

## Pengembangan Aplikasi Katalog *Wedding Organizer* berbasis Android menggunakan *Restful Web Service* di Ruang Lingkup Kota Malang

Fernanda Yerisha Hartinah Ridwan<sup>1</sup>, Bayu Priyambadha<sup>2</sup>, Heru Nurwarsito<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya  
Email: <sup>1</sup>fernandayhidwan@gmail.com, <sup>2</sup>bayu\_priyambadha@ub.ac.id, <sup>3</sup>heru@ub.ac.id

### Abstrak

Pernikahan ialah sebuah kegiatan yang akan dilakukan oleh hampir seluruh masyarakat yang sudah dewasa. Menurut data dari Badan Pusat Statistik pada tahun 2015 terdapat 1.958.394 pasangan yang menikah di Indonesia, sedangkan untuk di Jawa Timur sendiri terdapat 313.150 pasangan nikah yang tercatat. Angka yang cukup besar jika dibandingkan dengan keseluruhan dari jumlah pasangan yang menikah di Indonesia. Pernikahan sendiri bertujuan untuk menyatukan dua insan dalam janji nikah. Dalam suatu pernikahan, akan ada sebuah perayaan untuk merayakan terucapnya janji nikah tersebut. Biasanya upacara pernikahan ini memiliki berbagai macam variasi yang mengikuti tradisi budaya, adat, suku, agama hingga jenjang sosial. Agar upacara pernikahan ini menjadi suatu momen yang tidak terlupakan baik untuk mempelai maupun keluarga mempelai. Di dalam sebuah upacara pernikahan akan terdapat jamuan-jamuan makan untuk para tamu undangan, dekorasi-dekorasi unik untuk memeriahkan pesta, riasan untuk mempercantik kedua mempelai hingga pernak-pernik unik lain untuk membuat pesta pernikahan yang meriah. Akan tetapi banyak dari calon mempelai yang sulit mengatur waktu untuk mencari *vendor* maupun *wedding organizer* yang sesuai dengan keinginan mereka untuk mengatur prosesi pernikahannya. Juga informasi dari pihak *vendor* maupun *wedding organizer* itu sendiri sulit untuk diakses karena kurangnya promosi. Sehingga dibuatlah penelitian ini, yang mengembangkan sebuah sistem untuk menjadi jembatan antara pihak *vendor* atau *wedding organizer* dengan calon mempelai. Dimana sistem tersedia dalam 2 platform, yaitu platform website dan *mobile*. Untuk pihak *vendor* dapat mengakses pada platform website untuk menginputkan informasi detail bisnisnya, sedangkan untuk calon mempelai dapat mengakses pada platform *mobile* untuk mencari dan mendapatkan informasi daftar *vendor*. Dalam membangun sistem ini, peneliti menggunakan metode *Restful Webservice* dalam pertukaran data, sehingga sistem tidak langsung mengelola data yang ada di basis data, melainkan mengakses data dalam bentuk *JSON* yang ada di *API* tersebut.

**Kata kunci:** *Wedding Organizer, Vendor, Pernikahan, Restful webservice, Android*

### Abstract

*Married is activity that everyone will do someday when they grown up. Based on Badan Pusat Statistik 's data on 2015 there's 1.958.394 couples who had married in Indonesia, in East Java itself, there's 313.150 couples who had married and had recorded. It's a lot of number if its compared to a number of all couples who had married in Indonesia. The marriage itself, has an aim to unite two persons in marriage vows. Theres must be a wedding ceremony to celebrate that marriage. That wedding ceremony have some kind of variant that based on the tradition, tribe, the habit and the religion of the couple. So the couple wants the wedding ceremony be the unforgettable one. In wedding ceremony, theres banquets for the guests, decorations, make up for the bride and the groom and the gifts. But, theres a lot of couples that hard to arrange their time to find the right vendor or wedding organizer that suit their wedding's theme and to arrange the wedding ceremony. Beside, information about the vendor or wedding organizer are difficult to find, because of the lack of promotion. So, that's why this research are made to develop a system to connect the vendor with the couples. There's 2 platform in this system, that are in website and mobile. For the vendor, they can access the website platform to input detail information about their bussiness. For the couple, they can access the mobile platform to search and get information about vendor. While developing this system, the researcher use the Restful Webservice method to share data between that platform, so the system can't directly access the data in database. And the system access data from API that it provided in JSON form.*

**Keywords:** *Wedding Organizer, Vendor, Marriage, Restful webservice, Android*

---

## 1. PENDAHULUAN

Menurut data dari Badan Pusat Statistik pada tahun 2015 terdapat 1.958.394 pasangan yang menikah di Indonesia, sedangkan untuk di Jawa Timur sendiri terdapat 313.150 pasangan nikah yang tercatat, angka ini termasuk pernikahan diluar KUA, yang berarti seperdelapan dari angka jumlah pasangan nikah di Indonesia. Angka ini cukup terbilang besar dibandingkan dengan jumlah pernikahan di Sumatera Utara yang hanya mencapai 86.896 pasangan. Ini berarti di Jawa Timur sendiri minat pernikahan cukup tinggi (Badan Pusat Statistik, 2015).

