

E-ISSN: 2460-8998

Jurnal Informatika Terpadu

Vol. 8 No. 1 Tahun 2022



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

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

Daftar Isi

Perancangan Sistem Pengelola Inventaris berbasis Web menggunakan <i>Framework MVC</i> Annisa Rinjani, Sirojul Munir	01
Perancangan dan Implementasi <i>Booking System</i> Lapangan menggunakan <i>Framework MVC</i> berbasis Web Haya Rasikhah, Ahmad Rio Adriansyah	08
Analisis dan Implementasi Modul Akuntansi pada Aplikasi ERP berbasis <i>Open Source</i> Fadilla Arfikriyana, Suhendi	13
Penerapan <i>Log Analyzer</i> untuk Mengetahui Lalu Lintas Jaringan berbasis <i>Elasticsearch, Logstash, dan Kibana</i> Muhammad Jafier Rama Putra, Henry Saptono	21
Analisis dan Perancangan Aplikasi <i>Mobile</i> untuk Donasi menggunakan Metode <i>Hybrid</i> berbasis React Native Muhammad Abdul Karim, Ahmad Rio Adriansyah	26
Analisis dan Perancangan Aplikasi Penganggaran Barang berbasis Web pada Unit Sarana Prasarana Perguruan Tinggi Siti Zahra Maharani, Ahmad Rio Adriansyah	35
Pengembangan Permainan Edukasi guna Pengenalan Budaya Tak Benda di Indonesia berbasis <i>Mobile</i> Setyo Nugroho, Ahmad Rio Adriansyah	41
Analisis dan Implementasi RESTful API guna Pengembangan Sistem Informasi Akademik pada Perguruan Tinggi Mohammad Akmaluddin Novianto, Sirojul Munir	47



PERANCANGAN SISTEM PENGELOLA INVENTARIS BERBASIS WEB MENGUNAKAN FRAMEWORK MVC

Annisa Rinjani¹, Sirojul Munir²

^{1,2}Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jakarta Selatan, DKI Jakarta, Indonesia 12640

annisarinjani@student.nurulfikri.ac.id, rojulman@nurulfikri.ac.id

Abstract

Inventory Management System is an activity to record and borrow goods that runs manually at PT. Lima Titik Satu. A web-based inventory management system makes using a framework with the MVC method at PT. Lima Titik Satu to minimize the negative impact of a system still running manually. In building an inventory processing system, data was collected using the observation method, then analyzed and designed the system development using the waterfall method and UML diagrams to implement system development using the PHP framework Codeigniter with a MySQL database. The results of feature testing on the built system found that 93.21% of the inventory management system features can run well and use at PT. Lima Titik Satu.

Keywords: *Inventory Management System, Waterfall, Code Igniter, Blackbox Testing, User Acceptance Testing*

Abstrak

Sistem Pengelolaan Inventaris adalah kegiatan mendata serta peminjaman barang yang berjalan secara manual di PT. Lima Titik Satu. Untuk meminimalisir dampak negatif dari sistem yang masih berjalan manual maka dibuatlah sistem pengelola inventaris berbasis *web* menggunakan *framework* dengan metode MVC di PT. Lima Titik Satu. Dalam membangun sistem pengolahan inventaris dilakukan pengumpulan data menggunakan metode observasi, lalu menganalisa dan mendesain pengembangan sistem menggunakan metode *waterfall* dan diagram UML. Pada implementasi pengembangan sistem menggunakan PHP *framework codeigniter* dengan *database* MySQL, Dari hasil pengujian fitur pada sistem yang dibangun didapat hasil bahwa 93,21% fitur *sistem pengelola inventaris* dapat berjalan dengan baik dan digunakan di PT. Lima Titik Satu.

Kata kunci: *Sistem Pengelola Inventaris, Waterfall, Codeigniter, Blackbox Testing, User Acceptance Testing*

1. PENDAHULUAN

Manfaat teknologi informasi pada saat ini tidak hanya dipergunakan untuk kepentingan organisasi saja, tetapi juga untuk kebutuhan individu. Bagi organisasi, teknologi informasi dapat digunakan untuk mencapai keunggulan kompetitif, sedangkan untuk kebutuhan individu, teknologi dipergunakan untuk kepentingan pribadi seperti mencari pekerjaan dan lain lain. Berdasarkan data Badan Pusat Statistik, teknologi informasi dimanfaatkan dalam berbagai aspek, seperti perusahaan yang mempunyai *website* yang memiliki tujuan tertentu sebagai kegiatan utama pada tahun 2017 didaerah perkotaan [1]. Pada penelitian ini penulis akan memanfaatkan teknologi informasi sebagai upaya meningkatkan kemudahan dalam pengelolaan manajemen organisasi dengan memanfaatkan teknologi dalam pengolahan data inventaris pada PT Lima Titik Satu yang masih bersifat manual agar meminimalisir

kesulitan dalam mencari data, seperti data barang dan peminjaman.

Penelitian ini bertujuan untuk menjawab masalah yang diangkat dalam penelitian ini yaitu “Bagaimana merancang dan membangun sistem inventaris berbasis *web* menggunakan *framework* Codeigniter di PT Lima Titik Satu dan bagaimana fiturnya berjalan?”. Penelitian ini memiliki manfaat menghasilkan suatu informasi yang cepat dan akurat dalam membantu dan memudahkan proses peminjaman barang inventaris milik perusahaan. Sementara batasan masalah dari penulisan ini adalah hanya membahas tentang merancang serta membangun aplikasi pendataan dan peminjaman inventaris yang dikhususkan untuk perusahaan PT Lima Titik Satu, serta tidak membahas terkait cakupan keamanan.

2. TINJAUAN PUSTAKA

2.1 Website

Website adalah “Sebuah sistem informasi yang disajikan dalam bentuk teks, gambar, suara, dan lain-lain yang tersimpan dalam sebuah server Web Internet yang disajikan dalam bentuk hiperteks. Informasi *Web* dalam bentuk teks umumnya ditulis dalam format HTML (*Hypertext Markup Language*). Informasi lainnya disajikan dalam bentuk grafis (dalam format GIF, JPG, PNG), suara (dalam format AU, WAV), dan objek multimedia lainnya (seperti MIDI, Shockwave, Quicktime Movie, 3D World) [2]. Pada bagian ini dapat diberikan penjelasan tentang teori 2 yang menjadi dasar penelitian. Teori yang dituliskan bersifat khusus [2].

2.2 Inventaris

Inventaris atau persediaan adalah bahan atau barang yang disimpan yang akan digunakan untuk memenuhi tujuan tertentu. Setiap perusahaan yang melakukan kegiatan usaha umumnya memiliki persediaan. Keberadaannya tidak saja dianggap sebagai beban karena merupakan pemborosan (*waste*), tetapi sekaligus juga dapat dianggap sebagai kekayaan (*asset*) yang dapat segera dicairkan dalam bentuk uang tunai (*cash*) [2].

2.3 Model Pengembangan

Penelitian ini menggunakan model pengembangan *Waterfall* yang memiliki tahapan analisis, desain, implementasi, pengujian serta pemeliharaan sistem [3].

2.4 Tools Pengembangan

Penelitian ini menggunakan beberapa *tools* yang digunakan antara lain:

1. PHP
Penelitian ini menggunakan bahasa pemrograman PHP. PHP adalah sebuah bahasa pemrograman yang berjalan dalam sebuah web server (*server side*) [4].
2. Framework MVC (*Model View Control*)
Penelitian ini menggunakan Framework MVC. MVC adalah sebuah pattern/teknik pemrograman yang memisahkan bisnis logic (alur pikir), data logic (penyimpanan data) dan *presentation logic* (antarmuka aplikasi) atau secara sederhana adalah memisahkan antara desain, data dan proses [5]. Pola Desain pengembangan perangkat lunak (design pattern) merupakan deskripsi dari class-class dan object-object yang saling berkomunikasi yang tersusun untuk memecahkan masalah perancangan secara umum pada sebuah konteks tertentu [12]. Salah satu implementasi aplikasi berbasis web yang menggunakan pola desain MVC adalah Codeigniter.
3. MySQL
Penelitian ini menggunakan MySQL sebagai database sistem. MySQL adalah sebuah perangkat lunak

sistem manajemen basis data SQL (*database management system*) atau DBMS yang *multiithread*, *multiuser*, dengan sekitar 6 juta instalasi diseluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaanya tidak cocok dengan penggunaan GPL [4].

2.5 Pengujian Sistem

Penelitian ini menggunakan 3 metode pengujian sistem, yaitu :

1. *Blackbox Testing* adalah metode dimana penguji atau tester hanya mengetahui apa yang harus dilakukan suatu *software*. Penguji tidak mengetahui bagaimana *software* tersebut beroperasi [6][13]. Menurut Vibnu.C [12], *black-box testing* adalah teknik pengujian yang dilakukan tanpa harus memiliki pengetahuan tentang internal kerja aplikasi. .
2. UAT adalah proses untuk mendapatkan konfirmasi bahwa sebuah sistem memenuhi persyaratan yang disepakati dan mengetes apakah semua fungsi dan fitur berjalan dengan baik [6].
3. Skala Likert adalah adalah pengukuran skala yang menggunakan beberapa butir pertanyaan untuk mengukur perilaku individu dengan merespon 5 titik pilihan pada setiap butir pertanyaan, yaitu sangat setuju, setuju, tidak memutuskan, tidak setuju, dan sangat tidak setuju [6].

2.6 Penelitian Terkait

Untuk menyusun penelitian ini penulis menggunakan beberapa penelitian yang berkaitan namun memiliki beberapa hal yang dapat dibandingkan

1. Analisis dan Implementasi Sistem Aplikasi *Inventory Items* Berbasis *Web* Menggunakan *Framework Codeigniter* [7].
2. Perancangan dan Implementasi Sistem *Inventory Sayur Organik* dengan Menggunakan *Framework Codeigniter* [8].
3. Sistem Informasi Inventori Barang PT. Tissan Nugraha Globalindo Berbasis *Web* [9].
4. Analisa dan Perancangan Aplikasi Sistem *Inventory* (Studi Kasus: PT. Cakra Medika Utama) [10].

3. METODE PENELITIAN

Penelitian ini menggunakan metode deskriptif yaitu dimana masalah yang diselidiki menggambarkan keadaan subyek atau objek dalam penelitian yang berupa orang, lembaga, masyarakat dan lainnya yang mendasari fakta-fakta saat ini tampak atau apa adanya.

3.1 Metode Pengumpulan Data

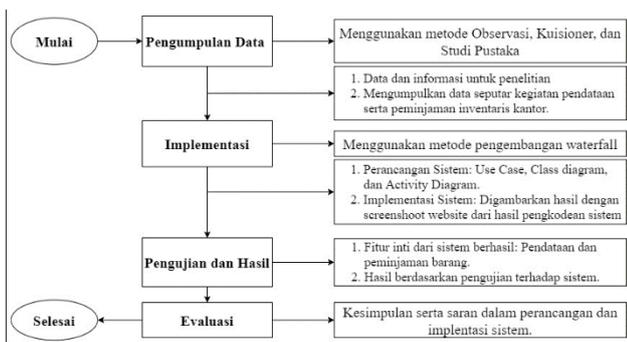
Dalam penelitian ini, ada tiga metode pengumpulan data yang digunakan yaitu:

1. Observasi : Melakukan analisis terhadap masalah yang ada dengan cara mengamati sumber dan mengumpulkan data yang berhubungan dengan data Inventaris pada PT. LIMA TITIK SATU.
2. Kuisisioner : Melakukan pengumpulan data dengan cara memberikan pertanyaan-pertanyaan dan melakukan tanya jawab menggunakan kuisisioner kepada bagian tester PT. LIMA TITIK SATU yang berhubungan dengan melakukan kegiatan peminjaman dan pengembalian device kantor.
3. Studi Pustaka : Studi pustaka dilakukan menggunakan cara membaca dan mempelajari berbagai sumber pustaka, seperti paper, buku, jurnal dan website yang berkaitan dengan penelitian, yang dapat mempermudah proses perancangan website ini. Adapun daftar referensi yang penulis gunakan untuk penulisan penelitian ini, dapat dilihat pada halaman daftar pustaka.

Penelitian ini menggunakan metode deskriptif yaitu dimana masalah yang diselidiki menggambarkan keadaan subyek atau objek dalam penelitian yang berupa orang, lembaga, masyarakat dan lainnya yang mendasari fakta-fakta saat ini tampak atau apa adanya [11].

3.2 Tahapan penelitian

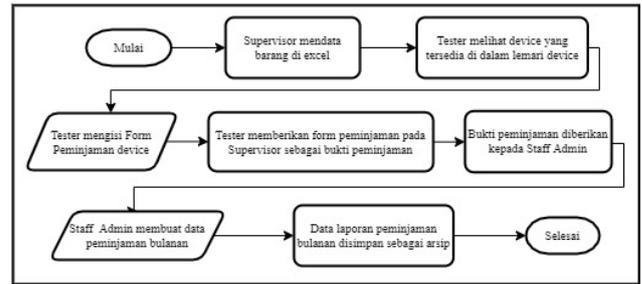
Tahapan penelitian ini yang secara umum menggambarkan proses dalam melakukan perancangan hingga pembuatan aplikasi inventaris kantor, dapat dilihat pada gambar di bawah ini:



Gambar 1. Tahap Penelitian

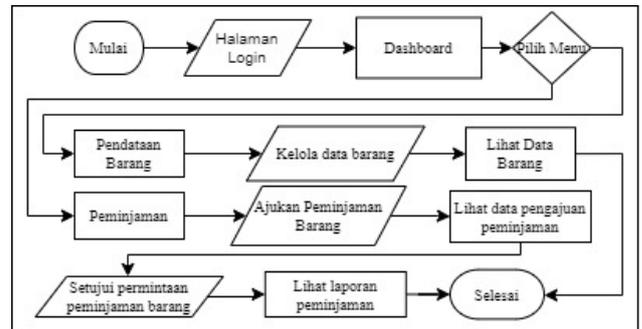
4. ANALISA DAN PERANCANGAN SISTEM

Berdasarkan observasi penulis PT. Lima Titik Satu dalam melakukan proses peminjaman inventaris hingga membuat laporan bulanan masih menggunakan sistem manual. Alur proses peminjaman manual inventaris di PT. Lima Titik Satu digambarkan pada gambar berikut:



Gambar 2. Alur Sistem Berjalan

Karena sistem manual memiliki dampak negatif, maka penulis merencanakan alur sebuah sistem yang dapat menghindari dampak akibat sistem manual, gambaran alur sistem tersebut dapat di lihat pada gambar di bawah ini:



Gambar 3. Alur Rancangan Sistem

Dari alur tersebut penulis dapat menjabarkan kebutuhan serta perancangan sistem yang akan di jabarkan berikut ini:

A. Kebutuhan Sistem

Dari hasil analisa kebutuhan sistem pengelola inventaris berbasis web penulis melalui observasi di PT. Lima Titik Satu penulis dapat menyimpulkan end user pada aplikasi tersebut, yaitu :

1. Super Admin
2. Supervisor
3. Tester
4. Staf Admin

Dari empat *user* tersebut memiliki masing-masing *requirement* yang dapat di uraikan pada tabel dibawah ini:

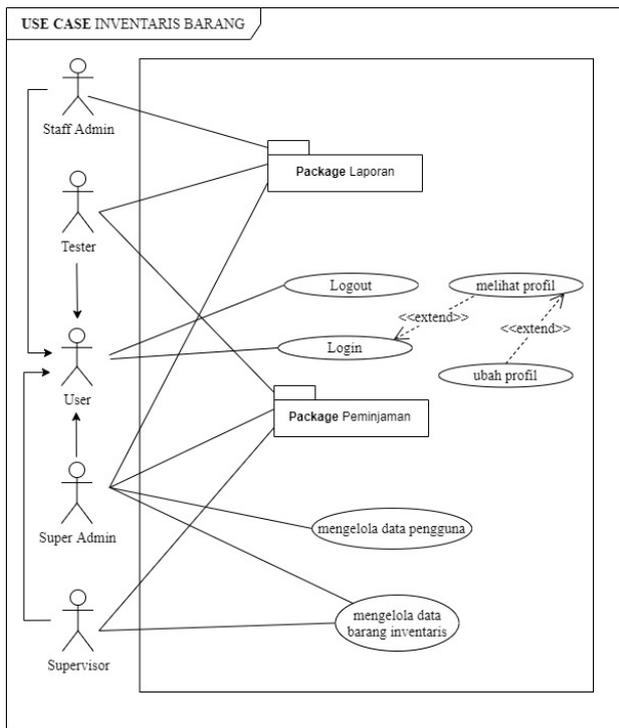
Tabel 1. User Requirement

Kode	Deskripsi
R. 001	Pengguna dapat login
R. 002	Super Admin dapat mengelola data pengguna
R. 003	Super Admin dan Staf Admin dapat melihat laporan peminjaman
R. 004	Super Admin dan Staf Admin dapat mencetak laporan peminjaman
R. 005	Super Admin, Supervisor dapat mengelola data barang inventaris

Kode	Deskripsi
R. 006	Super Admin, Supervisor dapat melihat permintaan pengajuan peminjaman barang inventaris
R. 007	Super Admin, Supervisor dapat menyetujui peminjaman barang inventaris
R. 008	Super Admin, Supervisor dapat menyelesaikan proses peminjaman
R. 009	Super Admin, Tester dan Supervisor dapat melihat profil
R. 010	Super Admin, Tester dan Supervisor dapat mengubah profil
R. 011	Super Admin, Tester dapat melihat data ketersediaan barang inventaris
R. 012	Super Admin, Tester dapat mengajukan peminjaman barang inventaris
R. 013	Super Admin, Tester dapat melihat hasil persetujuan pengajuan peminjaman barang inventaris
R. 014	Super Admin, Tester dapat melihat laporan peminjaman mandiri tester
R. 015	Pengguna dapat <i>logout</i>

B. Perancangan Sistem

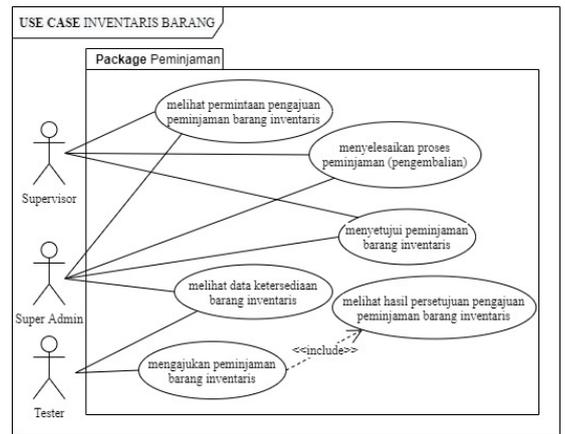
Dari *requirement* pada kebutuhan sistem dapat digambarkan sebuah *use case* seperti berikut ini:



Gambar 4. Use Case Diagram

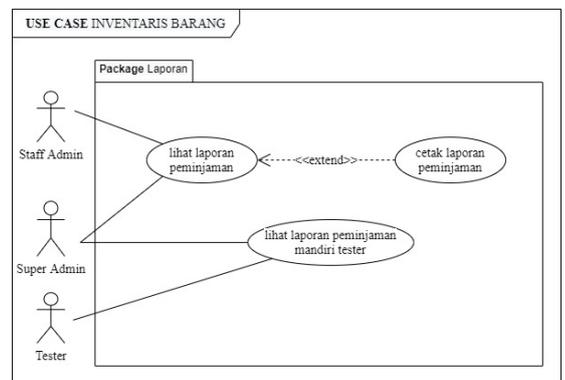
Pada *Use Case* tersebut terdapat dua *package diagram* yaitu:

• Package Peminjaman



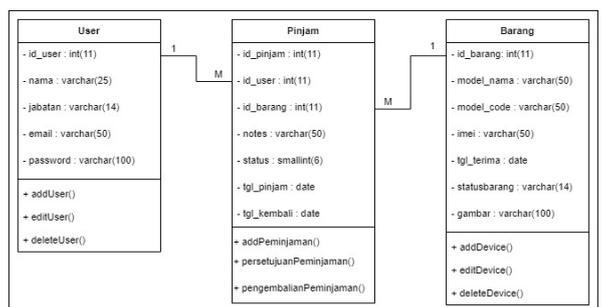
Gambar 5. Package Peminjaman

• Package Laporan



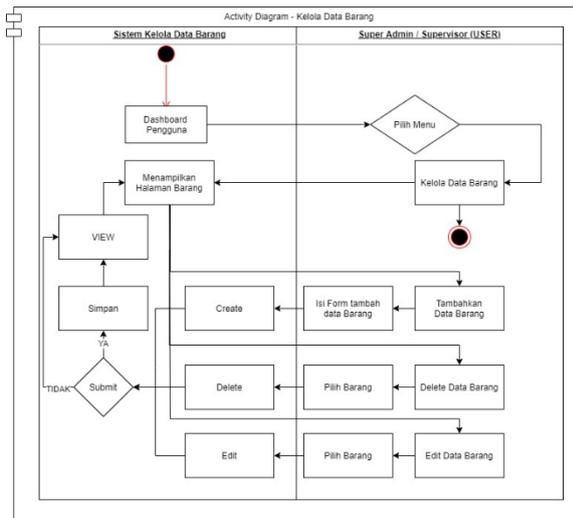
Gambar 6. Package Laporan

Relasi antara entitas yang ada pada sistem pengelolaan inventaris digambarkan pada gambar *Class Diagram* berikut ini:

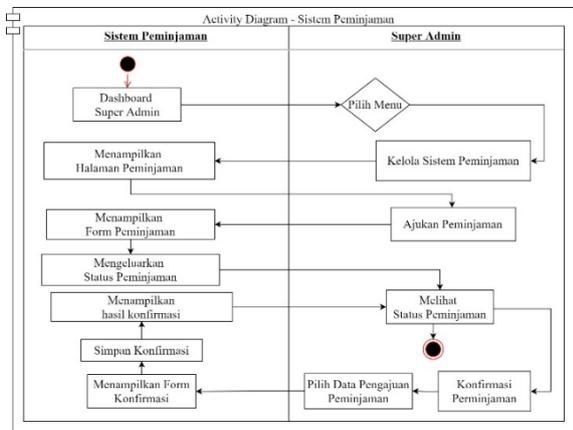


Gambar 7. Class Diagram

Activity Diagram yang menggambarkan proses inti sistem pengelola inventaris pada penelitian ini yaitu pendataan barang serta peminjaman barang dapat digambarkan sebagai berikut:

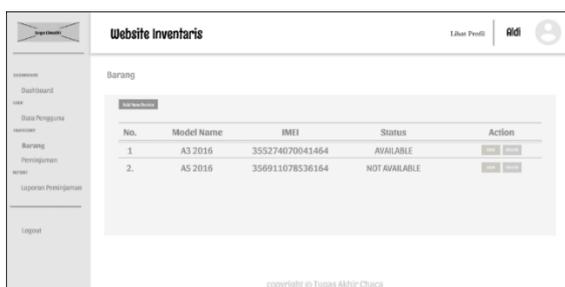


Gambar 8. Pendataan Barang

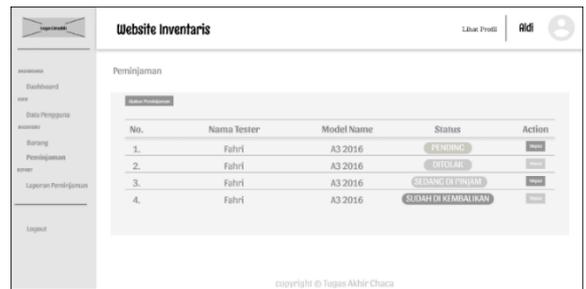


Gambar 9. Peminjaman Barang

Dari *activity diagram* tersebut dapat digambarkan pada rancangan tampilan antarmuka sebagai berikut ini:



Gambar 10. Rancangan Tampilan Antar Muka Pendataan Barang



Gambar 11. Rancangan Tampilan Antar Muka Peminjaman Barang

Pengujian sistem direncanakan menggunakan tiga metode pengujian *Blackbox Testing*, *User Acceptance Testing* dan *Skala Likert*. Untuk pengujian *User Acceptance Testing*, dan *Skala Likert* ditujukan untuk enam orang yaitu: empat orang *Tester*, satu *Supervisor* dan satu *Staff Admin*.

Pertanyaan terkait pengujian tersebut akan di uraikan pada tabel 2 dan 3:

- *Blackbox Testing*

Tabel 2. *Blackbox Testing*

No	Summary	Step	Expected Result	Fail / Pass
1	Kelola Data Barang	Melihat data barang	Pengguna dapat melihat data barang	
		Menambah data barang	Pengguna dapat menambah data barang	
		Membatalkan menambah data barang	Pengguna dapat membatalkan menambah data barang	
		Memilih data barang, lalu mengubah data barang	Pengguna dapat mengubah data barang	
		Membatalkan menambah data barang	Pengguna dapat membatalkan menambah data barang	
		Memilih data barang lalu menghapus data barang	Pengguna dapat menghapus data barang	
		Memilih data barang lalu melihat data gambar barang	Pengguna dapat melihat data gambar barang	

No	Summary	Step	Expected Result	Fail / Pass
2	Kelola Proses Peminjaman	Mengisi form pengajuan peminjaman lalu submit pengajuan. Mengisi form pengajuan peminjaman lalu submit pengajuan sebanyak dua kali. Melihat permintaan pengajuan peminjaman. Melihat permintaan pengajuan peminjaman, memilih data yang akan di konfirmasi lalu berikan konfirmasi pengajuan peminjaman. Menyelesaikan proses peminjaman. Melihat hasil persetujuan peminjaman.	Pengguna dapat mengajukan peminjaman barang inventaris Pengguna dapat mengajukan peminjaman barang inventaris lebih dari satu kali. Pengguna dapat melihat permintaan pengajuan peminjaman barang inventaris. Pengguna dapat menyetujui peminjaman barang inventaris. Pengguna dapat menyelesaikan proses peminjaman (pengembalian). Pengguna dapat melihat hasil persetujuan pengajuan peminjaman barang inventaris.	

• Skala Likert

Tabel 3. Skala Likert

No	Pertanyaan
1	Tampilan aplikasi inventaris berbasis <i>web</i> menarik.
2	Fitur-fitur dari aplikasi inventaris berbasis <i>web</i> mudah dipahami.
3	Fungsional dari aplikasi inventaris berbasis <i>web</i> berjalan dengan baik.
4	Fitur-fitur dari aplikasi inventaris berbasis <i>web</i> sudah sangat memudahkan pengguna dalam hal peminjaman inventaris.
5	Fitur-fitur dari aplikasi inventaris berbasis <i>web</i> untuk kedepannya perlu dilakukan pengembangan agar lebih baik lagi.

5. HASIL DAN PEMBAHASAN

Dalam implementasi sistem dari perancangan sistem yang telah dijabarkan pada pembahasan sebelumnya dibutuhkan

beberapa perangkat keras serta perangkat lunak untuk menjalankan sistem pengelolaan inventaris.

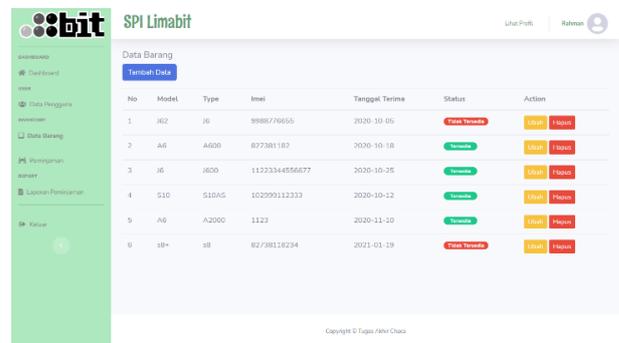
Adapun perangkat keras yang digunakan dalam implementasi aplikasi SPI adalah sebuah laptop dengan spesifikasi:

1. Model : Lenovo G40-45
2. Processor : AMD A8
3. RAM : 8 GB
4. Hard Disk : 500 GB

Perangkat lunak yang digunakan dalam implementasi aplikasi SPI diantaranya sebagai berikut :

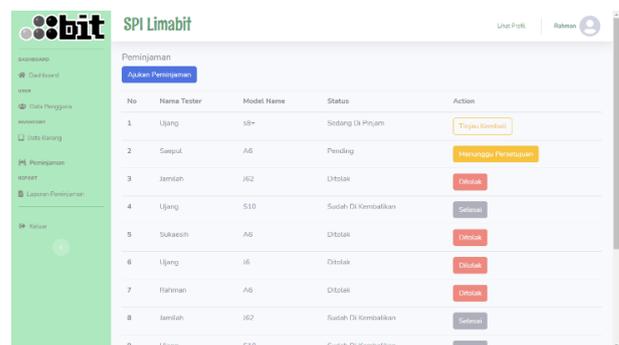
1. Google Chrome dengan versi 88.0.4324.104 digunakan untuk melihat hasil implementasi pada saat pengembangan.
2. Sublime Text dengan versi 3.2.2 digunakan untuk *text editor (coding)*.
3. Xampp dengan versi 3.2.4 digunakan server lokal untuk menyimpan database atau data website yang sedang dalam proses pengembangan.
4. Sistem Operasi yang digunakan adalah Windows 10 64bit.

Implementasi dari fitur inti tampilan antar muka dapat di tampilkan sebagai berikut:



Gambar 12. Implementasi Pendataan Barang

Halaman Data Barang terdapat pada pengguna tester, supervisor dan super admin yang di tampilkan sesuai kebutuhan tiap pengguna yaitu mengolah serta melihat data barang yang terdapat pada sistem pengelolaan inventaris.



Gambar 13. Implementasi Peminjaman Barang

Halaman Peminjaman terdapat pada pengguna tester, supervisor dan super admin yang di tampilkan sesuai kebutuhan tiap pengguna yaitu mengajukan dan meninjau proses peminjaman inventaris.

Hasil pengujian sistem menghasilkan:

- 89,65% fitur dinyatakan berjalan dengan baik pada pengujian *Blackbox Testing*
- 90% fitur dinyatakan berjalan dengan baik pada data kuisioner menggunakan *Skala Likert*.

6. KESIMPULAN

Sistem pengolahan inventaris dirancang dan dibangun dengan tahapan pengumpulan data menggunakan metode deskriptif dan desain pengembangan sistem menggunakan metode *waterfall* dan diagram UML. Pada tahapan implementasi pengembangan sistem menggunakan PHP *framework codeigniter* dengan *database MySQL*, serta nilai presentase hasil pengujian sistem pengelola inventaris dapat disimpulkan bahwa 93,21% fitur dapat berjalan dengan baik.

DAFTAR PUSTAKA

- [1] B. P. Statistik, "Penggunaan dan Pemanfaatan Teknologi Informasi dan Komunikasi Sektor Bisnis 2018," 2018.
- [2] E. Prasetyo, "Pemrograman Web PHP & MySQL," Yogyakarta: Graha Ilmu, 2008.
- [3] R. S. Pressman, "Rekayasa Perangkat Lunak Pendekatan Praktisi," Andi, 2002.
- [4] A. P. Basuki, "Membangun Web berbasis PHP dengan *Framework CodeIgniter*," Yogyakarta: Lokomedia, 2010.
- [5] I. Daqiqi, "Framework CodeIgniter: Sebuah Panduan dan Best Practice," Pekanbaru, 2011.
- [6] K. Septiani, "Rancang Bangun Modul Regulasi Dan Pembinaan Pada Aplikasi Umkm Juara (Wira Usaha Jawa Barat) Berbasis Web Menggunakan Reactjs dan Nodejs," Skripsi S1, Program Studi Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri, 2020.
- [7] A. Rahman, "Analisis dan Implementasi Sistem Aplikasi Inventory Items Berbasis Web Menggunakan Framework Codeigniter," *Al Ulum Sains dan Teknologi*, Vol. 2, No. 1, pp. 35–38, 2016.
- [8] R. Rendi, "Perancangan dan Implementasi Sistem Inventory Sayur Organik dengan Menggunakan Framework CodeIgniter," Skripsi S1, Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana, 2017.
- [9] N. F. Utami, "Sistem Informasi Inventori Barang PT. Tissan Nugraha Globalindo Berbasis Web," Skripsi S1, Program Studi Informatika, Fakultas Komunikasi dan Informatika, Universitas Muhammadiyah Surakarta, 2018.
- [10] S. Zalukhu, "Analisa dan Perancangan Aplikasi Sistem Inventory (Studi Kasus: Pt. Cakra Medika Utama)," *JSAI (Journal Sci. Appl. Informatics)*, Vol. 2, No. 1, pp. 116–122, 2019, doi: 10.36085/jsai.v2i1.153
- [11] M. Nazir, "Metode Penelitian," Jakarta: Ghalia Indonesia, 1988.
- [12] S. Munir et.al., "Perancangan Sistem Informasi Akademik Berbasis Web menggunakan Framework MVC pada Sekolah Tinggi Teknologi Terpadu Nurul Fikri," *j. inform. terpadu*, vol. 2, no. 1, Jul 2016.
- [13] S. Munir, A. Aufah, K. Septiani, dan S. T. Fauziah, "Analisis Dan Rancang Bangun Prototype Web Marketplace UMKM Juara," *j. teknologi terpadu*, vol. 6, no. 2, hlm. 66-71, Des 2020



PERANCANGAN DAN IMPLEMENTASI *BOOKING SYSTEM* LAPANGAN MENGGUNAKAN FRAMEWORK MVC BERBASIS WEB

Haya Rasikhah¹, Ahmad Rio Adriansyah²

^{1,2} Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jakarta Selatan, DKI Jakarta, Indonesia 12640

hayarrasikhah@gmail.com , arasy@nurulfikri.ac.id

Abstract

The community needs sports field rental to facilitate sports activities that require special equipment such as basketball hoops, net, goalpost, and others. The ordering system for sports fields is still less efficient in terms of time, effort, and cost because people have to go to the field to see field availability and place field orders. In addition to inconveniencing the field owner, the manual field ordering system also hinders the field owner in storing order data. Therefore, researchers built a website-based sports field ordering application that aims to simplify ordering sports fields. The method that researchers use in developing the application is the waterfall method. Researchers used questionnaires and interviews to collect data and tested the application using a Likert Scale and User Acceptance Testing (UAT). Functional test results show the application runs 100% with a feasibility level of 89.5% on the user system and 92.1% on the admin system, which means the application is feasible and can be used.

Keywords: Booking sports fields, Website, Waterfall, Laravel Framework, PIECES

Abstrak

Penyewaan lapangan olahraga sangat dibutuhkan masyarakat untuk memfasilitasi kegiatan olahraga yang membutuhkan alat-alat khusus seperti: Ring basket, Net, Gawang dan lainnya. Pada saat ini sistem pemesanan lapangan olahraga masih kurang efisien dalam hal waktu, tenaga, dan biaya karena penyewa harus mendatangi langsung lapangan untuk melihat kesediaan lapangan dan memesan lapangan. Selain merepotkan pihak penyewa lapangan, sistem pemesanan lapangan secara manual juga menghambat pemilik lapangan dalam proses menyimpan data pesanan. Oleh karena itu peneliti membangun aplikasi pemesanan lapangan olahraga berbasis *website* yang bertujuan untuk mempermudah proses pemesanan lapangan olahraga. Metode yang peneliti gunakan dalam membangun aplikasi adalah metode *waterfall*. Pada pengumpulan data peneliti menggunakan metode kuesioner dan wawancara serta melakukan pengujian aplikasi menggunakan Skala Likert dan *User Acceptance Testing* (UAT). Hasil pengujian fungsional menunjukkan aplikasi berjalan 100% dengan tingkat kelayakan penggunaan 89,5% pada sistem user, dan 92,1% pada sistem admin yang artinya aplikasi layak dan dapat digunakan.

