yang dapat dibaca oleh semua orang. Secara umum, *website* dibagi menjadi 3 jenis, yaitu *website* statis, dinamis, dan interaktif [1].

### 2.2 Text Editor

Text Editor memiliki fungsi yang sama dengan *web authoring* yaitu untuk mengetikkan skrip program, hanya saja tidak bersifat WYSIWYG. Sehingga untuk menampilkan hasilnya memerlukan *web browser*. Namun, karena lebih ringan dan lisensinya yang kebanyakan *open source* banyak juga yang lebih memilih menggunakan text editor dibandingkan aplikasi sekelas *Adobe Dreamweaver*. Yang termasuk dalam aplikasi jenis ini yaitu *Notepad*, *Notepad++*, *Sublime Text*, *Bracket*, dan sebagainya [1].

### 2.3 Web Browser

*Web browser* digunakan untuk menampilkan dan mengetes hasil program. Beberapa skrip CSS3 dan HTML5 ada yang hanya *support* pada *web browser* tertentu dan tidak *support* pada yang lain, sehingga menggunakan lebih dari satu *web browser* akan lebih baik. walaupun untuk kebanyakan *browser* versi baru sudah *support* hampir semua fitur CSS3 dan HTML5 [1].

### 2.4 Web Server

*Web server* merupakan perangkat lunak yang berfungsi untuk menerima permintaan (*request*) melalui protokol HTTP atau HTTPS dari *client* kemudian mengirimkan kembali dalam bentuk halaman-halaman *web* contoh yang termasuk *web server* adalah *Apache*. Dalam penggunaannya, biasanya sudah jadi satu paket dengan *PHP* dan *MySQL* diantaranya *XAMPP* dan *Appserv* [1].

### 2.5 HTML

HTML merupakan singkatan *Hypertext Markup Language* yaitu bahasa standar *web* yang dikelola penggunaannya oleh W3C (*World Wide Web Consortium*) berupa *tag-tag* yang menyusun setiap elemen dari *website*. HTML berperan sebagai penyusun struktur halaman *website* yang menempatkan setiap elemen *website layout* yang diinginkan.

HTML biasanya disimpan dalam sebuah file berekstensi *.html*. Untuk mengetikkan skrip HTML dapat menggunakan text editor seperti *Notepad* sebagai bentuk paling sederhana atau text editor khusus yang dapat mengenali setiap unsur

skrip HTML dan menampilkannya dengan warna yang berbeda sehingga mudah di baca, seperti *Notepad++*, *Sublime Text* dan masih banyak lagi aplikasi lain yang sejenisnya [1].

## 2.6 CSS

CSS adalah singkatan dari *Cascading Style Sheet* yaitu dokumen *web* yang berfungsi mengatur elemen HTML dengan berbagai property yang tersedia sehingga dapat tampil dengan berbagai gaya yang diinginkan. Sebagian orang menganggap CSS bukan termasuk salah satu bahasa pemrograman karena memang strukturnya yang sederhana, hanya berupa kumpulan-kumpulan aturan yang mengatur *style* elemen HTML.

Cara kerja CSS dalam memodifikasi HTML dengan memilih elemen HTML yang akan diatur kemudian memberikan property yang sesuai dengan tampilan yang diinginkan. Dalam memberikan aturan pada elemen *HTML*, skrip CSS terdiri atas 3 bagian yaitu *Selector* untuk memilih elemen yang akan diberi aturan, *property* yang merupakan aturan yang diberikan dan *value* sebagai nilai dari aturan yang diberikan [1].

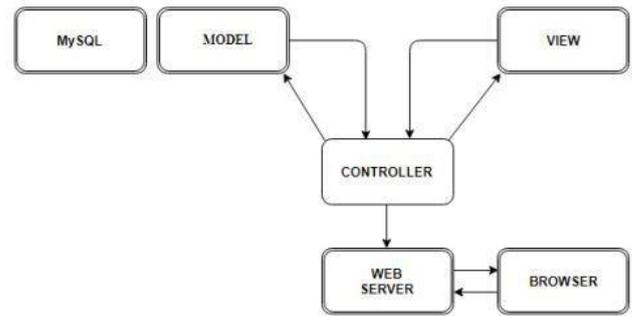
## 2.7 Pengertian Bootstrap

*Bootstrap* merupakan salah satu *framework* CSS paling populer dari sekian banyak *framework* CSS yang ada. *Bootstrap* memungkinkan desain sebuah *web* menjadi responsif sehingga dapat dilihat dari berbagai macam ukuran *device* dengan tampilan tetap menarik. *Bootstrap* juga membuat proses penganturan desain menjadi lebih cepat karena tidak perlu lagi banyak menulis *css*, bahkan hampir tidak perlu kecuali jika memerlukan pengaturan desain yang berbeda dengan *style Bootstrap*. *Bootstrap* telah didukung oleh hampir semua *browser* baik pada desktop maupun *mobile* [1].

## 2.8 Pengertian Framework CodeIgniter

*CodeIgniter* adalah sebuah *framework PHP* yang dapat membantu mempercepat *developer* dalam pengembangan aplikasi *web* berbasis *PHP* dibanding jika menulis semua kode program dari awal. *CodeIgniter* pertama kali dibuat oleh Rick Ellis, CEO *Ellislab, Inc.* (<http://ellislab.com>), sebuah perusahaan yang memproduksi *CMS (Content Management System)* yang cukup handal, yaitu *Expression Engine* (<http://www.expressionengine.com>). Saat ini, *CodeIgniter* dikembangkan dan dimaintain oleh *Expression Engine Development Team* [2].

## 2.9 Flow Diagram CodeIgniter



Gambar 1. Flow Diagram CodeIgniter

## 3. METODOLOGI PENELITIAN

Penelitian ini dilakukan dengan menggunakan tahapan metode :

1. Studi Pendahuluan  
Peneliti memahami *teori-teori* dan pengamatan terhadap pengelolaan data di *Hartini Cinema Management* dalam proses melakukan pendataan yang hanya tersimpan di *Instagram* dan tidak tersusun rapih.
2. Analisis  
Peneliti melakukan analisis data dan informasi berupa foto-foto dan data *talent* yang tersimpan di *instagram*, dan akun-akun *talent* yang penyimpanan datanya kurang tersusun rapih.
3. Perancangan Sistem  
Peneliti membuat perancangan *website* pendataan menggunakan *CodeIgniter*, untuk tabel data *talent* melalui *diagram activity*.
4. Implementasi  
Peneliti melakukan pengembangan untuk mewujudkan desain menjadi suatu *website* pendataan. Teknologi yang digunakan tergantung dengan kebutuhan yang telah dirumuskan pada tahap analisis *website* pendataan menggunakan *CodeIgniter*.
5. Testing  
Peneliti melakukan *testing* bertujuan untuk menguji tabel pendataan apakah sudah terstruktur dengan rapih dan terakurat untuk menyimpan data para *talent* pada *website* pendataan menggunakan *CodeIgniter*.
6. Evaluasi  
Dari hasil uji coba oleh pihak *Hartini Cinema Management* agar mengetahui apakah aplikasi pendataan biodata yang berbasis *website* harus ditingkatkan kembali atau sudah memenuhi kebutuhan.
7. Kesimpulan  
Penarikan kesimpulan dari hasil evaluasi yang telah di uji coba oleh pihak *Management*, untuk pendataan sudah cukup baik dan tersusun rapih.

#### 4. ANALISIS DAN PERANCANGAN

Dalam bab ini akan menjelaskan mengenai Analisis dan Rancangan yang berkaitan dengan permasalahan pendataan yang ada pada Hartini *Cinema Management*. Berikut adalah Analisis dan Rancangan yang mendasari dari perancangan *website* pendataan menggunakan *CodeIgniter Framework* pada Hartini *Cinema Management*.

##### 4.1 Metode Implementasi dan Evaluasi

Analisis adalah penguraian dari suatu masalah atau objek yang akhirnya menghasilkan suatu kesimpulan, hal ini dimaksudkan untuk mengidentifikasi dan mengevaluasi suatu masalah seperti sistem pendataan, Perancangan Aplikasi pengelolaan data talent pada Hartini *Cinema Management* menggunakan *CodeIgniter Framework* ini dimaksudkan untuk merubah sistem pendataan yang tidak beraturan /tidak tersusun di *Instagram* menjadi tersusun dan beraturan di *website* pendataan menggunakan Aplikasi pendataan ini.

##### 4.2 Rancangan Database

Rancangan *database* tabel menu dalam membuat *website* pengelolaan data menggunakan *CodeIgniter* untuk pendataan biodata para talent Hartin *Cinema Management*.

##### 1. Tampilan

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra	Tindakan
1	talent_id	int(11)	Tidak	Tidak ada	Tidak ada	AUTO_INCREMENT		Ubah Hapus Kardi Utama Unk Indeks
2	name	varchar(255)	Tidak	Tidak ada				Ubah Hapus Kardi Utama Unk Indeks Lainnya
3	gender	enum('Male','Female')	Tidak	Tidak ada				Ubah Hapus Kardi Utama Unk Indeks Lainnya
4	address	varchar(255)	Tidak	Tidak ada				Ubah Hapus Kardi Utama Unk Indeks Lainnya
5	umur	varchar(255)	Tidak	Tidak ada				Ubah Hapus Kardi Utama Unk Indeks Lainnya
6	image	varchar(255)	Tidak	Tidak ada				Ubah Hapus Kardi Utama Unk Indeks Lainnya
7	description	text	Tidak	Tidak ada				Ubah Hapus Kardi Utama Unk Indeks Lainnya
8	akun_ig	varchar(20)	Tidak	Tidak ada				Ubah Hapus Kardi Utama Unk Indeks Lainnya
9	phone	varchar(15)	Tidak	Tidak ada				Ubah Hapus Kardi Utama Unk Indeks Lainnya
10	tgl_daftar	date	Tidak	Tidak ada				Ubah Hapus Kardi Utama Unk Indeks Lainnya

Gambar 2. Tampilan Database Talent

##### 2. Tabel History Talent

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra	Tindakan
1	historytalent_id	int(11)	Tidak	Tidak ada	Tidak ada			Ubah Hapus Kardi Utama Unk Indeks Spasial Lainnya
2	talent_id	int(11)	Tidak	Tidak ada				Ubah Hapus Kardi Utama Unk Indeks Spasial Lainnya
3	film_id	int(11)	Tidak	Tidak ada				Ubah Hapus Kardi Utama Unk Indeks Spasial Lainnya
4	peran	varchar(45)	Tidak	Tidak ada				Ubah Hapus Kardi Utama Unk Indeks Spasial Lainnya
5	episod	varchar(45)	Tidak	Tidak ada				Ubah Hapus Kardi Utama Unk Indeks Spasial Lainnya

Gambar 3. Tabel History Talent

##### 3. Tabel Film

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra	Tindakan
1	film_id	int(11)	Tidak	Tidak ada	Tidak ada			Ubah Hapus
2	judul	varchar(50)	Tidak	Tidak ada				Ubah Hapus
3	jenis	varchar(50)	Tidak	Tidak ada				Ubah Hapus
4	tahun	year(4)	Tidak	Tidak ada				Ubah Hapus

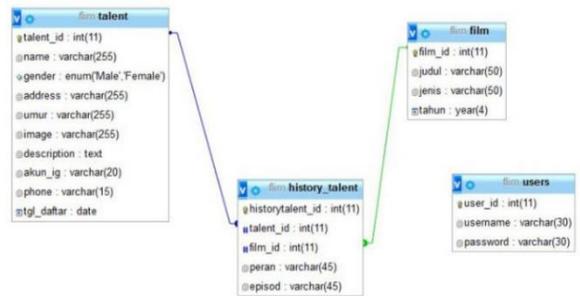
Gambar 4. Tabel Film

##### 4. Tabel Users

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra	Tinda
1	user_id	int(11)	Tidak	Tidak ada	Tidak ada	AUTO_INCREMENT		Ub
2	username	varchar(30)	Tidak	Tidak ada				Ub
3	password	varchar(30)	Tidak	Tidak ada				Ub