Pernikahan sendiri adalah suatu hal yang sangat manusiawi dimana pernikahan ini bertujuan menyatukan dua insan berbeda dalam sebuah janji nikah, yang biasanya akan dirayakan dengan beberapa upacara dan adat istiadat, yang akan dihadiri oleh keluarga maupun kerabat dari kedua mempelai. Biasanya upacara pernikahan ini memiliki berbagai macam variasi yang mengikuti tradisi budaya, adat, suku, agama hingga jenjang sosial. Upacara pernikahan dengan adat tertentu biasanya berkaitan dengan aturan suku maupun ajaran agama tertentu.

Seiring berkembangnya zaman, banyak pebisnis yang melihat adanya peluang dibidang ini untuk dapat mengatur jalannya sebuah pesta, sehingga mulailah banyak bermunculan bisnis yang berkecimpug di dunia jasa pernikahan yang biasanya disebut Wedding Organizer. Wedding Organizer disini dimaksudkan untuk mengatur segala keperluan pesta pernikahan. Alasan adanya bisnis ini biasanya adalah dikarenakan banyak calon pengantin yang kurang berpengalaman dan kurangnya waktu untuk mengatur sebuah pesta. Akan tetapi tak sedikit dari calon pengantin yang merasa kecewa dengan hasil kinerja dari beberapa Wedding Organizer yang mereka pilih. Ini dikarenakan minimnya informasi mengenai suatu Wedding organizer.

Oleh karena itu, penulis akan mengajukan penelitian ini dengan membuat sebuah sistem yang berjudul “Pengembangan Aplikasi Katalog Wedding Organizer Berbasis Android Menggunakan RESTful Web Service di Ruang Lingkup Kota Malang”. Dalam penelitian ini,

aplikasi dibangun pada platform mobile Android dan juga pada platform website. Penulis akan menerapkan teknologi RESTful webservice untuk menjembatani antara server-side dengan client-side agar diharapkan data yang ada pada server-side tidak mudah langsung diakses oleh user begitu saja.

Istilah RESTful sendiri pertama kali diperkenalkan oleh Roy Fielding pada tahun 2000. RESTful web service adalah kumpulan dari resource yaitu URI, tipe media internet yang di dukung dan sesuai standar HTTP, fungsi-fungsi operasional yang didukung web service (seperti : GET, POST, PUT, DELETE) dan API yang berbasis hypertext. Dalam pembuatan API itu sendiri, peneliti menggunakan sebuah *framework* PHP, yaitu Laravel.

Diharapkan penelitian ini dapat membantu masyarakat, khususnya untuk pasangan yang memutuskan untuk menikah, sehingga akan mendapatkan beberapa inspirasi dan informasi yang sesuai dengan konsep pesta pernikahan masing-masing. Juga untuk vendor maupun wedding organizer dapat menjadi ajang mempromosikan jasa maupun barang yang mereka tawarkan, sehingga dapat memberikan informasi produk yang akurat dan sesuai kepada calon pelanggan mereka.

## 2. LANDASAN KEPUSTAKAAN

Sebagai dasar teori dalam pelaksanaan penelitian ini, maka dilakukanlah sebuah studi literatur dengan meninjau beberapa penelitian serupa yang telah dilakukan untuk dijadikan bahan acuan dan referensi.

### 2.1. Kajian Literatur

Terdapat beberapa literatur yang digunakan pada penelitian ini yaitu, yang pertama adalah penelitian dengan judul Pengembangan Sistem Informasi 3 Pilar Dalam Penyelesaian Perkara Tilang Di Kota Kediri Menggunakan Restful Web Service. Penelitian ini membahas mengenai penyelesaian proses bisnis pelanggaran lalu lintas yang dikenakan denda berupa tilang dengan menerapkan framework Spring Boot sebagai web service yang mengimplementasikan teknologi RESTful (Nur, 2018).

Penelitian kedua yang digunakan sebagai referensi dalam penelitian ini ialah Pengembangan Perangkat Lunak Aplikasi

Monitoring Klimatologi Menggunakan Metode Restful Web Service Berbasis Android (Studi Kasus : Stasiun Klimatologi Karangploso Malang). Penelitian ini membahas mengenai pengembangan aplikasi perangkat bergerak untuk mengakses data dari salah satu stasiun BMKG di Kota Malang dengan cara merequest data dari web service yang menggunakan metode RESTful. Data yang ditampilkan akan berupa tabel dan chart pada sisi client (Rohman, 2018).