Kata kunci: Pemesanan lapangan olahraga, *Website*, *Waterfall*, *Framework* Laravel, PIECES

1. PENDAHULUAN

Olahraga merupakan aktivitas yang memiliki tujuan tertentu, seperti halnya melatih tubuh untuk kesehatan jasmani maupun rohani. Sehingga olahraga yang rutin dapat memiliki manfaat untuk menjaga kesehatan tubuh. Masyarakat bisa berolahraga dimana saja, tetapi ada beberapa kegiatan olahraga yang harus memiliki tempat khusus atau sarana yang menyediakan fasilitas olahraga sesuai dengan kebutuhannya. Beberapa contoh olahraga yang membutuhkan tempat khusus atau sarana untuk berolahraga yaitu basket, voli, futsal, dan badminton. Olahraga tersebut membutuhkan tempat khusus karena butuh fasilitas pendukung seperti, ring, net, gawang, dan lainnya. Untuk bisa memenuhi kebutuhan agar bisa

olahraga, banyak masyarakat melakukan penyewaan lapangan. Berdasarkan survey yang dilakukan dengan penyebaran *questioner* yang diisi oleh 94 responden. Responden terdiri dari 53.8% mahasiswa, 20.4% siswa SMA, 17.2% karyawan dan 1.1% lainnya bahwasannya responden pernah melakukan penyewaan lapangan olahraga untuk memenuhi kebutuhan olahraga dan diantaranya responden melakukan penyewaan pada lapangan futsal, voli, basket, dan badminton. Responden tersebut menginginkan inovasi dari sistem penyewaan lapangan olahraga secara online supaya membantu dalam informasi lapangan olahraga yang dapat digunakan dan mengefisienkan proses penyewaan lapangan olahraga. Oleh karena itu responden harus melakukan sewa lapangan

untuk dapat melakukan kegiatan olahraga tersebut. Pada umumnya sistem pemesanan lapangan masih menggunakan cara manual yaitu pengguna harus mendatangi langsung tempat penyewaan lapangan atau menghubungi lewat telepon untuk melakukan pengecekan terhadap lapangan yang kosong, sedangkan pemesanan melalui telepon ini memiliki kendala dalam hal validasi yang akurat, oleh karena itu dari pihak lapangan juga tidak sembarangan menerima pemesanan melalui telepon. Sistem pemesanan lapangan secara manual ini cukup merepotkan bagi pihak pengguna lapangan dan menjadi kurang efisien dalam hal waktu, tenaga, dan biaya karena pengguna harus mendatangi langsung setiap lapangan yang ada untuk melakukan pengecekan jadwal dan pemesanan lapangan.

Oleh karena itu berdasarkan uraian latar belakang di atas, maka sangat dibutuhkan otomatisasi sistem dengan merancang serta membuat aplikasi pemesanan lapangan olahraga berbasis web, dalam hal validasi yang akurat untuk masalah penjadwalan dan pemesanan lapangan. Supaya dapat mengefisienkan waktu dan mengurangi biaya transportasi. Dengan adanya pemesanan lapangan olahraga berbasis web juga akan mempermudah *user* karena dapat diakses diberbagai macam perangkat.

Framework

Bedasarkan kamus bahasa Inggris — Indonesia yang di buat oleh Yohanes Aristianto, *framework* adalah rangka atau kerangka kerja, disebut kerangka kerja sebab semua hal yang *programmer* lakukan untuk membangun sebuah aplikasi harus sesuai dengan aturan yang ada pada kerangka kerja dan tidak boleh keluar dari kerangka kerja tersebut [1]. Menurut Ralph E. Johnson, Ketua UIUC partens/Software Architecture Group dan koordinator program proyek senior di Department of Computer Science pada University of Illinois, menyatakan bahwa *framework* adalah desain yang *reuseable* yang dinyatakan sebagai suatu set abstraksi class yang mengatur bagaimana class saling terhubung [2]. *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. Berdasarkan penjelasan tersebut dengan adanya *framework* memungkinkan *programmer* untuk membangun sebuah aplikasi dengan lebih mudah dan efisien dengan *source code* yang di hasilkan lebih rapih untuk pengembangan di kemudian hari. *Framework* dapat diartikan sebagai sebuah kerangka kerja yang digunakan untuk mempermudah penyelesaian sebuah pekerjaan [3].

Software Development Life Cycle (SDLC)

Software 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 [4]. Dengan SDLC membuat pengembangan aplikasi menjadi terstruktur dan metodis karena harus dilakukan secara berurutan. Salah satu tipe SDLC yang paling banyak digunakan adalah *Waterfall*.

Waterfall method merupakan model pengembangan sistem informasi yang sistematis dan sekuensial dengan tujuan yang berbeda untuk setiap fase pengembangannya [5]. Setelah salah satu fase selesai, maka berlanjut ke fase berikutnya dan tidak bisa melakukan revisi ke fase yang sudah dilakukan. Metode *Waterfall* memiliki tahapan-tahapan sebagai berikut [6]:

1) Requirements Analysis

Seluruh kebutuhan *software* harus terkumpul pada fase ini, serta kegunaan *software* yang diharapkan pengguna dan batasan *software* yang didapat dari hasil konsultasi dengan pengguna dan berfungsi sebagai spesifikasi sistem.

2) Software Design

Tahapan ini dilakukan sebelum programmer melakukan coding. Tahapan ini bertujuan untuk perancangan sistem mengalokasikan kebutuhan-kebutuhan sistem baik dari perangkat keras serta perangkat lunak dengan membentuk arsitektur secara keseluruhan melalui penggambaran abstraksi sistem dan hubungannya menggunakan *mockup*.

3) Implementation

Dalam tahap ini programmer melakukan pembuatan aplikasi. Pembuatan aplikasi dipecah menjadi serangkaian program atau unit program yang nantinya akan digabungkan dalam tahap selanjutnya. Selain itu dalam tahap ini juga dilakukan verifikasi bahwa setiap unit memenuhi spesifikasinya.

4) Integation and Testing

Di tahapan ini unit-unit program digabungkan dan dilakukan pengujian sebagai sebuah sistem tunggal untuk memastikan apakah aplikasi yang dibuat telah sesuai dengan kebutuhannya atau tidak.

5) Operation and Maintenance

Ini adalah tahapan terakhir dalam metode *waterfall* dan biasanya (walau tidak selalu) tahapan ini merupakan tahapan yang paling panjang. Aplikasi dijalankan serta dilakukan *maintenance* secara berkala melibatkan perbaikan kesalahan yang tidak ditemukan pada tahap-tahap sebelumnya, meningkatkan implementasi dari unit-unit program serta layanan program sebagai kebutuhan baru.

Laravel

Laravel adalah sebuah MVC *web development framework* untuk sebuah pengembangan aplikasi yang diharapkan dapat meningkatkan kualitas aplikasi yang dihasilkan, dengan mengurangi biaya pengembangan dan perbaikan

serta menghasilkan *source code* yang rapih dan fungsional yang dapat mengefesienkan untuk implementasinya [7]. Laravel merupakan *framework* PHP yang menekankan pada kesederhanaan dan fleksibilitas pada desainnya [8]. Laravel merupakan *framework* PHP yang selalu *up-to-date* karena Laravel bersifat *open-source* dan dikembangkan secara bersama. Laravel sebuah kerangka kerja open source yang diciptakan oleh Taylor Otwell [9]. Laravel memberikan kemudahan untuk berinteraksi dengan *database* yang disebut *migration*. Dengan adanya *migration*, *programmer* sangat mudah untuk melakukan modifikasi sebuah *database* secara independen karena implementasi skema *database* dipresentasikan dalam sebuah Class. Ada beberapa basis data yang mendukung *migration* di Laravel seperti (MySQL, PostgreSQL, MSSQL, dan SQLITE) dan untuk menggunakan *framework* Laravel sebelumnya *programmer* harus menguasai metode dalam membuat program dengan istilah OOP (*Object Oriented Programming*).

2. METODE PENELITIAN

Metode yang digunakan dalam penelitian ini yaitu metode *Waterfall*. *Waterfall* merupakan model pengembangan sistem informasi yang sistematis dan sekuensial dengan tujuan yang berbeda untuk setiap fase pengembangannya[9]. 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* [10].

2.1 Pengumpulan Data

Dalam penelitian ini, untuk mengumpulkan data dibutuhkan masyarakat umum sebagai objek penelitian untuk dilakukannya pengumpulan data berdasarkan jenis data primer dan data sekunder yang bersifat kualitatif. Adapun data yang di kumpulan sebagai berikut :

Data primer adalah data yang menggunakan metode penelitian lapangan yaitu penelitian yang dilakukan dengan cara mendatangi langsung tempat yang dijadikan objek penelitian. Dalam hal ini penulis melakukan pengumpulan data dengan cara pengamatan langsung (observasi).

Dengan mengumpulkan data dan informasi yang diperlukan dari sumber sumber kebanyakan dari materi sejenis dokumen yang berkaitan dengan masalah yang diteliti. Cara yang dilakukan dengan mempelajari literatur-literturnya itu membaca informasi dari jurnal ilmiah, buku dan juga memanfaatkan media internet sebagai sumber informasi melalui situs-situs web. Dari studi literatur tersebut maka akan diperoleh referensi dalam mendukung konsep mengenai bagaimana dalam merancang sistem layanan pemesanan penggunaan lapangan futsal, badminton dan voli secara *online*. Dari studi literatur yang telah diperoleh maka dapat digunakan sebagai alat bantu

dalam mempermudah menerapkan teori maupun teknik-teknik pemrograman *website* dalam membangun sistem. Metode penelitian digunakan untuk dapat membantu menyelesaikan permasalahan sehingga hasil yang didapat lebih sistematis dan terarah. Metode pengumpulan data yang digunakan dalam penelitian ini dilakukan dengan penelitian lapangan yaitu penelitian dilakukan dengan cara mendatangi tempat secara langsung dan dengan pengumpulan data sekunder.

2.2 Tahapan penelitian

Berikut tahapan-tahapan yang dilakukan:

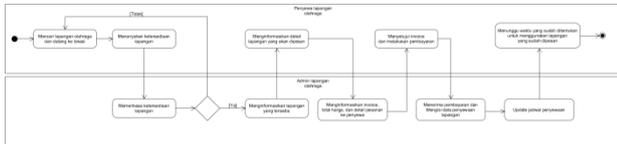
1. *Requirements definition* atau analisis kebutuhan, pada tahap ini dilakukan pengumpulan data melalui observasi dari aplikasi terkait dan wawancara kepada pemilik lapangan dalam mengelola lapangan dengan sistem manual. Peneliti juga melakukan survei kepada calon pengguna terhadap pemesanan lapangan secara manual dengan hasil yang diharapkan dari tahapan ini adalah daftar fitur-fitur yang dibutuhkan dalam aplikasi web.
2. *System and design* atau desain sistem, pada tahap ini mengolah hasil data yang didapat dari analisa kebutuhan dengan hasil yang diharapkan berupa *use case diagram*, *collaboration diagram*, ERD diagram dan *mockup* aplikasi.
3. *Implementation and unit testing* atau implementasi program, pada tahap ini peneliti akan menggunakan salah satu *framework* yaitu laravel untuk perancangan sistem pada bagian *backend* dan pada bagian *frontend* menggunakan Bootstrap dengan *database* MySQL.
4. *Integration and testing* atau pengujian program, pada tahap ini dilakukan uji fungsionalitas sistem menggunakan *Black Box Testing* dengan hasil yang diharapkan adalah mengetahui apakah semua fungsi dan fitur sesuai dengan kebutuhan.
5. *Operation and maintenance* atau penerapan dan pembaharuan program, pada tahap ini peneliti akan meluncurkan aplikasi web pemesanan lapangan olahraga.

3. HASIL DAN PEMBAHASAN

3.1 Analisis Sistem Berjalan

Peneliti melakukan wawancara dengan Ibu Meta, selaku admin lapangan futsal Arena. Tujuan dari wawancara ini adalah untuk mendapatkan informasi yang dibutuhkan untuk mengumpulkan *user requirement* dan gambaran perancangan sistem pemesanan lapangan olahraga secara online. Berdasarkan hasil wawancara yang telah dilakukan, maka didapatkan beberapa informasi terkait sistem yang berjalan saat ini. Informasi-informasi tersebut akan digunakan dalam perancangan sistem pemesanan lapangan olahraga secara *online* pada penelitian ini. Sistem yang telah berjalan sekarang ini dilakukan dengan cara manual atau kegiatan transaksi harus langsung ke lokasi lapangan

olahraga. Pengguna datang ke lokasi untuk melakukan penyewaan lapangan olahraga. Menginformasikan detail pemesanan sewa lapangan. Setelah lapangan dipesan, maka admin lapangan menginformasikan *invoice*, total harga dan detail pesanan penyewa. Kemudian admin menunggu uang pembayaran total sewa lapangan. Ketika pengguna sudah melakukan pembayaran, admin lapangan akan mencatat pesanan di buku penyewaan dan member bukti pembayaran.

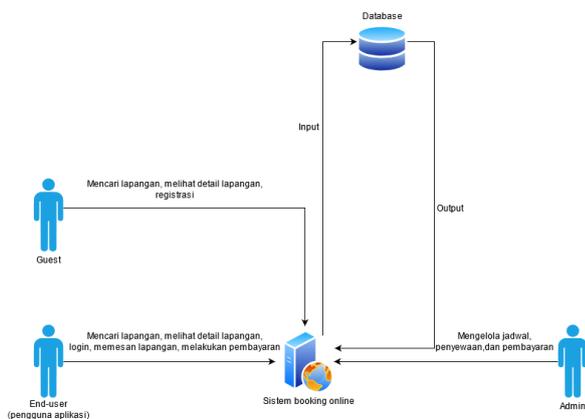


Gambar 1. Alur Diagram Sistem Berjalan

3.2 Analisis Kebutuhan Sistem dan User

Kelemahan pada sistem yang berjalan mengakibatkan proses penyebaran informasi jadwal lapangan kurang maksimal karena saat ini belum tersedianya website khusus yang menyediakan semua informasi terkait detail lapangan yang ada. Selain itu proses transaksi yang ada saat ini juga kurang efektif karena masih dilakukan secara langsung datang ke lokasi. Berdasarkan permasalahan diatas maka diperlukan sebuah sistem yang dapat membantu mempermudah pemesanan lapangan yang dilakukan oleh penyewa sehingga dapat memaksimalkan kinerja pemesanan lapangan.

Pada kategori *user* aplikasi dijelaskan mengenai hak-hak akses yang dimiliki oleh setiap *user*. Kategori *user* dibedakan menjadi tiga yaitu *guest*, *end-user*, dan *admin*. *User guest* hanya memiliki menu melihat jenis lapangan, *register* dan *login*. *End-User* yang telah login ke sistem akan memiliki *Dashboard User* untuk keperluan penyewaan lapangan dan pembayaran. *Admin* memiliki *Dashboard* sendiri yang bertujuan untuk mengelola penjadwalan lapangan dan data *user* pada *website*.



Gambar 2. Rich Picture Sistem yang Diusulkan

3.3 Perbandingan Hasil Analisis PIECES

Berdasarkan aplikasi *web sport corner* yang telah dibuat, peneliti menyimpulkan beberapa analisis terkait perbandingan antara sistem yang sudah berjalan dengan sistem yang telah dikembangkan. Hasil analisis dapat dilihat pada tabel.

Tabel 1. Hasil Perbandingan Analisis PIECES

Analisis PIECES	Sistem Yang Sedang Berjalan	Sistem yang telah dikembangkan
Kinerja (<i>Performance</i>)	Kinerja dari sistem pemesanan lapangan secara manual masih kurang, karena pengguna masih harus datang ke lokasi untuk melakukan pemesanan. Cara tersebut kurang begitu efektif ketika proses pemesana ingin dilakukan dengan cepat.	Proses pemesanan lebih cepat karena dilakukan melalui aplikasi sportcorner tanpa harus datang ke lokasi pemesanan.
Informasi (<i>Information</i>)	Sistem yang ada saat ini walaupun sudah menggunakan media promosi seperti whatsapp, facebook, instagram dan media social tim yang terlibat, masih dirasa kurang dalam penyebaran informasi nya. Hal ini dikarenakan belum adanya platform khusus yang menyediakan informasi detail lapangan maupun jadwal lapangan sehingga masih banyak masyarakat tidak mengetahui lapangan yang dapat langsung digunakan.	Informasi detail lapangan dan jadwal lapangan tersedia pada aplikasi sportcorner.
Analisis Ekonomi (<i>Economic</i>)	Banyaknya biaya-biaya yang harus dikeluarkan untuk media promosi, seperti media cetak. Karena masih menggunakan media cetak, maka tidak mudah untuk mengubah informasi yang sudah tersebar. Apabila ada perubahan informasi maka harus mengeluarkan biaya dan waktu lagi .	Perubahan informasi produk bisa langsung dilakukan pada aplikasi web sehingga tidak membutuhkan waktu dan biaya promosi lagi.
Keamanan atau Kontrol (<i>Security or Control</i>)	Kurang maksimalnya kontrol terhadap data penyewaan. Karena saat ini data masih disimpan pada lembaran kertas tertulis.	Admin dengan mudah melakukan control terdapat data pesanan . Perubahan data dapat langsung dilakukan pada aplikasi web dan akan langsung tersimpan secara otomatis pada database.

Analisis PIECES	Sistem Yang Sedang Berjalan	Sistem yang telah dikembangkan
Efisiensi (<i>Eficiency</i>)	Admin harus melakukan penulisan pemesanan setiap ada penyewaan lapangan, sehingga akan membutuhkan waktu yang lebih dalam proses pencatatan transaksi. Penyampaian informasi juga tidak bisa diakses setiap saat. Sehingga pemanfaatan sumber daya operasional dan teknologi masih kurang.	Proses <i>booking</i> lapangan akan langsung tercatat pada <i>database</i> tanpa harus manual admin menulis form order sehingga menjadi lebih efektif
Layanan (Service)	Pelayanan yang diberikan kepada penyewa menjadi kurang maksimal, karena keterbatasan media promosi menyebabkan proses pelayanan hanya sebatas kepada orang-orang yang sebelumnya sudah mengetahui keberadaan lapangan tersebut. Selain itu apabila pelayanan hanya menggunakan komunikasi via whatsapp terkadang dapat merugikan pihak pemilik lapangan karena pemesanan lapangan yang belum jelas kepastiannya.	Semua layanan yang diberikan oleh sportcorner secara keseluruhan dapat ditemukan pada website dan validasi pemesanan lebih pasti karena otomatisasi pada system aplikasi.

3.4 Pengujian *Blackbox Testing*

Dari 13 butir uji *black box testing* didapat 13 butir uji berjalan dengan baik atau 100% fitur berjalan dengan baik.

4. KESIMPULAN

Berdasarkan hasil ini maka dapat disimpulkan bahwa proses yang dilakukan dalam melakukan pemesanan olahraga masih banyak menggunakan cara pemesanan melalui telepon, mendatangi langsung ke lokasi dan lain sebagainya. Tentunya hal ini dapat membuang waktu penyewa, biaya dan tenaga. Dari permasalahan tersebut, penulis telah berhasil melakukan penelitian dengan kesimpulan yang dapat diambil sebagai berikut:

1. Perancangan sistem lapangan *online* dilakukan dengan tahapan-tahapan: pengumpulan data dan survei, studi pustaka untuk mendapatkan referensi aplikasi sejenis, desain aplikasi, pengembangan aplikasi, pengujian aplikasi dan implementasi.
2. Penelitian ini berhasil membuat suatu sistem berbasis web yang dapat mempermudah pemilik lapangan dalam mengelola dan mempublikasikan informasi terkait lapangan serta mempermudah penyewa dalam melakukan pemesanan lapangan, mengetahui info jadwal dan info perlombaan olahraga yang sedang diadakan.

DAFTAR PUSTAKA

- [1] S. Sabti, "Eksplorasi Yii *Framework* Sebagai Pendukung Pembuatan *Software* Berbasis Web (Studi Kasus Aplikasi Forum)," JBPTUNPASPP, 2012.
- [2] R. E. Johnson, "*The RTL System: A Framework for Code Optimization*," 1991.
- [3] Shalahuddin and A. S Rosa, "Modul Pembelajaran Rekayasa Perangkat Lunak," Modula, Bandung, 2011.
- [4] Tuteja and 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.
- [5] R. S. Pressman, "Rekayasa Perangkat Lunak: Pendekatan Praktisi (Buku Dua)," Penerbit Andi, Yogyakarta, 2002.
- [6] Sommerville, "*Software Engineering 9th Edition*," Addison-Wesley, 2011.
- [7] Widodo and Purnomo, "Perancangan Aplikasi Pencarian Layanan Kesehatan Berbasis HTML 5 *Geolocation*," *Jurnal Sistem Komputer*, 2016.
- [8] Naista, "Bikin *Framework* PHP Sendiri dengan Teknik OOP dan MVC," Lokomedi, Jakarta, 2016.
- [9] R. Hardiansyah dan S. Munir, "Analisis dan Pengembangan Sistem Reservasi *Online* untuk Wisata berbasis *Website* Studi Kasus Pulau Harapan Kepulauan Seribu", *j. inform. terpadu*, vol. 6, no. 1, hlm. 45-52, Mar 2020.
- [10] Ady dan Z. Imaduddin, "Rancang Bangun Aplikasi Pemesanan SEMBAKO berbasis *Mobile* menggunakan Android", *j. inform. terpadu*, vol. 3, no. 2, hlm. 67-74, Sep 2017.



ANALISIS DAN IMPLEMENTASI MODUL AKUNTANSI PADA APLIKASI ERP BERBASIS *OPEN SOURCE*

Fadilla Arfikriyana¹, Suhendi²

^{1,2}Sistem Informasi, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jakarta Selatan, DKI Jakarta, Indonesia 12640

arfikriyana220898@gmail.com, suhendi@nurulfikri.ac.id

Abstract

This study discusses implementing an accounting information system using the Odoo application at PT Delta Solusi Konsultama. PT Delta Solusi Konsultama is a consulting company that offers transfer pricing assistance/services, tax advisory, and tax compliance. In its business, PT Delta Solusi Konsultama only carries out service sales transactions but with a significant enough value. PT Delta Solusi Konsultama still records manually at the current rapid development of technology. In addition, there is a buildup of company financial records, so it takes a long time to search for data. Therefore, by applying the Accounting and Finance module to the Odoo application, it is hoped to overcome the problems at PT Delta Solusi Konsultama. The results of this study include: making a Chart Of Account (COA), recording customer data, recording service sales data, recording financial transaction journals, and printing financial reports. The Odoo application can connect users so that between divisions, there is system integration that can facilitate the control process. This research concludes that the PT Delta Solusi Konsultama Accounting Information System can run in real-time and accurately after using the Odoo application. The advice given from this research is to eliminate unused menus and apply developer mode to change user roles.

Keywords: Accounting, Financial Reports, Enterprise Resource Planning (ERP), Odoo, Open Source

Abstrak

Penelitian ini membahas tentang Implementasi Sistem Informasi Akuntansi Menggunakan Aplikasi Odoo pada PT Delta Solusi Konsultama. PT Delta Solusi Konsultama merupakan perusahaan konsultan yang menawarkan bantuan/ jasa *transfer pricing*, *tax advisory*, dan *tax compliance*. Dalam usahanya, PT Delta Solusi Konsultama hanya menjalankan transaksi penjualan jasa namun dengan nilai yang cukup besar. Pada pesatnya perkembangan teknologi saat ini, PT Delta Solusi Konsultama masih melakukan pencatatan secara manual. Selain itu terdapat penumpukan catatan keuangan perusahaan, sehingga membutuhkan waktu cukup lama untuk melakukan pencarian data. Maka dari itu, dengan diterapkan modul Accounting and Finance pada aplikasi Odoo diharapkan dapat mengatasi masalah yang terjadi di PT Delta Solusi Konsultama. Hasil dari penelitian ini diantaranya: pembuatan *Chart Of Account* (COA), pencatatan data *customer*, pencatatan data penjualan jasa, pencatatan jurnal transaksi keuangan dan pencetakan laporan keuangan. Aplikasi Odoo dapat menghubungkan *user*, sehingga antar divisi terdapat integrasi sistem yang dapat memudahkan proses pengontrolan. Penelitian ini selesai dengan kesimpulan bahwa Sistem Informasi Akuntansi PT Delta Solusi Konsultama dapat berjalan secara *real time* dan akurat setelah menggunakan aplikasi Odoo. Saran yang diberikan dari penelitian ini ialah menghilangkan menu-menu yang tidak terpakai, serta menerapkan mode pengembang untuk mengubah *role user*.

Kata kunci: Akuntansi, Laporan Keuangan, Enterprise Resource Planning (ERP), Odoo, Open Source

1. PENDAHULUAN

Peningkatan penggunaan teknologi komputer merupakan dampak dari perkembangan teknologi informasi. Dampak yang diperoleh adalah teknologi informasi telah memberikan kemudahan bagi karyawan dalam melakukan pemrosesan data. Teknologi merupakan alat yang berguna untuk membantu individu dalam penyelesaian

pekerjanya. Hubungan yang positif dan signifikan antara kecanggihan teknologi informasi dengan karakteristik informasi akuntansi. Teknologi informasi digunakan untuk mengubah data mentah menjadi suatu informasi yang diperlukan oleh pihak internal dan eksternal. Informasi akuntansi dapat membantu manajemen untuk memperjelas tugas-tugas mereka sebelum mengambil keputusan.

Kemajuan teknologi informasi dan komunikasi telah membuat sistem informasi akuntansi menjadi suatu alat penting dalam dunia bisnis yang sangat kompetitif. Penerapan sistem informasi akuntansi merupakan investasi yang penting untuk perusahaan. Sistem informasi akuntansi merupakan bagian yang sangat penting untuk meningkatkan efisiensi organisasi dan mendukung daya saing dengan menyediakan informasi keuangan dan akuntansi bagi manajemen. Sistem dapat dikatakan efektif apabila sistem mampu menghasilkan informasi yang dapat diterima dan mampu memenuhi harapan informasi secara tepat waktu (*timely*), akurat (*accurate*), dan dapat dipercaya (*reliable*) [1].

PT Delta Solusi Konsultama merupakan salah satu perusahaan jasa konsultan di bidang perpajakan. Sistem keuangan perusahaan antara pihak Direktur Keuangan dengan Staf Administrasi tidak terdapat integrasi. Berdasarkan hasil wawancara dengan Staf Administrasi, proses pencatatan keuangan perusahaan masih menggunakan aplikasi Microsoft Excel. Sehingga masalah yang terjadi adalah antara pencatatan dan bukti transaksi disimpan secara terpisah. Hal tersebut membuat sistem keuangan PT Delta Solusi Konsultama tidak berjalan secara *real time* dan akurat antara kedua pihak.

Perusahaan ini merupakan perusahaan yang baru berjalan kurang lebih 1 tahun dan tidak mempunyai anggaran khusus untuk menerapkan sistem *Enterprise Resource Planing* (ERP). Sehingga dengan permasalahan tersebut, penulis memberi solusi untuk menerapkan sistem *Enterprise Resource Planing* (ERP) berbasis *open source*. Pengintegrasian ini akan menggunakan aplikasi Odoo untuk menginput data transaksi keuangan perusahaan.

Oleh karena ini, penulis menyusun skripsi dengan judul “Analisis dan Implementasi Modul Akuntansi pada Aplikasi ERP berbasis *Open Source*” dengan harapan sistem informasi keuangan perusahaan dapat berjalan secara *real time* dan akurat. Sehingga dapat memudahkan Direktur Keuangan mengontrol keuangan perusahaan.

1.1 Rumusan Masalah

Berdasarkan latar belakang yang sudah dipaparkan sebelumnya, maka dapat dirumuskan permasalahannya sebagai berikut:

1. Bagaimana cara implementasi *software* Odoo pada Sistem Informasi Akuntansi di PT Delta Solusi Konsultama?
2. Apakah implementasi Odoo dapat berjalan secara *real time* dan akurat pada Sistem Informasi Akuntansi di PT Delta Solusi Konsultama?

1.2 Tujuan dan Manfaat

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

1. Memahami bagaimana cara implementasi *software* Odoo pada Sistem Informasi Akuntansi di PT Delta Solusi Konsultama.
2. Menghasilkan implementasi Odoo yang dapat berjalan secara *real time* dan akurat pada Sistem Informasi Akuntansi di PT Delta Solusi Konsultama.

Sedangkan manfaat dari penelitian ini adalah:

1. Perusahaan mampu implementasi *software* Odoo pada Sistem Informasi Akuntansi.
2. Perusahaan mampu menjalankan Sistem Informasi Akuntansi yang berjalan secara *real time* dan akurat.

2. TINJAUAN PUSTAKA

2.1 Pengertian Akuntansi

Menurut Sumarsan, pengertian akuntansi adalah suatu seni untuk mengumpulkan, mengidentifikasi, mengklasifikasikan, mencatat transaksi serta kejadian yang berhubungan dengan keuangan sehingga dapat menghasilkan informasi yaitu laporan keuangan yang dapat digunakan oleh pihak-pihak yang berkepentingan [2].

Kieso [3] menyatakan, bahwa akuntansi terdiri dari tiga kegiatan yang mendasar yaitu identifikasi, pencatatan, dan pengkomunikasian peristiwa ekonomi suatu organisasi kepada pihak yang berkepentingan. Perusahaan mengidentifikasi peristiwa ekonomi sesuai dengan kegiatan usahanya dan mencatat peristiwa tersebut untuk menyediakan catatan kegiatan keuangan. Pencatatan dilaksanakan secara sistematis, kronologis setiap peristiwa, dalam satuan mata uang.

2.2 *Enterprise Resource Planning* (ERP)

Enterprise Resources Planning (ERP), merupakan *Enterprise Resource Planning* (ERP) menurut [4] adalah sistem perusahaan yang meliputi semua fungsi yang terdapat di dalam perusahaan yang didorong oleh beberapa modul *software* yang terintegrasi untuk mendukung proses bisnis internal perusahaan. *Enterprise Resource Planning* (ERP) adalah suatu sistem yang terintegrasi yang digunakan oleh organisasi atau perusahaan sehingga dapat mendukung proses bisnis utamanya [9].

Russel [5] menyatakan, bahwa ERP adalah perangkat lunak yang mengorganisasi dan mengelola proses bisnis sebuah perusahaan dengan cara sharing informasi antar area fungsional. ERP adalah sebuah kerangka kerja transaksi keseluruhan perusahaan dengan berbagai hubungan ke pemrosesan pesanan penjualan, manajemen dan pengendalian persediaan, perencanaan produksi dan distribusi, serta keuangan.

2.3 *Open Source*

Open source merupakan alternatif yang signifikan untuk pengembangan produk komoditas perangkat lunak atau

aplikasi layanan. *Open source* jauh lebih cepat, lebih efektif, dan lebih rendah biayanya dibandingkan dengan *software engineering* lainnya, ini menunjukkan bahwa pendekatan baru untuk rekayasa, pengembangan produk dan inovasi lebih memungkinkan. *Open source software* adalah alat dan sistem aplikasi yang murah/gratis untuk diperoleh, relatif mudah untuk dipelajari dan dapat diakses secara global dan dapat di transfer ke internet, banyak *tools* yang diberikan oleh pengembang dengan cuma-cuma [6].

2.4 Odoo

Odoo adalah aplikasi web yang dibangun menggunakan bahasa pemrograman python, XML dan javascript dan postgresql sebagai *database*-nya [7]. 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 *Entreprise 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. 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*.
5. Modul CRM (*Customer Relationship Management*) dan EDI [8].

3. METODE PENELITIAN

Penelitian ini dilakukan dengan menggunakan tahapan metode:

1. Studi Pendahuluan

Pada tahap ini dilakukan 2 jenis studi guna mendapatkan data-data yang dibutuhkan dalam penelitian ini, 2 jenis studi yaitu: Studi Literatur dilakukan dengan mengumpulkan teori-teori dari berbagai referensi dan Studi Lapangan yang dilakukan oleh peneliti adalah mengadakan wawancara dengan Staf Administrasi.

2. Analisis Kebutuhan

Pada tahap ini data dan informasi yang didapatkan dari hasil studi literatur dan lapangan, menjadi sebuah acuan terkait perancangan konfigurasi sistem yang akan diterapkan oleh PT Delta Solusi Konsultama. Konfigurasi sistem ini juga mencakup kebutuhan dari Odoo sebagai sistem yang akan diterapkan.

3. Kustomisasi Sistem

Pada tahap ini hal yang dilakukan adalah kustomisasi sistem sesuai dengan kebutuhan yang diinginkan oleh perusahaan (*user*). Pada sistem Odoo terdapat kustomisasi yang bermaksud untuk mengecilkan ruang lingkup penelitian. Sehingga ruang lingkup yang di kustomisasi hanya untuk proses keuangan di PT Delta Solusi Konsultama dengan modul *Accounting and Finance*.

4. Implementasi dan Uji Coba

Pada tahap ini implementasi sistem Odoo dengan modul *Accounting and Finance* dilakukan setelah perancangan sistem selesai. Setelah itu proses uji coba dilakukan oleh Staf Administrasi dengan menggunakan satu unit laptop. Melalui uji coba ini diharapkan dapat mengetahui kekurangan dari sistem yang telah dibuat.

5. Evaluasi Sistem

Pada tahap ini dilakukan wawancara langsung dengan Staff Administrasi setelah proses uji coba selesai. Wawancara tersebut mengenai apakah sistem yang diimplementasikan sesuai dengan kebutuhan dan apakah sistem dapat berjalan dengan baik.

6. Kesimpulan dan Saran

Pada tahap ini dilaksanakan penarikan kesimpulan dari hasil implementasi Odoo dengan modul *Accounting and Finance* di PT Delta Solusi Konsultama. Serta saran atas kekurangan penelitian yang dilakukan guna pembembangan sistem Odoo untuk penelitian selanjutnya

4. ANALISIS DAN PERANCANGAN SISTEM

Pada bagian ini berisikan langkah-langkah yang dilakukan untuk mendapatkan informasi atau data mengenai sistem yang saat ini sudah diimplementasikan. Serta mengidentifikasi hasil pengumpulan data mengenai permasalahan sistem tersebut. Sehingga data tersebut dianalisis untuk dapat membantu proses implementasi sistem yang sesuai dengan kebutuhan PT Delta Solusi Konsultama.

4.1 Analisis Masalah

Dalam melakukan analisis sistem pencatatan Dalam analisis sistem informasi akuntansi, proses pencatatan keuangan PT Delta Solusi Konsultama ditemukan beberapa masalah. Proses keuangan masih menerapkan

sistem pencatatan secara manual (menggunakan Microsoft Excel). Dan harus mengunggah file terbaru ke penyimpanan berbasis online. Sehingga proses tersebut dapat menimbulkan beberapa masalah, diantaranya:

1. Bukti transaksi disimpan terpisah dengan pencatatan transaksi, sehingga proses pencariannya membutuhkan waktu lama.
2. Sulitnya pembuatan laporan keuangan karena antara transaksi kas kecil dan keuangan lainnya dicatat dalam file yang terpisah serta bagian yang berbeda.
3. File terbaru harus langsung diunggah ke penyimpanan berbasis *online* agar Direktur dapat mengontrol keuangan.