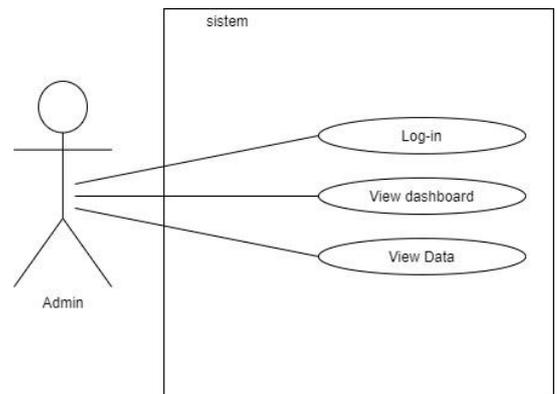
Gambar 5. Tabel Users

#### 4.3 Relasi Tabel

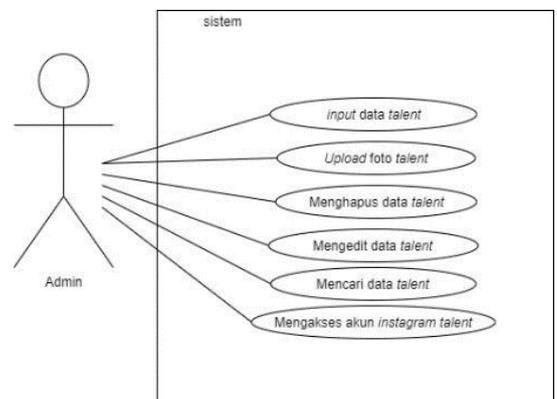


Gambar 6. Relasi Tabel

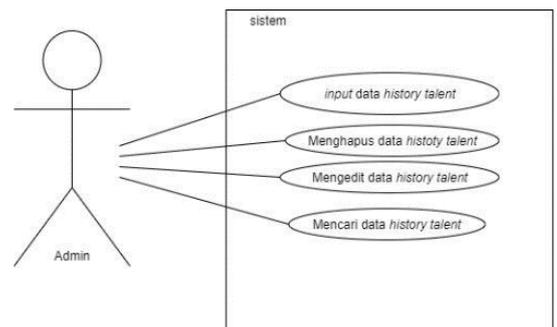
#### 4.4 Use Case



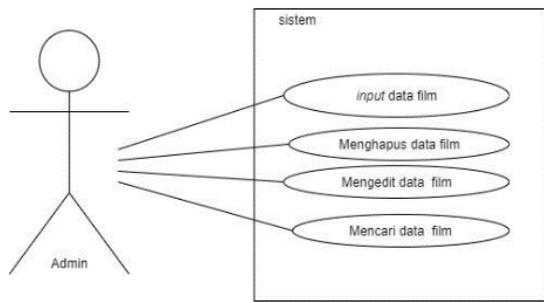
Gambar 7. Use Case Login



Gambar 8. Use Case Menu Talent

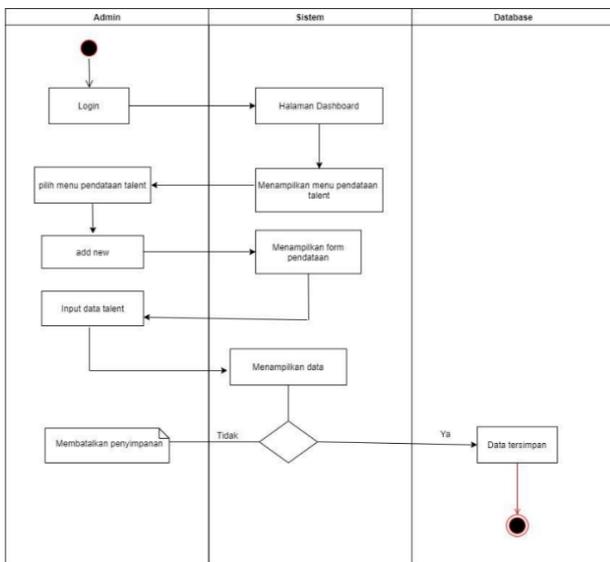


Gambar 9. Use Case Menu History Talent



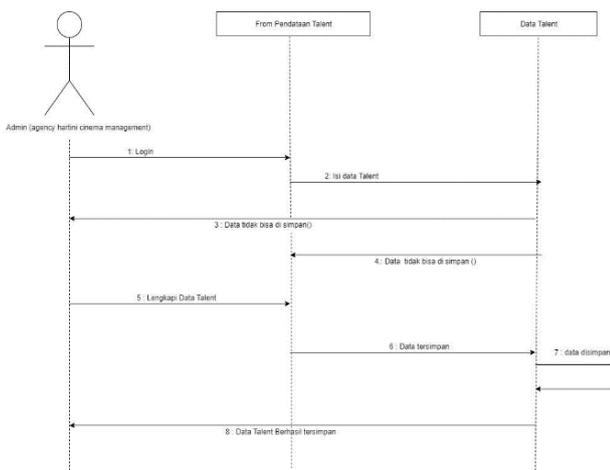
Gambar 10. Use Case Menu Film

4.5 Activity Diagram



Gambar 11. Activity Diagram Website Pendataan Talent

4.6 Sequence Diagram



Gambar 12. Sequence Diagram

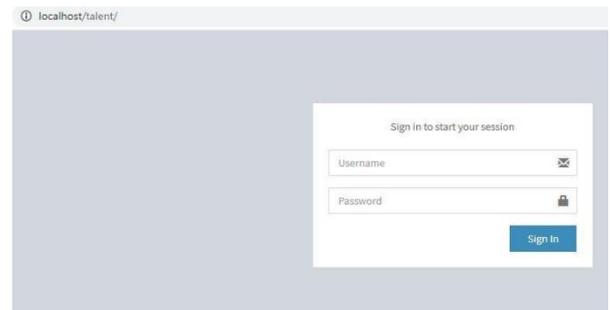
5. PEMBAHASAN DAN HASIL

Tahap ini menerjemahkan perancangan berdasarkan hasil analisis dalam bahasa yang dapat dimengerti oleh mesin serta penerapan perangkat lunak pada keadaan yang

sesungguhnya. Hasil antarmuka (*interface*) dari perangkat lunak dilakukan berdasarkan rancangan yang telah dilakukan. Hasil ditampilkan dari *screenshot* dari halaman *website* yang digunakan sebagai alat dan bahan penelitian yang telah dirincikan pada bab ini.

Bab ini merupakan tahapan-tahapan pengembangan sistem dan disertakan yang hasil menggambarkan program.

5.1 Halaman Login



Gambar 13. Halaman Login

5.2 Halaman Home



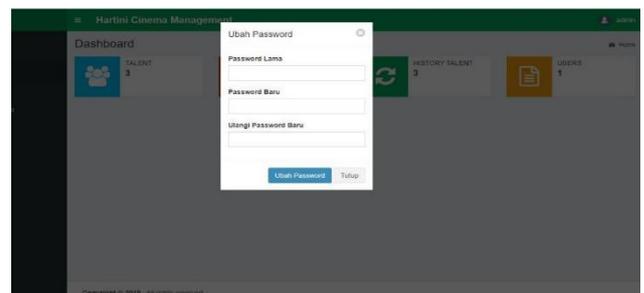
Gambar 14. Halaman Home

5.3 Kolom Admin



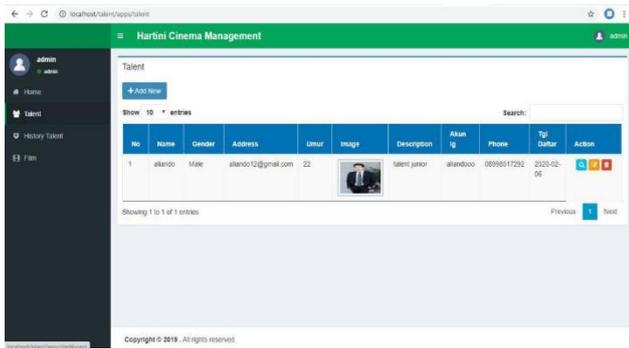
Gambar 15. Kolom Admin

5.4 Kolom Mengubah Password Admin



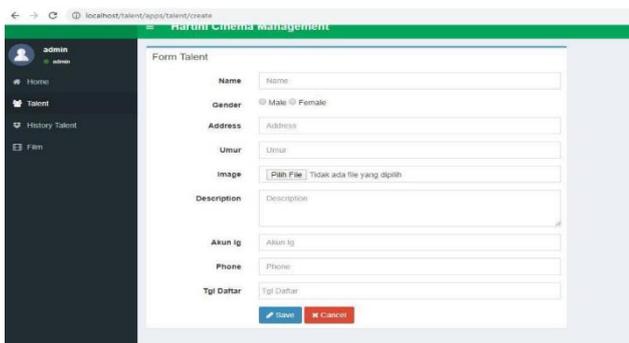
Gambar 16. Kolom Mengubah Password Admin

5.5 Halaman *Talent*



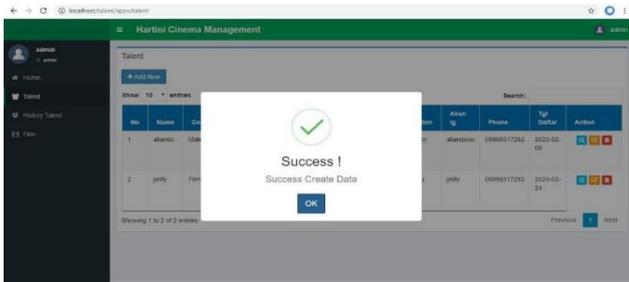
Gambar 17. Halaman *Talent*

5.6 *Add New* (Form Menambah *Data Talent*)



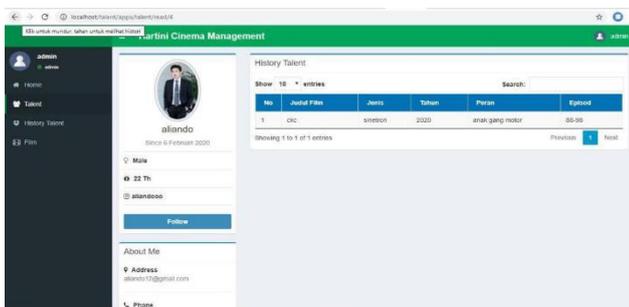
Gambar 18. *Add New* (Form Menambah *Data Talent*)

5.7 *Success Create Data*



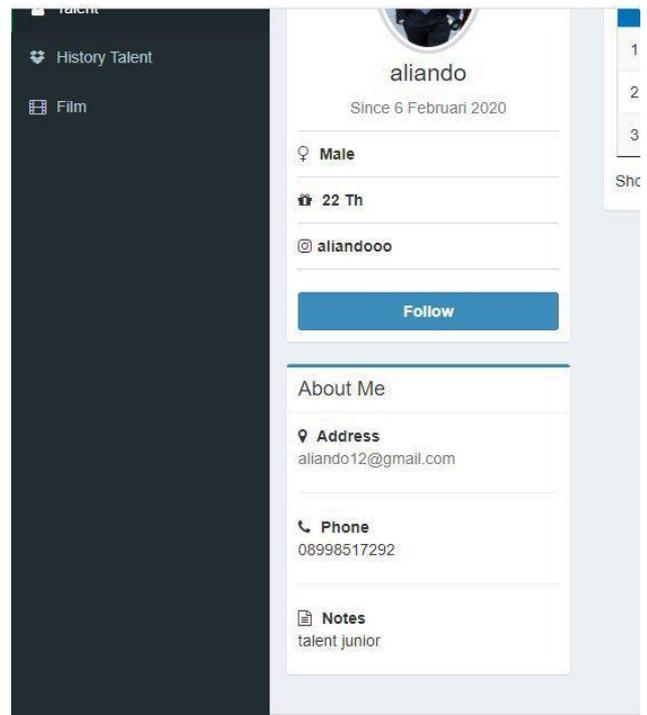
Gambar 19. *Success Create Data*

5.8 Tombol Search di dalam *Action*



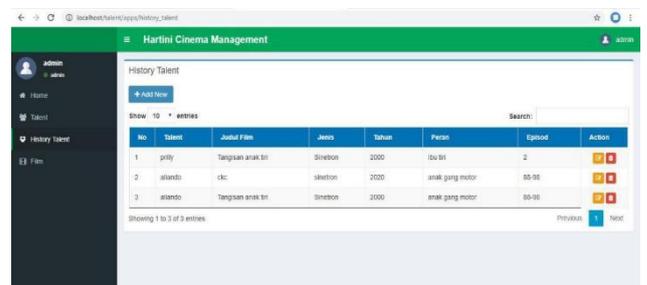
Gambar 20. Tombol Search di dalam *Action*

5.9 Tombol *Follow*



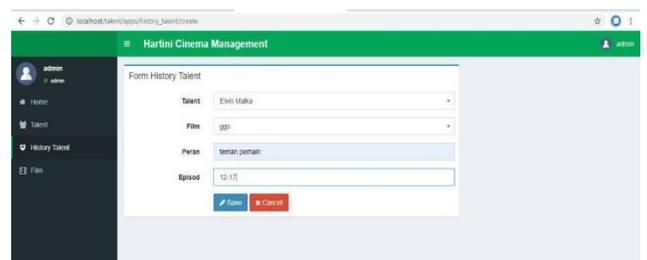
Gambar 21. Tombol *Follow*

5.10 Halaman *History Talent*



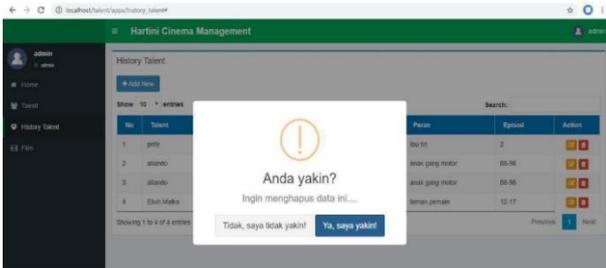
Gambar 22. Halaman *History Talent*

5.11 Halaman *Add New (History Talent)*



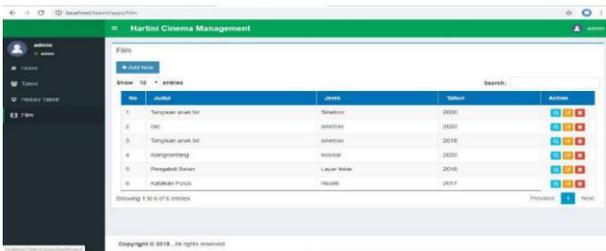
Gambar 23. Halaman *Add New (History Talent)*