Penelitian ketiga yang juga digunakan sebagai referensi pada penelitian ini adalah Rancang Bangun Aplikasi Penentuan Vendor Pernikahan Untuk Usaha *Wedding Organizer* Menggunakan Metode Smarter Berbasis Web. Penelitian ini membahas mengenai pengembangan aplikasi yang dapat menentukan vendor yang baik untuk usaha di bidang wedding organizer, penelitian ini menggunakan metode smarter dalam pengambilan keputusannya (Cahyo, 2018). Ketiga penelitian tersebut menjadi referensi dalam penelitian ini dikarenakan memiliki kesamaan objek ataupun metode pembahasan yang digunakan yaitu teknologi *restful webservice* dan wedding organizer.

## 2.2. Wedding Organizer

*Wedding Organizer* atau biasa disebut juga dengan istilah *Wedding Planner* adalah seseorang atau sekelompok orang yang akan membantu dalam merencanakan dan mengatur setiap atau semua aspek dari upacara pernikahan dan resepsi (Goulet, et al., 2012). WO disini berperan sebagai pemberi layanan untuk menyiapkan dan memenuhi segala kebutuhan pernikahan. WO akan menyusun konsep sebuah pesta, yang nantinya akan disesuaikan dengan keinginan mempelai. Kemudian, mereka akan merencanakan kebutuhan pesta termasuk anggaran dari pesta, dan WO akan mengatur jalannya sebuah pesta. Sehingga wedding organizer dapat dikatakan suatu usaha yang bergerak di bidang jasa untuk membantu merencanakan pesta pernikahan dan mengatur pesta pernikahan di hari H.

## 2.3. Vendor

Kata *vendor* memiliki arti seseorang yang menjual rumah, tanah dan lain sebagainya, atau bisa disebut sebagai penjual, berdasarkan dari Kamus Besar Bahasa Indonesia (KBBI) (KBBI, 2018). Vendor dalam hal ini adalah penyedia

barang atau perlengkapan pernikahan. Contohnya adalah penyedia layanan dekorasi tempat pesta dan penyedia catering pernikahan.

## 2.4. Katalog

Katalog ialah bahasa serapan yang diambil dari kata *Katalogos* yang merupakan frase Yunani, dimana terdiri dari kata *Kata* dan kata *Lagos*. Arti kata *Kata* ialah sebuah sarana atau menurut dan arti kata *Lagos* mempunyai banyak makna, yaitu kata, susunan, alasan dan nalar. Sehingga katalog dapat diartikan sebagai sebuah karya beserta isinya yang tersusun menurut cara yang masuk akal.

Katalog ialah carik kartu, daftar, atau buku yang memuat nama benda atau informasi tertentu yang ingin disampaikan, disusun secara berurutan, teratur, dan alfabetis, merupakan definisi dari kata Katalog berdasarkan Kamus Besar Bahasa Indonesia (KBBI, 2018). Katalog dapat menjadi sebuah dokumentasi untuk sebuah bisnis. Daftar koleksi tersebut dapat dalam bentuk kartu, selebaran, buku, aplikasi atau yang lainnya. Katalog berfungsi untuk menyimpan informasi mengenai produk sehingga mudah dibaca dan lebih menarik.

## 2.5. Android Studio

Android Studio merupakan *Integrated Development Environment (IDE)* resmi yang dikeluarkan oleh pihak Android. Pada *IDE* ini dapat dilakukan penulisan kode, *debugging*, pengujian, dan alat pembuatan aplikasi (Google Developers, 2018).

Android Studio sendiri menggunakan bahasa pemrograman *Java* milik *Oracle*, akan tetapi Android Studio sendiri kini telah mengembangkan bahasa pemrogramannya sendiri yaitu *Kotlin*. Dibandingkan dengan *IDE Eclipse* yang sama-sama menerapkan bahasa pemrograman *Java*, kini Android Studio lebih terkenal dalam pengembangan aplikasi android, dikarenakan Eclipse kini sudah tidak didukung lagi oleh Google, yang disampaikan oleh manajer produk Android Studio sendiri melalui website resmi *IDE* tersebut (Eason, 2015). Selain itu, Android Studio hadir dengan dokumentasi yang lebih lengkap, sehingga lebih memudahkan pengembang dalam proses pengembangan aplikasi.

## 2.6. Laravel

*Laravel* adalah sebuah *web application framework* berbasis PHP yang *open source*.