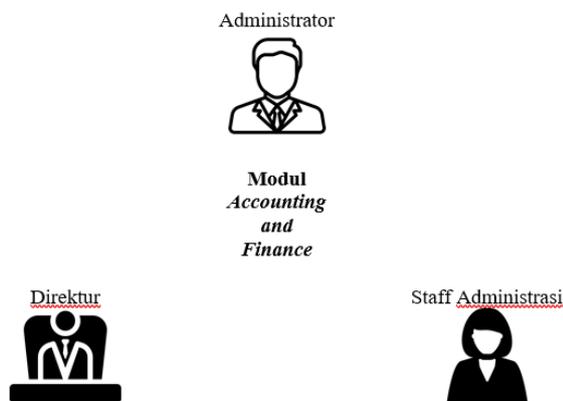
Masalah tersebut dapat menyebabkan adanya ketidakefektifan dan ketidakefisiensi dari setiap proses yang ada. Selain itu sistem yang ada sekarang mengharuskan *user* selalu *backup* data, agar dapat menghindari adanya kerusakan atau kehilangan file. Dan juga belum terbentuknya suatu sistem yang dapat mengatasi setiap permasalahan tersebut.

4.2 Analisis Perancangan Sistem

Pada tahap ini penulis melakukan perancangan terhadap sistem *Accounting and Finance* yang akan digunakan untuk mengatasi permasalahan sistem informasi akuntansi sebelumnya.

1. Identifikasi Aktor

Pada tahap ini akan mengidentifikasi aktor-aktor yang terlibat pada implementasi sistem informasi akuntansi menggunakan Odoo. Hal tersebut dilihat dari daftar kebutuhan *Software Requirement Specification (SRS)* yang telah dibuat pada subbab berikutnya. Berikut ini adalah deskripsi pada masing-masing aktor yang telah teridentifikasi. Di bawah ini adalah gambaran aktor yang teridentifikasi.

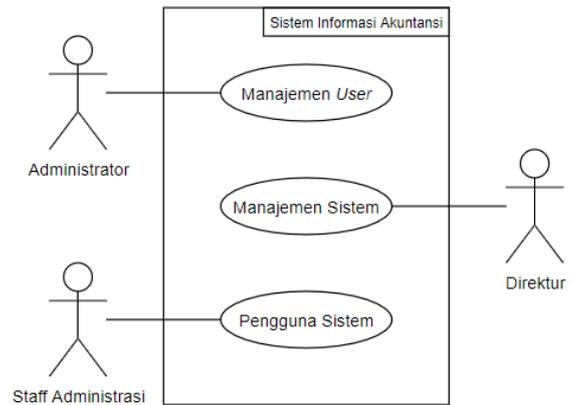


Gambar 1. Identifikasi Aktor

2. Use Case

Pada tahap ini menjelaskan tentang gambaran aktifitas aktor pada sistem yang akan dibuat. Berikut ini adalah diagram *use case* utama dari aplikasi Odoo

dalam proses implementasi modul *Accounting and Finance*.



Gambar 2. Use Case Utama

5. IMPLEMENTASI

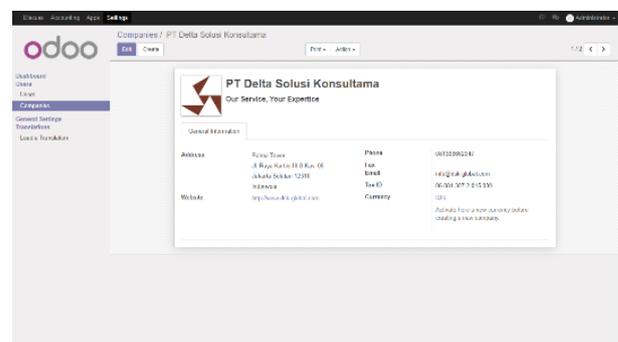
5.1 Konfigurasi Database

Sistem Odoo telah menyediakan ruang fitur manage database untuk memudahkan pembuatan database sesuai dengan kebutuhan user. Berikut ini tampilan login database:

Gambar 3. Form Login Aplikasi Odoo

5.2 Penginputan Informasi Perusahaan (Companies)

PT Delta Solusi Konsultama dijadikan sebagai *companies* atas *database* di aplikasi Odoo ini. Berikut ini adalah tampilan *Companies* PT Delta Solusi Konsultama yang telah diinput:

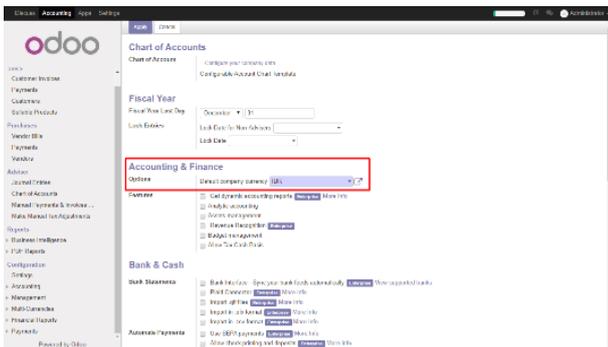


Gambar 4. Tampilan Informasi Companies

5.3 Kustomisasi Sistem

Sistem Odoo menyediakan berbagai modul untuk mendukung proses bisnis suatu perusahaan yang ingin mengimplementasikan ERP. Modul *Accounting and Finance* ini membantu PT Delta Solusi Konsultama dalam mengelola transaksi keuangan, baik penerimaan maupun pengeluaran.

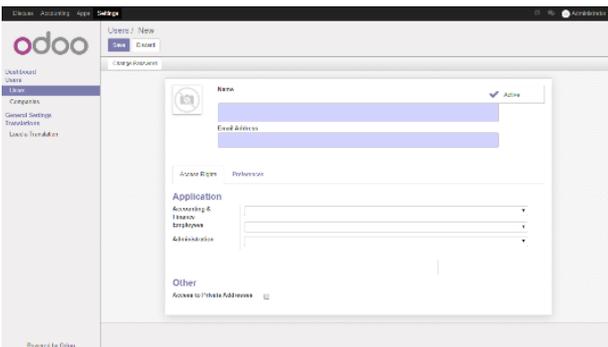
Pada modul *Accounting and Finance* terdapat konfigurasi khusus yaitu mata uang atau *currencies*. PT Delta Solusi Konsultama berada di Indonesia dan menjalankan transaksi menggunakan mata uang Rupiah. Maka konfigurasi mata uang sistem Odoo yang *default* menggunakan USD harus diubah menjadi IDR



Gambar 5. Konfigurasi Pengaturan Modul

5.4 Kustomisasi Hak Akses

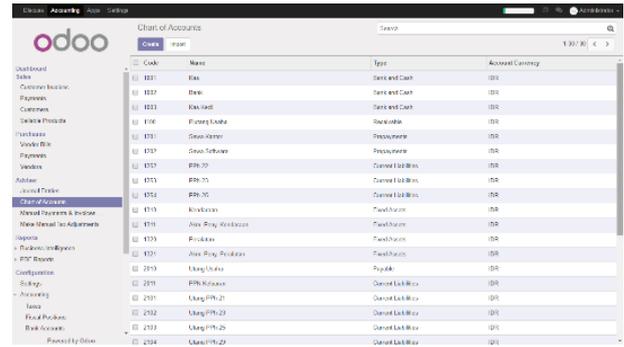
Odoo mempunyai fitur *User Permission* dimana *role* yang telah dibuat dapat digunakan sesuai dengan tugas masing-masing pengguna. Pada sistem Odoo ini terdapat 3 (tiga) aktivitas *user*, yaitu Manajemen *User*, Manajemen Sistem dan Pengguna Sistem. Berikut ini adalah tampilan form *user*:



Gambar 6. Hak Akses Pengguna

5.5 Penginputan Bagan Akun (Chart of Account)

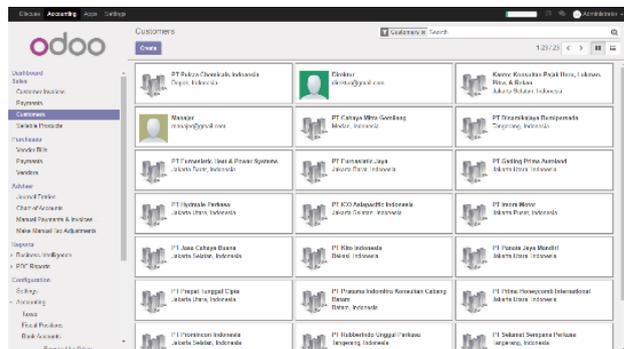
Chart of Accounts adalah satu daftar rangkaian akun-akun yang sudah dibuat atau disusun secara sistematis dan teratur dengan menggunakan simbol-simbol huruf, angka, atau paduan antara keduanya. Berikut ini adalah tampilan *Chart of Accounts* PT Delta Solusi Konsultama yang telah diinput:



Gambar 7. Tampilan List Chart of Accounts

5.6 Penginputan Pelanggan (Customer)

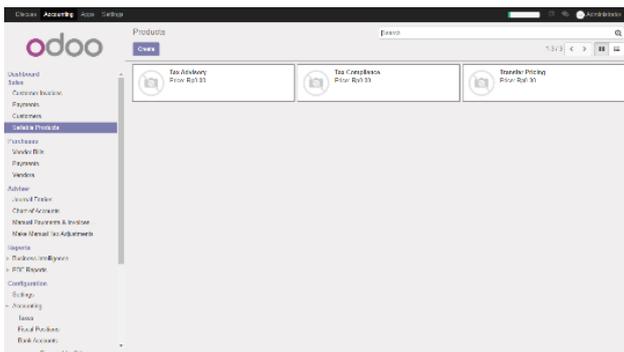
PT Delta Solusi Konsultama hanya melakukan transaksi penjualan saja dan sebagian besar merupakan sebuah perusahaan. Berikut ini adalah tampilan *customers* PT Delta Solusi Konsultama yang telah diinput:



Gambar 8. Tampilan List Customers

5.7 Penginputan Produk (Product)

PT Delta Solusi Konsultama menghasilkan *product* berupa jasa. Berikut ini adalah tampilan *customers* PT Delta Solusi Konsultama yang telah diinput:



Gambar 9. Tampilan List Product

5.8 Penginputan Jurnal (Journal)

Journal difungsikan sebagai wadah pencatatan transaksi keuangan PT Delta Solusi Konsultama. Berikut ini adalah tampilan *Journal* PT Delta Solusi Konsultama yang telah diinput:

Gambar 10. Tampilan List Journal Entries

5.9 Laporan Keuangan

Laporan keuangan merupakan hasil dari pencatatan transaksi yang akan dicetak setiap bulan oleh PT Delta Solusi Konsultama. Berikut ini tampilan Trial Balance PT Delta Solusi Konsultama dengan COA yang terdapat transaksi:

Code	Account	Debit	Credit	Balance
1001	Kas	Rp 6,000,000.00	Rp 5,236,940.00	Rp 763,060.00
1002	Bank	Rp 194,000,000.00	Rp 11,000,000.00	Rp 183,000,000.00
1100	Piutang Usaha	Rp 240,000,000.00	Rp 144,000,000.00	Rp 96,000,000.00
1201	Sewa Kantor	Rp 11,000,000.00	Rp 0.00	Rp 11,000,000.00
2011	PPN Keluaran	Rp 0.00	Rp 4,800,000.00	Rp -4,800,000.00
3001	Modal	Rp 0.00	Rp 56,000,000.00	Rp -56,000,000.00
4001	TP - Penjualan Jasa	Rp 0.00	Rp 235,200,000.00	Rp -235,200,000.00
5001	Beban Perlengkapan	Rp 90,000.00	Rp 0.00	Rp 90,000.00
5004	Beban Pengiriman Dokumen	Rp 57,000.00	Rp 0.00	Rp 57,000.00
5005	Beban Konsumsi	Rp 851,940.00	Rp 0.00	Rp 851,940.00
5006	Beban Parkir	Rp 18,000.00	Rp 0.00	Rp 18,000.00
5007	Beban Sewa Ruang Seminar	Rp 4,200,000.00	Rp 0.00	Rp 4,200,000.00
5008	Beban BBM	Rp 20,000.00	Rp 0.00	Rp 20,000.00

Gambar 11. Laporan Keuangan (Trial Balance)

5.10 Hasil Pengujian Black Box

Tabel 1. Hasil Pengujian Black Box

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Kesimpulan
1	Login	Mengisi username dan password yang benar	Login berhasil	[X] Valid [] Invalid
		Mengisi username dan password yang benar	Login gagal	[X] Valid [] Invalid
2	Buat Profil Perusahaan	Mengisi data profil perusahaan	Berhasil disimpan	[X] Valid [] Invalid
3	Buat User	Mengisi data user dan memilih hak akses aplikasi	Registrasi dan hak akses user berhasil	[X] Valid [] Invalid

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Kesimpulan
4	Buat Bagan Akun	Mengisi data bagan akun	Bagan akun berhasil disimpan	[X] Valid [] Invalid
		Mengisi data pelanggan	Pelanggan berhasil disimpan	[X] Valid [] Invalid
6	Buat Jurnal	Mengisi data transaksi dalam jurnal	Jurnal berhasil disimpan	[X] Valid [] Invalid
7	Cetak Laporan Keuangan	Memilih dan mencetak laporan keuangan	Laporan berhasil dicetak	[X] Valid [] Invalid

Berdasarkan tabel di atas, seluruh pengujian black box yang dilakukan oleh penulis, sistem dapat berjalan dengan baik (valid).

5.11 Hasil Pengujian UAT pada Administrator

Tabel 2. Hasil Pengujian UAT pada Administrator

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Kesimpulan
1	Login	Mengisi username dan password yang benar	Login berhasil	[X] Valid [] Invalid
		Mengisi data profil perusahaan	Berhasil disimpan	[X] Valid [] Invalid
3	Buka User	Mengisi data user dan memilih hak akses aplikasi	Registrasi dan hak akses user berhasil	[X] Valid [] Invalid

Berdasarkan tabel di atas, seluruh pengujian UAT yang dilakukan oleh Administrator, sistem dapat berjalan dengan baik (valid).

5.12 Hasil Pengujian UAT pada Direktur

Tabel 3. Hasil Pengujian UAT pada Direktur

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Kesimpulan
1	Login	Mengisi username dan password yang benar	Login berhasil	[X] Valid [] Invalid
		Mengisi username dan password yang benar	Login gagal	[X] Valid [] Invalid

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Kesimpulan
2	Buat Bagan Akun	Mengisi data bagan akun	Bagan akun berhasil disimpan	[X] Valid [] Invalid
3	Buka Pelanggan	Mengisi data pelanggan	Pelanggan ditampilkan	[X] Valid [] Invalid
4	Buka Jurnal	Mengisi data transaksi dalam jurnal	Jurnal ditampilkan	[X] Valid [] Invalid
5	Cetak Laporan Keuangan	Memilih dan mencetak laporan keuangan	Laporan berhasil dicetak	[X] Valid [] Invalid

Berdasarkan tabel di atas, seluruh pengujian UAT yang dilakukan oleh Direktur, sistem dapat berjalan dengan baik (valid).

5.13 Hasil Pengujian UAT pada Staf Administrasi

Tabel 4. Hasil Pengujian UAT pada Staf Administrasi

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Kesimpulan
1	Login	Mengisi <i>username</i> dan <i>password</i> yang benar	Login berhasil	[X] Valid [] Invalid
2	Buat Bagan Akun	Mengisi data bagan akun	Bagan akun berhasil disimpan	[X] Valid [] Invalid
3	Buat Pelanggan	Mengisi data pelanggan	Pelanggan berhasil disimpan	[X] Valid [] Invalid
4	Buat Jurnal	Mengisi data transaksi dalam jurnal	Jurnal berhasil disimpan	[X] Valid [] Invalid
5	Cetak Laporan Keuangan	Memilih dan mencetak laporan keuangan	Laporan berhasil dicetak	[X] Valid [] Invalid

Berdasarkan tabel di atas, seluruh pengujian UAT yang dilakukan oleh Staf Administrasi, sistem dapat berjalan dengan baik (valid).

5.14 Hasil Kuisisioner

Tabel 5. Hasil Kuisisioner

No.	Pertanyaan	1	2	3	4	Total	Rata-rata
1	Apakah aplikasi ini mudah dioperasikan?			2x3	1x4	10	3,3

No.	Pertanyaan	1	2	3	4	Total	Rata-rata
2	Apakah aplikasi ini dapat memudahkan pekerjaan pengguna?			2x3	1x4	10	3,3
3	Apakah aplikasi ini dapat menghubungkan para pengguna dalam pekerjaan?				3x4	12	4,0
4	Apakah aplikasi ini dapat menghasilkan data atau informasi secara <i>real time</i> ?			1x3	2x4	11	3,7
5	Apakah aplikasi ini dapat menghasilkan data atau informasi yang akurat?			1x3	2x4	11	3,7

6. KESIMPULAN

6.1 Kesimpulan

Dari hasil penelitian penulis yang berjudul Analisis dan Implementasi Modul Akuntansi pada Aplikasi ERP berbasis *Open Source*, dapat diambil beberapa kesimpulan sebagai berikut:

1. Dalam proses implementasi sistem informasi akuntansi menggunakan aplikasi Odoo, secara garis besar dilakukan dengan beberapa tahapan, yaitu:
 - a. Studi Literatur dan Lapangan
Studi literatur dan lapangan ini dilakukan untuk melakukan analisis terhadap kebutuhan-kebutuhan dalam implementasi sistem informasi akuntansi. Sehingga dari analisis tersebut menghasilkan informasi bahwa masalah yang dialami oleh PT Delta Solusi Konsultama, yaitu terdapat penumpukan file pencatatan transaksi keuangan yang dilakukan secara manual dan bukti transaksi disimpan terpisah dengan catatan.
 - b. Analisis dan Perancangan
Pada tahap ini dilakukan analisis sistem yang dapat mengatasi masalah PT Delta Solusi Konsultama. Serta perancangan alur sistem tersebut menggunakan *flowchart*, perancangan desain aplikasi yang mudah digunakan.

c. Implementasi Sistem

Pada tahap ini aplikasi diimplementasikan sesuai dengan yang telah kebutuhan *user*. Pembuatan aplikasi ini dirancang secara sederhana dengan desain antarmuka yang praktis, agar mudah menjalankannya.

2. Hasil kuesioner yang dilakukan oleh *user* menghasilkan kesimpulan bahwa Sistem Informasi Akuntansi PT Delta Solusi Konsultama dapat berjalan secara *real time* dan akurat setelah menggunakan aplikasi Odoo.

6.2 Saran

Setelah melakukan Implementasi Sistem Informasi Akuntansi menggunakan Aplikasi Odoo pada PT Delta Solusi Konsultama, penulis mempunyai beberapa saran yaitu:

1. Pada implementasi modul *Accounting and Finance* ini masih terdapat menu-menu yang tidak terpakai seperti *Purchases*. Sedangkan PT Delta Solusi Konsultama dalam bisnis usahanya tidak melakukan pembelian. Maka diharapkan untuk implementasi selanjutnya dapat menghilangkan menu-menu tersebut.
2. Direktur sebagai manajemen sistem yang seharusnya hanya dapat melihat, saat ini masih dapat membuat, menghapus dan mengedit sistem Odoo. Maka diharapkan untuk implementasi selanjutnya dapat mengubah *role user* tersebut menggunakan mode pengembang.
3. Dalam penggunaan aplikasi Odoo ini masih menggunakan *localhost*, sehingga user diharuskan menginstall terlebih dahulu untuk mengaksesnya. Maka diharapkan untuk implementasi selanjutnya dapat menggunakan hosting perusahaan untuk mengakses aplikasi Odoo.

DAFTAR PUSTAKA

- [1] K. I. Ratnaningsih dan I. G. N. A. Suaryana, "Pengaruh Kecanggihan Teknologi Informasi, Partisipasi Manajemen, dan Pengetahuan Manajer Akuntansi pada Efektivitas Sistem Informasi Akuntansi," *e-Jurnal Akuntansi*, Vol. 6, No. 1, hal. 1-16, 2014.
- [2] T. Sumarsan, "Perpajakan Indonesia," PT. Indeks, Jakarta, 2013.
- [3] D. E. Kieso et al., "*Intermediate Accounting*," 14th Edition, John Wiley & Sons Inc, Asia, 2011.
- [4] O'Brien, "*Management System Information*," McGraw Hill, New York, 2010.
- [5] Russel, "*Operation Management*," Prentice Hall, New Jersey, 2003.
- [6] M. Rifqi, "Kajian Pengembangan Sistem Informasi Berbasis *Open Source* Joomla dengan menggunakan Auto Generate," *Jurnal Telematika MKOM*, Vol. 4, No. 1, 2012.
- [7] A. Nasir dan Suhendi, "Penerapan Pengelolaan Transaksi Keuangan menggunakan Modul Accounting and Finance Odoo 10 Studi Kasus Yayasan SDIT Bahrul Fikri", *J. Inform. Terpadu*, Vol. 4, No. 1, hlm. 01-06, 2018.
- [8] C. A. Lestari, "Implementasi Odoo dengan Modul *Accounting and Finance* di SD Islam Tunas Mandiri," *J. Inform. Terpadu*, Vol. 3, No. 1, 2017.
- [9] L. Cahya Putri dan S. Suhendi, "Analisis dan Implementasi ERP pada Modul Point of Sale Studi Kasus Toko Tas Apidah", *J. Inform. Terpadu*, vol. 7, no. 1, hlm. 01-07, Mar 2021.



PENERAPAN *LOG ANALYZER LOG* UNTUK MENGETAHUI LALU LINTAS JARINGAN BERBASIS *ELASTICSEARCH, LOGSTASH, DAN KIBANA*

Muhammad Jafier Rama Putra¹, Henry Saptono²

^{1,2}Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jakarta Selatan, DKI Jakarta, Indonesia 12640
jafier7@gmail.com, henry@nurulfikri.ac.id

Abstract

Recent advances in information technology have changed the way people view the development of existing applications or networks. New information technology intends to help people divert time and energy to other jobs in their daily lives. What is needed for information technology is management, maintenance, and monitoring. Management, maintenance, and monitoring of information technology or computer networks and the services expected to run as expected by individuals, organizations, or institutions to assist and support work processes within the organization. This research intends to implement a log analyzer and network traffic data based on ELK Stack (Elasticsearch Logstash Kibana), which can send logs, collect logs and visualize logs from the server. The implementation of the log analyzer and network traffic data in this study uses Ubuntu 18.04 Server and Ubuntu 18.04 as a client. The results of testing the implementation of the log analyzer and network traffic data that have been carried out with a success rate of 100% show that all log tests that occur on the server can be sent in real-time to the central server of the ELK Stack.

Keywords: Log, Information Technology, Monitoring, Network, ELK Stack

Abstrak

Kemajuan dalam bidang teknologi informasi terkini telah mengubah cara pandang masyarakat mengenai pengembangan aplikasi atau jaringan yang ada sekarang. Sebuah teknologi informasi baru yang dimaksudkan untuk membantu manusia dalam hidup kesehariannya sehingga waktu dan tenaga dapat dialihkan pada pekerjaan lainnya. Hal yang diperlukan untuk teknologi informasi adalah pengelolaan, pemeliharaan, dan pemantauan. Pengelolaan, pemeliharaan, dan pemantauan teknologi informasi atau jaringan komputer dan layanan yang ada di dalamnya diharapkan dapat berjalan sebagaimana yang diharapkan oleh individu, organisasi atau institusi untuk membantu dan menunjang proses kerja dalam organisasi tersebut. Dalam penelitian ini bermaksud untuk melakukan implementasi *analyzer log* dan data lalu lintas jaringan berbasis ELK Stack (*Elasticsearch Logstash Kibana*) yang dapat melakukan pengiriman log, mengumpulkan log dan memvisualisasi log dari *server*. Implementasi *analyzer log* dan data lalu lintas jaringan dalam penelitian kali ini menggunakan Ubuntu 18.04 Server, dan Ubuntu 18.04 sebagai *client*. Dari hasil pengujian implementasi *analyzer log* dan data lalu lintas jaringan yang telah dilakukan dengan tingkat keberhasilan 100% menunjukkan bahwa semua pengujian log terjadi pada *server* dapat dikirimkan secara *realtime* ke server utama ELK Stack.

Kata Kunci : Log, Teknologi Informasi, Pemantauan, Jaringan, ELK Stack

1. PENDAHULUAN

Pemanfaatan teknologi informasi merupakan sarana penunjang atau pendorong bagi masyarakat dalam mencapai tujuannya. Pemanfaatan teknologi yang efektif dapat meningkatkan kinerja. Kinerja berhubungan dengan pencapaian serangkaian tugas-tugas yang dilaksanakan oleh individu-individu di dalam organisasi. Sehingga semakin tinggi kinerja individu semakin meningkat pula efektifitas, produktivitas dan kualitas pelayanan individu tersebut. Hal yang diperlukan untuk teknologi informasi

adalah pengelolaan, pemeliharaan, dan pemantauan dengan cara yang lebih efektif dan efisien.

Hal yang dibutuhkan keberlangsungan jaringan komputer untuk keandalan, keamanan, dan ketepatan layanan yang ada didalamnya. Untuk memastikan keandalan, keamanan, dan ketepatan maka dibutuhkan suatu proses analisis tersebut.

2. LANDASAN TEORI

2.1 Log

Log adalah catatan tentang peristiwa yang terjadi dalam sistem organisasi dan jaringan. Log terdiri dari entri log; setiap entri berisi informasi yang berkaitan dengan peristiwa tertentu yang terjadi dalam sistem atau jaringan.

2.2 Analisis Log

Analisis log adalah proses mengubah data log mentah menjadi informasi untuk memecahkan masalah [8]. Sistem digital menghasilkan banyak jenis log files, yang memberikan informasi penting tentang sistem ini. Beberapa jenis file log, seperti pemantauan jaringan log, interaksi layanan web, atau penggunaan web log dieksploitasi secara luas [7].

2.3 Lalu Lintas Jaringan

Suatu jaringan dapat dikatakan *traffic*-nya padat atau tinggi, apabila banyak host yang melakukan koneksi ke *server* di dalam jaringan tersebut. Fungsi pengawasan terhadap unjuk kerja jaringan dan pengambilan tindakan untuk mengendalikan aliran trafik agar diperoleh kapasitas jaringan dengan pengoperasian yang maksimum. Manajemen lalu lintas jaringan adalah sebuah pekerjaan untuk memelihara seluruh sumber jaringan dalam keadaan baik.

2.4 Domain Name System (DNS)

Domain Name System (DNS) adalah *distribute database system* yang digunakan untuk pencarian nama komputer (*name resolution*) di jaringan yang menggunakan TCP/IP (*Transmission Control Protocol/Internet Protocol*) [1]. TCP/IP (*Transmission Control Protocol/Internet Protocol*) adalah sekumpulan protocol yang terdapat di dalam jaringan computer yang digunakan untuk berkomunikasi atau bertukar data antar komputer. DNS biasanya digunakan sebuah Layanan Nama Domain untuk menyelesaikan permintaan untuk nama-nama website menjadi alamat IP untuk tujuan menemukan layanan komputer serta perangkat di seluruh dunia.

2.5 Dynamic Host Configuration Protocol (DHCP)

DHCP (*Dynamic Host Configuration Protocol*) adalah protokol yang berbasis arsitektur *client/server* yang dipakai untuk memudahkan pengalokasian alamat IP dalam satu jaringan [2]. Kemudahannya dalam pemberian alamat IP kepada komputer *client* atau perangkat jaringan (walau dalam jumlah yang banyak) secara otomatis.

2.6 Web Server

Web server adalah sebuah *software* yang memberikan layanan berbasis data dan berfungsi menerima permintaan dari HTTP atau HTTPS pada *client* yang dikenal dan biasanya kita kenal dengan nama *web browser* (Mozilla Firefox, Google Chrome) dan untuk mengirimkan kembali yang hasilnya dalam bentuk beberapa halaman web dan

pada umumnya akan berbentuk dokumen HTML [3]. Dalam Penggunaannya, biasanya sudah jadi satu paket dengan PHP dan MySQL diantaranya XAMPP dan *Appserv* [9].

2.7 Elasticsearch

Elasticsearch adalah mesin pencari *open-source* yang dibangun di atas Apache Lucene™, fulltext perpustakaan mesin pencari. Lucene bisa dibilang yang paling maju, kinerja tinggi, perpustakaan mesin pencari berfisalkan lengkap yang ada saat ini—baik sumber terbuka maupun berpemilik [4].

2.8 Logstash

Logstash adalah *software* gratis dan *open source* (berlisensi Apache 2.0) dan dikembangkan oleh American pengembang, Jordan Sissel dan tim dari Elastic. Mudah diatur, berkinerja, terukur dan mudah diperluas. *Logstash* memiliki berbagai macam mekanisme input: ia dapat mengambil input dari TCP/UDP, file, *Syslog*, Microsoft Windows *EventLogs*, STDIN, dan berbagai jenis sumber lain [5].

2.9 Kibana

Kibana adalah alat yang merupakan bagian dari tumpukan ELK, yang terdiri dari *Elasticsearch*, *Logstash*, dan Kibana. Itu dibangun dan dikembangkan oleh Elastic. Kibana adalah *platform* visualisasi yang dibangun di atas *Elasticsearch* dan memanfaatkan fungsionalitas *Elasticsearch* [6].

3. METODE PENELITIAN

3.1 Studi Literatur

Pada tahapan awal ini dilakukan dengan mencari, mengumpulkan, serta membaca artikel di *website* dan beberapa skripsi penelitian lainnya yang berhubungan dengan implementasi *analyzer log* dan lalu lintas jaringan berbasis ELK (*Elasticsearch*, *Logstash*, Kibana) Stack. Hasil dari studi literatur yaitu menerapkan ELK stack untuk memudahkan proses analisis data dan bagaimana penelitian harus dilakukan dan bahan apa saja yang diperlukan untuk tujuan penelitian ini agar dapat tercapai.

3.2 Pengujian dan Evaluasi

Pada tahapan ini akan dilakukan pengujian terhadap proses untuk memudahkan proses analisis data log dan data lalu lintas jaringan yang akan kita analisis dengan menggunakan ELK (*Elasticsearch*, *Logstash*, Kibana) Stack. serta mengetahui seberapa efektifitas dari manfaat log analisis dan lalu lintas jaringan. Setelah menguji dan melakukan evaluasi dari beberapa faktor, maka peneliti akan menganalisis hasil yang di dapat dari pengujian yang telah dilakukan. Pengujian ini dilakukan untuk

memberikan jawaban dari rumusan masalah, apa benar efektif atau tidaknya sistem analisis ini bekerja sebenarnya.

3.3 Rancangan Penelitian

3.3.1 Metode Pengumpulan Data

Metode pengumpulan data yang dilakukan oleh peneliti yaitu dengan metode deskriptif, metode ini akan mempelajari aspek-aspek apa yang akan di tinjau untuk mengetahui analisis implementasi *analyzer log* dan lalu lintas dalam suatu jaringan.

3.3.2 Studi Pusaka

Metode pengumpulan data dilakukan dari suatu jaringan yang akan di analisis dan diimplementasikannya berdasarkan rumusan masalah yang ada.

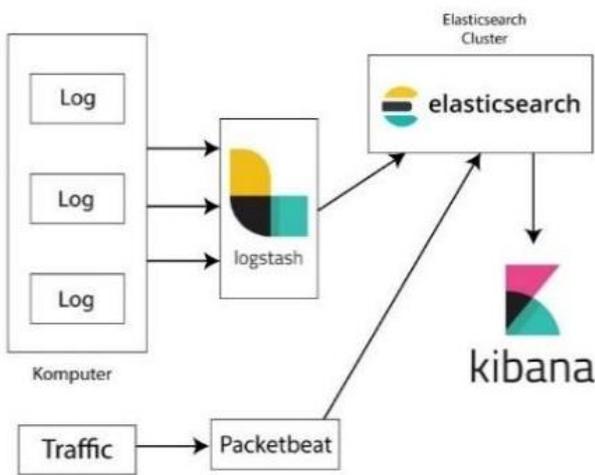
4. ANALISIS DAN PENELITIAN

4.1 Analisis Sistem

Analisis Sistem adalah menganalisa sistem yang sedang ingin di bangun yang bertujuan untuk mengetahui kebutuhan sistem dalam menganalisanya, sehingga memudahkan peneliti pada tahapan selanjutnya yaitu perancangan sistem.

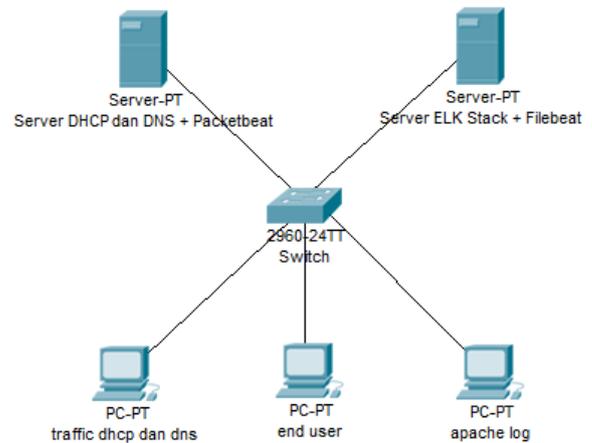
4.2 Perancangan Sistem

Perancangan sistem secara umum dilakukan dengan maksud untuk memberikan gambaran umum tentang sistem yang baru atau sistem yang akan diusulkan. Dalam penelitian ini, peneliti melakukan perancangan arsitektur sistem yang terdiri dari masing-masing menggunakan sistem operasi Linux Ubuntu Server 18.04 dan Ubuntu Desktop 18.04.



Gambar 1. Perancangan Sistem Logic

Perancangan sistem ini merupakan gambaran keseluruhan dari proses analisis secara fisik yang akan dijalankan. Dalam perancangan ini, digambarkan suatu sistem dengan tampilan yang *real* dalam mengimplementasikannya.



Gambar 2. Perancangan Sistem Fisik

5. HASIL DAN PEMBAHASAN

5.1 Pengujian

Pengujian ini peneliti akan memaparkan pengujian yang peneliti lakukan setelah pengimplementasian ELK Stack. Peneliti melakukan pengujian efektivitas dan Pengujian *Analyzer Log* dan Lalu lintas Jaringan berbasis ELK Stack.

Efektifitas diukur dengan cara sebagai berikut :

$$\text{Efektifitas Pengujian} = (\text{Output/Input}) * 100\%$$

$$\text{Rerata Efektifitas} = (\text{Total Efektifitas} / \text{Jumlah pengujian})$$

$$\text{Tingkat efektifitas} \sim \text{Rerata Efektifitas}$$

A. Ujian Pengiriman Data Log (*Logstash - Elasticsearch*)

Pada tahap pengujian ini dilakukan pengujian untuk memeriksa apakah data log tersimpan dari *logstash* ke *Elasticsearch* dan akan mengirim 10 log dalam kurun waktu tertentu. Selama waktu yang di tujukan akan terlihat berapa jumlah log yang di dihasilkan dan apa log akan tersimpan di *Elasticsearch*. Jadi mengirim log dari `/var/log/syslog`.