5.12 Tombol Pilihan Hapus *History Talent*



Gambar 24. Tombol Pilihan Hapus *History Talent*

5.13 Halaman Film



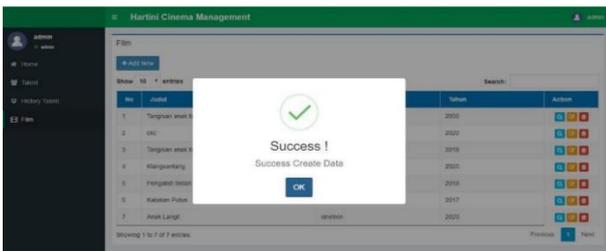
Gambar 25. Halaman Film

5.14 Add New (Film)



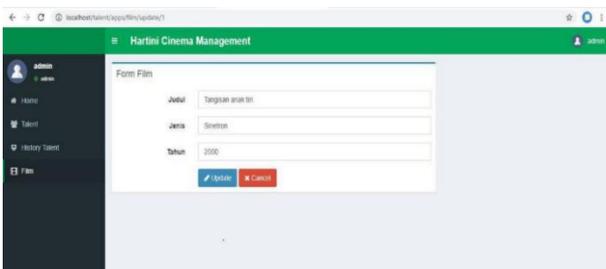
Gambar 26. Add New (Film)

5.15 Berhasil Menyimpan Data Film



Gambar 27. Berhasil Menyimpan Data Film

5.16 Tombol Edit



Gambar 28. Tombol Edit Film

5.17 Hasil Pengujian *User Acceptance Test* (UAT)

*User Acceptance Test* (UAT) atau Uji Penerimaan Pengguna adalah suatu proses pengujian oleh pengguna yang dimaksudkan untuk menghasilkan dokumen pendataan yang dijadikan bukti bahwa *software* yang telah dikembangkan telah dapat diterima oleh pengguna, apabila hasil pengujian (*testing*) sudah bisa dianggap memenuhi kebutuhan dari pengguna. Proses dalam UAT adalah pemeriksaan dan pengujian terhadap hasil pekerjaan. Diperiksa apakah item-item yang ada dalam dokumen pendataan sudah ada dalam *software* yang diuji atau tidak. Diuji apakah semua item yang telah ada telah dapat memenuhi kebutuhan Admin untuk pendataan talent di Hartini *Cinema Management*. Berikut *User Acceptance Test* (UAT) atau Uji Penerimaan Pengguna *website* pendataan pada Hartini *Cinema Management*:

Tabel 1. Hasil UAT

Aktivitas Pengujian	Realisasi yang diharapkan	Kesimpulan	
		Berhasil	Tidak Berhasil
Login	Sistem berhasil proses <i>input data username</i> dan <i>password admin</i>	100%	
Masuk halaman Home	Sistem bisa mengakses ke halaman Home untuk melihat <i>data talent</i> , film dan jumlah users	100%	
Masuk halaman talent	Sistem dapat menginput <i>data talent</i> yang dimasukan pada <i>form talent</i>	100%	
Input data, mengedit dan menghapus data talent pada menu talent	Sistem dapat menyimpan data setelah mengisi data, sistem bisa mengedit dan menghapus <i>data talent</i>	100%	
Klik tombol <i>search</i> pada tabel <i>action</i>	Sistem dapat menampilkan data talent beserta <i>history talent</i> dan akses ke <i>instagram talent</i>	100%	
Masuk ke halaman <i>History talent</i>	Sistem dapat memasukan data <i>history talent</i> seperti judul film, jenis film dan tahun	100%	
Input data, mengedit dan menghapus data talent pada menu <i>History talent</i>	Sistem dapat menyimpan data setelah mengisi data <i>history talent</i> , sistem bisa mengedit, menghapus dan menambah data <i>history talent</i>	100%	

Aktivitas Pengujian	Realisasi yang diharapkan	Kesimpulan	
		Berhasil	Tidak Berhasil
Berhasil di data <i>History Talent</i> dan dapat mencari data	Sistem memverifikasi proses input data untuk di simpan dan dapat mencari data	100%	
Klik <i>back</i> untuk kembali	Sistem dapat kembali ke tabel data setelah mengisi data	100%	
Masuk ke menu film	Sistem dapat mengisi data film apa saja yang ada dan tahun berapa saja film tersebut	100%	
<i>Input data</i> , mengedit dan menghapus <i>data talent</i> pada menu film	Sistem akan memunculkan data-data film seperti nama, jenis, dan tahun film	100%	
Klik <i>logout</i> untuk keluar <i>website</i>	Sistem keluar <i>website</i> setelah di klik <i>logout</i> dan kembali ke halaman depan <i>login</i>	100%	

Berdasarkan hasil uji coba UAT di dapatkan berhasil untuk semua pengujian yang meliputi : *Login*, masuk ke halaman *home*, menginput data, mengedit dan menghapus data talent pada setiap menu seperti menu *talent*, *history talent* dan film yang berhasil menampilkan data talent dan di simpulkan *website* sudah efektif untuk semua fungsi aplikasi pendataan sesuai kebutuhan *management*. Dan sudah efisien dalam efisiensi biaya karna di *website* pengelolaan data ini tidak memerlukan kuota karna *website* ini tidak online tetapi *localhost* sedangkan pendataan di *Instagram* sangat merlukan kuota, dan *website* ini lebih cepat dalam memasukan pendataan dibandingkan di *instagram* yang memerlukan waktu *loading* yang tidak tentu karena jaringan.

5.18 Permyataan Kuesioner

No	Pernyataan	A	B	C	D
1	tampilan <i>website</i> sebagai <i>website</i> pendataan menarik ?				
2	Menu-menu <i>website</i> pendataan ini , sudah sesuai dengan kebutuhan ?				
3	<i>form</i> pendataan pengisiin udah sesuai dengan kebutuhan <i>managem ent</i> ?				
4	Halaman Informasi data <i>managem ent</i> sudah sesuai <i>managam ent</i> ?				
5	Sistem pendataan dengan <i>website</i> ini sudah teratur ?				

Gambar 29. Pernyataan Kuesioner

Tabel 2. Hasil Pernyataan Kuesioner

No. Responden	Item Pernyataan					Skor Total
	1	2	3	4	5	
1	4	4	3	4	4	19
2	3	4	4	4	4	19
3	4	4	3	4	4	19
4	4	3	4	4	4	19
5	3	4	4	4	4	19
6	4	4	3	4	4	19
7	4	3	4	4	4	19
8	3	4	4	4	4	19
9	3	4	4	4	4	19
10	3	4	4	4	4	19
						190

- a. Responden yang menjawab sangat setuju (skor 4) berjumlah 40 orang
- b. Responden yang menjawab setuju (skor 3) berjumlah 10 orang
- c. Responden yang menjawab tidak setuju (skor 2) berjumlah 0 orang
- d. Responden yang menjawab sangat tidak setuju (skor 1) berjumlah 0 orang

1. Rumus:  $T \times P_n$

T = Total jumlah responden yang memilih

P<sub>n</sub> = Pilihan angka skor Likert

- a. Responden yang menjawab sangat setuju (4) = 4 x 40 = 160
- b. Responden yang menjawab setuju (3) = 3 X 10 = 30
- c. Responden yang menjawab tidak setuju (2) = 2 X 0 = 0
- d. Responden yang menjawab sangat tidak setuju (1) = 1 X 0 = 0
- e. Semua hasil dijumlahkan, total skor = 130

2. Interpretasi Skor Perhitungan

Agar mendapatkan hasil *interpretasi*, terlebih dahulu harus diketahui skor tertinggi (X) dan skor terendah (Y) untuk item penilaian dengan rumus sebagai berikut:

$$Y = \text{skor tertinggi likert} \times \text{jumlah responden}$$

$$X = \text{skor terendah likert} \times \text{jumlah responden}$$

Jumlah skor tertinggi untuk item “Sangat Suka” adalah 4x 100 = 400, sedangkan item “Sangat Tidak Suka”

adalah  $1 \times 100 = 100$ . Jadi, jika total *skor* penilaian *responden* diperoleh angka 190, maka penilaian *interpretasi responden* terhadap *website* pendataan pada Hartini *cinema management* tersebut adalah hasil nilai yang dihasilkan dengan menggunakan rumus *Index %*.

#### Penyelesaian Akhir

$$= \text{Total skor} / Y \times 100$$

$$= 190 / 400 \times 100$$

$$= 47.5\% \text{ berada dalam kategori "Cukup/Netral"}$$

## 6. KESIMPULAN DAN SARAN

### 6.1 Kesimpulan

Perancangan aplikasi pengelolaan data *talent* pada Hartini *Cinema Management* menggunakan *CodeIgniter Framework*, berhasil dibuat dalam penelitian ini. Hal itu ditunjukkan oleh hasil pengujian dengan angka keberhasilan 47.5% fungsi aplikasi dapat diterima oleh pengguna dengan hasil UAT di atas. *Website* ini ada efisiensi biaya

karena di *website* pengelolaan data ini tidak memerlukan kuota (biaya internet) karena *website* ini tidak online tetapi *localhost*, sedangkan pendataan di *instagram* sangat memerlukan kuota (biaya). *Website* ini lebih cepat untuk memasukan pendataan dibandingkan dengan *instagram*, yang memerlukan waktu *loading* yang tidak tentu karena jaringan.

### 6.2 Saran

Dari kegiatan penelitian ini terdapat saran yang ditujukan ke dalam penelitian ini, yakni diharapkan untuk pengembang atau peneliti selanjutnya untuk mengembangkan *website* pendataan ini untuk membuat sistem penggajian para talent pada Hartini *CinemaManagement*

### DAFTAR PUSTAKA

- [1] R. Abdulloh, "7 in 1 Pemrograman Web untuk Pemula," Jakarta: PT Elex Media Komputindo kelompok Gramedia, 2018.
- [2] Hakim, "Membangun Web berbasis PHP dengan *Framework CodeIgniter*," 2010.



## ANALISIS DAN PENERAPAN ENTERPRISE RESOURCE PLANNING (ERP) ODOO 10 GUNA MEMBUAT LAPORAN KEUANGAN PADA YAYASAN DARUL JANNAH

Rosy Annisa Firlana<sup>1</sup>, Suhendi<sup>2</sup>

<sup>1,2</sup> Sistem Informasi, Sekolah Tinggi Teknologi Terpadu Nurul Fikri  
Jakarta Selatan, DKI Jakarta, Indonesia 12640  
[rosyannisafirlana@gmail.com](mailto:rosyannisafirlana@gmail.com), [suhendi@nurulfikri.ac.id](mailto:suhendi@nurulfikri.ac.id)

### Abstract

*The rapid development of technology has made many foundations adopt the latest information technology to help run their business. This writing discusses the application of Odoo application to input payment transactions for students to create financial reports. This application is implemented to facilitate student payment transactions and make it easier to make financial reports. This application uses the Odoo application version 10 and uses the Accounting and Finance module. The research method used is the method of field studies using collecting data from interviews by the principal of TPA Darul Jannah. Testing of this system was carried out using Microsoft Edge. A survey by filling out a questionnaire was conducted to determine user satisfaction with the Odoo system. So it can be concluded that the Odoo system is quite helpful in inputting payment transactions for santri to print financial reports.*

**Keywords:** Information systems, computer networks, human and computer interactions

### Abstrak

Pesatnya perkembangan teknologi pada saat ini membuat banyak Yayasan mengadopsi teknologi informasi terbaru untuk membantu kelancaran bisnis. Penulisan ini membahas mengenai penerapan aplikasi Odoo yang digunakan untuk *input* transaksi pembayaran santri hingga pembuatan laporan keuangan. Aplikasi ini diterapkan dengan tujuan untuk mempermudah dalam *input* transaksi pembayaran santri dan mempermudah dalam membuat laporan keuangan. Penerapan ini menggunakan aplikasi Odoo versi 10 dan menggunakan modul *Accounting and Finance*. Metode penelitian yang digunakan yaitu metode studi lapangan dengan cara melakukan proses pengumpulan data dari hasil wawancara oleh kepala sekolah TPA Darul Jannah. Pengujian sistem ini dilakukan dengan menggunakan *Microsoft Edge*. Survei dengan pengisian kuisioner dilakukan untuk mengetahui tingkat kepuasan pengguna terhadap sistem Odoo. Sehingga dapat disimpulkan bahwa sistem Odoo ini sudah cukup membantu dalam proses *input* transaksi pembayaran santri hingga mencetak laporan keuangan.

**Kata kunci:** *Accounting and Finance*, Odoo, *Enterprise Resource Planning* (ERP)

### 1. PENDAHULUAN

Pesatnya perkembangan teknologi pada saat ini, membuat banyak Yayasan mengadopsi teknologi informasi terbaru untuk membantu kelancaran bisnis. Teknologi tidak hanya dipandang sebagai pelengkap, tetapi sudah menjadi salah satu penentu atas pelaksanaan sasaran atau strategi pengembangan pelaporan keuangan yayasan. Sehingga menimbulkan tantangan baru bagi yayasan untuk menyediakan suatu sistem yang mampu mengintegrasikan informasi –informasi yang dibutuhkan Yayasan.