B. Ujian menggunakan *Shell Script (syslog)*

Pengujian ini menggunakan *Shell Script* untuk membantu me-*generate* isi log yang akan input ke *Elasticsearch* sesuai isi dan jumlah log-nya. *Shell Script* adalah kumpulan beberapa *command* yang ditulis pada teks file yang nantinya akan di-*execute* oleh *shell*.

Tabel 1. Pengujian Data Log (syslog)

Pengujian ke	Jumlah Log Input	Waktu Awal	Selisih Waktu (rata-rata) detik	Waktu Akhir	Jumlah Log yang diterima (Output)	Efektifitas
1	10	06:33:27	1	06:33:28	10	100%
2	20	06:39:10	2	06:39:12	20	100%
3	30	06:44:00	4	06:44:04	30	100%
4	40	06:48:42	6	06:48:48	40	100%
5	50	07:08:31	7	07:08:38	50	100%
6	60	07:12:36	8	07:12:44	60	100%
7	70	07:20:51	8	07:20:59	70	100%
8	80	07:23:26	8	07:23:44	80	100%
9	90	07:28:21	9	07:28:30	90	100%
10	100	07:32:11	9	07:32:20	100	100%
Rata-rata						100%

Rerata Efektifitas =

$$(100\% + 100\% + 100\% + 100\% + 100\% + 100\% + 100\% + 100\% + 100\% + 100\%)/10 = 100\%$$

Tabel 2. Pengujian Traffic Log (packetbeat)

Pengujian ke	IP Pengirim	Port	IP Tujuan	Waktu Awal	Selisih Waktu (rata-rata) detik	Traffic	Waktu Akhir
1	192.168.1.1	9200	192.168.1.6	08:30:31	20	08:30:31	TCP
2	192.168.1.6	5601	192.168.1.3	08:42:25	-	08:42:25	TCP
3	192.168.1.6	53	192.168.1.1	08:45:40	10	08:45:50	UDP
4	192.168.1.6	5044	192.168.1.3	08:51:07	3	08:51:10	TCP

Data pengujian dapat dikatakan memiliki tingkat efektifitas **100%**.

5.2 Pengujian Log Apache menggunakan Apache Benchmarks

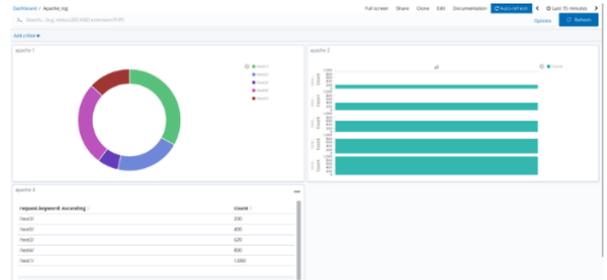
Pengujian ini menggunakan Apache Benchmark untuk mengukur kapabilitas Apache untuk melayani request dari client yang akan inputkan ke Elasticsearch sesuai isi dan jumlah lognya. Apache Benchmark adalah tool untuk mengukur kapabilitas Apache untuk melayani request dari client yang nantinya akan tertulis di Logstash dan tersimpan di Elasticsearch. Log tersebut akan terlihat di Kibana.

$$\text{Rerata Efektifitas} = (100\% + 100\% + 100\% + 100\% + 100\%) / 5 = 100\%$$

Data pengujian dapat dikatakan **efektif** atau dalam persentase memiliki tingkat efektifitas **100%**.

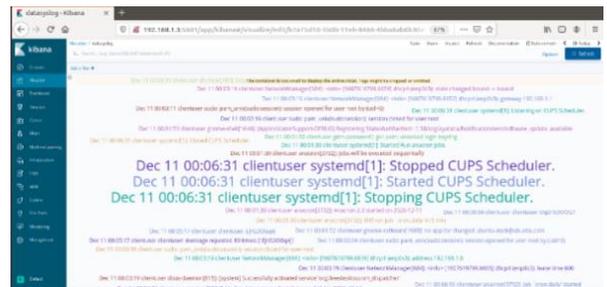
5.3 Hasil Visualisasi

- Hasil visualisasi Apache menggunakan tipe visual Pie, Horizontal Bar, Data Table



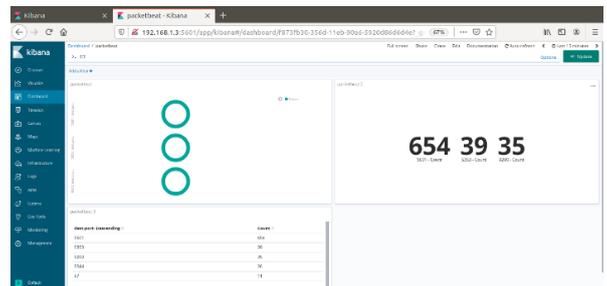
Gambar 3. Visualisasi Apache

- Hasil visualisasi syslog (data log) menggunakan tipe visual Tag Cloud



Gambar 4. Visualisasi Syslog

- Hasil visualisasi Packetbeat menggunakan tipe visual Pie, Metrics, Data Table



Gambar 5. Visualisasi Packeybeat

6. PENUTUP

6.1 Kesimpulan

Kesimpulan yang dapat diambil oleh peneliti dalam penelitian ini yaitu:

1. Rancangan yang dilakukan dalam implementasi ini sebagai proses untuk dapat mengelompokkan log secara tepat adalah menggunakan Elasticsearch, Logstash serta Kibana. Untuk rancangan sistem lalu lintas jaringan, menggunakan Packetbeat langsung ke Elasticsearch dan divisualisasikan dengan Kibana.
2. Log yang digunakan adalah log pada mesin linux dan dikirim di /var/log yang akan diproses melalui Logstash.
3. Efektivitas ELK Stack untuk (syslog) memiliki tingkat efektifitas 100%, sedangkan untuk Apache log yaitu memiliki efektifitas 100%. Kemudian untuk lalu lintas jaringan juga memiliki efektifitas 100%.

4. ELK Stack dapat menjadi solusi untuk implementasi sebuah *Log* yang membantu dalam memvisualisasikan hasil.

6.2 Saran

Beberapa saran yang dapat dilakukan untuk penelitian selanjutnya, yaitu:

1. Penelitian selanjutnya dapat diimplementasikan pada jaringan *real* atau tidak dalam lingkungan virtualisasi.
2. Penelitian ini memungkinkan dapat dikembangkan dengan jumlah data log dan menambah metode beragam dengan jumlah *node* (komputer).
3. Penelitian ini memungkinkan untuk diuji kembali dengan jumlah data lalu lintas jaringan yang lebih beragam dengan jumlah *node* (komputer) lebih banyak.

DAFTAR PUSTAKA

- [1] D. Ardiantoro, "Pengantar DNS (*Domain Name System*)," 2003.
- [2] M. R. Andargini, "Panduan Instalasi & *Setup* DHCP Server," 2008.
- [3] C. Tarigan, *et al.*, "Sistem Pengawasan Kinerja Jaringan *Server Web* Apache dengan Log Management System ELK (*Elasticsearch, Logstash, Kibana*)," pp. 7–14, 2018.
- [4] C. Gormley, Z. Tong, "*Elasticsearch: The Definitive Guide*," 2015.
- [5] J. Turnbull, "*The Logstash Book*," 2014.
- [6] Y. Gupta, "*Kibana Essentials*," 2015.
- [7] H. Saneifar, *et al.*, "*Terminology Extraction from Log Files*," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, 2009.
- [8] S. Alspaugh, *et al.*, "*Analyzing Log Analysis: An Empirical Study of User Log Mining*," LISA'14 Proc. 28th USENIX Conf. Large Install. Syst. Adm., pp. 53–68, 2014.
- [9] A. Permatasari dan S. Suhendi, "Rancang Bangun Sistem Informasi Pengelolaan *Talent Film* berbasis Aplikasi Web", *J. Inform. Terpadu*, vol. 6, no. 1, hlm. 29-37, Mar 2020.
- [10] D. O. Saputra dan H. Saptono, "Implementasi *Network Monitoring System* terintegrasi dengan Ticketing System menggunakan Nagios dan osTicket", *J. Inform. Terpadu*, vol. 5, no. 1, hlm. 06-17, Mar 2019



ANALISIS DAN PERANCANGAN APLIKASI MOBILE UNTUK DONASI MENGGUNAKAN METODE *HYBRID* BERBASIS *REACT NATIVE*

Muhammad Abdul Karim¹, Ahmad Rio Adriansyah²

^{1,2} Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jakarta Selatan, DKI Jakarta, Indonesia 12640

alkariim.tridek@gmail.com, arasy@nurulfikri.ac.id

Abstract

This study discusses the design of mobile-based goods donation applications using the framework React Native. In Indonesia, there are many online donation platforms, but most donations are money. However, the first needs of victims of natural disasters are not money but primary goods. In addition, based on Carousell's Preloved Indonesia 2017 Trends Report, 82% of the 1,000 respondents piled up unused items in their homes in Indonesia. Therefore, a mobile-based item donation application makes, where donated goods are items that have not been used. The goal is to make it easier for users to donate goods, wherever and whenever. The application builds using the framework React Native with RESTful API and MySQL as a database programming language. Tests carried out are functional system testing using black-box testing and User Acceptance Test (UAT) and testing using a questionnaire. The functional system runs according to the defined requirements based on the test results. The test results are based on 100% black-box testing and 100% UAT and questionnaires with an average value of 87.4%.

Keywords: Mobile Applications, Goods Donation, MySQL, Web Service, React Native

Abstrak

Penelitian ini membahas tentang rancang bangun aplikasi donasi barang berbasis *mobile* menggunakan *framework react native*. Di Indonesia terdapat banyak *platform* donasi *online*, namun kebanyakan donasi yang diberikan berbentuk uang. Padahal kebutuhan pertama para korban bencana alam bukanlah uang melainkan barang primer. Selain itu, berdasarkan Laporan Tren Preloved Indonesia 2017 oleh Carousell, di Indonesia terdapat 82% dari 1.000 responden menumpuk barang yang tak terpakai dirumahnya. Maka dari itu, dibuatlah aplikasi donasi barang berbasis *mobile*, dimana barang yang didonasikan merupakan barang-barang bekas yang sudah tidak terpakai. Tujuannya yaitu memudahkan pengguna untuk berdonasi barang, dimana pun dan kapan pun. Aplikasi dibangun menggunakan *framework React Native* dengan *RESTful API* dan *MySQL* sebagai bahasa pemrograman database. Pengujian yang dilakukan yaitu pengujian fungsional sistem dengan menggunakan teknik *black-box testing* dan *User Acceptance Test (UAT)* serta pengujian dengan menggunakan kuesioner. Berdasarkan hasil pengujian, fungsional sistem berjalan sesuai dengan requirement yang telah didefinisikan. Hasil pengujian tersebut berdasarkan dari hasil *black-box testing* 100% dan *UAT* 100% serta kuesioner dengan rata rata nilai 87,4%.

Kata kunci: Aplikasi *Mobile*, Donasi Barang, *MySQL*, *Web Service*, *React Native*

1. PENDAHULUAN

Menurut BNPB [1] pada awal tahun 2019 sampai 30 April 2019 terdapat 656 kali puting beliung, 466 banjir, 453 tanah longsor, 12 kali gempa bumi, 4 kali letusan gunung berapi di Indonesia. Bencana alam ini mengakibatkan kerusakan rumah dan fasilitas umum, maka perlu adanya perbaikan dan pembangunan kembali lingkungan serta bangunan yang telah rusak. Hal ini tentunya membuat organisasi yang bergerak di bidang donasi dan kemanusiaan seperti Aksi Cepat Tanggap, Dompet Dhuafa

Republika, PKPU Human Interative, Rumah Zakat, Indorelawan dan sebagainya mulai banyak dan berkembang di Indonesia.

Salah satu cara yang ditawarkan oleh organisasi tersebut yaitu dengan menghimpun donasi dari masyarakat untuk para korban bencana. Donasi merupakan kegiatan kemanusiaan yang bertujuan untuk keperluan kegiatan sosial [2]. Pemberian donasi dapat berupa pakaian, makanan, barang, maupun uang.

Seiring berkembangnya teknologi, solusi lain yang ditawarkan adalah hadirnya beberapa *platform* donasi berbasis *online*, yang membuat masyarakat bisa berdonasi dimana pun dan kapan pun. Akan tetapi dari sekian banyak *platform* tersebut, kebanyakan donasi yang diberikan berbentuk uang. Padahal yang dibutuhkan para korban bencana alam bukan hanya uang, akan tetapi juga barang-barang primer. Seperti yang dilansir oleh *voaindonesia.com* bahwa pada saat terjadi gempa bumi di Dongala dan Palu pada tahun 2018, yang dibutuhkan oleh para korban adalah makanan, minuman, obat-obatan, pakaian dan transportasi [3].

Disisi lain, tak sedikit masyarakat Indonesia yang memiliki barang-barang yang tak terpakai menumpuk di rumah. Dibuktikan dengan Laporan Tren Preloved Indonesia 2017 oleh Carousell Indonesia, 82% dari 1.000 responden menumpuk barang yang tak terpakai di rumahnya [4]. Hal ini menunjukkan potensi besar yang dapat dilakukan masyarakat Indonesia untuk mendonasikan barang-barang mereka yang tak terpakai di rumah.

Berdasarkan masalah yang telah dijelaskan serta melihat perkembangan teknologi saat ini, maka penulis menawarkan solusi berupa pembuatan aplikasi donasi barang berbasis *mobile*. Aplikasi *mobile* dipilih karena *smartphone* merupakan salah satu perangkat yang paling banyak digunakan di Indonesia. Berdasarkan data dari *We are social* dan *Hootsuite* pada Januari 2019, ada 355.5 juta *smartphone* yang digunakan oleh masyarakat di Indonesia atau sekitar 133% lebih banyak dibandingkan jumlah penduduk Indonesia yang hanya 269 juta jiwa [5]. Aplikasi ini dibuat dengan memanfaatkan teknologi *framework React Native*. *React Native* adalah sebuah *framework Javascript* yang dikembangkan oleh *facebook* yang memungkinkan kita membuat aplikasi *mobile* dengan Sistem Operasi Android maupun IOS [6].

2. TINJAUAN PUSTAKA

2.1 Donasi Barang

Donasi adalah gerakan kemanusiaan dengan memberi bantuan secara suka rela kepada orang-orang yang membutuhkan seperti para korban bencana alam, anak yatim piatu, dan sebagainya [7]. Donasi dapat dikatakan sebagai pertolongan yang diberikan oleh para dermawan. Barang adalah segala sesuatu yang berwujud atau berjasad (KBBI). Barang memiliki nilai guna. Nilai guna dari barang ini yang dibutuhkan oleh para calon penerima donasi. Jadi, donasi barang adalah gerakan kemanusiaan dengan memberi bantuan secara suka rela dalam bentuk barang. Donasi tidak hanya berupa uang melainkan dapat berupa barang. Barang yang diberikan merupakan barang yang masih layak pakai.

2.2 Aplikasi Mobile

Perangkat lunak aplikasi atau *application software* terdiri dari beberapa program yang didesain untuk membuat

penggunanya lebih produktif dan/atau membantu dalam melakukan tugas [8]. Program aplikasi merupakan *software* yang banyak digunakan untuk membantu menyelesaikan tugas tertentu. Aplikasi *mobile* adalah aplikasi perangkat lunak yang dibuat khusus untuk dijalankan di dalam tablet dan juga *smartphone*. Aplikasi *mobile* memiliki karakteristik yang berbeda dengan aplikasi lainnya. Aplikasi *mobile* yang baik mampu menyajikan informasi dan layanan secara efisien dan praktis [9]. Jadi, aplikasi *mobile* adalah perangkat lunak yang dibuat dengan program-program aplikasi untuk tujuan tertentu, yang dijalankan khusus di dalam *tablet* atau *smartphone*.

2.3 Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti *smartphone*. Android adalah sistem operasi *open source*, dan Google merilis kodenya di bawah Lisensi Apache.

2.4 Framework React Native

Framework adalah kerangka kerja. *Framework* juga dapat diartikan sebagai kumpulan *script* (terutama *class* dan *function*) yang dapat membantu *developer/programmer* dalam menangani berbagai masalah-masalah pemrograman, sehingga pekerja *developer* lebih fokus dan lebih cepat dalam membangun aplikasi [10]. Jadi, *framework* adalah kumpulan perintah dasar untuk membantu *developer* dalam membangun aplikasi.

Banyak macam *framework*, salah satunya yaitu *react native*. *React native* adalah *framework javascript* untuk membuat aplikasi *mobile*, seperti iOS dan android [11]. *React native* adalah *platform* yang gratis untuk mengembangkan aplikasi *mobile native*, sebagian besar dikembangkan oleh Facebook. Bagian yang keren bekerja menggunakan *react native* adalah aplikasi yang dibangun seluruhnya *native* [12]. *React native* berfungsi sebagai *framework* untuk membangun aplikasi berbasis *mobile* dengan menggunakan bahasa pemrograman *javascript* [19]. Fitur pada *react native*, yaitu [11]:

- a. *React*, ini adalah *framework* untuk membangun aplikasi *web* dan *mobile* menggunakan *JavaScript*.
- b. *Native*, dapat menggunakan *native components controlled* dari *JavaScript*.
- c. *Platform*, mendukung *platform* iOS dan Android.

Keuntungan menggunakan *react native* yaitu [11]:

- a. *JavaScript*, dapat menggunakan pengetahuan *JavaScript* yang ada untuk membangun aplikasi seluler.
- b. *Code sharing*, dapat membagikan sebagian besar kode di berbagai *platform*.
- c. *Community*, banyak komunitas *react* dan *react native*, sehingga memudahkan ketika ingin bertanya mengenai kendala dan masalah yang dihadapi selama proses *develop* aplikasi menggunakan *react native*.

2.5 RESTful API

Web service adalah sistem perangkat lunak untuk menyediakan suatu layanan kepada sistem lain dari berbagai *framework* dan *platform*. Terdapat beberapa jenis *web service*, salah satunya yaitu *RESTful web service*. *RESTful web service* atau juga dikenal dengan nama *RESTful Web API* merupakan sebuah *web service* yang diimplementasikan menggunakan http dengan prinsip-prinsip *REST*. Service yang digunakan menggunakan method milik http antara lain GET, PUT, POST or DELETE [13].

REST adalah pola desain untuk berinteraksi dengan sumber daya yang disimpan di server [18]. Empat metode HTTP utama menentukan empat operasi yang umumnya dilaksanakan oleh *RESTful web service*. Metode HTTP POST digunakan untuk membuat sumber data, GET digunakan untuk *query*, PUT digunakan untuk mengubah data, dan DELETE digunakan untuk menghapus data.

2.6 Penelitian Terkait

Penelitian yang dilakukan oleh [14] merancang aplikasi berbasis *crowdsourcing* dalam upaya penanggulangan penduduk miskin. Aplikasi dibuat dalam bentuk *website*, menggunakan bahasa pemrograman PHP. *Framework* yang digunakan adalah *CodeIgniter*. Proses bisnis aplikasi yang dibuat adalah kegiatan donasi. Donasi yang diberikan berupa uang. Donasi ditujukan untuk membantu penduduk miskin.

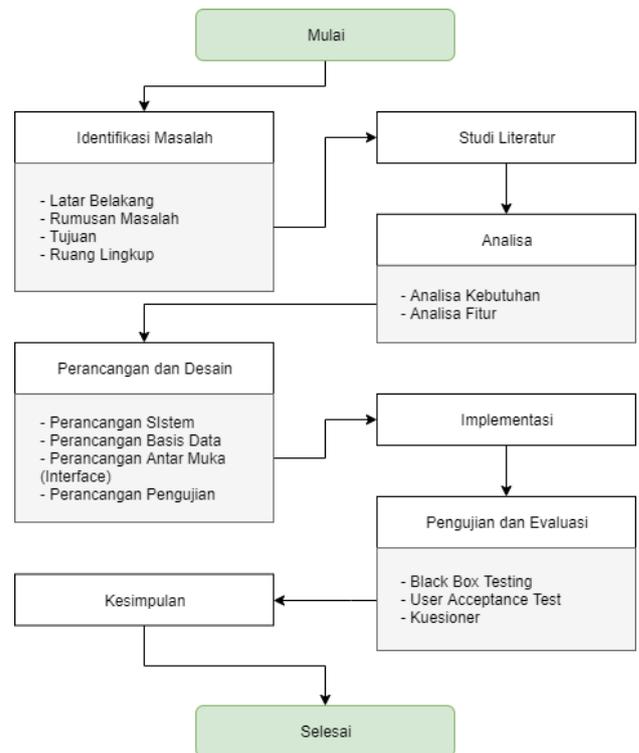
Selanjutnya, terdapat penelitian yang dilakukan oleh [15] melakukan rancang bangun sistem informasi pengumpulan dana panti asuhan menggunakan metode *crowdfunding* dengan model situs donasi. Aplikasi dibuat dalam bentuk *website*, menggunakan pemrograman PHP dengan *framework Yii2*. Aplikasi dibuat untuk penggalangan donasi bagi panti asuhan. Donasi yang diberikan berupa uang.

Sementara itu, penelitian yang dilakukan oleh [16] melakukan pengembangan sistem informasi penggalangan donasi pada Yayasan Gerakan Nurani Orang Tua Asuh (GN-OTA) Kabupaten Lumajang. Aplikasi yang dibuat pada penelitian ini berbasis web, menggunakan *framework CodeIgniter*, dengan model MVC. Aplikasi ini dibuat untuk penggalangan donasi dan publikasi bagi yayasan GN-OTA. Donasi yang diberikan berupa uang.

3. METODE PENELITIAN

3.1 Tahapan Penelitian

Tahapan penelitian adalah proses penelitian yang terdiri dari tahapan identifikasi masalah, studi literatur, analisa, perancangan dan desain, implementasi, pengujian dan evaluasi, serta yang terakhir adalah kesimpulan. Berikut pada gambar tersebut adalah tahapan-tahapan penelitian:



Gambar 1. Flowchart Metodologi Penelitian

3.2 Pengumpulan Data

Metode pengumpulan data yang dilakukan pada penelitian ini adalah studi kepustakaan dan observasi. Studi kepustakaan berfokus untuk mengkaji lebih dalam mengenai aplikasi *mobile* dengan menggunakan *framework react native*. Observasi dilakukan dengan memperhatikan lingkungan sekitar, kejadian-kejadian yang berkaitan dengan penelitian seperti kegiatan sosial dan berdonasi untuk membantu korban bencana alam.

3.3 Jenis Penelitian

Jenis dari penelitian ini adalah pembangunan aplikasi berbasis *mobile*. Metode penelitian yang digunakan adalah metode penelitian kualitatif.

3.4 Metode Pengujian

Pengujian dilakukan dengan metode *black box testing* [17]. Pendekatan yang dilakukan dalam pengujian aplikasi ini yaitu *alpha testing* [17]. *Alpha testing* yaitu pengujian yang dilakukan oleh pihak pengembang sebelum dirilis ke pengguna eksternal atau pelanggan. Selain *black box testing* [17], pengujian yang dilakukan yaitu UAT dan kuesioner.

4. IMPLEMENTASI DAN PENGUJIAN

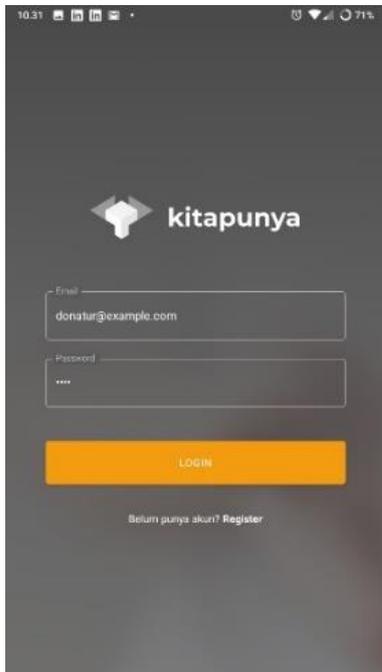
4.1 Implementasi

Implementasi adalah proses pengembangan aplikasi yang telah dirancang pada tahapan sebelumnya, sehingga diperoleh hasil berupa aplikasi yang sesuai dengan

kebutuhan yang telah didefinisikan pada tahapan Analisis dan Perancangan Sistem.

4.1.1 Implementasi Antarmuka untuk Donatur

a. Halaman *Login dan Register*



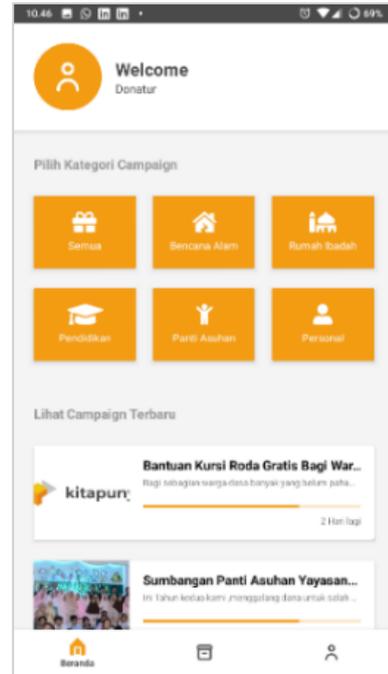
Gambar 2. Halaman *Login*



Gambar 3. Halaman *Register*

Gambar 2 dan 3 merupakan implementasi antarmuka *login* dan *register* donatur. *User* dapat melakukan *login* ataupun *register* dengan mengisi informasi pada *form* yang disediakan. *User* yang berhasil *login* maupun *register* akan diarahkan ke halaman *Home Screen*.

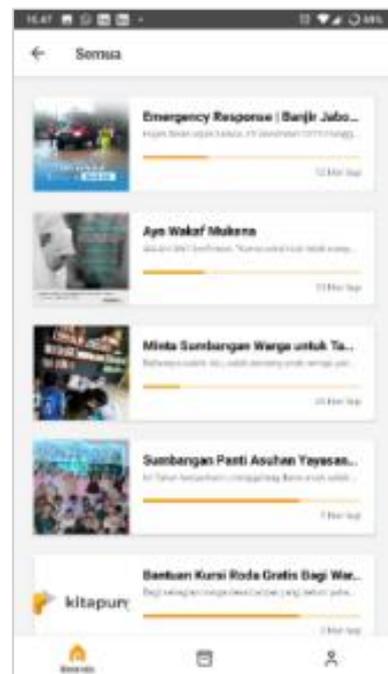
b. Halaman Beranda



Gambar 4. Halaman Beranda

Gambar 4 merupakan implementasi halaman beranda. Halaman beranda terdiri dari beberapa bagian yaitu *header* yang berisikan informasi user berupa nama dan foto profil user. Kemudian menu kategori berisikan menu dari kategori dari *campaign* yang ada, dimana apabila dipilih salah satunya maka akan menampilkan *list campaign* berdasarkan kategori yang dipilih. Terakhir adalah *list campaign* terbaru, dimana berisikan *campaign* terbaru.

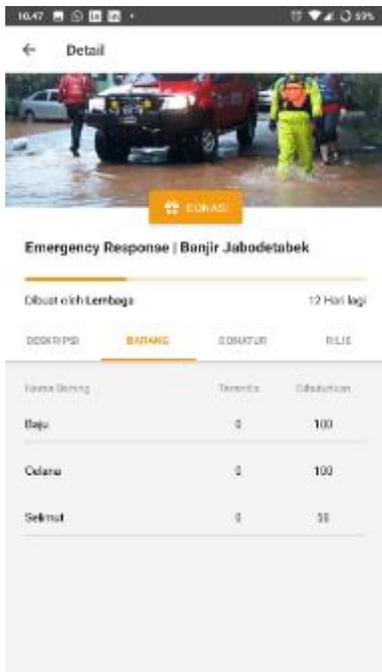
c. Halaman *List Campaign*



Gambar 5. Halaman *List Campaign*

Halaman *list campaign* berisikan daftar *campaign* yang ada berdasarkan kategori yang dipilih. Daftar *campaign* berisikan judul, foto, deskripsi dengan pembatasan karakter, serta sisa hari dari setiap *campaign*

d. Halaman Detail *Campaign*



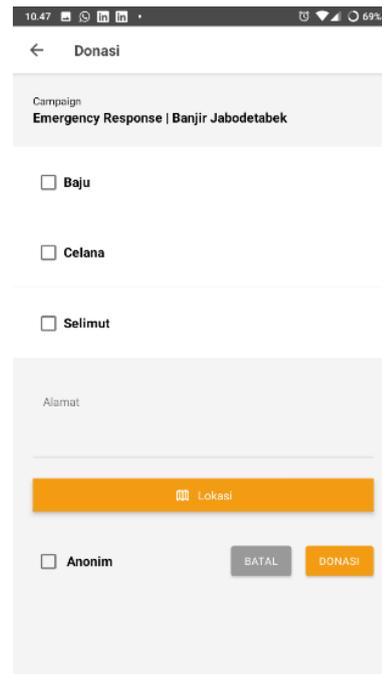
Gambar 6. Halaman Detail *Campaign* (a)



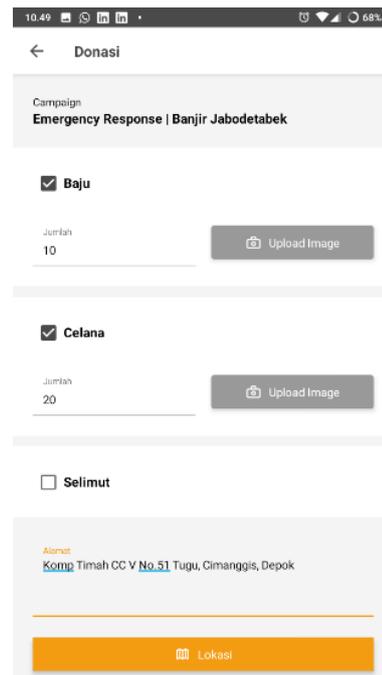
Gambar 7. Halaman Detail *Campaign* (a)

Halaman detail *campaign* berisikan informasi dari *campaign* yang dibuat seperti judul, foto, *campaigner*, sisa hari, deskripsi, daftar barang yang dibutuhkan, daftar donatur, serta rilis *campaign*. Di halaman ini juga terdapat sebuah tombol donasi yang selanjutnya akan mengarah ke Halaman Donasi

e. Halaman Donasi



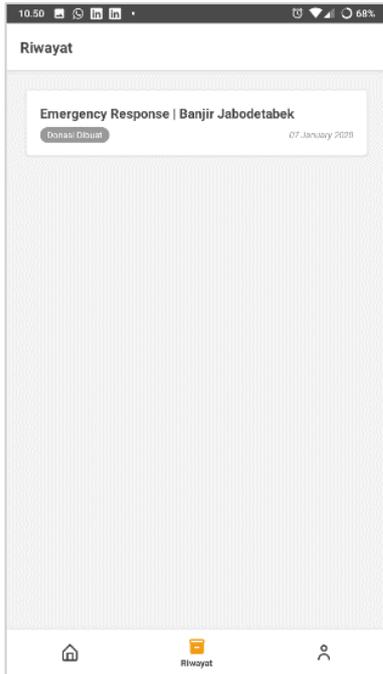
Gambar 8. Halaman Donasi (a)



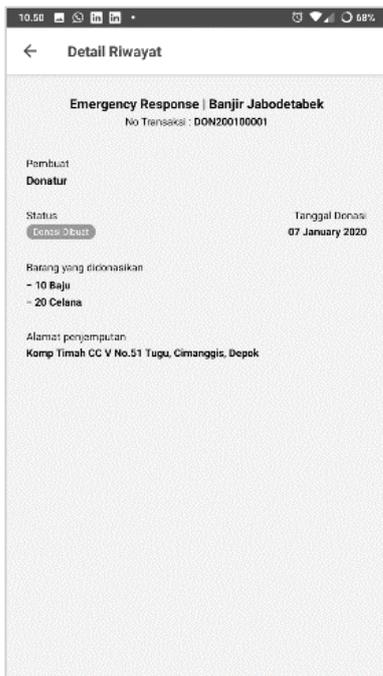
Gambar 9. Halaman Donasi (b)

Pada halaman ini *user* dapat memasukkan daftar barang yang akan didonasikan. Pertama *user* akan memilih barang yang akan didonasikan dengan menekan *checkbox* yang ada pada setiap barang, kemudian *user* memasukkan jumlah barang dan foto barangnya. Setelah itu *user* memasukkan alamat dan lokasi GPS yang menjadi titik penjemputan barang. Jika anonim dipilih maka donasi yang dilakukan bersifat anonim, jadi nama *user* tidak akan terlihat di detail *campaign*.

f. Halaman Riwayat Donasi



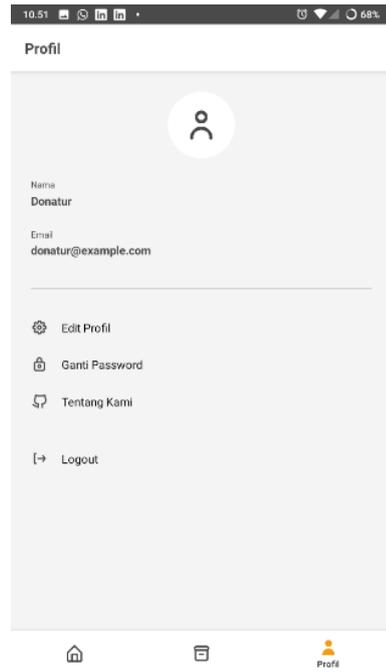
Gambar 10. Halaman Riwayat Donasi (a)



Gambar 11. Halaman Riwayat Donasi (b)

Gambar 10 dan 11 merupakan implementasi halaman riwayat donasi dan detailnya. Setiap daftar riwayat berisi judul *campaign*, status donasi, dan tanggal dilakukannya donasi. Halaman detail riwayat berisi detail riwayat donasi yang dibuat oleh *user* yang terdiri dari judul *campaign*, pembuat *campaign*, status donasi, tanggal dilakukannya donasi, barang dan jumlah barang yang didonasikan serta titik jemput barang.

g. Halaman Profil



Gambar 12. Halaman Profil

Pada gambar 12 dapat dilihat implementasi dari halaman profil yang berisi detail nama, email dan profil *user*.

4.1.2 Implementasi Antarmuka untuk *Driver*

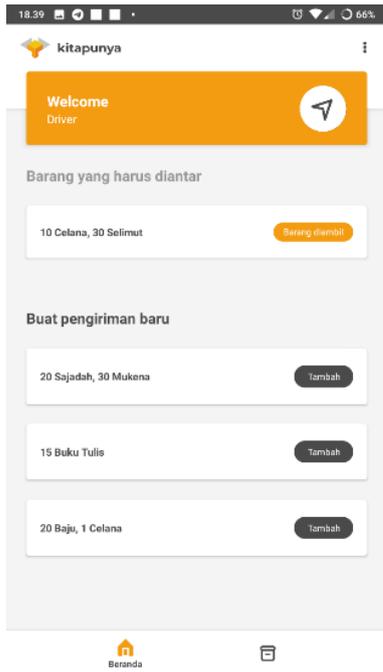
a. Halaman *Login*



Gambar 13. Halaman *Login*

Pada gambar 13 dapat dilihat implementasi antarmuka dari halaman *login*. *User* dapat melakukan *login* dengan mengisi informasi pada *form* yang disediakan. *User* yang berhasil *login* akan diarahkan ke halaman beranda.

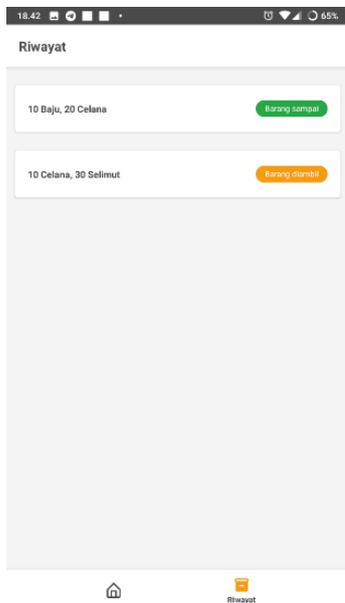
b. Halaman Beranda



Gambar 14. Halaman Beranda

Halaman beranda terdiri dari beberapa bagian. Bagian pertama berisikan nama *user*, daftar donasi yang harus diantar serta daftar donasi yang belum diambil. Daftar donasi berisikan nama dan jumlah barang tiap donasi, serta terdapat status dari donasi tersebut.

c. Halaman Riwayat Donasi



Gambar 15. Halaman Riwayat

Pada gambar 15 dapat dilihat implementasi antarmuka dari Riwayat Screen yang berisikan riwayat pengantaran donasi dari setiap *user*. Daftar donasi berisikan nama dan jumlah barang tiap donasi, serta terdapat status dari donasi tersebut.

4.2 Pengujian

Berdasarkan fitur-fitur yang telah dikembangkan, dilakukan pengujian dengan metode *black-box* sebagai berikut:

Tabel 1. Hasil Pengujian menggunakan *Black-Box*

Rencana Pengujian	Hasil yang diharapkan	Hasil Pengujian
User melakukan registrasi	User dapat melakukan registrasi	Sesuai
	Jika email sudah terdaftar maka registrasi ditolak	Sesuai
User melakukan login	User berhasil melakukan login untuk masuk kedalam aplikasi	Sesuai
User melakukan logout	User berhasil melakukan logout	Sesuai
Donatur melihat beranda aplikasi	User dapat melihat informasi dasar berupa nama dan avatar <i>user</i> yang telah login	Sesuai
	Terdapat 6 kategori <i>campaign</i>	Sesuai
	User dapat melihat <i>campaign</i> terbaru sebanyak 5 <i>campaign</i>	Sesuai
	User dapat melihat daftar <i>campaign</i> berdasarkan kategori yang dipilih sebelumnya	Sesuai
Donatur melihat daftar <i>campaign</i>	Setiap item dari daftar <i>campaign</i> menampilkan judul, deskripsi, sisa waktu serta menampilkan <i>progress bar</i> sesuai dengan hari yang telah dilalui per batas waktu <i>campaign</i>	Sesuai
	User dapat melihat informasi dasar <i>campaign</i> seperti judul, gambar, <i>campaigner</i> , sisa hari <i>campaign</i>	Sesuai
	User dapat melihat deskripsi <i>campaign</i>	Sesuai
Donatur melihat detail <i>campaign</i>	User dapat melihat detail barang yang dibutuhkan	Sesuai
	User dapat melihat siapa saja donatur yang telah melakukan donasi di <i>campaign</i> tersebut	Sesuai
Donatur melakukan proses donasi	User berhasil melakukan donasi	Sesuai
Donatur melihat riwayat donasi	User dapat melihat riwayat donasi	Sesuai
	User dapat melihat detail donasi	Sesuai
Donatur melakukan perubahan data akun	User berhasil mengubah data akun	Sesuai
	User berhasil mengganti password	Sesuai
Driver melihat beranda aplikasi	User dapat melihat informasi dasar berupa nama <i>user</i> yang telah login	Sesuai
	User berhasil melihat daftar barang yang harus di antarkan	Sesuai
Driver melihat riwayat penjemputan	User dapat melihat daftar riwayat penjemputan	Selesai
	User dapat melihat detail riwayat penjemputan	Selesai
Driver melakukan penjemputan barang	User berhasil melihat detail barang yang harus diantar	Sesuai
	User berhasil mengubah status donasi barang	Sesuai

Selanjutnya dilakukan pengujian menggunakan metode UAT dimana pengujiannya dibagi berdasarkan aktor yang sudah didefinisikan sebelumnya yaitu donatur dan *driver*.

UAT untuk donatur memiliki 13 pertanyaan yang diuji oleh *user*, dimana setiap fitur akan memiliki hasil sesuai atau tidak sesuai berdasarkan dari hasil pengujian aplikasi. Dari 3 user yang melakukan UAT untuk donatur, semua fitur yang diuji oleh user memiliki hasil sesuai. Berikut adalah persentase keberhasilan dari UAT untuk donatur:

$$\text{Presentase Keberhasilan} = \frac{39}{13 \times 3} \times 100\% = 100\%$$

UAT untuk *driver* memiliki 9 pertanyaan yang diuji oleh *user*, di mana setiap fitur akan memiliki hasil sesuai atau tidak sesuai berdasarkan dari hasil pengujian aplikasi. Dari 2 user yang melakukan UAT untuk *driver*, semua fitur yang diuji oleh user memiliki hasil sesuai. Berikut adalah persentase keberhasilan dari UAT untuk *driver*

$$\text{Presentase Keberhasilan} = \frac{27}{9 \times 3} \times 100\% = 100\%$$

Terakhir dilakukan pengujian dengan menyebar *form* kuisisioner. Kuisisioner terdiri dari 7 pertanyaan yang akan membuktikan tercapat atau tidaknya tujuan penelitian ini. Berikut adalah hasil interpretasi dari kuisisioner

Tabel 2. Hasil Interpretasi Kuisisioner

Pertanyaan	Skor	Rumus	Hasil Interpretasi
Pernyataan 1	50	$50 / (5 \times 12) \times 100\%$	83,3%
Pernyataan 2	50	$50 / (5 \times 12) \times 100\%$	83,3%
Pernyataan 3	57	$57 / (5 \times 12) \times 100\%$	95%
Pernyataan 4	55	$55 / (5 \times 12) \times 100\%$	92%
Pernyataan 5	48	$48 / (5 \times 12) \times 100\%$	90%
Pernyataan 6	53	$53 / (5 \times 12) \times 100\%$	88,3%
Pernyataan 7	54	$54 / (5 \times 12) \times 100\%$	90%

Berdasarkan hasil dari kuisisioner yang telah dilakukan maka didapatkan skor rata-rata sebesar 87,4%.

5. KESIMPULAN

Berdasarkan hasil penelitian, dapat diambil beberapa kesimpulan dari penelitian ini, yaitu sebagai berikut:

1. Melalui metodologi *waterfall* penulis mampu merancang aplikasi donasi barang berbasis *mobile* menggunakan *framework react native* melalui tahapan tahapan antara lain, identifikasi masalah, studi kepustakaan, analisa, perancangan dan desain, implementasi, pengujian dan evaluasi serta kesimpulan.
2. Berdasarkan hasil kuisisioner dan UAT untuk donatur yang telah dilakukan bahwa aplikasi yang dikembangkan mampu memudahkan masyarakat untuk melakukan donasi berupa barang dengan rata rata hasil kuisisioner sebesar 87,4% dan hasil UAT sebesar 100%.
3. Berdasarkan hasil UAT untuk *driver* yang telah dilakukan bahwa aplikasi yang dikembangkan mampu mempermudah proses penjemputan barang dari lokasi

penjemputan ke tujuan dengan rata rata hasil UAT sebesar 100%.

4. Berdasarkan hasil survey kuisisioner menyatakan bahwa 90% responden menyatakan bahwa aplikasi yang dikembangkan mempermudah melakukan donasi barang serta 92% responden menyatakan bahwa aplikasi yang dikembangkan dapat memperoleh informasi sebuah *campaign* dengan jelas.

DAFTAR PUSTAKA

- [1] B. N. P. B. (BNPB), "Tren Kejadian Bencana 10 Tahun Terakhir," Data Informasi Bencana Indonesia (DBI), 30 Maret 2019. [Online]. Available: <https://bnpb.cloud/dibi/>. [Accessed 22 Maret 2019].
- [2] R. F. Hilda Amalia, "Sistem Informasi Pengolahan Dana Donasi," *Perspektif*, pp. Vol. XV, No. 1, Maret 2017, 2017.
- [3] VOA Indonesia, "Kami Lapar, Habis Kami": 5 Kebutuhan Darurat & Cara Bantu Korban Gempa Palu," 3 Oktober 2018. [Online]. Available: <https://www.voaindonesia.com/a/gempa-tsunami-palu-donggala-kebutuhan-darurat-cara-bantu-korban/4596886.html>.
- [4] A. Anggoro, "Transaksi Barang Preloved Makin Digemari," 14 Desember 2017. [Online]. Available: <https://swa.co.id/swa/trends/transaksi-barang-preloved-makin-digemari>.
- [5] W. A. S. d. Hootsuite, "Digital In 2019," Januari 2019. [Online]. Available: <https://wearesocial.com/global-digital-report-2019>
- [6] React-Native, "React Native," 2019. [Online]. Available: <https://facebook.github.io/react-native/>.
- [7] D. R. Anggraini, "Rancang Bangun Aplikasi Donasi Berbasis Website pada BSMI Cabang Surabaya," Institut Bisnis dan Informatika STIKOM Surabaya: Surabaya, 2018.
- [8] E. Irwansyah and J. V. Moniaga, "Pengantar Teknologi Informasi," Yogyakarta: Deepublish, 2014.
- [9] H. Tolle, A. Pinandito, A. P. Kharisma and R. K. Dewi, "Pengembangan Aplikasi Perangkat Bergerak (Konsep dan Implementasi)," Malang: UBMedia, 2017.
- [10] Y. Yudhanto and H. A. Prasetyo, "Mudah Menguasai Framework Laravel," Jakarta: PT Elex Media Komputindo, 2019.
- [11] AMC College, "React Native: Hybrid Mobile App Development," Information & Communication Technology (ICT), 2017.

- [12] A. Paul and A. Nalwaya, "React Native for Mobile Development: Harness the Power of React Native to Create Stunning iOS and Android Application," India: Apress, 2019.
- [13] S. Mutmainah, "Pengembangan Pessta+ Berbasis Yii Framework Sebagai Media Validasi Sertifikat dalam Penilaian Objektif Sidang di Perguruan Tinggi," Sekolah Tinggi Manajemen dan Ilmu Komputer (STMIK) Raharja, Tangerang, 2017.
- [14] A. K. A. Pradana, "Pembuatan Aplikasi Berbasis *Crowdsourcing* dalam Upaya Penanggulangan Penduduk Miskin," *Jurnal Informatika & Multimedia*, vol. 08, pp. 1-9, 2016.
- [15] I. D. P. Deni Fadjri, "Rancang Bangun Sistem Informasi Pengumpulan Dana Panti Asuhan menggunakan Metode *Crowdfunding* dengan Model Situs Donasi," *Coding : Jurnal Komputer dan Aplikasi*, vol. 07, no. 01, pp. 64-73, 2019.
- [16] A. H. B. Y. T. M. Yuris Socio Perestroika, "Pengembangan Sistem Informasi Penggalangan Donasi pada Yayasan Gerakan Nurani Orang Tua Asuh (GN-OTA) Kabupaten Lumajang," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 5, pp. 1833-1840, 2018.
- [17] J. Simarmata, "Rekayasa Perangkat Lunak," Yogyakarta: Andi Offset, 2010.
- [18] M. A. Arianto, "Analisis dan Perancangan *Representational State Transfer (REST)* Web Service Sistem Informasi Akademik STT Terpadu Nurul Fikri Menggunakan Yii Framework", *J. Teknologi Terpadu*, Vol. 2, No. 2, 2017.
- [19] I. F. Abiyyu, dan H. A. Tawakal, "Pengembangan Aplikasi Pendeteksi Kematangan Buah Melon: Studi Kasus Aplikasi Melonku," *J. Informatika Terpadu*, Vol. 7, No. 1, pp. 27-32, 2021. <https://doi.org/10.54914/jit.v7i1.331>



ANALISIS DAN PERANCANGAN APLIKASI PENGANGGARAN BARANG BERBASIS WEB PADA UNIT SARANA PRASARANA PERGURUAN TINGGI

Siti Zahra Maharani¹, Ahmad Rio Adriansyah²

^{1,2} Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jakarta Selatan, DKI Jakarta, Indonesia 12640
maharanisitizahra@gmail.com, arasy@nurulfikri.ac.id

Abstract

This study discusses the budget approval process at the Nurul Fikri Integrated Technology High School Facilities and Infrastructure Unit, which is inflexible because it requires a physical supervisor's signature, management of the procurement of goods that have not been carried out centrally and is expected to reduce delays in the fulfillment of goods and repetitive work. Therefore, an STT-NF Goods Budgeting application is needed to facilitate submissions and streamline time in meeting the needs of goods and data backup. The method used in designing the STT-NF Goods Budgeting application uses a literature review, interviews with the head of the Integrated High School Facilities and Infrastructure Section, Nurul Fikri, and the Incremental Development System method in stages. The study results showed the suitability of the features as expected; 96% of users stated that the features in the information system were by the process of submitting existing goods..

Keywords: *Budgeting of Goods, Facilities and Infrastructure Unit, Incremental Development System, Website*

Abstrak

Penelitian ini membahas tentang proses persetujuan anggaran di Unit Sarana dan Prasarana Sekolah Tinggi Teknologi Terpadu Nurul Fikri yang tidak fleksibel karena memerlukan tanda tangan atasan secara fisik, pengelolaan pengadaan barang yang belum dilakukan secara terpusat dan diharapkan dapat mengurangi keterlambatan dalam pemenuhan barang serta pengerjaan yang berulang. Oleh karena itu, diperlukan aplikasi Penganggaran Barang STT- NF yang dapat memudahkan pengajuan serta mengefektifkan waktu dalam memenuhi kebutuhan barang serta pencadangan data. Metode yang digunakan dalam perancangan aplikasi Penganggaran Barang STT-NF menggunakan kajian literatur, wawancara kepada kepala Bagian Sarana dan Prasarana Sekolah Tinggi Terpadu Nurul Fikri dan juga menggunakan metode *Incremental Development System* secara bertahap. Hasil dari penelitian diketahui bahwa kesesuaian fitur dengan yang diharapkan, 96% pengguna menyatakan bahwa fitur dalam sistem informasi sesuai dengan proses pengajuan barang yang sudah ada.

Kata Kunci: *Penganggaran Barang, Unit Sarana dan Prasarana, Incremental Development System, Website*

1. PENDAHULUAN

Teknologi informasi seperti memanjakan kita sebagai penggunanya untuk menyelesaikan pekerjaan perusahaan baik negeri maupun swasta dalam skala kecil, sedang, ataupun besar akan mudah dikerjakan dengan bantuan teknologi informasi. Karena dengan teknologi informasi yang dihasilkan dapat membantu pekerjaan manusia dengan kecepatan dan akurasi yang tinggi.

Sekolah Tinggi Teknologi Terpadu Nurul Fikri (STT-NF) merupakan Sekolah Tinggi Teknologi yang berdiri sejak 2012 di Depok. STT-NF mempunyai struktur organisasi dalam menjalankan kegiatan operasional institusi yaitu Ketua, Wakil Ketua I Bidang Akademik,

Wakil Ketua II Bagian Sarana dan Prasarana (BSP) dan Keuangan, serta Wakil Ketua III Bidang Penerimaan Mahasiswa Baru (PMB) dan Kemahasiswaan [1].

Wakil Ketua II Bagian Sarana dan Prasarana (BSP) mempunyai beberapa kegiatan operasional antara lain proses pengajuan penganggaran barang dan dana. Saat ini sistem pengajuannya masih secara manual yaitu menggunakan kertas atau belum sepenuhnya dengan komputerisasi. Hal tersebut cukup menyulitkan pegawai Bagian Sarana dan Prasarana karena memerlukan proses tambahan dalam melakukan pencatatan barang yaitu secara manual dan mem-backup ke *Dashboard* Anggaran Barang di komputer.

Untuk mengefisienkan proses pengajuan barang tersebut, dapat dilakukan pengisian secara online. Dimana pengaju hanya perlu memilih barang-barang yang diperlukan dengan harga yang sudah tercantum dan persetujuan atasan hanya dengan memilih tombol *approved* atau *not approved*.

2. METODE PENELITIAN

2.1 Alur Tahapan Penelitian

Alur tahapan penelitian yang digunakan pada perancangan aplikasi Penganggaran Barang STT-NF menggunakan metode *Incremental Development*. Alur tahapan pada model *incremental* ini harus dilakukan secara berurutan dari tahapan pertama, kemudian dapat dilanjutkan ke tahapan-tahapan berikutnya seperti berikut ini [2]:

- a. Tahap *Requirements* atau Kebutuhan
Tahapan pertama pada model *Incremental Development* adalah *requirements* yang bertujuan untuk mengumpulkan data serta informasi yang dibutuhkan agar menjawab perumusan masalah sebelumnya [8]. Pengumpulan data pada penelitian ini dapat berupa wawancara dan sumber penelitian yang terkait.
- b. Tahap *Incremental* atau Tambahan
Pada tahap *incremental* ini, beberapa tahapan yang dilakukan dalam perancangan aplikasi penganggaran barang STT-NF yaitu membuat beberapa *incremental*. *Incremental* (1) adalah versi awal yang dibangun dari produk atau fitur terakhir. Sedangkan *incremental* (2) merupakan hasil implementasi dari *incremental* sebelumnya yaitu *incremental* (1) dengan berbagai tambahan yang diminta oleh pengguna. Begitu juga dengan *incremental* (3) dan seterusnya, dimana *incremental* tersebut merupakan hasil pengolahan dari *incremental-incremental* sebelumnya.
- c. Tahap *Specification* atau Spesifikasi
Setelah membuat beberapa *incremental*, selanjutnya adalah proses spesifikasi setiap *incremental* yang sudah dibuat. Dimana menggunakan analisis kebutuhan sebagai acuannya [7].
- d. Tahap *Architecture Design* atau Desain
Arsitektur Perancangan desain arsitektur pada *software* berfungsi sebagai penjelasan bentuk sistem yang dimulai dengan desain model *database*, desain antarmuka, desain tabel, desain input dan desain output dalam mengembangkan aplikasi Penganggaran Barang STT-NF.
- e. Tahap *Code* atau Kode
Tahapan kode ini adalah implementasi dari tahapan desain sistem pada program. Pengkodean sangat dibutuhkan dalam pengembangan aplikasi Penganggaran Barang STT-NF untuk membuat suatu fitur atau *incremental* yang sudah dirancang dapat

berjalan dengan baik dan sesuai.

- f. Tahap Tes atau Uji
Tahap terakhir pada model *incremental development* ini adalah tahap pengujian. Hal ini dilakukan untuk memastikan semua fitur pada aplikasi Penganggaran Barang STT-NF yang sudah dirancang dan dibangun dari tahap *Requirements*, tahap *Incremental*, tahap *Specification*, tahap *Architecture Design* sampai tahap *Code* dapat berfungsi sesuai dengan yang pengguna inginkan.

2.2 Teknik Penelitian

- a. Studi Literatur
Studi literatur dibutuhkan untuk menelusuri sumber-sumber tulisan yang telah disusun sebelumnya yang akan dipergunakan oleh penulis untuk membandingkan, melengkapi ataupun mengumpulkan suatu literatur.
- b. Wawancara
Penulis akan melakukan wawancara dengan memberikan pertanyaan-pertanyaan seputar aktifitas yang dilakukan oleh pengaju kepada bagian sarana dan prasarana serta kebutuhan apa saja yang diinginkan pengguna dalam aplikasi penganggaran barang.

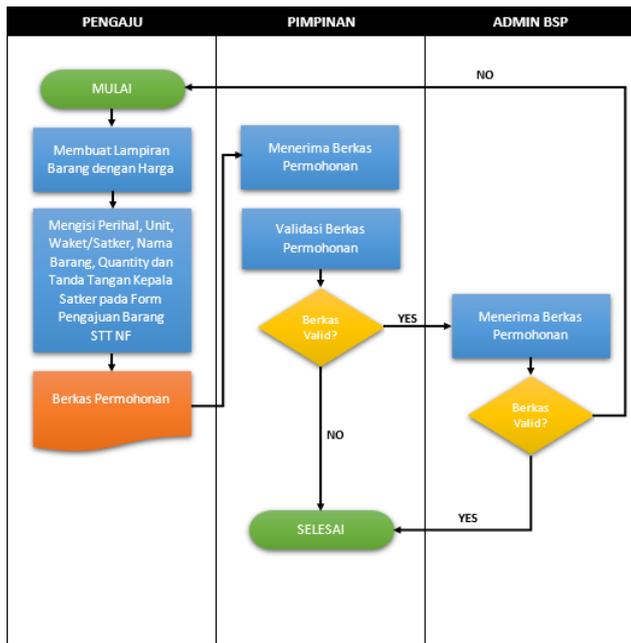
2.3 Teknik Pengujian

Tahap pengujian ini dilakukan apabila proses implementasi sudah dilakukan. Tujuannya untuk menjawab perumusan masalah yang sudah dijelaskan sebelumnya. Pengujian aplikasi pengajuan barang ini menggunakan *Black-box Testing* yang meliputi tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi dan kesesuaian alur fungsi dengan bisnis proses yang sudah ditentukan [3]. *Black-box* yaitu melakukan pengamatan hasil eksekusi melalui data uji dan memeriksa fungsional aplikasi, apakah berjalan baik atau tidak [6].

3. HASIL DAN PEMBAHASAN

3.1 Analisis Sistem

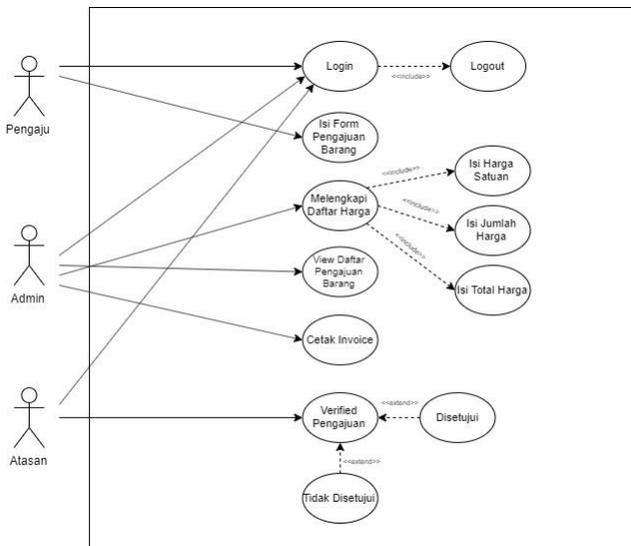
- a. Alur Pengajuan Barang
Hasil dari wawancara dengan beberapa pengguna yaitu alur proses dalam pengajuan barang di STT-NF seperti pada tabel berikut:



Gambar 1. Alur Pengajuan Barang di STT-NF

b. Use Case Diagram

Pada gambar dibawah ini menjelaskan mengenai Use Case Diagram aplikasi Penganggaran Barang STT-NF berdasarkan kebutuhan user dari hasil wawancara kepada pihak terkait.



Gambar 2. Use Case Diagram

c. Perancangan Pengujian

Berikut adalah tabel yang berisi dengan kebutuhan-kebutuhan dalam pengujian aplikasi Penganggaran Barang STT-NF:

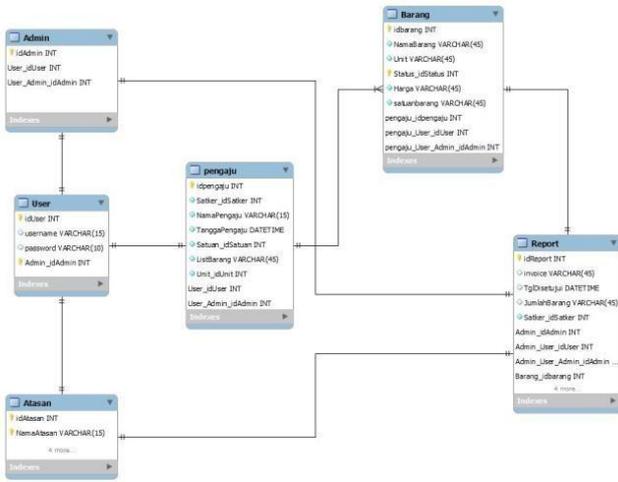
Tabel 1. Perancangan Pengujian

No	Aktor	Keterangan	Berhasil	Gagal
1	Pengaju	Melakukan login	Pindah ke halaman Pengajuan	Menampilkan notifikasi berisi kesalahan pada <i>username</i> atau <i>password</i>
2	Pengaju	Mengisi data pada form pengajuan barang	Data dikirim ke <i>database</i>	Menampilkan notifikasi untuk melengkapi data yang kosong
3	Atasan	Melakukan login	Pindah ke halaman Histori Pengajuan	Menampilkan notifikasi berisi kesalahan pada <i>username</i> atau <i>password</i>
4	Atasan	Menyetujui data pada form pengajuan barang	Data dikirim ke <i>database</i>	Menampilkan notifikasi untuk melengkapi data yang kosong
5	Admin	Melakukan login	Pindah ke halaman Histori Pengajuan	Menampilkan notifikasi berisi kesalahan pada <i>username</i> atau <i>password</i>
6	Admin	Melihat histori pengajuan barang	Pindah ke halaman berikutnya	-
7	Admin	Melengkapi data harga pada form pengajuan barang	Data dikirim ke <i>database</i>	Menampilkan notifikasi untuk melengkapi data yang kosong
8	Admin	Mengunduh form pengajuan barang	Pindah ke halaman PDF	-
9	Admin	Mencetak form pengajuan barang	Mencetak form pengajuan barang	-

3.2 Domain Model

Domain model adalah sekumpulan dari *entity* di *database* yang saling berelasi [4]. *Domain Model* adalah penggambaran objek dari suatu lingkup area atau domain melalui perwujudan ide abstraksi kelas (*conceptual class*) yang mempunyai realsi objek [5]. Domain model juga bisa disebut sebagai penggambaran obyek nyata terhadap situasi riil dari suatu lingkup area/domain melalui perwujudan ide abstraksi kelas.

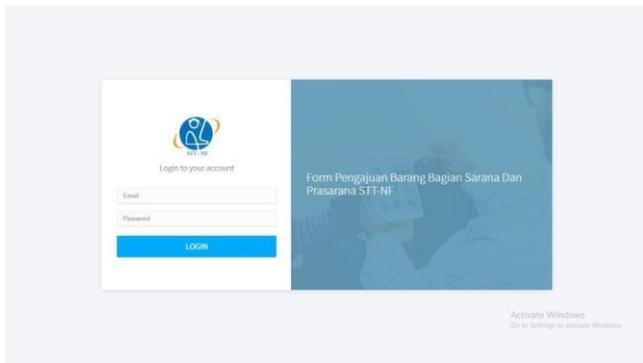
Pada gambar berikut merupakan domain model pada sistem yang menggambarkan keterhubungan antar objek, keterhubungan antar objek tersebut dihubungkan dengan link berupa garis disertai informasi *multiplicity* pada masing-masing objek.



Gambar 3. ERD Database Penganggaran Barang STT-NF

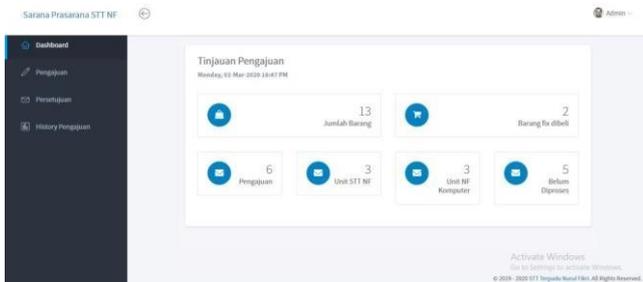
3.3 Implementasi

a. Halaman Login User



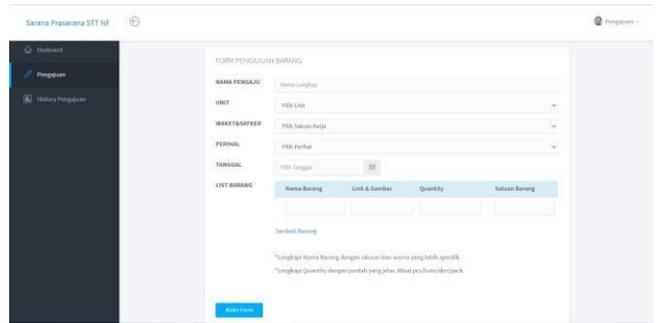
Gambar 4. Tampilan Halaman Login

b. Halaman Dashboard



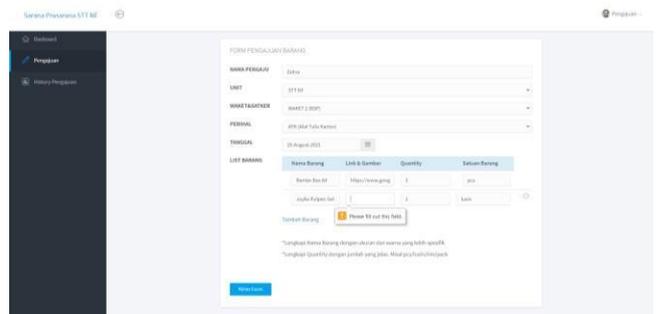
Gambar 5. Antar Muka Tinjauan Pengajuan

c. Halaman Pengajuan pada User Pengaju



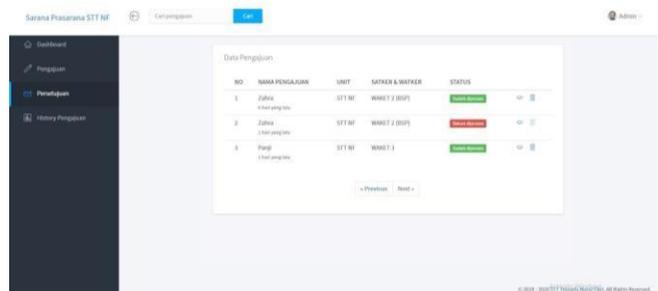
Gambar 6. Antar Muka Halaman Pengajuan pada User Pengaju

d. Notifikasi Kelengkapan Data



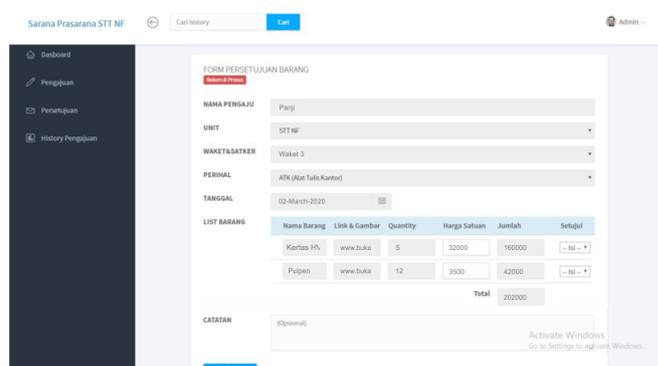
Gambar 7. Antar Muka Notifikasi Kelengkapan Data

e. Halaman Persetujuan dan Form Pengajuan Barang pada User Admin



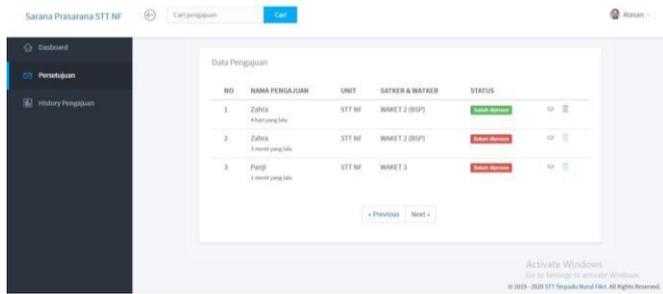
Gambar 8. Antar Muka Halaman Persetujuan pada User Admin

f. Halaman Pengajuan pada User Admin



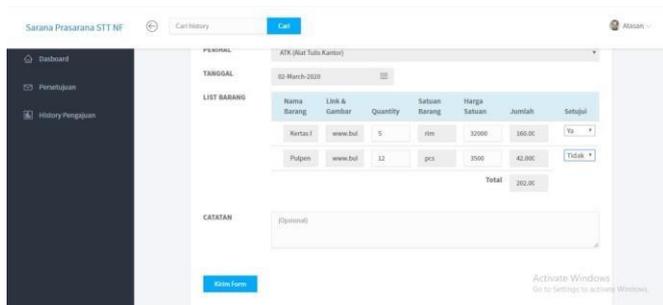
Gambar 9. Antar Muka Halaman Pengajuan pada User Admin

g. Halaman Persetujuan Barang pada *User* Atasan



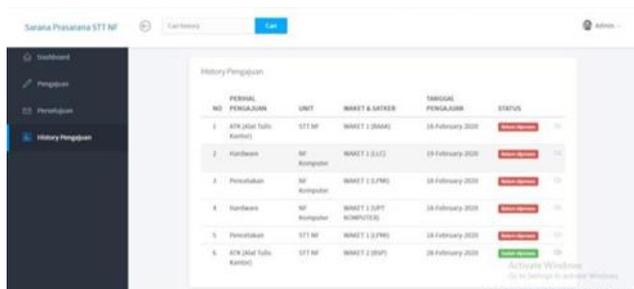
Gambar 10. Antar Muka Halaman Persetujuan pada *User* Atasan

h. Halaman Persetujuan pada *User* Atasan



Gambar 11. Antar Muka Halaman Persetujuan pada *User* Atasan

i. Halaman Pengajuan Barang



Gambar 12. Antar Muka Halaman Histori Persetujuan pada *User* Admin

j. Form Pengajuan Barang



Gambar 13. Antar Muka Halaman Pengajuan Barang Berupa PDF

3.4 *Usability Testing*

Pengujian *Usability* Aplikasi Penganggaran Barang Berbasis Web Studi Kasus Unit Sarana dan Prasarana Sekolah Tinggi Teknologi Terpadu Nurul Fikri dilakukan terhadap staff di beberapa bagian terkait informasi dengan jumlah responden sebanyak 3 orang. Pengujian dilakukan dengan menggunakan kuisioner yang berjumlah 5 pertanyaan. Adapun hasil pengujian sebagai berikut:

Tabel 2. Hasil Pengujian

Skala Penilaian	Jumlah	Skor	Jumlah x Skor
Sangat Setuju (SS)	12	5	60
Setuju (S)	3	4	12
Ragu-Ragu (R)	0	3	0
Tidak Setuju (TS)	0	2	0
Sangat Tidak Setuju (STS)	0	1	0
Nilai Total			72
Nilai Maksimal			75

Rumus untuk aspek *Usability* sebagai berikut:

$$\text{Presentase} = \frac{\text{nilai total}}{\text{nilai maksimal}} \times 100\%$$

Nilai maksimal yang didapat jika responden memilih jawaban sangat setuju dengan skor 5 dan skor minimal yang didapat adalah 1. Sehingga didapatkan persentase dari aspek *Usability* sebagai berikut:

$$\text{Presentase} = \frac{\text{nilai total}}{\text{nilai maksimal}} \times 100\% = 96\%$$

Hasil dari persentase dari pengujian aspek *Usability* sebesar 96%.

4. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan analisis, perancangan sistem, implementasi sistem hingga pengujian terhadap Aplikasi Penganggaran Barang Berbasis Web Studi Kasus Unit Sarana dan Prasarana Sekolah Tinggi Teknologi Terpadu Nurul Fikri dapat disimpulkan sebagai berikut:

1. Perancangan sistem penganggaran barang ini memiliki beberapa tahapan yaitu mencari informasi dari kajian literatur dan wawancara dengan beberapa pengguna yang akan menggunakan aplikasi tersebut. Selain itu, perancangan sistem memiliki tampilan yang sederhana dan jelas.
2. Sistem penganggaran barang di STT-NF sebelumnya masih dilakukan secara manual, dengan adanya aplikasi ini diharapkan dapat memudahkan proses pengajuan barang di STT-NF.
3. Dari hasil pengujian kesesuaian fitur dengan yang

diharapkan, 96% pengguna menyatakan bahwa fitur dalam sistem informasi sesuai dengan proses pengajuan barang yang sudah ada.

4.2 Saran

Berdasarkan analisis, perancangan sistem, implementasi sistem hingga pengujian terhadap Aplikasi Penganggaran Barang Berbasis Web Studi Kasus Unit Sarana dan Prasarana Sekolah Tinggi Teknologi Terpadu Nurul Fikri masih terdapat kekurangan yang perlu dilakukan pengembangan lebih lanjut. Penulis menyarankan beberapa hal yang dapat dikembangkan sebagai berikut:

1. Tampilan pada aplikasi penganggaran barang diharapkan lebih *user friendly*.
2. Akses aplikasi penganggaran barang tidak hanya dapat dilakukan pada jaringan internal.
3. Aplikasi penganggaran barang STT-NF diharapkan dapat dikembangkan pada *mobile*.

DAFTAR PUSTAKA

- [1] “Profil STT Terpadu Nurul Fikri,” Diakses pada 31 Oktober 2019, dari STT Terpadu Nurul Fikri, 2014.
- [2] I. Aji, “Tentang Incremental Model,” ilhamaji.blogspot.com, November 24 2014, [Online]. Tersedia: <http://ilhamaji.blogspot.com/2014/11/tentang-incremental-model.html>
- [3] Syafnidawaty, “Blackbox Testing,” *raharja.ac.id*, Oktober 20 2020, [Online]. Tersedia: <https://raharja.ac.id/2020/10/20/black-box-testing/>
- [4] T. Hartono, “Entity Relationship Diagram (ERD): Apa dan Bagaimana Cara Buatnya?” *dewaweb.com*, Jan 13 2022, [Online]. Tersedia: <https://www.dewaweb.com/blog/entity-relationship-diagram/>
- [5] R. Hardiansyah dan S. Munir, “Analisis dan Pengembangan Sistem Reservasi Online untuk Wisata berbasis Website Studi Kasus Pulau Harapan Kepulauan Seribu”, *J. Inform. Terpadu*, Vol. 6, No. 1, hlm. 45-52, Mar 2020.
- [6] M. Miftachurrohman, & S. Munir, “Analisis dan Perancangan Aplikasi Pengelolaan Perbaikan Kendaraan Dinas Badan Pengkajian dan Penerapan Teknologi berbasis Desktop,” *J. Inform. Terpadu*, Vol. 3, No. 2, pp. 81-87, 2017. <https://doi.org/10.54914/jit.v3i2.108>
- [7] R. T. Tedja, “Software Development Model: Increment Model,” *sis.binus.ac.id*, Juli 2 2019, [Online]. Tersedia: <https://sis.binus.ac.id/2019/07/02/software-development-model-incremental-model/>
- [8] E. D. Wahyuni, M. R. K. Azis, I. Nuryasin, “Implementasi Metode *Incremental* pada Sistem Informasi Administrasi Desa Jambuwer,” *J. Teknokompak*, Vol. 15, No. 2, pp. 156-167, 2021.



PENGEMBANGAN PERMAINAN EDUKASI GUNA PENGENALAN BUDAYA TAK BENDA DI INDONESIA BERBASIS *MOBILE*

Setyo Nugroho¹, Ahmad Rio Ardiansyah²

^{1,2} Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jakarta Selatan, DKI Jakarta, Indonesia 12640
sayasetyo008@gmail.com, arasy@nurulfikri.ac.id

Abstract

The game "Ksatria Budaya" is an Android-based mobile game, designed as a medium for introducing Indonesian cultural heritage. This game uses Unity as a game engine and JSON as a database. This game is a combination of side scrolling game and quiz game. To see the feasibility of this game, the author will do 3 tests, namely black-box testing, compatibility testing, and usability testing. The development of this game begins with conducting a literature study related to research, culture and also the cultural heritage itself. Questions that appear in the game are based on data on the Ministry of Education and Culture website. The Agile SCRUM method was chosen by the author in the development of this game. The stages are carried out through creating user stories, designing product backlogs, sprint planning, sprints, and sprint reviews. The results of this game test are, all the functions in the game run well, the game can be installed and run on different devices, and gets a score of 83% for usability testing. Where if it is categorized by the score interpretation criteria table, this game is included in the very feasible criteria for use.

Keywords: Informatics Engineering, Game, Android, Unity, Cultural Heritage

Abstrak

Game "Ksatria Budaya" adalah *game mobile* berbasis android, yang dirancang sebagai media pengenalan warisan budaya Indonesia. *Game* ini memanfaatkan Unity sebagai *game engine* dan JSON sebagai *database*. *Game* ini merupakan perpaduan dari *side scrolling game* dan *quiz game*. Untuk melihat kelayakan *game* ini, penulis akan melakukan 3 pengujian yaitu *black-box testing*, *compatibility testing*, dan *usability testing*. Pengembangan *game* ini diawali dengan melakukan studi literatur terkait penelitian, budaya dan juga warisan budaya Indonesia itu sendiri. Pertanyaan yang muncul dalam *game* dibuat berdasarkan data yang ada di dalam situs Kemdikbud. Metode Agile SCRUM dipilih penulis dalam pengembangan *game* ini. Tahap yang dilakukan melalui membuat *user stories*, merancang *product backlog*, *sprint planning*, *sprint*, dan *sprint review*. Hasil dari pengujian *game* ini adalah, semua fungsi dalam *game* berjalan dengan baik, *game* dapat diinstal dan dijalankan pada perangkat yang berbeda, dan mendapatkan nilai 83% untuk *usability testing*. Dimana jika dikategorikan dengan tabel kriteria interpretasi skor, *game* ini masuk dalam kriteria sangat layak untuk digunakan.

Kata kunci: Teknik Informatika, Game, Android, Unity, Warisan Budaya

1. PENDAHULUAN

Indonesia menjadi negara kepulauan terbesar di dunia dengan 5 pulau besar, dan belasan ribu pulau kecil yang terbentang dari Sabang – Merauke. Hal ini menjadikan Indonesia memiliki beraneka ragam budaya dengan ciri dan keunikan tersendiri di setiap daerah. Menurut data dari Kemdikbud, ada lebih dari 500 Warisan Budaya Tak Benda yang telah ditetapkan, lebih dari 70000 yang tercatat, dan 10 diantaranya telah diakui oleh UNESCO [1].

Kemajuan teknologi bagaikan kapak bermata dua, yang dapat mempengaruhi eksistensi warisan budaya. Di satu sisi, teknologi dapat memudahkan bahkan menghilangkan budaya. Dan di sisi lain teknologi juga dapat dimanfaatkan sebagai media untuk upaya pengenalan dan pelestarian budaya.

Upaya pengenalan budaya memanfaatkan teknologi dapat dilakukan dengan menggunakan game sebagai mediana. Penulis memilih Android karena pangsa pasar Android di Indonesia mencapai 90% [2]. Ditambah lagi, *trend* untuk aplikasi edukasi sedang meningkat. Dikutip dari Appbrain

aplikasi dengan kategori edukasi menempati peringkat pertama untuk jumlah, dengan 272.591 aplikasi dari total aplikasi yang tersedia di Google Play sebanyak 2,847,871 [3]. Hal ini membuktikan bahwa teknologi menciptakan peluang tersendiri untuk menjadi media pembelajaran.

Rumusan Masalah

1. Bagaimana merancang *game* edukasi “Ksatria Budaya” sebagai media pengenalan warisan budaya Indonesia ?
2. Bagaimana kelayakan *game* edukasi “Ksatria Budaya” ?

Tujuan dan Manfaat Penelitian

Tujuan dari pengembangan aplikasi ini :

1. Membuat dan memanfaatkan *game* sebagai media untuk pengenalan warisan budaya Indonesia.
2. Memenuhi segala aspek pengujian, baik *black-box testing*, *compatibility testing*, dan *usability testing*.

Manfaat Dari Pengembangan aplikasi ini :

1. Memberikan pengetahuan kepada pengguna tentang waian budaya tak benda Indonesia.
2. Mengetahui kelayakan *game* edukasi “Ksaria Budaya”.

Batasan Masalah

1. *Game* ini hanya berjalan pada ponsel berbasis Android.
2. *Game* ini dibuat dengan menggunakan Unity.
3. *Game* ini menggunakan JSON sebagai *database*.
4. *Game* ini hanya dapat dimainkan perseorangan (*single player*).
5. *Game* ini berformat 2D *Game* ini dapat dimainkan tanpa memerlukan koneksi internet (*offline*).
6. *Database* soal merupakan daftar dari Warisan Budaya Tak Benda (WBTB) Indonesia.

2. TINJAUAN PUSTAKA

2.1 Warisan Budaya Tak Benda

Budaya dalam Kamus Besar Bahasa Indonesia dapat diartikan sebagai, pikiran, akal budi serta adat istiadat [4]. Sedangkan warisan budaya adalah keseluruhan peninggalan kebudayaan yang memiliki nilai penting sejarah, ilmu pengetahuan dan teknologi, dan/atau seni. Warisan budaya dimiliki bersama oleh suatu komunitas atau masyarakat dan mengalami perkembangan dari generasi ke generasi, dalam alur suatu tradisi [5]. Warisan budaya tak benda atau *Intangible Cultural Heritage* bersifat tak dapat dipegang (*intangible/abstrak*), seperti konsep dan teknologi. dan sifatnya dapat berlalu dan hilang dalam waktu seiring perkembangan zaman seperti misalnya bahasa, musik, tari, upacara, serta berbagai perilaku terstruktur lain [6].

2.2 Android

Android adalah sistem operasi *mobile* berbasis linux, yang dibangun oleh Android Inc. Di tahun 2005, Google membeli android dan menjadikannya *open source* dibawah lisensi Apache [7]. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak [14].

2.3 Media Pembelajaran

Media pembelajaran adalah sarana pendidikan yang dapat digunakan sebagai perantara dalam proses pembelajaran untuk mempertinggi efektifitas dan efisiensi dalam pengajaran [8].

2.4 Game

Awalnya *game* diartikan suatu aktivitas yang dilakukan hanya untuk kesenangan dan tanpa tujuan tertentu, namun nyatanya *game* adalah objek yang terdiri dari komponen dan aturan [9]. Permainan sebagai media pendidikan mempunyai banyak peranan. Permainan mendidik memungkinkan partisipasi aktif dari pembelajar untuk belajar, memberikan umpan balik secara langsung, dan mempermudah penerapan konsep dalam kehidupan sehari-hari [10].

2.5 Unity

Unity adalah software untuk membuat *game* dan aplikasi 2D, 3D, VR, dan AR yang cantik dan menarik [11]. Unity *Engine* merupakan salah satu *game engine* yang cukup populer karena aksesibilitas kesederanaan dan berbagai fitur yang dimilikinya [13].

2.6 JSON

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer [12].

3. METODE PENELITIAN

3.1 Tahapan Penelitian

Berikut adalah tahapan-tahapan yang penulis lakukan dalam pengembangan *game*.

1. Studi Literatur

Studi literatur dilakukan untuk mencari referensi dan metode yang akan digunakan untuk pengumpulan data dalam pengembangan aplikasi. Studi literatur didapatkan dari membaca beberapa jurnal, buku dan *website* yang berkaitan tentang pengembangan *game* ini.

2. Pengumpulan Data
Pada tahap ini, penulis akan melakukan pengumpulan data yang akan digunakan sebagai sumber konten untuk *game* “Ksatria Budaya”.
3. Analisis Kebutuhan Sistem
Pada tahap ini, penulis telah melakukan analisis terkait kebutuhan dalam pengembangan aplikasi yang akan dibangun.
4. Perancangan Produk *Backlog*
Pada tahap perancangan produk *backlog*, penulis akan membuat daftar fitur yang akan dikerjakan selama pengembangan *Game* “Ksatria Budaya”.
5. Testing Aplikasi
Penulis akan melakukan pengujian dengan 3 metode, yaitu *black-box testing*, *compatibility testing*, dan *usability testing*.
6. Penarikan Kesimpulan
Penarikan kesimpulan berdasarkan kuisioner yang diisi oleh responden.

4. IMPLEMENTASI DAN HASIL

4.1 Analisis Sistem

Penulis membuat *product backlog* dalam bentuk *user stories* sebagai rancangan untuk fitur yang akan dikembangkan dalam *game* yang akan dibangun. *Product backlog* merupakan sebuah bentuk perancangan fitur yang biasa digunakan dalam pendekatan dengan metode *scrum*. Berikut adalah *product backlog* yang penulis buat.

Sebagai	Item	Saya Ingin	Sehingga
Main Menu	Melihat Menu		Bisa melihat tampilan menu
	Memilih Menu Main		Bisa memulai permainan
	Memilih Menu Keluar		Bisa keluar dari permainan
	Menu Cara Main		Bisa melihat cara main
	Memilih Menu Informasi		Bisa melihat informasi
Pengguna	Mute Musik		Bisa mematikan musik
	Melihat Pertanyaan		Bisa melihat pertanyaan yang muncul
	Melihat dan Memilih Jawaban		Bisa menjawab pertanyaan
	Mempause game		Bisa menghentikan game sementara
	Resume game		Bisa kembali melanjutkan <i>game</i>
Menu Pause	Back to Main Menu		Bisa kembali ke menu utama
	Exit		Bisa keluar dari permainan
Menu Game Over	Memilih Restart		Bisa mengulang permainan
	Memilih Menu		Bisa kembali ke menu utama
	Memilih Exit		Bisa keluar dari permainan

Gambar 1. *User Stories*

Setelah merancang *product backlog*, penulis juga merancang peraturan permainan (*rules of game*). Dan berikut adalah aturan-aturan yang akan berlaku di dalam permainan:

- Setelah *game* dimulai, karakter akan berlari
- Akan ada musuh yang menghadang
- Jika karakter bertemu dengan karakter musuh, maka akan tampil pertanyaan beserta pilihan jawaban
- Pemain akan diberikan waktu +- 10 detik untuk menjawab pertanyaan

- Jika pertanyaan dijawab dengan benar, maka karakter akan menyerang musuh dan akan mendapatkan poin tambahan
- Jika pertanyaan tidak terjawab/dijawab salah, maka musuh akan menyerang karakter. Darah karakter akan berkurang dan juga pemain tidak mendapatkan poin tambahan
- Jika darah karakter habis, maka permainan selesai

Untuk *database* soal, penulis membuat berdasarkan data yang ada di situs Warisan Budaya Tak Benda Kemdikbud.

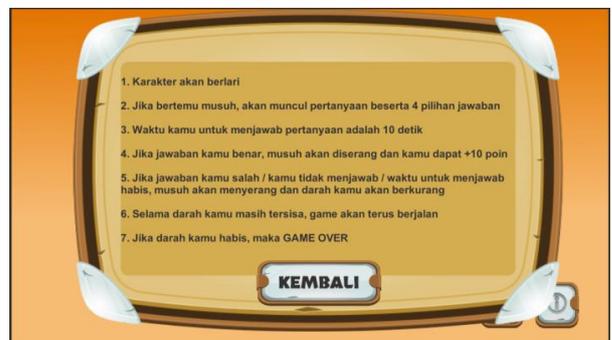
4.2 Implementasi *Product Backlog*

Berikut adalah tangkapan layar saat *game* pertama kali dimainkan. Akan ada beberapa pilihan menu, yaitu menu main, cara main, keluar, mute/unmute, dan lihat info.



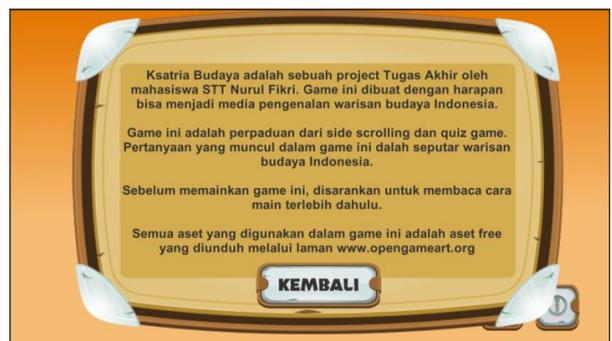
Gambar 2. Implementasi Melihat Menu

Saat memilih Menu cara main, akan muncul tampilan seperti di bawah ini.



Gambar 3. Implementasi Cara Main

Berikut adalah tampilan disaat kita memilih tombol info.



Gambar 4. Implementasi Lihat Info

Saat memilih menu Main, kita akan masuk ke *game loop*. Disana terdapat indikator darah, skor, tombol *pause*, waktu untuk menjawab, soal beserta pilihan jawaban.



Gambar 5. Implementasi *Game Loop*



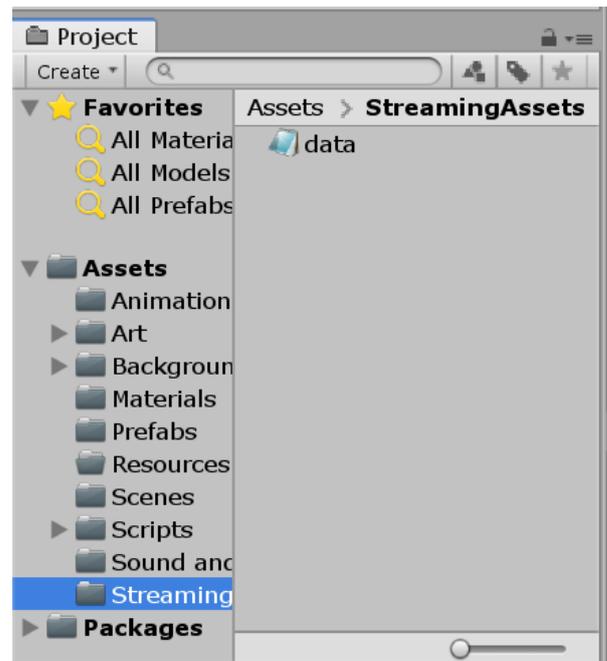
Gambar 6. Implementasi Menu *Pause*



Gambar 7. Implementasi Menu *Game Over*

4.3 Implementasi JSON

Seperti yang sudah dijelaskan sebelumnya, bahwa game yang sedang dikembangkan ini menggunakan JSON. Lebih tepatnya, JSON digunakan untuk menyimpan database soal yang terdapat di dalam game. File JSON terletak didalam folder *Assets > StreamingAssets*.



Gambar 8. Implementasi Database Soal

Berikut adalah contoh isi dari file JSON yang digunakan dalam *game* ini.

```
{
  "allRoundData": [
    {
      "name": "Buat TA",
      "questions": [
        {
          "questionText": "Senjata yang menjadi simbol keberanian dan kegagahan ureung Aceh adalah",
          "answers": [
            {
              "answerText": "Rencong",
              "isCorrect": true
            },
            {
              "answerText": "Badik",
              "isCorrect": false
            },
            {
              "answerText": "Clurit",
              "isCorrect": false
            },
            {
              "answerText": "Mandau",
              "isCorrect": false
            }
          ]
        },
        {
          "questionText": "Tarian Tor-Tor merupakan tarian yang berasal dari",
          "answers": [
            {
              "answerText": "Jawa Tengah",
              "isCorrect": false
            },
            {
              "answerText": "Sumatra Utara",
              "isCorrect": true
            },
            {
              "answerText": "Kalimantan Timur",
              "isCorrect": false
            },
            {
              "answerText": "Sumatra Barat",
              "isCorrect": true
            }
          ]
        },
        {
          "questionText": "Rumah Adat Sumatra Utara adalah",
          "answers": [
            {
              "answerText": "Rumah Gadang",
              "isCorrect": false
            },
            {
              "answerText": "Rumah Panggung",
              "isCorrect": false
            },
            {
              "answerText": "Rumah Adat Karo",
              "isCorrect": true
            },
            {
              "answerText": "Rumah Limas",
              "isCorrect": false
            }
          ]
        },
        {
          "questionText": "Salah satu makanan khas Minangkabau yang sangat terkenal adalah",
          "answers": [
            {
              "answerText": "Pempek",
              "isCorrect": false
            },
            {
              "answerText": "Gudeg",
              "isCorrect": false
            },
            {
              "answerText": "Limpung",
              "isCorrect": false
            },
            {
              "answerText": "Randang",
              "isCorrect": true
            }
          ]
        },
        {
          "questionText": "Rumah Gadang adalah rumah adat",
          "answers": [
            {
              "answerText": "Minangkabau",
              "isCorrect": true
            },
            {
              "answerText": "Belitong",
              "isCorrect": false
            },
            {
              "answerText": "Sumatra Utara",
              "isCorrect": false
            },
            {
              "answerText": "Aceh",
              "isCorrect": false
            }
          ]
        }
      ]
    }
  ]
}
```

Gambar 9. Implementasi JSON

4.4 Hasil Pengujian

Blackbox testing

Dalam *blackbox testing*, penulis memastikan semua *product backlog* dapat dijalankan dengan semestinya.

Sebagai	Item	Saya Ingin	Sehingga	Keterangan
Main Menu	Melihat Menu		Bisa melihat tampilan menu	Berhasil
	Memilih Menu Main		Bisa memulai permainan	Berhasil
	Memilih Menu Keluar		Bisa keluar dari permainan	Berhasil
	Menu Cara Main		Bisa melihat cara main	Berhasil
	Memilih Menu Informasi		Bisa melihat informasi	Berhasil
Pengguna	Mute Musik		Bisa mematikan musik	Berhasil
	Melihat Pertanyaan		Bisa melihat pertanyaan yang muncul	Berhasil
	Game Loop	Melihat dan Memilih Jawaban	Bisa menjawab pertanyaan	Berhasil
	Mempause game		Bisa menghentikan game sementara	Berhasil
	Resume game		Bisa kembali melanjutkan game	Berhasil
Menu Pause	Back to Main Menu		Bisa kembali ke menu utama	Berhasil
	Exit		Bisa keluar dari permainan	Berhasil
	Menu	Memilih restart	Bisa mengulang permainan	Berhasil
Game Over	Memilih menu		Bisa kembali ke menu utama	Berhasil
	Memilih Exit		Bisa keluar dari permainan	Berhasil

Gambar 10. Blackbox Testing

Compatibility Testing

Penulis juga mencoba untuk menginstal dan menjalankan *game* di 5 perangkat yang berbeda.

Perangkat	Spesifikasi	Instalasi	Dijalankan
Zenfone Max Pro M1	Android Oreo RAM 3 GB Layar 5,99"	v	v
Zenfone Max M1	Android Oreo RAM 3GB Layar 5,5"	v	v
Zenfone Max M2	Android Oreo RAM 3GB Layar 6,3"	v	v
Sony Xperia XZ	Android Marshmallow RAM 3GB Layar 5,2"	v	v
Xiaomi Pocophone F1	Android Oreo RAM 6GB Layar 6,18"	v	v

Gambar 11. Compatibility Testing

Usability Testing

No	Instrumen	Skala Nilai				
		SS	S	RG	TS	STS
1	Aplikasi ini mudah digunakan	4	0	0	1	0
2	Aplikasi ini mudah dipahami	4	0	0	1	0
3	Aplikasi ini bekerja sesuai yang saya inginkan	4	0	1	0	0
4	Aplikasi ini bermanfaat	3	2	0	0	0
5	Aplikasi ini membantu saya untuk mengenal Warisan Budaya Indonesia	3	1	1	0	0
6	Aplikasi ini menambah wawasan saya terhadap Warisan Budaya Indonesia	2	1	2	0	0
7	Saya merasa nyaman menggunakan aplikasi ini	0	4	1	0	0
8	Saya belajar menggunakan aplikasi ini dengan cepat	3	2	0	0	0
9	Saya mudah mengingat bagaimana cara menggunakan aplikasi ini	3	2	0	0	0
10	Saya tidak melihat adanya ketidak konsistenan selama saya menggunakannya	1	3	1	0	0
11	Saya selalu berhasil setiap menggunakan (memainkannya)	1	2	0	1	1
12	Saya akan merekomendasikan aplikasi ini kepada teman saya	0	4	1	0	0
13	Secara keseluruhan, saya puas dengan aplikasi ini	0	4	1	0	0
Jumlah		28	25	8	3	1
Jumlah Nilai		140	100	24	6	1
Total Nilai		271				
Presentase (%)		83%				

Gambar 12. Usability Testing

Total Nilai yang didapatkan *game* ini adalah 271 (83%) dari 325 (100%).

5. KESIMPULAN

Setelah melewati 3 pengujian, yaitu *black-box testing*, *compatibility testing*, dan *usability testing*. Didapatkan kesimpulan sebagai berikut,

1. Pada pengujian *black-box testing* dilakukan untuk menguji fungsionalitas. Dan *game* Ksatria Budaya dapat berjalan sesuai *requirement* dan berjalan sesuai harapan.
2. Berdasarkan pengujian *compatibility testing*, *game* Ksatria Budaya dapat dijalankan di perangkat yang beragam dengan ukuran dan resolusi layar yang berbeda.
3. Berdasarkan hasil pengujian *usability testing*, *game* Ksatria Budaya mendapatkan nilai 83% yang masuk dalam kriteria sangat layak.

DAFTAR PUSTAKA

- [1] "Warisan Budaya Tak Benda: Beranda." <http://warisanbudaya.kemdikbud.go.id/> (diakses Agu 18, 2021)
- [2] "Mobile Operating System Market Share Indonesia," *StatCounter Global Stats*. <https://gs.statcounter.com/os-market-share/mobile/indonesia> (diakses Agu 18, 2021)
- [3] "Top categories on Google Play," *AppBrain*. <https://www.appbrain.com/stats/android-market-app-categories> (diakses Agu 18, 2021)
- [4] "Hasil Pencarian - KBBI Daring." <https://kbbi.kemdikbud.go.id/entri/budaya> (diakses Agu 18, 2021)
- [5] A. GLN, "Warisan Budaya Tak Benda | Gerakan Literasi Nasional." <https://gln.kemdikbud.go.id/glnsite/formulir-warisan-budaya-tak-benda/> (diakses Agu 18, 2021)
- [6] "Warisan Budaya Tak Benda: Tentang." <https://warisanbudaya.kemdikbud.go.id/?tentang&active=pengertian%20dan%20domain%20%20budaya%20takbenda> (diakses Agu 18, 2021)
- [7] J. F. DiMarzio, "Beginning Android programming with Android Studio," Fourth edition, Indianapolis, IN: Wrox, a Wiley brand, 2017.
- [8] H. Sanaky, "Media Pembelajaran: Buku Pegangan Wajib Guru dan Dosen," Yogyakarta: Kaubaka, 2011.
- [9] "What is a Game?" <http://www.thegamesjournal.com/articles/WhatIsaGame.shtml> (diakses Agu 18, 2021)

- [10] A. S. Sadiman, R. Rahardjo, A. Haryono, dan Harjito, “Media Pendidikan: Pengertian, Pengembangan, dan Pemanfaatannya,” Jakarta: Rajawali Pers, 2009.
- [11] U. Technologies, “Unity Games Solutions - Create 2D And 3D Games: Unity.” <https://unity.com/solutions/game> (diakses Agu 18, 2021).
- [12] “JSON,” <https://www.json.org/json-en.html> (diakses Agu 18, 2021).
- [13] A. Fitriadi dan H. A. Tawakal, “Analisis dan Pengembangan Sistem Pembelajaran Matematika Berbasis Permainan pada Mode Operasi Dasar,” *J. Inform. Terpadu*, vol. 7, no. 2, hlm. 62-69, Sep 2021.
- [14] R. Wahyudi dan S. Munir, “Rancang Bangun Aplikasi *Mobile Client* CRM berbasis Android Studi Kasus CV Esindo Multi Tata,” *Jurnal Inform. Terpadu*, Vol. 4, No. 2, pp. 34-39, 2018 <https://doi.org/10.54914/jit.v4i2.151>



ANALISIS DAN IMPLEMENTASI RESTFUL API GUNA PENGEMBANGAN SISTEM INFORMASI AKADEMIK PADA PERGURUAN TINGGI

Mohammad Akmaluddin Novianto^{1,2}, Sirojul Munir²

^{1,2}Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jakarta Selatan, DKI Jakarta, Indonesia 12640

akmalseferagic@student.nurulfikri.ac.id, rojulman@nurulfikri.co.id

Abstract

In an organization, one of the most important is the processing of data and information, with the larger and complexity of an information system, the need for data processing and integration is a big concern for many organizations. STT-NF now has started to utilize technology and develop applications to support the ease of academic community. One of the applications developed is a mobile-based academic information system application. Currently STT-NF already has an academic system called AIS with a web platform located in ais.nurulfikri.ac.id. However, there is a problem when the academic information system will be developed in STT Terpadu Nurul Fikri with a platform or with a different programming language, the problem is the unavailability of web services that are able to integrate the system with other systems to be developed, thus complicating development, especially in terms of exchange, integration and processing of data. Therefore, it is necessary to design and create the service (web service) to solve the difficulties experienced by the AIS system. This study will design and create a web service API study result card module for academic information system mobile applications in STT NF using RESTful – Spring Framework, which in addition to providing services for other systems in need, also offers the ease of bridging data exchange without any concerns about the differences in platforms and programming languages. Design web service module KHS using waterfall development method, waterfall aims to produce a software or system that has been determined with the best quality. Waterfall prioritizes interaction with stakeholders at the beginning of the project to get a clear picture, and when using the waterfall process, each team completes a stage, then the team must advance to the next stage and be expected without repeating the previous process.

Keywords: *Web Service, Restful-Spring Framework, Academic Information Systems, Service Oriented Architecture*

Abstrak

Dalam suatu organisasi salah satu yang terpenting adalah pengolahan data dan informasi, dengan semakin besar dan kompleksnya suatu sistem informasi, maka kebutuhan akan akan pengolahan dan integrasi data menjadi perhatian besar bagi banyak organisasi. STT-NF saat ini sudah mulai melakukan pemanfaatan teknologi dan mengembangkan aplikasi-aplikasi penunjang kemudahan bagi para civitas akademiknya. Salah satu aplikasi yang dikembangkan adalah aplikasi sistem informasi akademik berbasis mobile. Saat ini STT-NF sudah mempunyai sistem akademik bernama AIS dengan platform web yang beralamat di ais.nurulfikri.ac.id. Namun muncul masalah pada saat akan dikembangkan sistem informasi akademik yang ada pada STT Terpadu Nurul Fikri dengan *platform* ataupun dengan bahasa pemrograman yang berbeda, yakni belum tersedianya layanan (*web service*) yang mampu mengintegrasikan sistem tersebut dengan sistem lain yang akan dikembangkan sehingga hal ini mempersulit pengembangan terutama dalam hal pertukaran, integrasi dan pengolahan data. Oleh karena itu perlu dilakukan perancangan dan pembuatan layanan (*web service*) tersebut untuk menyelesaikan kesulitan yang dialami oleh sistem AIS. Penelitian ini akan merancang dan membuat API *web service* modul kartu hasil studi untuk aplikasi mobile sistem informasi akademik di STT NF menggunakan RESTful – Spring Framework, yang selain karena menyediakan layanan bagi sistem lain yang membutuhkan juga menawarkan kemudahan dalam menjembatani pertukaran data tanpa memperlumahkan perbedaan platform dan bahasa pemrograman. Rancang bangun *web service* modul KHS ini menggunakan metode pengembangan *waterfall*, *waterfall* bertujuan untuk menghasilkan suatu *software* atau sistem yang telah ditentukan dengan kualitas terbaik. *Waterfall* mengutamakan interaksi dengan *stakeholders* di awal *project* dikerjakan untuk mendapat gambaran yang jelas, dan saat menggunakan proses *waterfall*, setiap tim menyelesaikan suatu tahap, maka tim harus lanjut ke tahap berikutnya dan diharapkan tanpa mengulang proses sebelumnya.

Kata kunci: *Web Service, RESTful-Spring Framework, Sistem Informasi Akademik, Service Oriented Architecture*

1. PENDAHULUAN

Salah satu bagian terpenting dalam suatu organisasi adalah pengolahan data dan informasi, dengan semakin besar dan kompleksnya suatu sistem informasi, maka kebutuhan akan pengolahan dan integrasi data menjadi perhatian besar bagi banyak organisasi, proses-proses bisnis terus mengalami perubahan mengikuti kebutuhan organisasi, mengakibatkan diperlukan pengembangan sistem dan aplikasi yang ada [1]. Namun tantangan muncul dalam pengembangan sistem dan aplikasi yang telah tersedia sebelumnya, diantaranya adalah bagaimana data dan informasi yang lama dapat diintegrasikan dan dapat dipergunakan kembali pada pengembangan sistem dan aplikasi baru, bagaimana membangun sistem dan aplikasi yang dapat digunakan kembali di masa mendatang yang dapat berjalan baik lintas *platform*, bahasa pemrograman, maupun berbagai sistem operasi.

Hal tersebut yang mendasari konsep dan pemikiran lama akan sistem informasi khususnya mengenai arsitektur perangkat lunak yang terus berkembang. *Service Oriented Architecture* merupakan salah satu konsep arsitektur perangkat lunak yang menyediakan layanan bagi suatu sistem untuk bisa digunakan pada sistem lain sesuai kebutuhan. SOA bertujuan untuk memberikan layanan yang dapat diakses sistem lain, sehingga mendukung integrasi antar sistem [2].

Dalam mengimplementasikan SOA, *web service* dapat digunakan untuk membuat pertukaran data yang diakses melalui standar *internet protocol*. Dalam perkembangan *web service* telah dikembangkan REST (*Representational State Transfer*) *web service*. Dengan mengimplementasikan REST *web service* dalam sistem maupun dengan bahasa pemrograman atau platform berbeda. *Web service* adalah standar yang digunakan untuk melakukan pertukaran data antar aplikasi atau sistem, karena aplikasi yang melakukan pertukaran data bisa ditulis dengan bahasa pemrograman yang berbeda atau berjalan pada platform yang berbeda [11]. Beberapa contoh implementasi *web service* antara lain adalah SOAP dan REST. *Web service* yang berbasis arsitektur REST kemudian dikenal sebagai RESTful *web services*, layanan ini menggunakan metode HTTP untuk menerapkan konsep arsitektur REST.

Sekolah Tinggi Teknologi Terpadu Nurul Fikri yang disingkat STT Terpadu Nurul Fikri merupakan perguruan tinggi teknologi yang memadukan antara keilmuan praktis di bidang teknologi informasi dengan pengembangan kepribadian islami. Teknologi informasi merupakan salah satu teknologi yang berkembang cepat pada saat ini. penggunaan alat bantu komputer sebagai salah satu bentuk teknologi informasi untuk menunjang sistem informasi agar dapat memberikan hasil lebih baik dan akurat untuk sebuah sistem.