TPA (Taman Pendidikan Al-Qur'an) Darul Jannah merupakan suatu Lembaga informal di bawah Yayasan

Darul Jannah yang dimana Lembaga tersebut menerima peserta yang ingin belajar Qur'an bagi kalangan anak-anak dari usia PAUD hingga SMP. Proses transaksi pembayaran di TPA Darul Jannah di bawah pengelolaan Yayasan Darul Jannah pada saat ini masih menggunakan cara manual, sehingga teknologi menjadi kunci sangat penting agar dalam pendataan lebih sistematis dan pengurangan penumpukan file yang disimpan dalam suatu folder.

Dalam mencapai hal tersebut penulis ingin mengimplementasikan konsep ERP (*Enterprise Resource Planning*) yang dimana dapat mengotomatisasi dan mengintegrasikan seluruh proses bisnis transaksi keuangan

Yayasan Darul Jannah khusus di TPA Darul Jannah. Dengan menggunakan sistem saat ini tidak akan mudah untuk mendapatkan efisiensi ketepatan waktu serta informasi yang akurat. Solusi yang ditawarkan oleh sistem ERP membuat yayasan mendapatkan otomatisasi proses pelaporan keuangan dan berefek kepada proses transaksi yang efisien.

Dalam melakukan proses penerapan sistem keuangan TPA Darul Jannah, penulis menggunakan aplikasi Odoo. Odoo sendiri terdapat berbagai macam modul maka dari itu penulis memilih modul *Accounting and Finance* dan sudah di design sedemikian rupa agar dapat disesuaikan dengan kebutuhan manajemen keuangan TPA Darul Jannah. Aplikasi Odoo ini sangat diharapkan mampu membantu dalam pengelolaan keuangan di TPA Darul Jannah jika dalam implementasinya dilakukan secara tepat, sehingga segala kebutuhan TPA Darul Jannah dapat terintegrasi dengan baik dan mempersingkat waktu karena yang pada awalnya dilakukan dengan cara manual.

### 1.1 Rumusan Masalah

Berdasarkan penjelasan latar belakang di atas, dapat dipaparkan beberapa permasalahan yaitu:

- Bagaimana alur proses usulan transaksi pembayaran yang akan diterapkan di TPA Darul Jannah menggunakan aplikasi Odoo?
- Bagaimana implementasi transaksi pembayaran di TPA Darul Jannah menggunakan aplikasi Odoo?
- Bagaimana hasil evaluasi dalam penerapan sistem pengelolaan transaksi keuangan menggunakan aplikasi Odoo untuk TPA Darul Jannah?

### 1.2 Tujuan dan Manfaat Penelitian

Dilihat dari rumusan masalah yang sudah dipaparkan, maka tujuan yang ingin dicapai, sebagai berikut:

- Memahami alur proses transaksi keuangan yang sudah diterapkan di TPA Darul Jannah.
- Memahami cara implementasi transaksi pembayaran menggunakan aplikasi Odoo.
- Mengetahui hasil evaluasi dari penerapan sistem transaksi pembayaran menggunakan aplikasi Odoo di TPA Darul Jannah.

Sedangkan manfaat yang diharapkan oleh peneliti adalah sebagai berikut:

- Memudahkan alur proses pembayaran di TPA Darul Jannah.
- Mengurangi penumpukan file pada folder-folder tertentu dalam bentuk kertas.
- Memudahkan pihak Yayasan dalam pengecekan data-data keuangan.

### 1.3 Batasan Masalah

Berdasarkan latar belakang dari permasalahan dalam penelitian ini, maka yang menjadi batasan masalah, sebagai berikut:

- Implementasi yang dilakukan hanya pada TPA Darul Jannah untuk melakukan laporan keuangan dalam bentuk jurnal umum.
- Proses transaksi pembayaran yang akan digunakan dalam aplikasi Odoo adalah pembayaran SPP, pendaftaran, daftar ulang, dan pembelian buku.

## 2. METODE PENELITIAN

### 2.1 Laporan Keuangan

Laporan keuangan adalah hasil proses pencatatan akuntansi keuangan. Laporan keuangan itu berisi informasi tentang prestasi perusahaan di bidang keuangan pada masa lampau. Laporan keuangan yang utama yaitu neraca dan perhitungan rugi-laba. Namun, dalam praktek sering diikutsertakan beberapa laporan lain untuk memperjelas, misalnya laporan perubahan modal atau laporan laba yang ditahan, laporan perubahan modal kerja, perhitungan harga pokok, dan lain-lain. Neraca memberikan gambaran mengenai posisi keuangan perusahaan pada tanggal tertentu. Perhitungan rugi-laba menunjukkan hasil aktivitas perusahaan selama satu periode. Laporan keuangan merupakan bagian penting dari informasi keuangan bagi pimpinan perusahaan, investor, kreditor, pemerintah dan pihak-pihak lain yang berkepentingan. Dua buah laporan keuangan yang terpenting adalah neraca (*balance sheet*) dan perhitungan rugi-laba (*income statement*).

Menurut standar akuntansi keuangan tujuan laporan keuangan adalah:

- Menyediakan informasi yang menyangkut posisi keuangan, kinerja serta perubahan posisi keuangan suatu perusahaan yang bermanfaat bagi sejumlah besar pemakai dalam mengambil keputusan ekonomi,
- Laporan keuangan disusun untuk memenuhi kebutuhan bersama oleh sebagian besar pemakainya, yang secara umum menggambarkan pengaruh keuangan dari kejadian masa lalu,
- Laporan keuangan juga menunjukkan apa yang telah dilakukan manajemen atau pertanggungjawaban manajemen atas sumber daya yang dipercayakan kepadanya [1].

### 2.2 Odoo

Odoo adalah aplikasi web yang dibangun menggunakan bahasa pemrograman Python, XML dan Javascript dan PostgreSQL sebagai *database*-nya. Dalam sejarah pertama kali Odoo bernama TinyERP, kemudian pada tahun 2009 berganti nama menjadi OpenERP setelah itu dikembangkan kembali menjadi versi 8 berganti nama menjadi Odoo. Odoo juga salah satu penerapan dari aplikasi *Enterprise Resource Planning* (ERP) yang dibuat dalam bentuk *Open Source*.

Odoo merupakan satu-satunya sistem manajemen yang tidak hanya digunakan oleh perusahaan perusahaan besar saja, tetapi juga digunakan oleh perusahaan kecil dan independen. Odoo pun dapat diaplikasikan pada berbagai macam sektor, seperti sektor perdagangan, tekstil, agrikultural, dan lainnya. Perbedaan tersebut menggambarkan tingkat fleksibilitas Odoo yang sangat tinggi sehingga dapat menjangkau seluruh jenis perusahaan yang ada [2].

Berikut kelebihan-kelebihan yang dimiliki oleh Odoo:

1. Akses informasi yang dapat dipercaya.
2. Menghindari redundansi dari pemasukan data dan operasi.
3. Mengurangi waktu jeda waktu penampilan informasi dan laporan.
4. Pengurangan biaya, penghematan waktu, dan peningkatan kontrol dengan analisis skala enterprise. Modul CRM (*Customer Relationship Management*) dan EDI [2].

### 2.3 Modul Accounting and Finance

Modul *Accounting and Finance* merupakan modul dasar dari kebanyakan dari sistem Odoo, karena modul ini mampu untuk mengumpulkan data keuangan dari berbagai *department* fungsional dan mampu untuk menghasilkan laporan keuangan.

Beberapa fungsi modul *Accounting and Finance*:

1. Meng-handle semua *account* yang berhubungan dengan masukan atau entri dari dampak dari entri tersebut terhadap sistem secara keseluruhan,
2. Mencatat semua proses keuangan yang masuk dan juga mencatat semua bagian keuangan tersebut digunakan.

Dengan adanya modul ini, diberikan gambaran mengenai posisi keuangan yang sangat membantu dalam mengambil keputusan [2].

### 2.4 Enterprise Resource Planning (ERP)

*Enterprise Resources Planning* (ERP), merupakan sebuah konsep atau perangkat lunak yang berguna untuk merencanakan dan mengelola sumber daya perusahaan yang berpengaruh luas mulai dari manajemen paling atas hingga operasional disebuah perusahaan agar dapat dimanfaatkan secara optimal untuk menghasilkan nilai tambah bagi seluruh pihak yang berkepentingan (stakeholder) atas perusahaan tersebut. Dalam *Enterprise Resources Planning* (ERP) biasanya memiliki modul-modul yang terintegrasi berdasarkan fungsi-fungsi operasional dalam bisnis, yaitu: akuntansi, keuangan, sumber daya manusia, pemasaran, logistik dan lainnya.

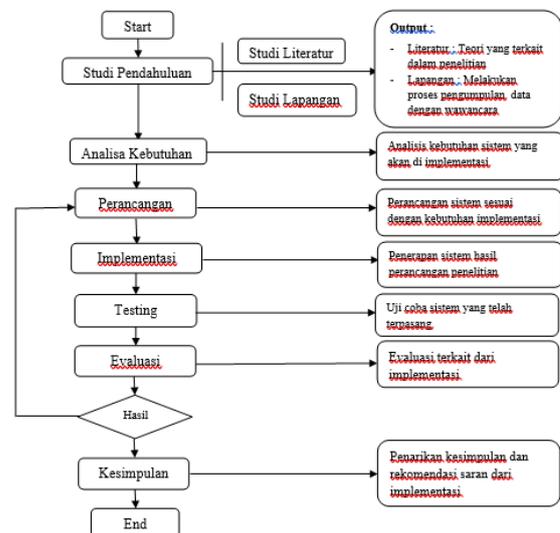
Salah satunya adalah modul yang menyangkut fungsi

akuntansi berisikan buku besar, piutang dagang, hutang dagang, aktiva tetap, manajemen kas dan akuntansi. Sedangkan fungsi keuangan berupa modul analisis portofolio, analisis resiko, analisis kredit, manajemen aktiva, sewa guna dan lainnya [2].

ERP (*Enterprise Resource Planning*) ialah perencanaan sumber daya perusahaan dan merupakan sebuah sistem informasi yang digunakan oleh sebuah perusahaan barang atau jasa yang berguna untuk mengintegrasikan semua proses jalannya perusahaan dari segala aspek baik proses produksi, operasional, distribusi, dan proses lainnya dari produk atau jasa dari perusahaan tersebut. ERP dirancang agar dapat mengkoordinasikan semua sumber daya, informasi dan aktivitas yang diperlukan untuk proses bisnis perusahaan [3].

### 3. METODOLOGI PENELITIAN

Berikut ini tahapan-tahapan penelitian yang akan dilakukan:



Gambar 1. Tahapan Penelitian

Adapun perangkat yang digunakan yaitu:

- a. Intel® Core™ i3-5005U CPU @2.00GHz 2.00 GHz,
- b. 4.00 GB (3.34 GB Usable),
- c. Tipe 64 bit,
- d. Sistem operasi windows 10.

Serta sistem aplikasi yang digunakan adalah:

- a. Aplikasi Odoo versi 10,
- b. PostgreSQL,
- c. Web Browser,
- d. Microsoft Office.

### 4. ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini berisikan langkah-langkah untuk mendapatkan informasi dari sistem yang akan diterapkan berdasarkan hasil pengumpulan data yang dibutuhkan melalui observasi sistem yang telah digunakan sebelumnya dan wawancara

kepada pihak yang terkait dalam keuangan TPA Darul Jannah. Informasi dan data yang telah didapatkan akan dianalisis dan digunakan untuk mempermudah mengidentifikasi suatu permasalahan sistem yang sebelumnya sudah diterapkan.

#### 4.1 Analisis Masalah

Dalam melakukan analisis sistem pencatatan akuntansi dan keuangan TPA Darul Jannah pada proses pembayaran dan pengeluaran biaya operasional santri yang berjalan, maka ditemukan beberapa masalah dalam pencatatan akuntansi dan keuangan tersebut, yang dimana sistem yang digunakan masih menerapkan sistem pencatatan secara manual (menggunakan *Microsoft Excel*). Dalam hal ini proses tersebut belum efektif sehingga dapat menimbulkan beberapa masalah, diantaranya sebagai berikut:

1. Terdapat tumpukan file data keuangan yang disimpan dalam folder,
2. Memungkinkan terjadinya data yang tercecer, rusak, atau hilang,
3. Proses pendataan keuangan menjadilebih lama,
4. Melakukan pencatatan transaksi keuangan serta laporan keuangan masih dalam sistem manual.

Dari berbagai masalah ini dapat disebabkan karena adanya kurang keefektifan dalam proses transaksi pembayaran hingga pencatatan keuangan, selain dari itu sistem yang digunakan sekarang ini mengharuskan admin membackup data-data agar data tersebut selalu tersedia dan menghindari terjadinya kehilangan data.