STT NF saat ini sudah mulai melakukan pemanfaatan dan pendayagunaan teknologi informasi guna mewujudkan misi dari STT-NF. Oleh karena itu STT-NF mengembangkan aplikasi-aplikasi penunjang kemudahan bagi para civitas akademiknya. Salah satu aplikasi yang dikembangkan adalah aplikasi sistem informasi akademik berbasis mobile. Saat ini STT-NF sudah mempunyai sistem akademik bernama AIS dengan platform web yang beralamat di ais.nurulfikri.ac.id. tetapi kesulitan akan timbul pada saat akan dikembangkan sistem informasi akademik yang ada pada STT Terpadu Nurul Fikri tersebut dengan *platform* ataupun dengan bahasa pemrograman yang berbeda, yakni belum tersedianya layanan (*web service*) yang mampu mengintegrasikan sistem tersebut dengan sistem lain yang akan dikembangkan sehingga hal ini mempersulit pengembangan terutama dalam hal pertukaran, integrasi dan pengolahan data. Oleh karena itu perlu dilakukan perancangan dan pembuatan layanan (*web service*) tersebut untuk menyelesaikan kesulitan yang dialami oleh sistem AIS.

Penelitian ini akan merancang dan membuat API *web service* modul kartu hasil studi untuk aplikasi mobile sistem informasi akademik di STT NF menggunakan RESTful – *Spring Framework*, yang selain karena menyediakan layanan bagi sistem lain yang membutuhkan juga menawarkan kemudahan dalam menjembatani pertukaran data tanpa mempermasalahkan perbedaan platform dan bahasa pemrograman.

a. Perumusan masalah

Berdasarkan latar belakang di atas, maka rumusan masalah dalam penelitian ini antara lain:

1. Bagaimana merancang API *web service* modul KHS AIS STT Terpadu Nurul Fikri dengan RESTful – Spring Framework?
2. Apakah REST *web service* yang telah dirancang pada sistem informasi akademik berhasil diimplementasikan?

b. Tujuan dan Manfaat Penelitian

Tujuan Penelitian :

1. Mengimplementasikan REST *web service* pada *database* utama sistem informasi akademik STT Terpadu Nurul Fikri.
2. Merancang model kebutuhan dan standarisasi API pada sistem informasi akademik STT Terpadu Nurul Fikri modul KHS.

Manfaat Penelitian :

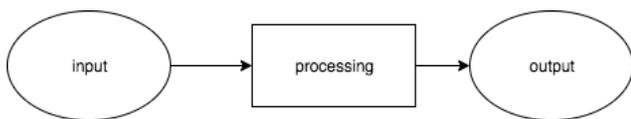
1. Memberikan kemudahan bagi *developer* dalam mengakses data sistem informasi akademik, baik pada platform yang sama ataupun berbeda.
2. Memberikan nilai tambah pada sistem informasi akademik STT Terpadu Nurul Fikri.

2. LANDASAN TEORI

2.1. Sistem Informasi

Sistem informasi memuat berbagai informasi penting mengenai orang, tempat, dan segala sesuatu yang ada dalam lingkungan sekitar organisasi tersebut.

Sistem informasi selalu menggambarkan, merancang, mengimplementasikan dengan menggunakan proses perkembangan sistematis dan merancang sistem informasi berdasarkan analisis kebutuhan. Jadi, bagian utama dari proses ini adalah mengetahui rancangan dan analisis sistem. Seluruh aktivitas utama dilibatkan dalam siklus perkembangan yang lengkap [3].



Gambar 1. Konsep Dasar Sistem Informasi

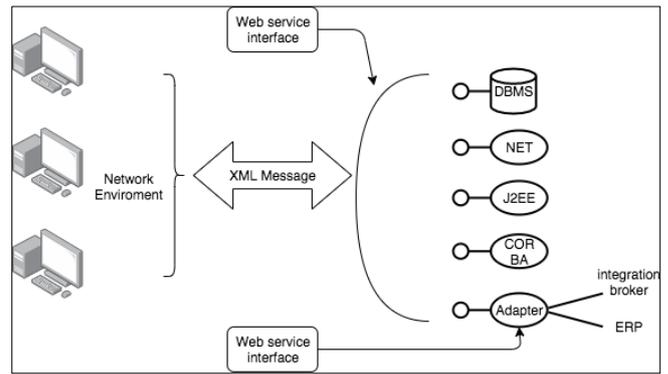
2.2. Web Service

2.2.1. Web Service

Sebuah *software* yang dirancang untuk mendukung interoperabilitas interaksi antar mesin melalui sebuah jaringan. *Web service* secara teknis memiliki mekanisme interaksi antar sistem sebagai penunjang interoperabilitas, baik berupa agregasi (pengumpulan) maupun sindikasi dan data kolaborasi informasi yang bisa diakses melalui internet oleh berbagai pihak menggunakan teknologi yang dimiliki oleh masing-masing pengguna [4].

Alasan menggunakan *web service* adalah kemudahan dalam penggunaan kembali dan dapat *sharing* logika yang sama dengan klien yang beragam seperti *mobile*, *desktop*, dan aplikasi *web*. Jangkauan *web service* yang luas karena *web service* bergantung pada standar yang terbuka, dapat beroperasi pada platform yang berbeda, serta tidak bergantung pada teknologi eksekusi yang mendasarinya. Semua *web service* setidaknya menggunakan HTTP dan format penukaran data standar berupa XML, JSON, atau media lain.

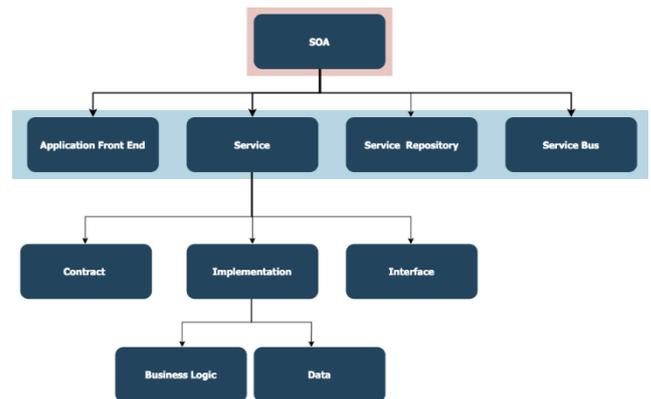
Selain itu, *web service* menggunakan HTTP dalam dua cara yang berbeda, yaitu sebagai protokol transportasi untuk menyampaikan data [5].



Gambar 2. Antarmuka Web Service dengan Sistem Lainnya.

2.2.2. SOA

Service Oriented Architecture atau SOA didefinisikan sebagai kebijakan, praktek, kerangka kerja yang memungkinkan fungsionalitas aplikasi disediakan dan dikonsumsi sebagai seperangkat *service* pada sebuah unit yang sesuai dengan kebutuhan *service customer*. *Service* dapat digunakan, dipublikasikan, ditemukan, dan diabstraksikan menggunakan standar antarmuka [6].



Gambar 3. Struktur Data Hirarki SOA

2.2.3. REST Web Service

REST merupakan singkatan dari *Representational State Transfer*. RESTful bukanlah sebuah standar protokol *web service*, melainkan hanya sebuah gaya arsitektur. Ide dasar dari arsitektur REST adalah bagaimana menghubungkan jalur komunikasi antar mesin atau aplikasi melalui HTTP sederhana. Sebelum adanya REST, mekanisme atau protokol *middleware* yang cukup kompleks seperti DCE, CORBA, RPC, ataupun SOA.

REST mampu mengeksploitasi berbagai kelebihan dari HTTP yang digunakan untuk kebutuhan *web service*. Walaupun SOAP juga dapat menggunakan protokol HTTP, namun hanya terbatas untuk kebutuhan transport saja, dengan adanya REST, aplikasi *client* dapat berupa aplikasi apa saja hanya dengan memanfaatkan HTTP. Berikut ini beberapa prinsip arsitektur dari REST yang dikutip dari sebuah buku berjudul “RESTful java with JAX-RS” [7] :

1. Addressability

Addressability merupakan sebuah ide dimana setiap objek *resource* pada suatu sistem dapat dicapai hanya dengan melalui sebuah *unique identifier*. Pada dunia REST, *addressability* dikelola dengan penggunaan *Uniform Resource Identifier* (URI).

2. Constrained & Uniform Interface

Pada sistem COBRA ataupun SOAP. Pengembangan *client* harus mengetahui *method* apa saja yang disediakan oleh *web service server*. Pemanggilan *method* tersebut dikenal dengan istilah RPC (*Remote Procedure Call*).

Namun pada sistem REST, *method* atau *procedure* yang digunakan untuk layanan apapun hanyalah *method-method* yang disediakan pada HTTP. Istilah yang biasa digunakan untuk menyatakan prinsip *uniform interface* pada REST adalah CRUD (*Create, Read, Update, Delete*). Berikut adalah ulasan detail mengenai *method-method* tersebut :

Tabel 1. HTTP Method dan Penggunaanya dalam REST

Metode	Deskripsi
GET	Mendapatkan (read) sebuah sumber daya (<i>resource</i>) yang diidentifikasi dengan URI (<i>Uniform Resource Identifier</i>).
POST	Mengirimkan sumber daya (<i>resource</i>) ke <i>server</i> . Digunakan untuk membuat (<i>create</i>) sumber daya baru.
PUT	Mengirimkan sumber daya (<i>resource</i>) ke <i>server</i> , digunakan untuk memasukkan (<i>insert</i>) atau memperbaharui (<i>update</i>) sumber daya yang tersimpan.
DELETE	Menghapus sumber daya (<i>resource</i>) yang diidentifikasi dengan URI.
HEAD	Mendapatkan metadata (<i>response header</i>) dari sumber daya (<i>resource</i>) yang diidentifikasi dengan URI.

Dengan menggunakan protokol HTTP, URI dapat dijadikan sebagai media yang digunakan untuk mengakses *resources* dari *server*. Hal ini disebut dengan URI *tunneling*. URI *tunneling* mempergunakan URI untuk mentransfer informasi pada antar sistem yang dalam jaringan dengan melakukan *encode* pada URI itu sendiri. Dengan mengirim HTTP *method* yang telah disebutkan sebelumnya, server dapat melakukan eksekusi terhadap suatu program yang menghasilkan atau mengambil suatu *resource* dan mengirimnya kembali ke *client*. Dalam proses ini terjadi proses mapping dan URI menjadi *method call* pada *server* yang dituju [8].

3. Stateless Communication

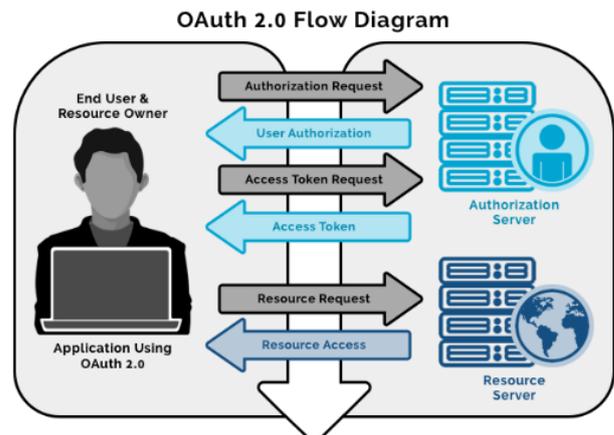
Pada dunia REST, seperti halnya pada dunia *World Wide Web*, *stateless* berarti tidak ada *client session* data yang disimpan pada *server*. *Server* hanya menyimpan dan mengelola *state* dari *resource* yang digunakan.

4. Format Pesan Pertukaran

Saat ini terdapat dua buah format pesan yang dipertukarkan (*data interchange format*) yang digunakan pada *web service*, yaitu XML, dan JSON. Pada aplikasi berbasis *javascript*, penggunaan JSON sebagai format pesan pertukaran pesan akan membawa dampak performa yang cukup signifikan dibandingkan bila menggunakan *library* tambahan untuk membaca data dari *Document Object Model* (DOM).

2.2.4. Oauth 2

OAuth (*Open Authorization*) adalah sebuah protokol standar otorisasi standar terbuka yang memungkinkan pengguna mengakses aplikasi tanpa perlu berbagi *password*. *OAuth* juga mengizinkan otorisasi API yang sudah terlindungi yang berasal dari desktop ataupun aplikasi web melalui metode sederhana dan standard. Mengatur lalu lintas data antar aplikasi dan digunakan saat pembuat API mengetahui siapa yang terlibat dan berkomunikasi di dalam sistem. *OAuth* menyediakan kemampuan terhubung dengan sistem secara aman, sehingga pengguna tidak perlu menyerahkan *password*nya.



Gambar 4. Oauth 2 Diagram

2.2.5. API

API Merupakan *software interface* yang terdiri atas kumpulan instruksi yang disimpan dalam bentuk *library* dan menjelaskan bagaimana agar suatu *software* dapat berinteraksi dengan *software* lain

2.2.6. Swagger

Swagger adalah salah satu *tools* untuk kita bisa mendokumentasikan API secara lebih efektif, dan memungkinkan kita mendeskripsikan struktur API kita, sehingga mesin dapat membacanya.

2.3. Spring Framework

2.3.1. Java

Java adalah bahasa pemrograman serbaguna. Java dapat digunakan untuk membuat suatu program sebagaimana kita membuatnya dengan bahasa seperti Pascal atau C++. Yang lebih menarik, java juga mendukung sumber daya internet yang saat ini sangat populer, yaitu *World Wide Web* atau yang sering disebut *web* saja. Java juga mendukung aplikasi klien atau server, baik dalam jaringan local (LAN) maupun jaringan berskala luas (WAN).

Program java bersifat tidak bergantung pada *platform*, artinya java dapat dijalankan pada beberapa jenis dan sistem operasi.

2.3.2. Spring Framework

Spring framework adalah *framework open source* yang menyediakan infrastruktur yang komprehensif dalam mengembangkan aplikasi *java* dengan mudah dan cepat. Spring akan membantu *programmer* dalam pengembangan aplikasi dengan *build* yang sederhana, *portable*, cepat dan sistem berbasis JVM yang fleksibel. Spring dapat digunakan untuk melakukan pengaturan deklarasi manajemen transaksi, *remote access* dengan menggunakan RMI atau *database*.

2.3.3. Data Source

Data Source adalah sumber utama dimana data berasal. Dalam pemrograman komputer, *data source* bisa berasal dari basis data, *data set*, *spreadsheet*, dan lain-lain. Ketika data ditampilkan dalam halaman web atau aplikasi apapun, data diambil dari sumber data dapat disajikan, dalam format yang sudah didefinisikan dalam pemrograman.

2.3.4. Spring Boot

Spring Boot merupakan sebuah *framework* Java yang digunakan untuk membuat aplikasi berbasis web dan aplikasi *enterprise*. Pada Spring Boot memberikan fleksibilitas untuk melakukan konfigurasi beans dengan berbagai cara seperti *XML*, *Annotations*, dan *JavaConfig*.

Spring Boot pada penelitian ini digunakan sebagai bagian *backend* yang digunakan untuk membuat API *web service*. Pada *Spring Boot* sendiri *layer-layer* nya jelas dan dapat digunakan untuk membuat sistem *microservice* yang *reusable* dan *scalable*.

3. METODOLOGI PENELITIAN

3.1. Metode Pengembangan Sistem

Rancang bangun *web service* modul KHS ini menggunakan metode pengembangan *waterfall*. *Waterfall* mengutamakan interaksi dengan *stakeholders* di awal *project* dikerjakan untuk mendapat gambaran yang jelas, dengan menggunakan proses *waterfall*, setelah menyelesaikan suatu tahap, maka peneliti harus lanjut ke tahap berikutnya dan diharapkan tanpa mengulang proses sebelumnya.

3.2. Metode Penelitian

Metode penelitian yang penulis lakukan dalam penelitian ini menggunakan pendekatan metode kualitatif. yaitu pendeskripsian masalah yang diambil, menjelaskan langkah-langkah apa saja yang diperlukan dalam penelitian atau tahapan analisis.

3.3. Metode Pengumpulan Data

1. Studi Kepustakaan

Studi kepustakaan dilakukan dengan mencari sumber-sumber pustaka yang mendukung penelitian dan memberikan informasi yang memadai dalam menyelesaikan penelitian ini. Studi kepustakaan yang digunakan antara lain, buku, jurnal, artikel dan *paper*.

2. Observasi

Observasi dilakukan dengan pengamatan langsung terhadap objek yang akan diteliti terkait permasalahan yang akan dibahas nantinya, yaitu dengan mengidentifikasi proses kegiatan akademik dengan melihat dan mengamati bisnis proses yang terjadi antara pengguna dan sistem.

3.4. Metode Pengujian

Dalam penelitian ini, metode yang digunakan untuk pengujian *web service* yang dirancang yaitu menggunakan pengujian *Blackbox* dan Postman.

Salah satu metode *Blackbox testing* yaitu *equivalence class partitioning* adalah metode uji coba *Blackbox* yang membagi domain *input* dari program menjadi beberapa kelas data dari kasus uji coba yang dihasilkan. *Equivalence class partitioning* berusaha untuk mendefinisikan kasus uji yang dapat menemukan sejumlah kesalahan, kasus uji yang didesain untuk *equivalence class partitioning* berdasarkan pada evaluasi dari ekuivalensi jenis atau *class* untuk kondisi *input*. *Class-class* yang ekuivalen merepresentasikan sekumpulan keadaan *valid* dan *invalid* untuk kondisi *input*. Kondisi *input* sendiri dapat berupa nilai numeric yang spesifik, kisaran nilai (*range*), sekumpulan nilai yang berhubungan (himpunan), atau kondisi *Boolean* [9].

Pengujian dengan menggunakan metode *Blackbox* berfokus pada persyaratan fungsional dari *web service* [12]. Dengan begitu, metode pengujian *Blackbox* memungkinkan *web service* mendapatkan serangkaian *input* yang sepenuhnya menggunakan persyaratan fungsional untuk *web service* yang dirancang. Pengujian API menggunakan alat bantu Postman dalam menguji fungsionalitas API dengan melakukan HTTP request (*GET*, *POST*, *PUT*, *DELETE*) Postman REST Client adalah salah satu aplikasi yang mendapatkan rating tertinggi dalam *chrome web store*. Lebih dari 348.000

pengguna unik dan lebih dari 63.000 koleksi berbagi via Postman [10].

4. ANALISIS DAN RANCANGAN

4.1. Analisis Kebutuhan

Proses analisis kebutuhan dibutuhkan untuk menentukan apa yang dibutuhkan dalam proses pengembangan *Web Service*. Hal ini didapatkan melalui kajian dan pengamatan terkait fitur dan fungsinya yang nantinya akan diintegrasikan pada aplikasi, berikut adalah analisisnya :

1. *Web service* dapat diimplementasikan pada platform *mobile* Android dan web.
2. *Web service* dapat menjadi penghubung akses dengan *database* utama sistem.
3. *Web service server* harus terpisah dari *database* dan *web server*.
4. *Web service* harus memperhatikan bandwidth dan resource yang tersedia.

Ketentuan otorisasi (keamanan) :

1. *Web service* harus memiliki sistem keamanan (*Authorization*).
2. Sistem menyediakan fitur token dalam memberikan akses *authorization* kepada *user* yang akan mengakses *web service*.

4.1.1. Analisis Perancangan REST *Web Service*

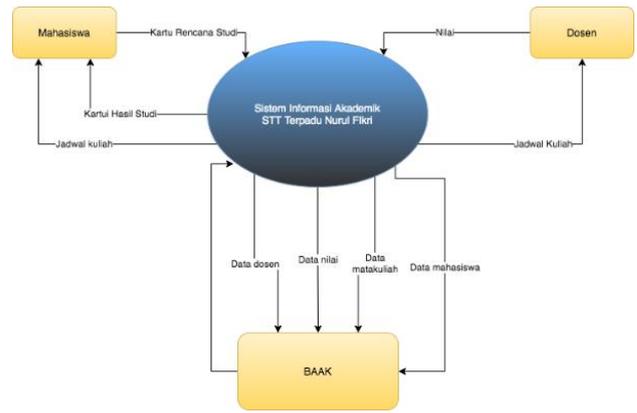
Berdasarkan hasil analisis kebutuhan yang dilakukan dengan observasi wawancara, hasilnya dapat dilihat pada tabel berikut :

Tabel 2. Hasil Analisis Perancangan *Web Service*

No.	Kebutuhan Perancangan	Kondisi Awal
1.	Basis data SIAK <i>postgreSQL</i>	Tersedia
2.	Framework untuk membangun web service	Belum tersedia
3.	Rancangan arsitektur <i>web service</i>	Belum dirancang
4.	Daftar modul <i>web service</i>	Sudah ditentukan
5.	Format data transfer yang digunakan	Sudah ditentukan
6.	Dokumentasi API	Belum tersedia

4.1.2. Analisis Sistem Berjalan

Setelah dilakukan pengamatan pada sistem informasi akademik STT Terpadu Nurul Fikri pada tahun 2021, maka terlihat sistem berjalan yang diilustrasikan melalui *data flow diagram* (DFD) berikut ini :

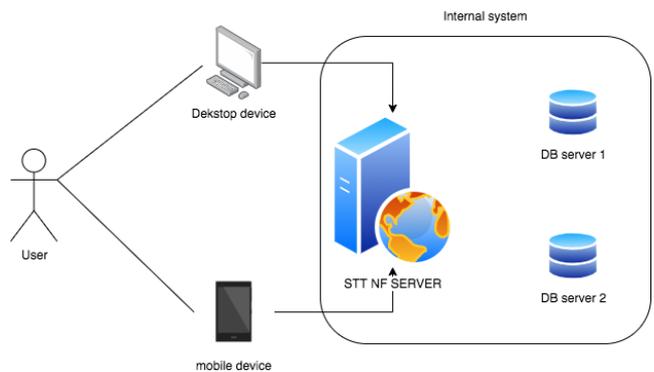


Gambar 5. Analisis Sistem Berjalan

4.1.3. Analisis Arsitektur Berjalan

Seperti yang tertulis pada landasan teori, bahwa sistem informasi akademik yang ada di STT Terpadu Nurul Fikri terdiri dari software berbasis web, yang di dalamnya ada beberapa web server dan terhubung dengan database server sebagai penyimpanan data. perangkat lunak tersebut hanya bisa diakses oleh user melalui web browser dengan jaringan internet lokal.

Jika disederhanakan dalam bentuk diagram, keadaan arsitektur berjalan pada sistem informasi akademik STT NF adalah seperti ini :



Gambar 6. Analisis Arsitektur Berjalan

4.2. Perancangan Sistem

4.2.1. Perancangan Kandidat Modul *Web Service*

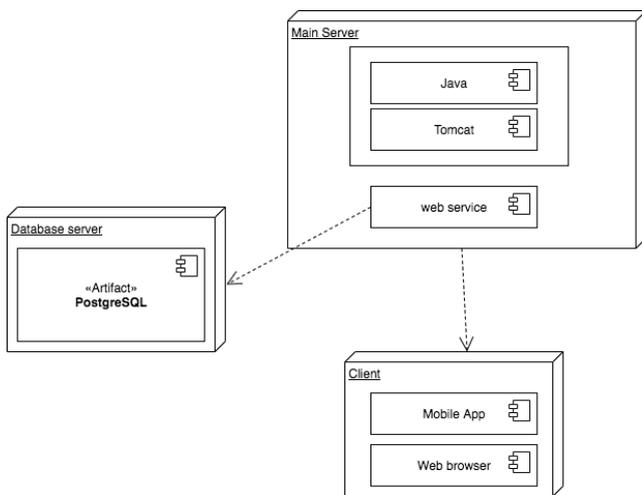
Bagian ini adalah perancangan melalui pendekatan dan prinsip-prinsip dari *SOA* untuk menghasilkan kandidat modul *web service* yang akan tersedia dan digunakan, berikut daftar kandidatnya :

Tabel 3. Kandidat Modul Web Service

No.	Kebutuhan	Deskripsi	User
1.	Modul Mahasiswa	Berfungsi untuk menampilkan data mahasiswa di STT Terpadu Nurul Fikri	Mahasiswa, dosen, BAAK.
2.	Modul Matakuliah	Berfungsi untuk menampilkan data yang berkaitan dengan mata kuliah yang ada di STT Terpadu Nurul Fikri.	Mahasiswa, dosen, BAAK
3.	Modul Kartu Hasil Studi	Berfungsi untuk menampilkan nilai hasil studi mahasiswa STT Terpadu Nurul Fikri	Mahasiswa, BAAK.

4.2.2. Perancangan *Deployment Diagram*

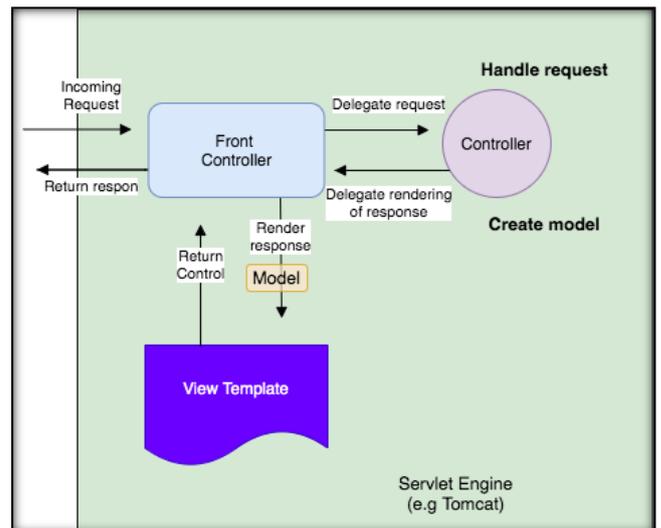
Deployment diagram dibutuhkan untuk menampakan bagian-bagian *software* yang berjalan dalam hardware, untuk mengimplementasikan sebuah sistem dan keterhubungan antara komponen-komponen tersebut, dan menggambarkan arsitektur fisik dari aplikasi yang melibatkan seluruh perangkat yang berkaitan, baik *software* atau *hardware* yang biasanya nanti disebut dengan *Node*, dan menunjukkan bagaimana komponen ini bekerja sama akan digambarkan dalam diagram deployment, berikut adalah diagram *deployment* tersebut :



Gambar 7. Deployment Diagram

4.2.3. Perancangan Model Web Service

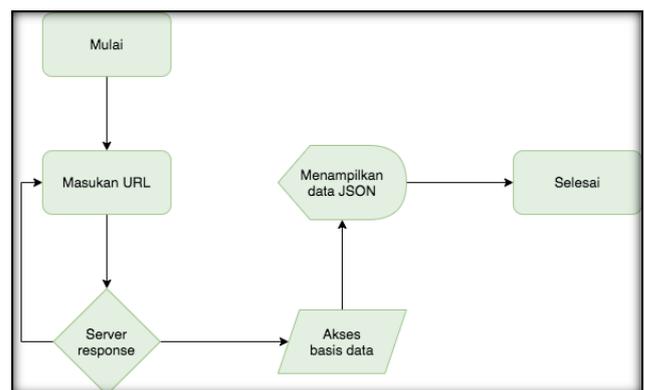
Web service sistem informasi akademik yang dirancang menggunakan Spring Framework, memiliki model seperti yang diilustrasikan pada diagram dibawah ini :



Gambar 8. Model Web Service

4.2.4. Perancangan Diagram Alir

Berikut adalah aliran dari operasi yang dirancang untuk REST *web service* sistem informasi akademik STT Terpadu Nurul Fikri, seperti yang ada pada gambar di bawah ini :



Gambar 9. Diagram Alir

4.2.5. Perancangan Akses

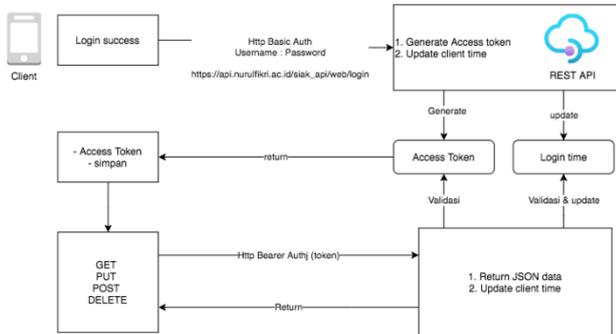
Pada bagian ini, penulis ingin membuat skema perancangan akses contoh *request* dan *respon* ketika *web service* berhasil diimplementasikan melalui tabel dibawah ini :

Tabel 4. Perancangan Akses

Method	Request	Respon yang diharapkan
GET	Input NIM dan Tahun Akademik.	Memunculkan daftar mata kuliah yang diambil pada semester tersebut.
POST	Input NIM dan password.	Berhasil login dan generate akses token yang digunakan untuk akses <i>web service</i> .
PUT	Mengubah data mahasiswa atau nilai pada KHS.	Berhasil untuk merubah data mahasiswa atau nilai pada KHS.
DELETE	Menghapus data tertentu pada sistem.	Berhasil untuk menghapus data tertentu.

4.2.6. Perancangan Skema Otentikasi

Bagian ini menunjukkan bagaimana skema otentikasi yang akan diterapkan pada rancangan API *web service* di sistem informasi akademik STT NF menggunakan OAuth2.



Gambar 10. Skema Otentikasi Token

Pada tahap ini, client akan login terlebih dahulu untuk meminta token untuk *authorization*, kemudian *server* mengautentikasi permintaan melalui *http basic auth*, *server* lalu menggenerasikan token dan diberikan kepada *client*, lalu *client* akan mendapatkan akses untuk masuk ke resource data. Ada 4 kondisi nantinya, yang pertama yaitu berhasil untuk menampilkan data JSON, yang kedua token salah, yang ketiga token mengalami kadaluarsa (*expired*) selama 60 menit.

4.2.7. Perancangan Pengujian

Sebelum *web service* ini layak digunakan pada aplikasi, maka perlu dilakukan rangkaian uji coba untuk memastikan bahwa *web service* ini dapat digunakan dengan baik, diantara metode pengujian yang dilakukan adalah :

Tabel 5. Tools Pengujian

No.	Jenis pengujian	Tujuan	Penguji
1.	Blackbox Testing	Untuk menguji fungsionalitas	Peneliti (pengembang)
2.	Postman	Proses development API	Peneliti (pengembang)

Menguji fungsionalitas fitur-fitur yang ada dengan menggunakan *blackbox* dan Postman, dan berfokus pada pengujian GET, POST, DELETE, UPDATE pada sistem *web service*, apakah sudah dapat berjalan sesuai dengan yang diharapkan atau belum, berikut adalah tabel pengujian yang telah dibuat :

Tabel 6. Rancangan Pengujian Black Box

No	Pengujian	Deskripsi uji	Hasil uji	Keterangan
1.	GET	Proses pengambilan data dari <i>web service</i>	Berhasil/ Tidak berhasil	Belum diuji
2.	POST	Proses Penambahan data ke <i>web service</i>	Berhasil/ Tidak berhasil	Belum diuji
3.	PUT	Proses perubahan data pada <i>web service</i>	Berhasil/ Tidak berhasil	Belum diuji
4.	DELETE	Proses penghapusan data dari <i>web service</i>	Berhasil/ Tidak berhasil	Belum diuji

5. HASIL IMPLEMENTASI DAN PENGUJIAN

Berikut ini adalah hasil implementasi perangkat lunak dari model REST API *web service* yang telah dirancang.

5.1. Hasil Pengujian Dengan Postman

Parameter yang diuji pada *web service* adalah bagian dari fungsional dari API yang dirancang, sedangkan untuk hasil pengujian yang sesuai dengan apa yang diinginkan, dirinci pada sub bab berikut ini :

5.1.1. Format URI API Web Service

Dikarenakan API *web service* yang akan dirancang menggunakan RESTful (*Representational State Transfer*) yang memungkinkan klien dapat melakukan request melalui protokol HTTP dengan mudah menggunakan URI, berikut daftar struktur dari URI yang akan digunakan :

1. Format URI Modul Mahasiswa

Tabel 7. Format URI Model Mahasiswa

No.	Method	Mapping
1.	GET	/Mahasiswa/get?(nim_mhs) /Mahasiswa/getall
2.	POST	/Mahasiswa/post
3.	PUT	/Mahasiswa/put
4.	DELETE	/Mahasiswa/delete?(nim_mhs)

2. Format URI Modul Matakuliah

Tabel 8. Format URI Modul Matakuliah

No.	Method	Mapping
1.	GET	/Matakuliah/get?(kode_mk) /Matakuliah/getall
2.	POST	/Matakuliah/post
3.	PUT	/Matakuliah/put
4.	DELETE	/Matakuliah/delete?(kode_mk)

3. Format URI Modul KHS

Tabel 9. Format URI Modul KHS

No.	Method	Mapping
1.	GET	/Khs/Khsdetail? (nim_mhs)&(tahun_ajaran)
		/Khs/getall
2.	POST	/Khs/post
3.	PUT	/Khs/put
4.	DELETE	/Khs/delete?(nim_mhs)&(kode_mk)

4. Format URI Modul Oauth

Tabel 10. Format URI Oauth

No.	Method	Mapping
1.	POST	/oauth/token

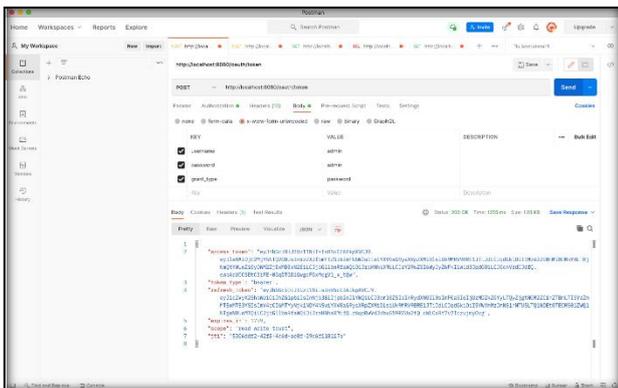
5. Format URI Modul Oauth

Tabel 11. Format URI User

No.	Method	Mapping
1.	POST	/users

5.1.2. Hasil dari Oauth2

Pengujian fungsional yang dilakukan menggunakan aplikasi Postman, data yang diuji adalah data *user* dengan *role admin* dan *user* untuk mendapatkan token agar dapat mengakses mapping lainnya, hasil pengujian dapat dilihat pada gambar dibawah ini :

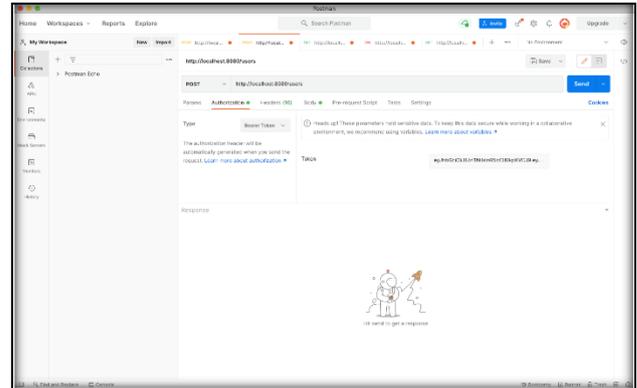


Gambar 11. Login User Admin dengan Oauth2

Pengujian dilakukan pada aplikasi Postman, pada alamat URI <http://localhost:8080/oauth/token>, dengan mengisi pada bagian *body* di kolom *x-www-urlecoded*, dengan parameter *username*, *password*, dan *grant_type*, *server* akan mengautentikasi permintaan melalui *http basic auth*, *server* lalu menggenerasikan token dan diberikan kepada *client*, dan akan habis masanya pada waktu 30 menit, dan bisa digunakan untuk mapping-mapping selanjutnya.