#### 4.2 Analisis Perancangan Sistem

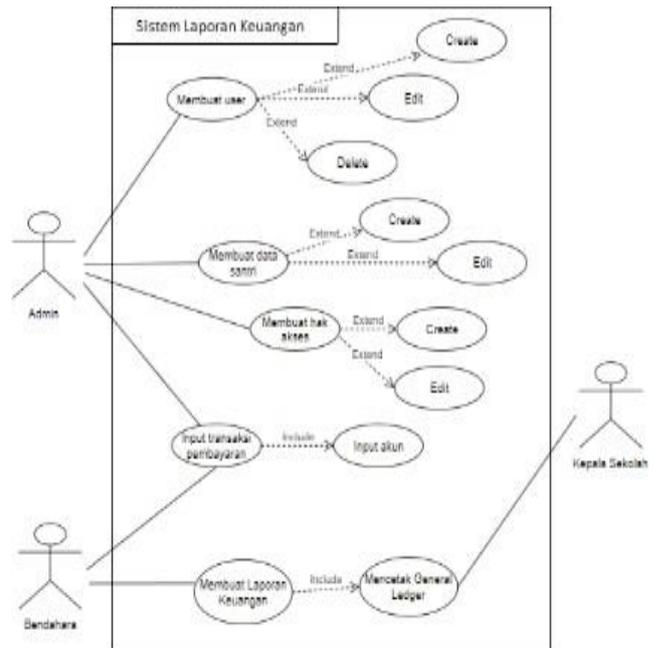
##### 4.2.1 Identifikasi Aktor

Pada tahap ini akan mengidentifikasi aktor-aktor yang terlibat pada implementasi aplikasi Odoo di TPA Darul Jannah. Berikut ini adalah aktor yang teridentifikasi dalam keuangan TPA Darul Jannah:

1. Administrator,
2. Bendahara,
3. Kepala Sekolah.

##### 4.2.2 Use Case

Pada gambar ini akan menjelaskan tentang *use case diagram* dalam implementasi Odoo modul keuangan, yang dimana *use case* sendiri berperan sebagai representasi dari fungsi-fungsi yang berdasarkan dengan kebutuhan *user*.

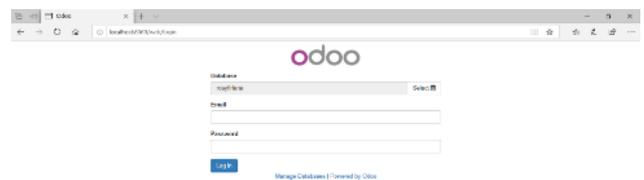


Gambar 2. Use Case

### 5. IMPLEMENTASI DAN EVALUASI

#### 5.1 Login

Berikut ini tampilan *login user* Odoo:



Gambar 3. Login

#### 5.2 Penginputan Data Perusahaan

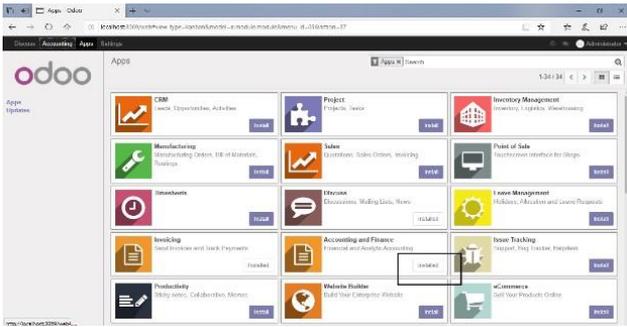
Berikut ini tampilan data perusahaan TPA Darul Jannah:



Gambar 4. Penginputan Data Perusahaan

#### 5.3 Kustomisasi Sistem

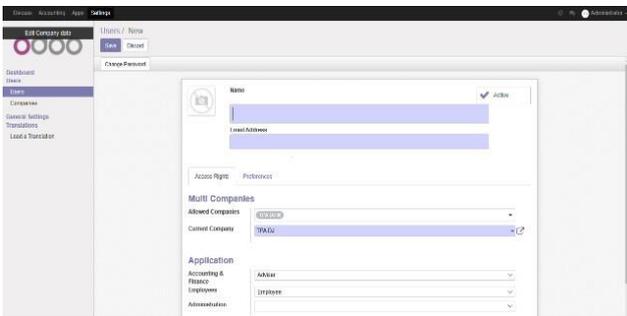
Terdapat banyak modul yang disediakan oleh sistem Odoo untuk menunjang kebutuhan integrasi data yang dibutuhkan oleh sebuah perusahaan. Disini penulis hanya berfokus pada modul *Accounting and Finance* untuk dapat menyelesaikan proses implementasi ini.



Gambar 5. Kustomisasi Sistem

5.4 Kustomisasi Hak Akses

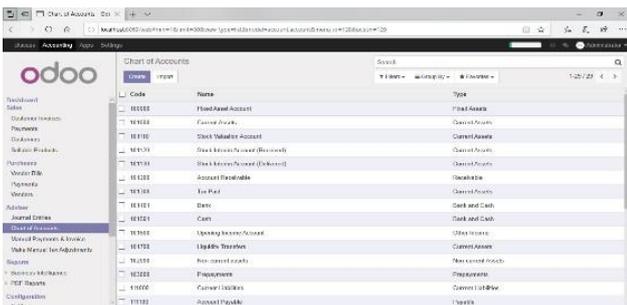
Odoo mempunyai fitur *User Permission* dimana *role* yang telah dibuat dapat digunakan sesuai dengantugas masing-masing peran.



Gambar 6. Kustomisasi Hak Akses

5.5 Kustomisasi Akun

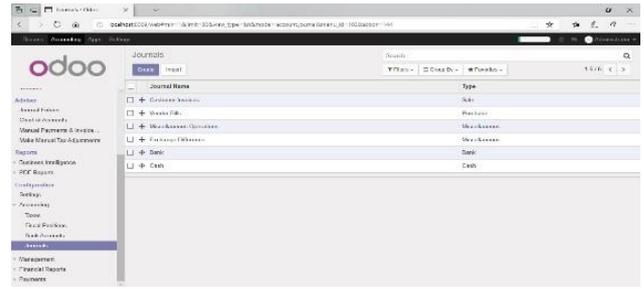
Pada dasar akuntansi memiliki akun-akun yang mempresentasikan dengan adanya transaksi. Dalam sistem Odoo pun telah tersedia konfigurasi akun-akun yang dapat disesuaikan dengan kebutuhan.



Gambar 7. Kustomisasi Akun

5.6 Kustomisasi Jurnal

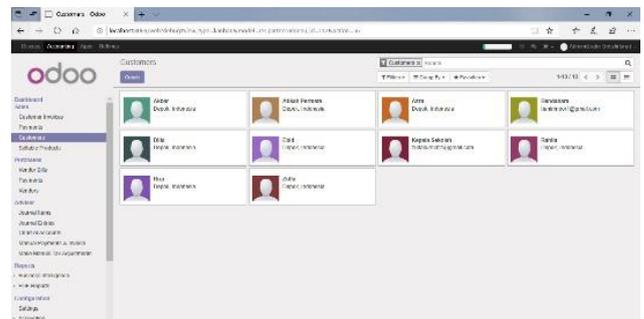
Pada tahap ini adalah pembuatan jurnal yang berfungsi untuk tempat penginputan transaksi yang ada.



Gambar 8. Kustomisasi Jurnal

5.7 Penginputan Data Santri (Customer)

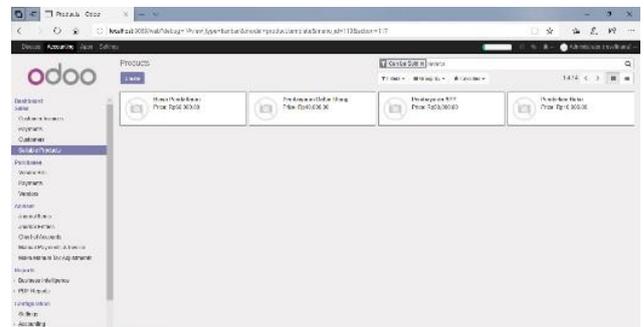
Berikut ini tampilan data *customers* TPA Darul Jannah yang telah diinput:



Gambar 9. Penginputan Data Santri

5.8 Penginputan Produk

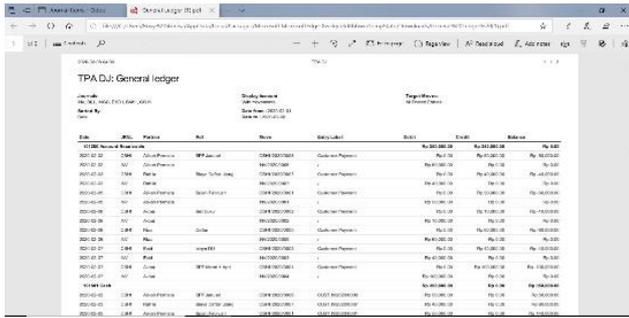
Berikut ini tampilan data produk TPA Darul Jannah:



Gambar 10. Penginputan Produk

5.9 Laporan Keuangan

Pada tahap terakhir implementasi ini adalah mencetak laporan keuangan dari penginputan transaksi-transaksi yang sudah dilakukan dalam kurun waktu tertentu dan sistem Odoo memfasilitasi hal tersebut. Hal ini dapat dilihat di dalam fitur *Reports*, konten tersebut sesuai dengan standart keuangan yang ada. Berikut ini tampilan *General Ledger* yang terdapat disetiap transaksi:



Gambar 11. Laporan Keuangan

5.10 Hasil Pengujian UAT Bendahara

Tabel 1. Pengujian UAT

No	Item Testing	Hasil Testing	
		Sukses	Tidak Sukses
1	Login	√	
2	Input transaksi pembayaran SPP	√	
3	Input transaksi pembayaran daftar ulang	√	
4	Input transaksi pembayaran kegiatan tahunan	√	
5	Input laporan keuangan	√	
6	Mencetak General Ledger	√	
7	Logout	√	

Dari hasil pengujian didapatkan kesimpulan bahwa hasil uji dapat digunakan sesuai kebutuhan yang meliputi login, input transaksi, input laporan, mencetak general ledger dan logout.

5.11 Hasil Pengujian UAT Kepala Sekolah

Tabel 2. Pengujian UAT Kepala Sekolah

No	Item Testing	Hasil Testing	
		Sukses	Tidak Sukses
1	Login	√	
2	Melihat laporan keuangan	√	
3	Mencetak General Ledger	√	
4	Logout	√	

Hasil Pengujian user kepala sekolah dapat disimpulkan bahwa hasil uji coba dapat digunakan sesuai dengan alur yang telah ditentukan.

5.12 Hasil Pengujian UAT Admin

Tabel 3. Pengujian UAT Admin

No	Item Testing	Hasil Testing	
		Sukses	Tidak Sukses
1	Login	√	
2	Meng-input data santri	√	
3	Membuat user	√	
4	Men-setting daftar akun	√	
5	Input transaksi pembayaran	√	
6	Input laporan keuangan	√	
7	Mencetak General Ledger	√	
8	Logout	√	

Hasil pengujian user Admin dapat disimpulkan bahwa dalam melakukan proses dari awal proses login, penginputan transaksi pembayaran santri, pengeditan data santri, hingga proses logout dapat berjalan sesuai dengan kebutuhan.

5.13 Hasil Evaluasi

Tabel 4. Hasil Evaluasi

No	Pernyataan	Hasil Penilaian			
		1 (Sangat Tidak Setuju)	2 (Tidak Setuju)	3 (Setuju)	4 (Sangat Setuju)
1	Aplikasi Odoo ini sangat membantu untuk proses input transaksi pembayaran				√
2	Dengan menggunakan aplikasi ini pencatatan sudah tersusun dengan rapih			√	
3	Aplikasi ini cepat dalam proses input transaksi			√	
4	Aplikasi ini dapat membantu dalam pembuatan laporan keuangan			√	
5	Aplikasi Odoo dapat membantu peng-inputan data santri			√	

Berdasarkan hasil evaluasi yang telah didapat dari kepala sekolah dan bendahara TPA Darul Jannah maka dapat disimpulkan bahwa aplikasi Odoo ini yang telah diterapkan di TPA Darul Jannah sudah cukup membantu proses

transaksi pembayaran santri hingga mencetak laporan keuangan.

## 6. KESIMPULAN

### 6.1 Kesimpulan

Dari hasil penelitian yang telah dilakukan oleh peneliti, maka dapat disimpulkan bahwa penelitian ini berhasil menerapkan sistem keuangan di TPA Darul Jannah menggunakan *software* Odoo. Hasil dari penelitian ini dapat menjawab rumusan masalah yang sudah dipaparkan sebelumnya, sebagai berikut:

1. Alur proses usulan transaksi pembayaran yang akan diterapkan di TPA Darul Jannah, yaitu:
  - a. Membuat bukti transaksi pembayaran santri melalui submenu *Customer Invoice*,
  - b. Melakukan penginputan transaksi melalui submenu *Payment*,
  - c. Membuat laporan keuangan otomatis yang berbentuk *General Ledger* di dalam menu *Reports*.
2. Implementasi transaksi pembayaran santri TPA Darul Jannah menggunakan aplikasi Odoo, berikut ini tahapan yang dilakukan oleh bendahara atau admin dalam proses *input* pembayaran hingga pembuatan laporan keuangan, sebagai berikut:
  - a. Pembuatan bukti transaksi,
  - b. Melakukan *input* transaksi ke dalam sub menu *payment*,
  - c. Melakukan cetak bukti transaksi,
  - d. Pembuatan laporan keuangan.
3. Dari hasil pernyataan kuisioner didapatkan hasil evaluasi bahwa penerapan sistem transaksi

pembayaran menggunakan aplikasi Odoo memudahkan proses transaksi, memahami cara implementasi pembayaran, dan mengetahui hasil evaluasi dari penerapan sistem Odoo.