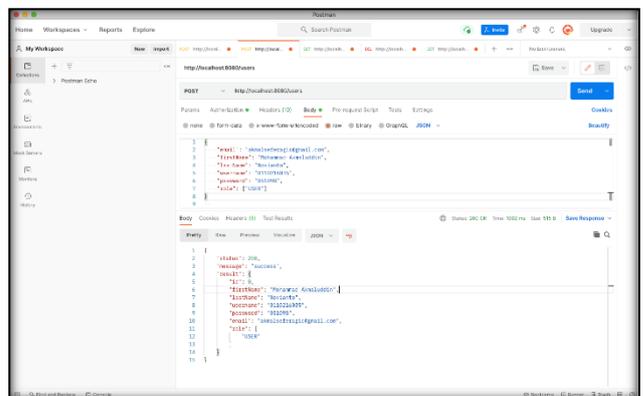
5.1.3. Hasil Modul User Model

Pengujian fungsional ini dilakukan dengan Postman, data yang diuji adalah data *user* yang akan diakses setelah mendapatkan token, token tersebut diletakkan pada kolom *Authorization* bagian *bearer* token, dan melakukan method *POST* dengan alamat URI <http://localhost:8080/users> untuk membuat akun *user*, hanya akun dengan *role admin* yang bisa membuat akun *user* lainnya.



Gambar 12. Memasukkan Token pada Bearer Token

Lalu memasukkan data *user* yang akan ditambahkan melalui kolom *body* dengan format *JSON* dengan isian *email*, *firstName*, *lastName*, *username*, *password*, dan juga *role user*. Setelah memasukkan *text* berformat *JSON* pada *raw body*, lalu melakukan *POST* dengan harapan berhasil untuk membuat *user* baru dengan *role [USER]*.



Gambar 13. Hasil dari Membuat Akun Role User dengan Method POST

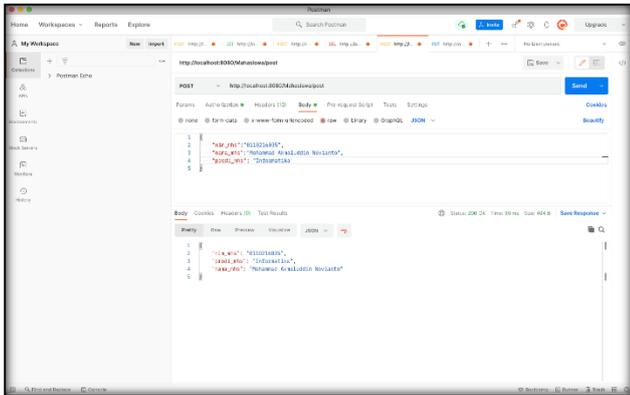
Data *user* baru berhasil ditambahkan dengan output status 200, dan *body* dengan *format text JSON* yang dikeluarkan.

5.1.4. Hasil Modul Mahasiswa Model

Pengujian fungsional ini dilakukan dengan Postman, data yang diuji adalah data modul mahasiswa yang akan diakses setelah mendapatkan token, token tersebut diletakkan pada kolom *Authorization* bagian *bearer* token, dan melakukan akses dengan alamat URI <http://localhost:8080/Mahasiswa>.

1. Melakukan *method* POST

Melakukan *method* POST untuk menambahkan data mahasiswa dengan URI <http://localhost:8080/Mahasiswa/post> dengan memasukkan *request body* seperti pada gambar 14 dengan memilih *radio button* jenis *raw* dan memasukkan *text* berformat JSON, akan menampilkan hasil seperti pada gambar 14.

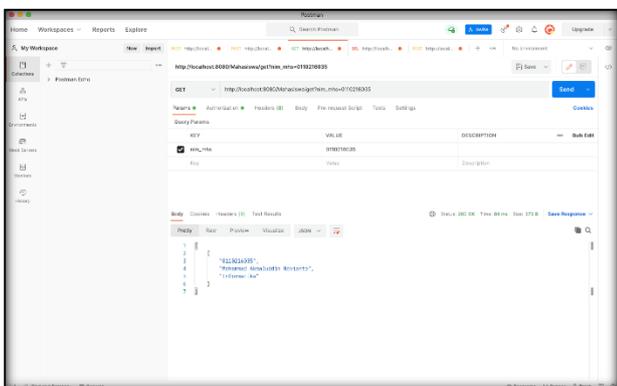


Gambar 14. Hasil dari POST Mahasiswa Model

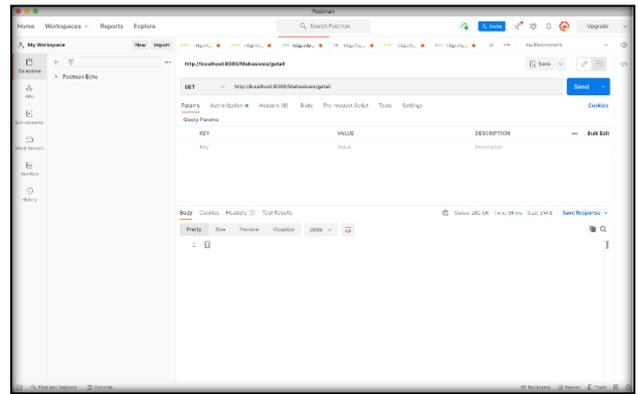
Data yang sudah dimasukkan sesuai dengan gambar diatas, maka data yang diharapkan masuk ke dalam sistem, akhirnya berhasil ditambahkan dengan status 200, dan *body* menampilkan ulang apa yang kita masukkan pada kolom *body* di atasnya.

2. Melakukan *method* GET

Melakukan *method* GET untuk mengecek atau melihat daftar mahasiswa dengan URI <http://localhost:8080/Mahasiswa/get> dengan memasukkan parameter *nim_mhs* dengan *syntax key* *nim_mhs* pada kolom *params*, akan menampilkan hasil seperti pada gambar 15, lalu dengan URI <http://localhost:8080/Mahasiswa/getall> untuk melihat semua data mahasiswa model yang ada pada database sistem.



Gambar 15. Hasil dari GET Mahasiswa model



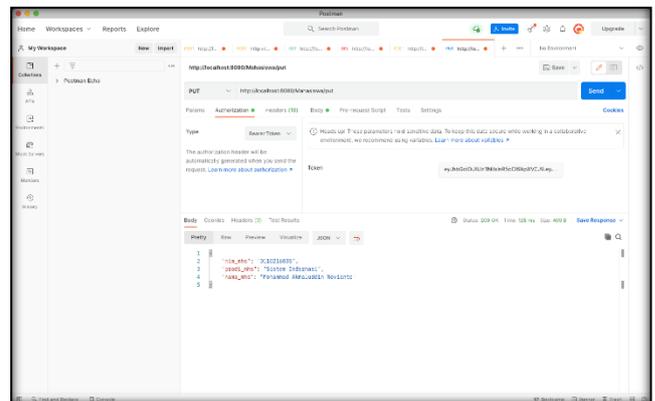
Gambar 16. Hasil dari *method* GET (all) Mahasiswa

Pada hasil keluaran dari *method* GET dengan parameter *nim* mahasiswa, didapatkan keluaran yang menampilkan data mahasiswa berformat JSON dengan struktur :

- *nim_mhs* : *nim* mahasiswa
- *nama_mhs* : *nama* mahasiswa
- *prodi_mhs* : *prodi* mahasiswa

3. Melakukan *method* PUT

Melakukan *method* PUT untuk mengubah data mahasiswa dengan URI <http://localhost:8080/Mahasiswa/put> dengan memasukkan *request body* seperti pada gambar 14 dengan memilih *radio button* jenis *raw* dan memasukkan *text* berformat JSON sama seperti akan melakukan *method* POST, hanya saja ditambahkan *id_mhs* agar sistem tau *id* mana yang akan diubah.



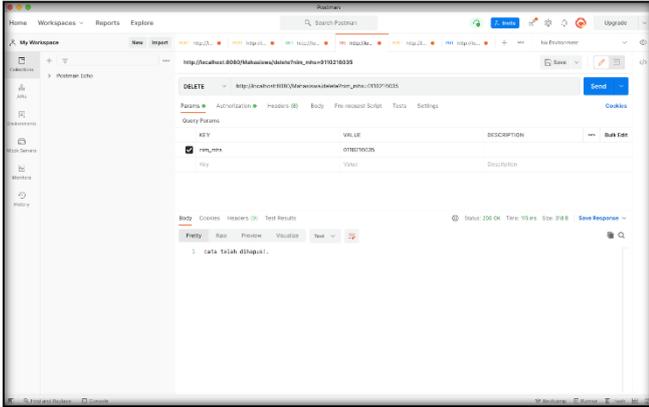
Gambar 17. Hasil dari *method* PUT Mahasiswa model

Jika *output* yang dihasilkan terlihat seperti pada gambar 17, maka perubahan data dengan *method* PUT berhasil dilakukan.

4. Melakukan *method* DELETE

Melakukan delete data dengan menggunakan *method* DELETE dengan URI <http://localhost:8080/Mahasiswa/delete> untuk menghapus data dari mahasiswa model, dengan memasukkan

parameter nim mahasiswa, dengan *syntax key* `nim_mhs` seperti gambar di bawah ini.



Gambar 18. Hasil dari *method* DELETE Mahasiswa model

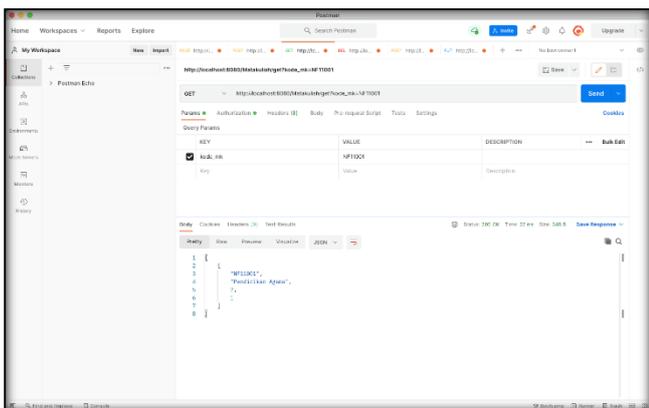
Setelah memasukkan parameter nim mahasiswa yang ada, maka output “data telah dihapus” akan muncul sebagai keluaran seperti yang ada pada gambar 18.

5.1.5. Hasil Modul Matakuliah Model

Pengujian fungsional ini dilakukan dengan *Postman*, data yang diuji adalah data modul matakuliah yang akan diakses setelah mendapatkan token, token tersebut diletakkan pada kolom *Authorization* bagian *bearer* token, dan melakukan akses dengan alamat URI <http://localhost:8080/Matakuliah>.

1. Melakukan *Method* GET

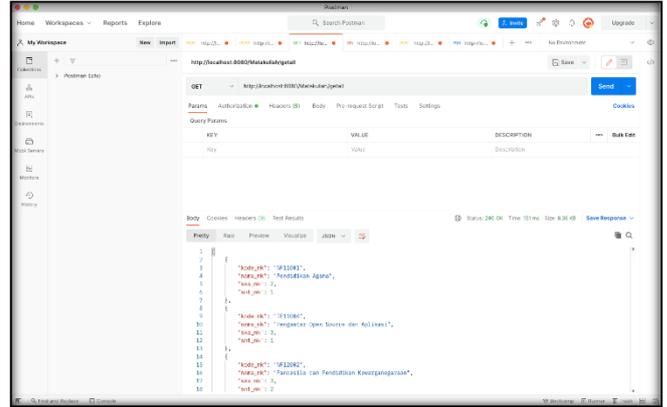
Melakukan *method* GET untuk mengecek atau melihat data yang ada pada daftar matakuliah dengan URI <http://localhost:8080/Matakuliah/get>, dengan memasukkan parameter kode matakuliah dengan *syntax key* `kode_mk` pada kolom *params*, akan menampilkan hasil seperti pada gambar 18



Gambar 19. Hasil dari *method* GET modul matakuliah model

Lalu jika ingin melihat data matakuliah secara keseluruhan, dapat menggunakan URI <http://localhost:8080/Matakuliah/getall>, lalu sistem akan

menampilkan semua daftar matakuliah seperti yang ada pada gambar 20



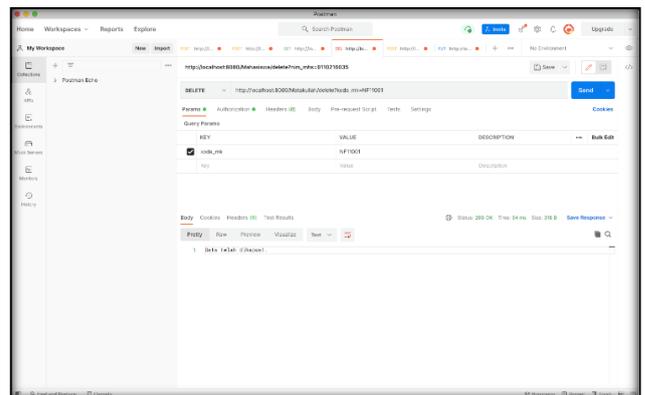
Gambar 20. Hasil dari *method* GET(all) pada modul Matakuliah model

Pada hasil keluaran dari *method* GET dengan atau tanpa parameter kode matakuliah, didapatkan keluaran yang menampilkan data matakuliah berformat JSON dengan struktur :

- `kode_mk` : kode matakuliah
- `nama_mk` : nama matakuliah
- `sks_mk` : jumlah sks matakuliah
- `smt_mk` : semester matakuliah

2. Melakukan *Method* DELETE

Melakukan delete data matakuliah dengan menggunakan *method* DELETE dengan URI <http://localhost:8080/Matakuliah/delete> untuk menghapus data dari matakuliah model, dengan memasukkan parameter kode matakuliah dan menggunakan *syntax key* `kode_mk` seperti pada gambar 21 di bawah ini.

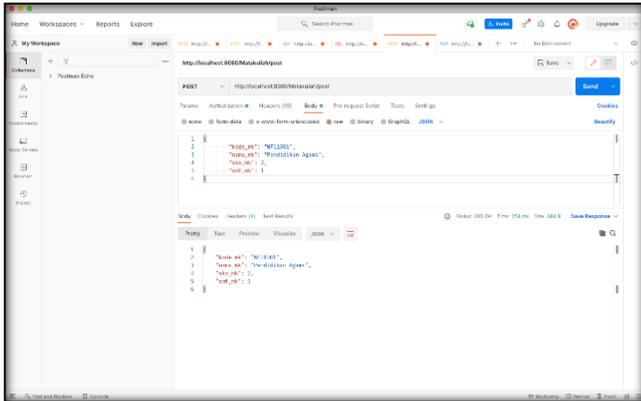


Gambar 21. Hasil dari *method* delete pada modul matakuliah model

Sistem akan mengembalikan keluaran dengan format string “Data telah dihapus!” seperti pada gambar 21, yang mana artinya berhasil menghapus data yang diinginkan.

3. Melakukan *method POST*

Melakukan *method POST* untuk menambahkan data matakuliah dengan URI <http://localhost:8080/Matakuliah/post> dengan memasukkan *request body* seperti pada gambar 22 dengan memilih *radio button* jenis *raw* dan memasukkan *text* berformat JSON, akan menampilkan hasil seperti pada gambar 22 di bawah ini.

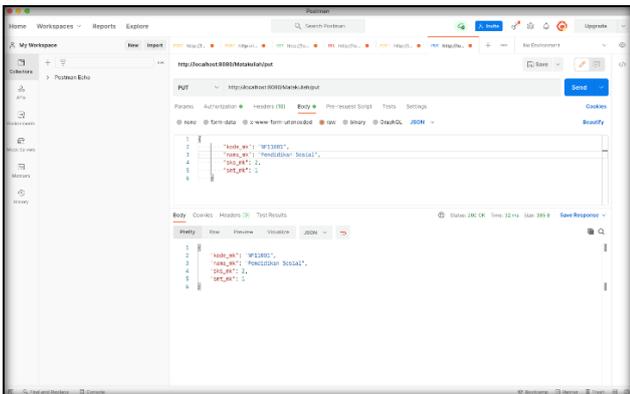


Gambar 22. Hasil dari *Method POST* pada Modul Matakuliah Model

Data yang sudah dimasukkan sesuai dengan gambar diatas, maka data yang diharapkan masuk ke dalam sistem, akhirnya berhasil ditambahkan dengan status 200, dan *body* menampilkan ulang apa yang kita masukkan pada kolom *body* diatasnya.

4. Melakukan *Method PUT*

Melakukan *method PUT* untuk mengubah data matakuliah dengan URI <http://localhost:8080/Matakuliah/put> dengan memasukkan *request body* seperti pada gambar 23 dengan memilih *radio button* jenis *raw* dan memasukkan *text* berformat JSON sama seperti akan melakukan *method POST*.



Gambar 23. Hasil Keluaran dari *Method PUT* pada Matakuliah

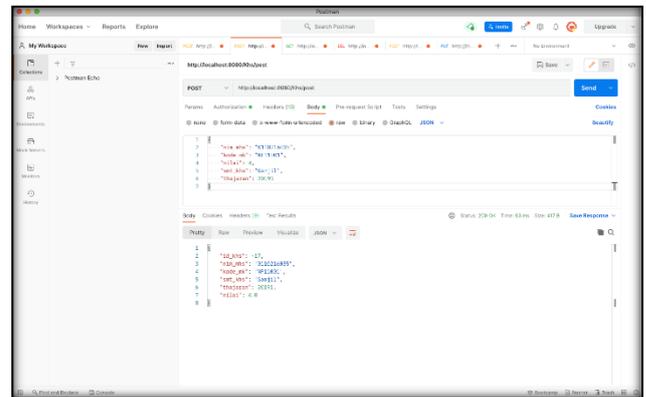
Jika output yang dihasilkan terlihat seperti pada gambar 26, maka pengubahan data dengan *method PUT* berhasil dilakukan.

5.1.6. Hasil Modul KHS Model

Pengujian fungsional ini dilakukan dengan Postman, data yang diuji adalah data modul kartu hasil studi (KHS) yang akan diakses setelah mendapatkan token, token tersebut diletakkan pada kolom *Authorization* bagian *bearer* token, dan melakukan akses dengan alamat URI utama <http://localhost:8080/Khs>.

1. Melakukan *Method POST*

Melakukan *method POST* untuk menambahkan data pada modul kartu hasil studi (KHS) dengan URI <http://localhost:8080/Khs/post> dengan memasukkan *request body* seperti pada gambar 24 dengan memilih *radio button* jenis *raw* dan memasukkan *text* berformat JSON, akan menampilkan hasil seperti pada gambar 18.

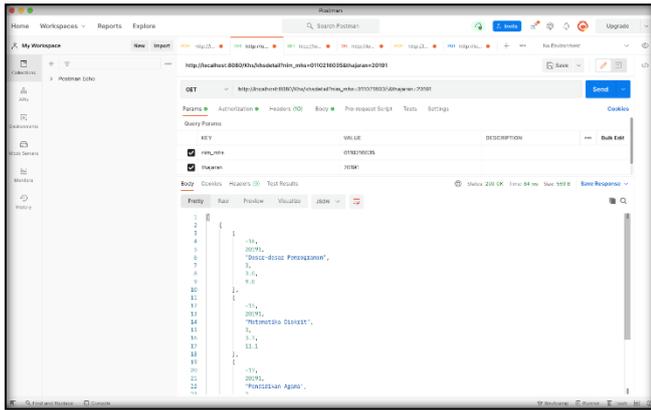


Gambar 24. Hasil dari *Method POST* pada Modul KHS Model

Data yang sudah dimasukkan sesuai dengan gambar diatas, maka data yang diharapkan masuk ke dalam sistem, akhirnya berhasil ditambahkan dengan status 200, dan *body* menampilkan ulang apa yang kita masukkan pada kolom *body* di atasnya.

2. Melakukan *Method GET*

Melakukan *method GET* untuk mengecek atau melihat data yang ada pada daftar kartu hasil studi (KHS) dengan URI <http://localhost:8080/Khs/Khsdetail>, dengan memasukkan parameter *nim* mahasiswa dan tahun ajaran dengan *syntax key* *nim_mhs* dan *tahun_ajaran* pada kolom *params*, akan menampilkan hasil dari hasil studi yang diambil oleh mahasiswa pada tahun ajaran yang *diinput seperti* pada gambar 25.



Gambar 25. Hasil dari *method* GET pada *modul* KHS model

```

19  [
20  |
21  |   -17,
22  |   "Pendidikan Agama",
23  |   2,
24  |   4,8,
25  |   8,8
26  | ]
27  ],
28  "Bobot X nilai" : 28.1,
29  "Total SKS" : 8,
30  "Total SKS yang telah ditempuh" : 16,
31  "Indeks Per Semester" : 3.5125"
32
    
```

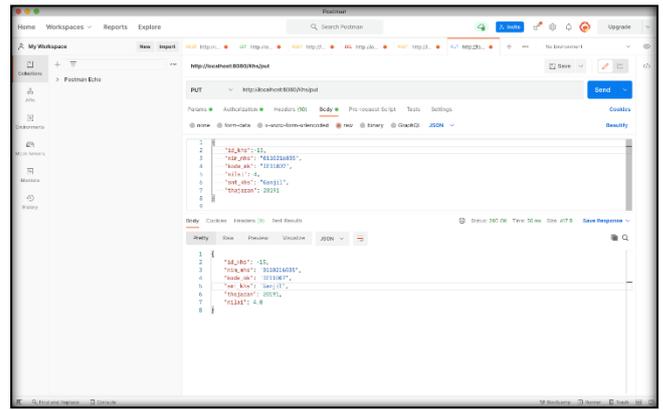
Gambar 26. Hasil dari *method* GET pada *modul* KHS model

Pada hasil keluaran dari *method* GET dengan atau tanpa parameter nim mahasiswa dan tahun ajaran, didapatkan keluaran yang menampilkan data kartu hasil studi berformat JSON dengan struktur :

- id_khs = id dari matakuliah pada modul khs
- tahun_ajaran = tahun ajaran kartu hasil studi
- nama_mk = nama matakuliah yang diambil
- sks_mk = jumlah sks yang ada pada matakuliah tersebut
- nilai = nilai yang didapatkan pada matakuliah tersebut
- bobot_sks dikali nilai
- total_sks pada semester tersebut
- total_sks keseluruhan
- indeks per semester yang didapatkan pada semester tersebut

3. Melakukan *Method* PUT

Melakukan *method* PUT untuk mengubah data kartu hasil studi (KHS) dengan URI <http://localhost:8080/Khs/put> dengan memasukkan *request body* seperti pada gambar 27 dengan memilih *radio button* jenis *raw* dan memasukkan *text* berformat JSON sama seperti akan melakukan *method* POST, hanya saja ditambahkan id_khs agar sistem tau id mana yang akan diubah.

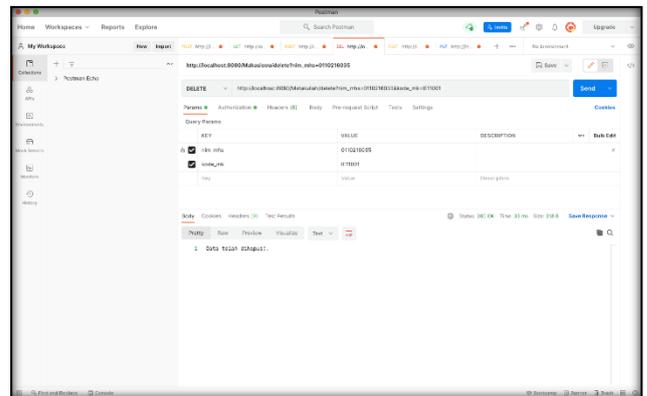


Gambar 27. Hasil dari *method* PUT pada *modul* KHS model

Jika output yang dihasilkan terlihat seperti pada gambar 27, maka pengubahan data dengan *method* PUT berhasil dilakukan.

4. Melakukan *Method* DELETE

Melakukan *delete* data dengan menggunakan *method* DELETE dengan URI <http://localhost:8080/Khs/delete> untuk menghapus data dari kartu hasil studi (KHS) model, dengan memasukkan parameter nim mahasiswa dan kode matakuliah, dengan *syntax key* nim_mhs dan kode_mk seperti gambar dibawah ini.

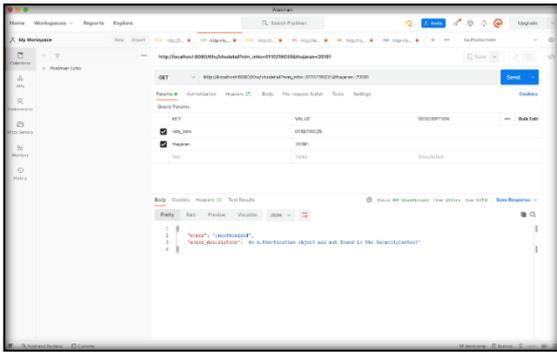


Gambar 28. Hasil dari *method* DELETE pada *modul* KHS model

Setelah memasukkan parameter nim mahasiswa dan kode matakuliah yang ada, maka *output* “data telah dihapus” akan muncul sebagai keluaran seperti yang ada pada gambar 28.

5.1.7. Hasil Pengujian Jika Tanpa Token

Setelah pengujian terhadap API berhasil dilakukan dengan token, kini saatnya menguji bagaimana API yang diakses tanpa menggunakan token pada *Authorization*, berikut adalah *output* yang ditampilkan pada aplikasi Postman saat mencoba untuk GET pada data KHS.

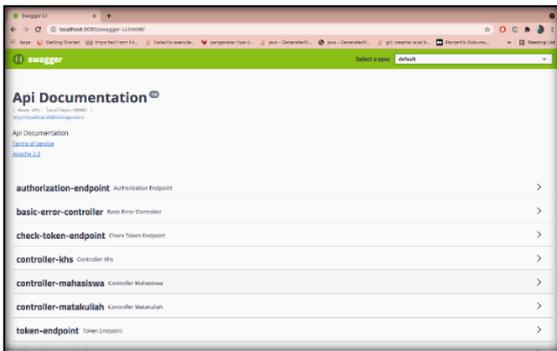


Gambar 29. Hasil Jika Tanpa Token

Pada saat kita tidak memasukkan token *authorization* pada saat mengakses API *web service*, maka hasil keluaran yang muncul adalah 401 *Unauthorized*.

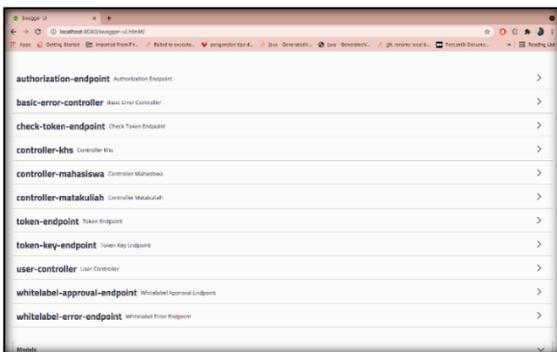
5.2. Dokumentasi API

Dokumentasi berikut bertujuan untuk memberikan penjelasan mengenai API yang telah dirancang, sehingga dapat dengan mudah digunakan oleh pengembang lainnya, berikut adalah tampilan dokumentasi dari API ini, dengan menggunakan Swagger 2.

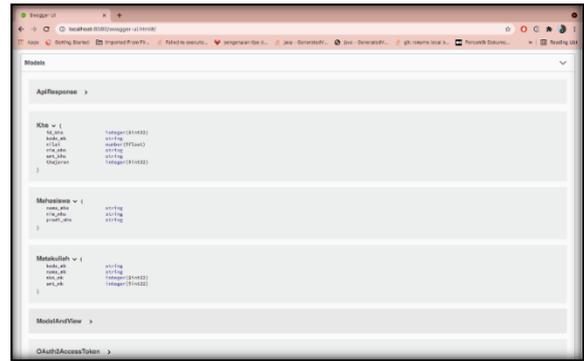


Gambar 30. Swagger Documentation (a)

Fungsi utama dalam pembuatan dokumentasi API ini adalah pengguna dapat melihat *method*, dan model apa saja yang ada dan tersedia, tipe data yang digunakan, mapping yang digunakan, dan apa saja parameter yang dibutuhkan untuk sebuah *method*, dengan mengakses <http://localhost:8080/Swagger-ui.html>.



Gambar 31. Swagger Documentation (b)



Gambar 32. Swagger Documentation (c)

5.3. Pengujian Menggunakan Black Box

Pengujian dengan *Blackbox testing* dengan 13 skenario yang diuji menggunakan metode pengujian *BlackBox* dan Postman, 13 skenario berhasil diuji dan berjalan 100%, dan sistem menghasilkan respon yang sesuai dengan yang diharapkan.

6. KESIMPULAN DAN SARAN

Pada bagian kali ini dijelaskan tentang kesimpulan dan saran dari penelitian tugas akhir yang telah dilakukan, bagian ini diakhiri dengan saran yang dapat dilakukan pada penelitian selanjutnya.

6.1. Kesimpulan

Berdasarkan hasil penelitian tugas akhir ini, dapat disimpulkan bahwa :

1. Rancang bangun API *web service* dilakukan dengan tahapan-tahapan: (1) Pendefinisian masalah dan ruang lingkup kebutuhan modul web KHS (Kartu Hasil Studi) pada sisitem informasi akademik dilakukan pada saat awal, (2) perancangan sistem mulai dari rancangan diagram alir, URI, hak akses dan skema otentikasi, dokumentasi API, (3) Implementasi program menggunakan Spring *Framework* dan database PostgreSQL, dengan sistem otentikasi menggunakan Oauth2, dan dokumentasi dengan menggunakan Swagger, (4) Pengujian aplikasi API *web service*, (5) Penarikan kesimpulan dan saran.
2. REST API *web service* telah berhasil diimplementasikan dengan hasil pengujian terhadap 4 fitur, yaitu GET, POST, PUT, DELETE pada 3 modul (mahasiswa, matakuliah, KHS), dengan 13 skenario yang diuji menggunakan metode pengujian *BlackBox* dan Postman, 13 skenario berhasil diuji dan berjalan 100%, dan sistem menghasilkan respon yang sesuai dengan yang diharapkan.

6.2. Saran

Berdasarkan hasil penelitian tugas akhir ini, beberapa hal dapat dijadikan saran untuk pengembangan penelitian berikutnya :

1. Implementasi *database* pada *API web service* dapat lebih kompleks dan menyerupai *database* utama AIS STT Terpadu Nurul Fikri.
2. Dapat menampilkan fitur Indeks Per Kumulatif (IPK) pada penelitian selanjutnya.
3. Pembuatan *API web service* untuk fitur aplikasi *mobile* AIS lainnya dibuat menggunakan RESTful Spring Framework.

DAFTAR PUSTAKA

- [1] F. Kapojos et.al., "Implemenatasi *Service-Oriented Architecture* dengan *Web Service* untuk Aplikasi Sistem Informasi Akademik," *Jurnal Teknik Elektro dan Komputer*, Vol. 1, No. 1, 2012.
- [2] T. Erl, "*Service Oriented Architecture: Concepts, Technology, and Design*," Prentice Hall Professional Technical Reference, 2005.
- [3] E Indrayani, "Pengelolaan Sistem Informasi Akademik Perguruan Tinggi berbasis Teknologi Informasi dan Komunikasi (TIK)," *Jurnal Penelitian Pendidikan*, Vol. 12, No. 1, pp. 51-67, 2011.
- [4] T Andriyanto, "Rancang Bangun Sistem Informasi Praktek Kerja Lapangan Terintegrasi menggunakan *Web Service*," *Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, Vol. 7, No. 2, pp. 551-558, 2016.
- [5] R Daigneau, "*Service Design Patterns: Fundamentals Design Solutions for SOAP/WSDL and RESTful Web Services*," Boston: Pearson educatin.Inc, 2012.
- [6] D. Sprott & L. Wilkes, "*Understanding Service-Oriented Architecture*," *Microsoft Architect Journal*, Vol. 1, No. 1, pp. 10-17, 2004.
- [7] B. Burke, "*RESTful Java with Jax-RS*," O'Reilly Media, Inc, 2009.
- [8] J. Webber, S. Parastatidis, & I. Robinson, "*REST In Practice: Hypermedia and Systems Architecture*," O'Reilly Media, Inc, 2010.
- [9] R. S. Pressman, "*Software Testing Strategies*," In *Software Engineering: A Practitioner's Approach*.: McGraw-Hill, 2001.
- [10] J. Hunt, "*Guide to the Unified Process featuring UML, Java and Design Patterns*," Springer Science & Business Media, 2006.
- [11] A. Mujahid et.al., "Analisis dan Pengembangan Sistem Informasi Pengelolaan Masjid berbasis

Mobile dengan Teknologi *API Web Service*," *Jurnal Informatika Terpadu*, Vol. 7, No. 2, hal. 80-86, 2021. <https://doi.org/10.54914/jit.v7i2.368>

- [12] M. R. A. Putra, & S. Munir, "Rancang Bangun *Website Customer Relationship Management (CRM)* pada Modul Akuntansi Studi Kasus CV Esindo Multi Tata," *Jurnal Informatika Terpadu*, Vol. 5, No. 1, hal. 24-29, 2019. <https://doi.org/10.54914/jit.v5i1.176>

Jurnal Informatika Terpadu

Vol. 8 No. 1 Tahun 2022

Daftar Isi

Perancangan Sistem Pengelola Inventaris berbasis Web menggunakan <i>Framework MVC</i>	01
Annisa Rinjani, Sirojul Munir	
Perancangan dan Implementasi <i>Booking System</i> Lapangan menggunakan <i>Framework MVC</i> berbasis Web	08
Haya Rasikhah, Ahmad Rio Adriansyah	
Analisis dan Implementasi Modul Akuntansi pada Aplikasi ERP berbasis <i>Open Source</i>	13
Fadilla Arfikriyana, Suhendi	
Penerapan <i>Log Analyzer</i> untuk Mengetahui Lalu Lintas Jaringan berbasis <i>Elasticsearch, Logtash, dan Kibana</i>	21
Muhammad Jafier Rama Putra, Henry Saptono	
Analisis dan Perancangan Aplikasi <i>Mobile</i> untuk Donasi menggunakan Metode <i>Hybrid</i> berbasis React Native	26
Muhammad Abdul Karim, Ahmad Rio Adriansyah	
Analisis dan Perancangan Aplikasi Penganggaran Barang berbasis Web pada Unit Sarana Prasarana Perguruan Tinggi	35
Siti Zahra Maharani, Ahmad Rio Adriansyah	
Pengembangan Permainan Edukasi guna Pengenalan Budaya Tak Benda di Indonesia berbasis <i>Mobile</i>	41
Setyo Nugroho, Ahmad Rio Adriansyah	
Analisis dan Implementasi RESTful API guna Pengembangan Sistem Informasi Akademik pada Perguruan Tinggi	47
Mohammad Akmaluddin Novianto, 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

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