### 6.2 Saran

Dalam penerapan sistem Odoo di TPA DarulJannah masih banyak kekurangan, oleh sebab itu perlu dengan adanya penyempurnaan penelitian lebih lanjut. Maka peneliti menyarankan untuk pengoptimalan penerapan aplikasi Odoo yang akan datang, sebagai berikut:

1. Hak *user* kepala sekolah dan bendahara masih belum sesuai, seharusnya di dalam tampilan kepala sekolah dan bendahara tidak ada menu *setting*, yang dimana menu tersebut digunakan untuk membuat hak *user* dan data perusahaan.
2. *Input* transaksi pengeluaran kas masih belum tersedia, diharapkan untuk penelitian selanjutnya dapat melengkapi *input* transaksi pengeluaran kas di dalam aplikasi Odoo untuk TPA Darul Jannah.

## DAFTAR PUSTAKA

- [1] K. D. Ramadhan, L. O. Syarfan, "Analisis Laporan Keuangan dalam Mengukur Kinerja Perusahaan," Riau: Universitas Islam Riau, 2016.
- [2] C. Ayu, "Implementasi Odoo Dengan Modul *Accounting and Finance*," Depok: STT Nurul Fikri, 2018.
- [3] M. R. Y. Putra, "Implementasi Modul *Accounting And Finance* menggunakan Odoo dengan ERP Metode *Iterative Waterfall* pada PT. Albasia Nusa Karya," Universitas Telkom, 2018.



## ANALISIS DAN PENGEMBANGAN SISTEM RESERVASI ONLINE UNTUK WISATA BERBASIS WEBSITE STUDI KASUS PULAU HARAPAN KEPULAUAN SERIBU

Rian Hardiansyah<sup>1</sup>, Sirojul Munir<sup>2</sup>

<sup>1,2</sup>Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri  
Jakarta Selatan, DKI Jakarta, Indonesia 12640  
[hardiansyahrian3@gmail.com](mailto:hardiansyahrian3@gmail.com), [rojulman@nurulfikri.ac.id](mailto:rojulman@nurulfikri.ac.id)

### Abstract

*The tourism sector is one of the largest and most influential industries in the world economy. One of the beach resorts in Indonesia that are often visited in DKI Jakarta is the Thousand Islands beach. The research designed and developed a web-based online travel reservation on Harapan Island, Thousand Islands. The development stages of this application include data collection, system analysis, application development using the waterfall method, and system design using UML diagrams. In the implementation of the program using PHP Framework Laravel and MySQL database. System functional testing uses Blackbox Testing, and system evaluation uses UAT (User Acceptance Test) and questionnaires by users. The operational test results show that the prototype application runs 100% with a 91.42% eligibility level, meaning that the Harapan Island travel reservation application is feasible to use.*

**Keywords:** Tourism, Harapan Island, Web Framework, Waterfall, UML

### Abstrak

Sektor pariwisata merupakan salah satu sektor terbesar dan terkuat dalam perekonomian dunia. Salah satu tempat wisata pantai di Indonesia yang sering dikunjungi di provinsi DKI Jakarta adalah pantai kepulauan seribu. Pada penelitian ini dilakukan rancang bangun aplikasi reservasi wisata *online* berbasis *web* studi kasus di Pulau Harapan, Kepulauan Seribu. Tahapan dalam pengembangan aplikasi ini meliputi pengumpulan data, analisis sistem, pengembangan aplikasi menggunakan metode *waterfall* dan perancangan sistem menggunakan diagram UML. Pada implementasi program menggunakan PHP Framework Laravel dan database MySQL. Pengujian fungsional sistem menggunakan *Blackbox Testing*, dan evaluasi sistem menggunakan UAT (*User Acceptance Test*) dan kuesioner oleh pengguna. Hasil pengujian fungsional menunjukkan *prototype* aplikasi berjalan 100% dengan tingkat kelayakan penggunaan 91.42%, artinya aplikasi reservasi wisata Pulau Harapan layak dan dapat digunakan.

**Kata kunci:** Pariwisata, Pulau Harapan, Web Framework, Waterfall, UML

### 1. PENDAHULUAN

Sektor pariwisata merupakan salah satu sektorterbekas dan terkuat dalam perekonomian dunia. Sektor pariwisata menjadi salah satu pendorongutama perekonomian dunia karena terdapat beberapa keuntungan yang mampu memberikan devisa cukup besar bagi negara, memperluas lapangan pekerjaan dan memperkenalkan budaya negara. Salah satu destinasi utama dari wisatawan mancanegara ke Indonesia adalah Pulau Bali [1]. Selain di Pulau Bali, masih banyak provinsi yang tidak kalah jauh dengan keindahan alam serta pantai yang begitu indah. Contohnya adalah provinsi DKI Jakarta. Salah satu tempat wisata pantai yang sering

dikunjungi di provinsi DKI Jakarta adalah pantai Kepulauan Seribu.

Pariwisata di Kepulauan Seribu terdiri dari 3 jenis wisata yang menjadi daya tarik dalam merespon motivasi wisatawan untuk datang. Ketiga jenis wisata ini adalah wisata pantai (pulau wisata umum) berjumlah 45 pulau, wisata cagar alam berjumlah 20 dan wisata sejarah berjumlah 4 pulau.

Salah satu pariwisata di Kepulauan Seribu adalah kawasan wisata Pulau Harapan. PulauHarapan merupakan salah satu pulau permukiman diKepulauan Seribu yang telah

mempersiapkan diri menjadi destinasi wisata baharinya. Letak geografis yang berada dikawasan utara, Pulau Harapan memiliki lingkungan laut yang indah dan permai karena lokasinya yang semakin menjauhi daratan Jakarta menuju laut lepas. Dari sini gugusan pulau- pulau kecil lainnya nampak berjejer rapih melengkapi keelokan pemandangan. Sebuah Taman terbuka juga terdapat pada pulau ini yang bermanfaat sebagai tempat berinteraksi tidak saja dengan pengunjung melainkan juga dengan masyarakat serta alam yang indah [2]. Akan tetapi, banyak wisatawan yang belum mengetahui tentang keberadaan Pulau Harapan, karena letak Pulau Harapan dalam peta Kepulauan Seribu (DKI Jakarta) terlihat sangat kecil.

Masalah di atas dapat diatasi dengan bantuan teknologi informasi. Dengan teknologi informasi data dapat dikelola dengan mudah, cepat dan akurat berkat kecanggihan komputer. Dengan adanya bantuan aplikasi (sistem informasi), data dapat digunakan sebagai informasi bahkan pengetahuan yang berguna bagi berbagai pihak yang berkepentingan. sistem informasi merupakan kesatuan elemen-elemen yang saling berinteraksi secara sistematis dan terstruktur untuk menciptakan dan membentuk aliran informasi yang mendukung pembuatan keputusan dan melakukan kontrol terhadap jalannya suatu organisasi/institusi [3].

Teknologi informasi yang bisa diterapkan pada kasus diatas adalah teknologi dalam bentuk *website*. Dengan adanya bantuan *website* dapat memudahkan penulis dalam mendapatkan informasi, memberikan informasi, serta membuat aplikasi secara cepat dan mudah. *Website* atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman [4].

Dalam pengembangan website yang kompleks dibutuhkan kerangka kerja aplikasi (*framework*) untuk kemudahan pengembangan. Konsep yang digunakan oleh *framework* yaitu dengan menggunakan konsep MVC (*Model View Controller*). MVC adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari persentasi. MVC memisahkan aplikasi berdasarkan komponen-komponen aplikasi, seperti manipulasi data, *controller* dan *user interface* [5].

Dengan latar belakang diatas penulis melakukan penelitian untuk merancang bangun aplikasi reservasi wisata berbasis *web* menggunakan MVC *framework* studi kasus wisata Pulau Harapan Kepulauan Seribu.

## 1.1 Rumusan Masalah

1. Bagaimana merancang dan membangun aplikasi berbasis *web* untuk reservasi pariwisata di pulau harapan kepulauan seribu?
2. Fitur-fitur apakah yang dapat diterapkan dalam pembuatan aplikasi reservasi pariwisata di Pulau Harapan Kepulauan Seribu?

## 1.2 Tujuan dan Manfaat

1. Membuat aplikasi berbasis web untuk reservasi pariwisata di Kepulauan Seribu kelurahan Pulau Harapan menggunakan MVC *Framework*.
2. Memberikan informasi wisata di Pulau Harapan kepada wisatawan.
3. Memudahkan wisatawan dalam pencarian tempat objek wisata.
4. Menghasilkan fitur-fitur yang berguna dan bermanfaat untuk wisatawan.

## 1.3 Batasan Masalah

1. Pembuatan rancang bangun *web* pariwisata di Kepulauan Seribu kelurahan Pulau Harapan hanya menggunakan konsep MVC *Framework* yaitu laravel 5.7.
2. Pembuatan aplikasi ini ditujukan hanya di Kepulauan Seribu kelurahan Pulau Harapan.
3. Pengembangan aplikasi tidak termasuk fitur untuk pembayaran.

## 2. LANDASAN TEORI

### 2.1 Pariwisata

Menurut definisi yang luas pariwisata adalah perjalanan dari satu tempat ke tempat lain, bersifat sementara, dilakukan perorangan maupun kelompok, sebagai usaha mencari keseimbangan atau keserasian dan kebahagiaan dengan lingkungan hidup dalam dimensi sosial, budaya, alam, dan ilmu [6].

### 2.2 Pulau Harapan

Tabel 1. Jumlah Penduduk Tiap Pemukiman

No	Nama Pulau	Luas (HA)	Penduduk		Jumlah	Penduduk		Jumlah
			Lk	Pr		Lk	Pr	
1	P. Harapan	6,70	927	957	1.884	403	59	462
2	P. Sabira	8,82	271	283	554	122	17	139
	Jumlah	15,52	1.198	1.240	2.438	525	76	601

(sumber: Kantor Kelurahan Pulau Harapan)

Menurut data yang penulis dapat dari kelurahan Pulau Harapan pada Tabel 1, dijelaskan bahwa luas Pulau Harapan 6.70 hektar. Jumlah penduduk Pulau Harapan yaitu sebesar 1884 orang, diantaranya jumlah laki-laki sebesar 927 orang dan jumlah perempuan sebesar 957 orang. Kemudian jumlah kartu keluarga di Pulau Harapan sebesar 462, diantaranya 403 kartu keluarga laki-laki dan 52 kartu keluarga perempuan.

### 2.3 Framework Laravel

Framework Laravel adalah sebuah kerangka kerja open source yang diciptakan oleh Taylor Otwell. Laravel merupakan sebuah *framework bundle*, migrasi dan artisan CLI (*Command Line Interface*) yang menawarkan seperangkat alat dan arsitektur aplikasi yang menggabungkan banyak fitur terbaik dari kerangka kerja seperti *CodeIgniter*, *YII*, *ASP NET MVC*, *Ruby on Rails*, *Sinatra* dan lain-lain. Laravel memiliki fitur yang akan meningkatkan kecepatan pengembangan *web* [5].

### 2.4 Alur Kerja MVC pada Laravel

MVC adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari persentasi. MVC memisahkan aplikasi berdasarkan komponen-komponen aplikasi, seperti manipulasi data, *controller*, dan *user interface* [5].

### 2.5 MySQL

MySQL adalah salah satu jenis database server yang sangat terkenal. MySQL menggunakan bahasa SQL untuk mengakses *database*-nya. Lisensi Mysql adalah *FOSS License Exception* dan ada juga yang versi komersial nya. Tag Mysql adalah “*The World's most popular open source database*”. MySQL tersedia untuk beberapa platform, di antaranya adalah untuk versi windows dan versi linux. Untuk melakukan administrasi secara lebih mudah terhadap MySQL, anda dapat menggunakan *software* tertentu, di antaranya adalah *PHPMyAdmin* dan *MySQL* [7].

### 2.6 Waterfall

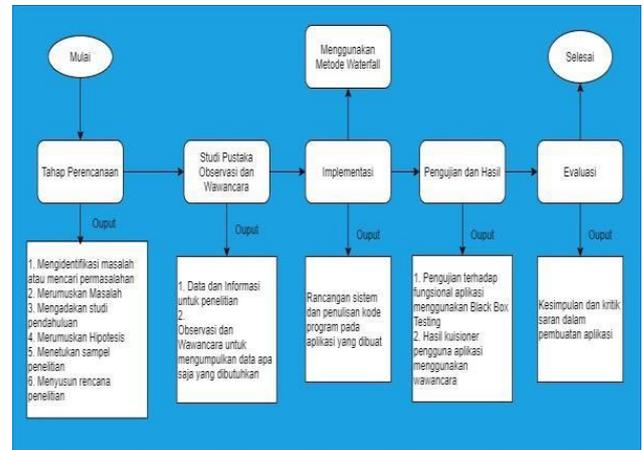
*Waterfall* merupakan merupakan model pengembangan sistem informasi yang sistematis dan sekuensial dengan tujuan yang berbeda untuk setiap fase pengembangannya. Setelah salah satu fase selesai, maka berlanjut ke fase berikutnya dan tidak bisa melakukan revisi ke fase yang sudah di lakukan [8]. Di bawah ini adalah 5 tahapan yang dimiliki oleh metode *waterfall*:

1. *Requirements Definition*
2. *System and Software Design*
3. *Implementation and Until Testing*
4. *Integration and System Testing*
5. *Operation and Maintenance*

## 3. METODE PENELITIAN

### 3.1 Tahapan Penelitian

Pada tahapan penelitian ini penulis akan membuat alur pengerjaan untuk mengembangkan aplikasi. Berikut ini contoh gambar pada tahapan penelitian:



Gambar 1. Tahapan Penelitian

Berikut ini penjelasan dari tahapan penelitian di atas yaitu sebagai berikut:

1. Tahapan Perencanaan  
Tahap perencanaan bertujuan untuk membantu penulis dalam menyelesaikan permasalahan yang akan diteliti.
2. Studi Pustaka Observasi dan Wawancara  
Studi pustaka bertujuan mengumpulkan data dan informasi yang berguna untuk membantu penulis dalam melakukan penelitian yang sedang dilakukan. Data dan informasi berasal dari buku, karya ilmiah maupun jurnal yang ada di perpustakaan ataupun yang bersumber dari internet. Observasi di lapangan bertujuan untuk mengumpulkan data dan informasi yang ada di pulau harapan. Wawancara bertujuan untuk menilai tingkat kepuasan wisatawan dalam berkunjung.
3. Implementasi  
Tahapan implementasi adalah tahapan untuk melakukan suatu perancangan sistem dan fitur-fitur pada aplikasi. Tahapan implementasi dapat membantu dalam melakukan penulisan kode program pada aplikasi yang ingin dibuat. Contohnya menggunakan metode *waterfall*. Dalam melakukan pengembangan aplikasi reservasi wisata berbasis *web* menggunakan *MVC framework* studi kasus Pulau Harapan Kepulauan Seribu, penulis menggunakan metode *Waterfall*, karena tahapan-tahapan dalam metode tersebut mudah untuk dimengerti.
4. Pengujian Hasil  
Tahapan ini merupakan pengujian aplikasi, yang bertujuan untuk mengetahui fungsional aplikasi

menggunakan *blackbox testing*. Sehingga didapatkan sebuah hasil yang jelas apakah aplikasi sudah sesuai dengan yang diinginkan atau masih ada beberapa fitur aplikasi yang belum berfungsi.

## 5. Evaluasi

Tahapan evaluasi bertujuan untuk mengetahui seberapa baik dan buruk aplikasi yang telah dibuat berdasarkan pada tahapan pengujian dan hasil. Tahapan evaluasi akan menjadi tahapan perbaikan dari aplikasi yang telah dikerjakan. Sehingga aplikasi tersebut dapat dikembangkan pada masa yang akan datang.

## 3.2 Rancangan Penelitian

### 3.2.1 Metode Pengembangan

Metode perencanaan aplikasi yang penulis gunakan adalah metode *waterfall* yang merupakan salah satu metode dalam SDLC (*Sistem Development Live Cycle*). Ciri khas pengerjaan dari metode *waterfall* adalah setiap fase dalam metode *waterfall* harus diselesaikan terlebih dahulu sebelum melanjutkan ke fase selanjutnya.

### 3.2.2 Metode Penelitian

Dalam metodologi penelitian yang penulis lakukan menggunakan metodologi kualitatif dalam pembuatan aplikasi yang penulis buat. Penelitian kualitatif merupakan penelitian tentang riset yang bersifat deskriptif dan cenderung menggunakan analisis. Proses dan makna (perspektif subjek) lebih ditonjolkan dalam penelitian kualitatif. Penelitian kualitatif memfokuskan mengumpulkan data di awal penelitian. Alasan penulis menggunakan metodologi kualitatif karena sangat efektif digunakan untuk landasan teori yang sesuai dengan fakta dan adanya pemahaman khusus dalam menganalisa.

### 3.2.3 Metode Pengumpulan Data

#### a. Studi Pustaka

Dalam penelitian ini penulis menggunakan sumber data yang berasal dari beberapa buku, jurnal, skripsi, tesis dan literatur untuk mengumpulkan data dan informasi yang dapat dijadikan landasan teori serta pembahasan dalam masalah yang dihadapi pada penelitian ini. Sumber data berasal dari internet dan perpustakaan. Penelitian ini berkaitan mengenai MVC *framework*, *website travel*, dan metode pengembangan aplikasi serta berbagai hal yang berkaitan dengan penelitian ini.

#### b. Studi Lapangan

##### 1. Observasi

Metode pertama yang digunakan oleh penulis dalam studi lapangan yaitu observasi. Dalam metode observasi dilakukan penelitian dan peninjauan langsung di lapangan agar dapat

mengumpulkan data dan informasi yang dibutuhkan. Hasil dari observasi yang penulis dapatkan yaitu bahwa para wisatawan masih kesulitan dalam mencari tempat penginapan dalam berkunjung serta kapal laut untuk menjelajah pulau-pulau. Maka dari itu penulis membuat rancang bangun aplikasi reservasi wisata *online* menggunakan *web framework* Laravel studi kasus: wisata Pulau Harapan Kepulauan Seribu. Dalam aplikasi ini memiliki sistem pemesanan online yang dapat membantu para wisatawan untuk mencari tempat penginapan dan hal apapun yang berhubungan dengan objek wisata di pulau harapan (paket wisata pulau harapan).

##### 2. Wawancara

Metode kedua yang digunakan penulis yaitu metode wawancara. Wawancara dilakukan dengan mewawancarai langsung pihak-pihak yang terkait dengan penelitian yang dilakukan. Tujuannya untuk mendapatkan data dan informasi agar membantu dalam penelitian ini. Wawancara dilakukan terhadap pengunjung atau wisatawan yang sedang berkunjung di Pulau Harapan. Tujuannya yaitu mengetahui tingkat kepuasan para wisatawan dalam berkunjung. Penulis mewawancarai 10 orang wisatawan dari Indonesia yang bersedia melakukan wawancara. Penulis melakukan wawancara pada saat acara *Snorkeling*, *BBQ*, dan sebelum wisatawan pulang ke pelabuhan Kali Adem.

## 4. ANALISIS DAN PERANCANGAN

### 4.1 Analisis Sistem

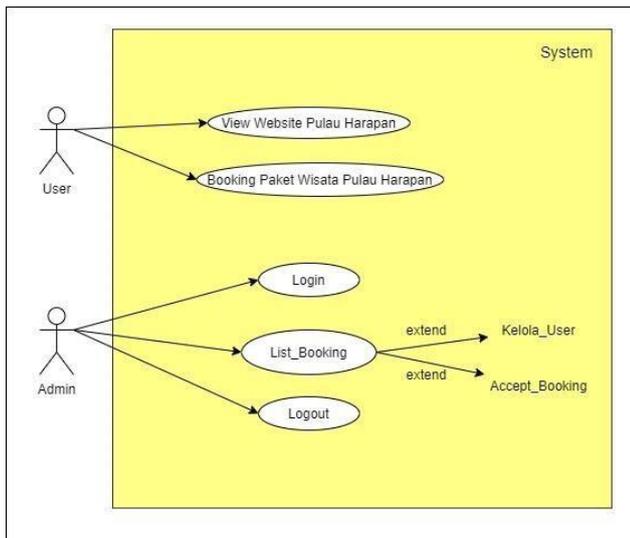
Analisis sistem adalah suatu tahapan pengembangan yang dilakukan untuk menganalisa sistem yang sedang berjalan saat ini, yang dimana bertujuan untuk mengetahui kebutuhan sistem dalam pembuatan aplikasi. Sehingga memudahkan penulis pada tahapan selanjutnya yaitu perancangan sistem. Adapun tahapan dalam analisis sistem terbagi menjadi dua tahapan, yaitu analisis sistem berjalan dan analisis kebutuhan sistem pemesanan online.

### 4.2 Analisis Sistem Berjalan

Di dalam analisis kebutuhan sistem pemesanan *online* penulis bagi menjadi dua bagian, diantaranya yaitu analisis fungsional dan analisis non-fungsional. Analisis fungsional yaitu kebutuhan yang berisi proses-proses apa saja atau layanan apa saja yang nantinya harus disediakan oleh sistem, mencakup bagaimana aplikasi nantinya merespon input dan *output* dari *user*.

### 4.3 Analisis Kebutuhan Sistem

Pada aplikasi ini memiliki 2 aktor, yaitu: Admin (*Travel*) dan *User* (Wisatawan).



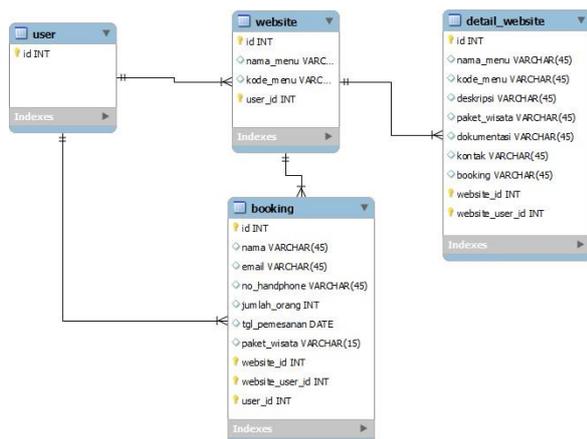
Gambar 2. Use Case Diagram

4.4 Perancangan Sistem

Desain sistem yang penulis gunakan pada penelitian ini adalah *domain model*, *sequence diagram*, dan *class diagram*. Pada tahap pengembangan desain ini dibuat berdasarkan sudut pandang kebutuhan sistem. Desain sistem yang dirancang adalah sebagai berikut:

A. Domain Model

*Domain Model* adalah penggambaran objek dari suatu lingkup area atau domain melalui perwujudan ide abstraksi kelas (*conceptual class*) yang mempunyai relasi antar objek. Berikut ini adalah domain model yang telah penulis buat untuk menggambarkan relasi antar objek yang terdapat pada sistem pemesanan *online* wisata Pulau Harapan.

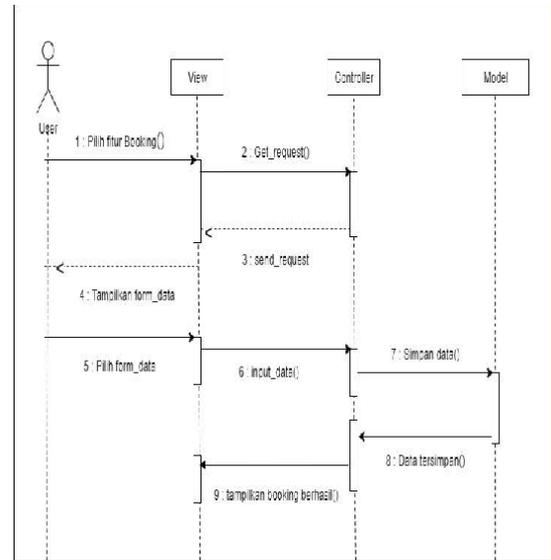


Gambar 3. ERD Aplikasi Website

B. Sequence Diagram

*Sequence Diagram* adalah diagram yang menggambarkan kolaborasi dinamis antara sejumlah objek. *Sequence diagram* bertujuan untuk menunjukkan rangkaian pesan yang dikirim antara objek dan juga interaksi antara objek. *Sequence*

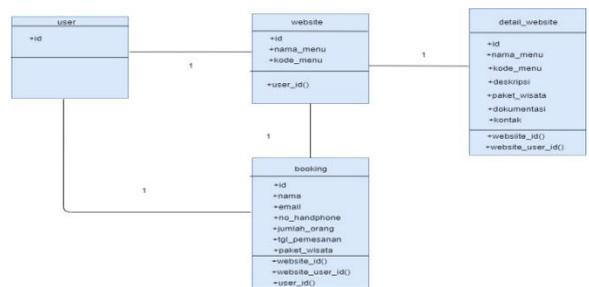
*diagram* memiliki komponen utama yang terdiri atas objek yang digambarkan dengan kotak persegi yang memiliki nama, pesan (*message*) yang digambarkan dengan garis tanda panah, dan waktu yang di tunjukan dengan garis terputus vertikal. Berikut ini gambaran *sequence diagram* yang penulis buat sesuai dengan konsep MVC (*model, view, and controller*).



Gambar 4. Sequence Diagram Booking

C. Class Diagram

*Class Diagram* adalah diagram UML yang menggambarkan kelas-kelas dalam sebuah sistem dan hubungannya antara satu dengan yang lain, serta dimasukkan pula atribut dan operasi (*method*). *Class diagram* bertujuan untuk memberikan gambaran tentang sistem dan relasi antara objek kepada *user*. Berikut ini adalah *class diagram* dari aplikasi pemesanan online paket wisata Pulau Harapan.



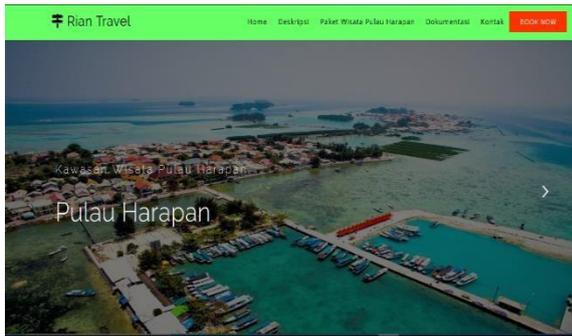
Gambar 5. Class Diagram

5. IMPLEMENTASI DAN EVALUASI

5.1 Implementasi Antar Muka Sistem

Implementasi antar muka sistem adalah tahapan dimana penulis akan menampilkan antar muka sistem dari aplikasi reservasi wisata Pulau Harapan yang mencakup tampilan untuk user dan tampilan untuk admin.

A. Halaman Utama



Gambar 6. Halaman Utama

B. Halaman Booking



Gambar 7. Halaman Booking

C. Halaman Login Admin



Gambar 8. Halaman Login Admin

5.2 Evaluasi Sistem

Evaluasi sistem adalah tahap dimana penulis melakukan pengujian terhadap aplikasi. Pada tahap ini penulis juga akan melakukan evaluasi terkait kesalahan dan kekurangan pada saat proses pengujian aplikasi.

A. Hasil Pengujian Black Box Testing

Black box testing merupakan pengujian yang berpusat pada kebutuhan fungsional perangkat lunak dimana memungkinkan untuk memperoleh sekumpulan kondisi input yang secara penuh memeriksa fungsional dari sebuah aplikasi. Black box testing nantinya akan diuji oleh pengembang aplikasi.

Tabel 2. Hasil Pengujian Black Box Testing

No.	Pengujian	Deskripsi Uji	Hasil Uji	Keterangan
1	Login Admin	Proses validasi akun admin	Berhasil	Admin berhasil login ke dalam aplikasi
2	List Booking	Menguji fungsionalitas	Berhasil	Admin berhasil melihat list booking
3	Kelola User	Menguji fungsionalitas tombol create, read, update, dan delete	Berhasil	Admin berhasil melakukan proses create, read, update, dan delete
4	View Menu Website Pulau Harapan	Menguji fungsionalitas	Berhasil	User berhasil melihat menu aplikasi
5	Booking Paket Wisata Pulau Harapan	Menguji fungsionalitas	Berhasil	User berhasil melakukan booking paket wisata Pulau Harapan
6	Accept Booking	Menguji fungsionalitas	Berhasil	Admin berhasil menerima booking paket wisata Pulau Harapan
7	Logout Admin	Menguji fungsionalitas tombol logout	Berhasil	Admin berhasil keluar aplikasi

Dari fungsionalitas aplikasi yang telah diuji pada Tabel 2 dengan black box testing didapatkan hasil bahwa 7 dari 7 fungsionalitas aplikasi berhasil. Artinya persentase keberhasilan aplikasi dapat berjalan sesuai fungsinya adalah sebesar 100%.

B. Hasil Pengujian UAT

1. Pengujian UAT User (Wisatawan)

Tabel 3. Hasil Pengujian UAT User

No	Pengujian	Url	Deskripsi	Hasil	Catatan
1	View Menu Website Pulau Harapan	http://website-pulau-harapan.test	User melihat tampilan Utama aplikasi	Berhasil	Aplikasi menampilkan halaman utama ke user

No	Pengujian	Url	Deskripsi	Hasil	Catatan
2	Booking Paket Wisata Pulau Harapan	http://website-pulau-harapan.test/booking/create	User booking paket wisata Pulau Harapan	Berhasil	User berhasil melakukan booking paket wisata Pulau Harapan dengan akun yang sudah terdaftar pada aplikasi

Pada tabel 3 dari hasil pengujian UAT kepada *user* didapatkan hasil bahwa 2 dari 2 fungsionalitas yang telah diuji adalah berhasil. Dengan hasil ini artinya keberhasilan fungsionalitas aplikasi pada sisi *user* memiliki persentase sebesar 100%.

## 2. Pengujian UAT Admin (Travel)

**Tabel 4.** Hasil Pengujian UAT Admin

No	Pengujian	Url	Deskripsi	Hasil	Catatan
1	Login Admin	http://website-pulau-harapan.test/login	Admin login pada aplikasi	Berhasil	Admin berhasil login ke dalam dashboard aplikasi
2	List Booking	http://website-pulau-harapan.test/booking	Admin melihat list booking	Berhasil	Admin berhasil melihat list booking
3	Kelola User	http://website-pulau-harapan.test/booking	Admin mengelola user	Berhasil	Admin berhasil melakukan create, read, update, dan delete pada user
4	Accept Booking	http://website-pulau-harapan.test/booking/edit	Admin menerima booking paket wisata Pulau Harapan dari user	Berhasil	Semua booking paket wisata Pulau Harapan dari user yang sudah login berhasil diterima oleh Admin
5	Logout Admin	Click logout	Admin keluar dari dashboard aplikasi	Berhasil	Admin berhasil keluar dari aplikasi

Pada tabel 4 dari hasil pengujian UAT kepada admin didapatkan hasil bahwa 5 dari 5 fungsionalitas yang telah diuji adalah berhasil. Dengan hasil uji ini artinya keberhasilan fungsionalitas aplikasi pada sisi admin berjalan 100%.

## C. Hasil Kuesioner

Penulis menggunakan kuesioner dalam mengumpulkan data karena ini adalah teknik yang paling banyak digunakan dalam *skala likert*. Kuesioner ini telah penulis berikan kepada *user* (10 wisatawan) untuk memperbaiki kekurangan-kekurangan yang ada pada aplikasi ini.

Hasil kuesioner yang penulis telah berikan kepada 10 responden didapatkan total skor sebesar 320. Maka dengan hasil ini didapatkan rata-rata keseluruhan sebesar 91.42 % dari total 100%. Kriteria interpretasi skornya berdasarkan interval adalah responden sangat setuju ketika mencoba aplikasi *web* ini.

## 6. PENUTUP

### 6.1 Kesimpulan

Kesimpulan yang dapat penulis dapatkan dalam penelitian terkait pembuatan aplikasi reservasi wisata Pulau Harapan berbasis *web* menggunakan MVC *framework* dengan menggunakan konsep MVC yaitu Laravel 5.7 adalah sebagai berikut:

- Pada rancang bangun aplikasi reservasi wisata Pulau Harapan berbasis web ini dilakukan dengan proses pengumpulan data menggunakan metode observasi dan wawancara. Dalam rancang bangun aplikasi ini penulis menggunakan metode *waterfall* untuk mengembangkan aplikasi reservasi wisata Pulau Harapan. Proses selanjutnya penulis menggunakan diagram UML untuk desain sistem aplikasi. Untuk implementasi aplikasi ini penulis menggunakan MVC *framework* PHP yaitu Laravel 5.7
- Pada rancang bangun aplikasi reservasi wisata Pulau Harapan berbasis web, terdapat fitur yang dapat diterapkan di dalam aplikasi yaitu fitur *booking* untuk user dan fitur *accept booking* untuk admin. Pada fitur *booking* terdapat form data yang harus di isi oleh *user*. Untuk melakukan *booking* paket wisata Pulau Harapan *user* tidak perlu *login* ke dalam aplikasi dan *user* bisa langsung *booking* dengan mengisi form terlebih dahulu. Setelah itu *user* berhasil *booking* paket wisata Pulau Harapan. Selanjutnya untuk fitur *accept booking*, admin melihat *list booking* dan kemudian *admin* menyetujui *booking* yang dilakukan oleh *user*. Selanjutnya *admin* berhasil menerima *booking* dari *user*.

- C. Berdasarkan hasil dari pengujian *black box* terhadap fungsionalitas aplikasi didapatkan persentase keberhasilan sebesar 100%. Untuk pengujian UAT didapatkan hasil 100% pada sisi admin dan 100% pada sisi *user*. Sedangkan untuk hasil kuesioner yang telah diberikan kepada 10 responden terkait kelayakan aplikasi, didapatkan hasil sebesar 91,42%. Hal ini dapat disimpulkan bahwa aplikasi Reservasi Wisata Pulau Harapan sangat layak dan bermanfaat bagi wisatawan dan *agent travel*.

## 6.2 Saran

Adapun saran-saran yang penulis dapatkan dari penelitian ini adalah sebagai berikut:

- A. Pada penelitian yang dibuat oleh penulis yaitu aplikasi reservasi wisata Pulau Harapan masih berbasis *web* dan belum mempunyai versi *mobile*. Penulis berharap ke depannya dapat mengembangkan aplikasi ini ke dalam versi Android.
- B. Tampilan aplikasi reservasi wisata Pulau Harapan masih ada beberapa orang yang mengatakan kurang menarik. Penulis berharap ke depannya dapat memperbaiki tampilan aplikasi agar terlihat menarik di lihat orang lain.
- C. Saat ini aplikasi reservasi wisata Pulau Harapan belum ada fitur untuk melakukan pembayaran. Ke depannya dapat menambahkan fitur untuk melakukan pembayaran di dalam aplikasi.
- D. Berdasarkan semua saran yang diberikan kepada penulis, aplikasi reservasi wisata Pulau Harapan ke depannya dapat memperbaiki segala kekurangan yang ada. Terutama untuk memaksimalkan tampilan *frontend* aplikasi dan memaksimalkan fitur-fitur yang tersedia pada aplikasi.

## DAFTAR PUSTAKA

- [1] M T. P. P. C. S. K. W. C. D. P. Victoria Lelu Sabon, "Strategi Peningkatan Kinerja Sektor Pariwisata Indonesia pada *Asean Economic Community*," *Jurnal Bisnis dan Manajemen*, vol. Volume 8 (2), p. 164, 2018
- [2] A. Razak and R. Suprihardjo, "Pengembangan Kawasan Pariwisata Terpadu di Kepulauan Seribu," *JURNAL TEKNIK POMITS*, vol. 2, 2013
- [3] M. D. Ajie, "Pengertian Sistem Informasi Manajemen," p. 8, 26 November 1996
- [4] R. Hidayat, "Website dan Perkembangannya," in *Cara Praktis Membangun Website Gratis Memanfaatkan Layanan Domain dan Hosting Gratis*, Jakarta, PT Elex Media Komputindo, 2010, p. 2.
- [5] I. G. Handika and A. Purbasari, "Pemanfaatan Framework Laravel dalam Pembangunan Aplikasi *E-Travel* berbasis *Website*," *Konferensi Nasional Sistem Informasi 2018 STMIK Atma Luhur Pangkalpinang*, pp. 1329- 1334, 2018.
- [6] D. J. J. S. SY, "Menenal Pariwisata," in *Ekonomi Pariwisata Sejarah dan Prospeknya*, pp. 15-22.
- [7] A. Sofwan, "*Database MySQL dan Phpmyadmin*," *Belajar MySQL dengan Phpmyadmin*, p. 2, 2007.
- [8] R. Pressman, *Rekayasa Perangkat Lunak: Pendekatan Praktisi (Buku Kedua)*, Yogyakarta: Andi, 2002.

# Jurnal Informatika Terpadu

Vol. 6 No. 1 Tahun 2020

## Daftar Isi

<b>Analisis dan Pengembangan Sistem Desain Asset untuk Keramik Lantai berbasis Aplikasi <i>Website</i></b>	<b>01</b>
Alvino Gartner, Sirojul Munir	
<b>Visualisasi Implementasi Autentifikasi <i>Digital Signature</i> menggunakan Fungsi Hash pada File Digital</b>	<b>07</b>
Rufaidah Taqiyyah, Ahmad Rio Adriansyah	
<b>Analisis dan Evaluasi Algoritma <i>MapReduce WordCount</i> pada <i>Cluster Hadoop</i> menggunakan Indikator Kecepatan</b>	<b>14</b>
Robiyatul Adawiyah, Sirojul Munir	
<b>Uji Performa dan Perbandingan RDBMS MySQL dan <i>HIVE -Hadoop</i></b>	<b>20</b>
Nurul Azizah, Henry Saptono	
<b>Rancang Bangun Sistem Informasi Pengelolaan <i>Talent Film</i> berbasis Aplikasi <i>Web</i></b>	<b>29</b>
Atikah Permata Sari, Suhendi	
<b>Analisis dan Penerapan <i>Enterprise Resource Planning (ERP)</i> Odoo 10 guna membuat Laporan Keuangan pada Yayasan Darul Jannah</b>	<b>38</b>
Rosy Annisa Firlana, Suhendi	
<b>Analisis dan Pengembangan Sistem Reservasi <i>Online</i> untuk Wisata berbasis <i>Website</i> Studi Kasus Pulau Harapan Kepulauan Seribu</b>	<b>45</b>
Rian Hardiansyah, Sirojul Munir	

### Published by:

LPPM STT Terpadu Nurul Fikri

Jln. Raya Lenteng Agung, no. 20, Srengseng Sawah,

Jagakarsa, Jakarta Selatan, DKI Jakarta 12640

Telp. 021 - 786 3191

Email : [lppm@nurulfikri.ac.id](mailto:lppm@nurulfikri.ac.id)

Website : <https://journal.nurulfikri.ac.id/index.php/jit>