*Laravel* menggunakan konsep model–view–controller (MVC) yang dapat memudahkan untuk pengembangan sebuah web service. *Laravel* sendiri berada dibawah lisensi MIT License. Pada *Laravel* versi 5.6, sudah terdapat tambahan fungsi untuk membuat sebuah *API* yang nantinya dapat dijadikan web service. *Laravel* juga menyediakan fitur *Eloquent ORM*, fitur ini menangani hubungan objek pada database, *Laravel* akan menyimpan record database yang telah dibuat dengan adanya fitur ini. Ini akan memudahkan untuk pengembangan aplikasi. Framework ini dikembangkan oleh Taylor Otwell dengan menggunakan PHP 5.3, framework ini juga memiliki dokumentasi yang lengkap dan baik sehingga memudahkan *developer* dalam pengembangan sebuah sistem. Mulanya pembuatan proyek ini dikarenakan Otwell tidak dapat menemukan *framework* yang selalu *update* dengan versi PHP. Oleh karena itu, Otwell memiliki ide untuk mengembangkan sebuah *framework* dengan nama *Laravel*. Oleh karena itu *Laravel* menisyrakat PHP versi 5.3 keatas (Rohman, 2014).

**2.7. Restful Web Service**

REST (*REpresentational State Transfer*) ialah sebuah arsitektur komunikasi yang kerap digunakan dalam sebuah pengembangan layanan berbasis web, dimana REST sendiri merupakan standar arsitektur komunikasi yang berbasis web juga. Pada umumnya, protokol untuk komunikasi datanya, REST menggunakan protokol *Hypertext Transfer Protocol* atau disingkat sebagai HTTP. Arsitektur REST ini dikenalkan oleh Roy Fielding pertama kali pada tahun 2000 (Fielding, 2000).

Arsitektur REST menyediakan sebuah *resources* (sumber daya/data), yang mana ketika *client* mengakses server, maka *resources* tersebut akan ditampilkan untuk penggunaan selanjutnya. Adapun setiap *resources* tersebut memiliki sebuah URIs (*Universal Resource Identifiers*) atau global ID untuk mengidentifikasi masing-masing *resources* tersebut. *Resources* tersebut juga direpresentasikan dalam sebuah format data, yang umum digunakan ialah format data JSON ataupun XML.

**3. Metodologi Penelitian**

Bab ini akan membahas bagaimana urutan dan metodologi yang digunakan dalam penelitian ini untuk menyelesaikan masalah

dalam penelitian ini. Metode penelitian sendiri artinya ialah menggunakan suatu cara untuk mencapai suatu tujuan demi diperolehnya suatu keberhasilan dalam penelitian dengan digunakannya suatu teknik tertentu, maka dalam pelaksanaannya, metodologi yang tepat dan istimewa harus digunakan dengan tujuan diadakannya penelitian tersebut berdasarkan fakta-fakta yang ada untuk menguji kebenaran sesuatu secara ilmiah. Adapun diagram alir metodologi penelitian ini digambar seperti pada Gambar 1 berikut.



Gambar 1. Diagram Alir Metodologi Penelitian

**4. Analisis Kebutuhan**

**4.1. Penggalan Kebutuhan Sistem**

Pada penelitian kali ini, peneliti melakukan pengumpulan data untuk menggali dan menganalisis kebutuhan dari sistem dengan cara melakukan wawancara dan menyebarkan kuisioner. Kuisioner disebarkan pada sampel dari sebuah populasi masyarakat yang sudah dan akan menikah. Begitupun pada metode wawancara dilakukan pada beberapa sampling masyarakat, yaitu beberapa masyarakat yang akan merayakan upacara pernikahan dan beberapa masyarakat yang telah atau pernah merayakan upacara pernikahan. Pada saat wawancara maupun kuisioner, responden diberikan beberapa pertanyaan mengenai apa saja yang perlu disiapkan untuk merayakan sebuah pesta pernikahan, berapa lama waktu

yang dibutuhkan untuk mengumpulkan informasi mengenai *vendor* maupun *wedding organizer* yang disewa, berapa lama waktu yang dibutuhkan untuk menyiapkan seluruh kebutuhan upacara pernikahan, *vendor* apa saja yang dibutuhkan dalam perayaan pernikahan, informasi apa saja yang penting untuk diketahui oleh calon pasangan menikah mengenai *vendor* dan *wedding organizer*, dan apa kesulitan yang dihadapi dalam menyiapkan kebutuhan tersebut.

**4.2. Identifikasi Aktor**

Terdapat beberapa *stakeholders* yang terlibat dalam aplikasi katalog *Wedding Organizer* ini salah satunya adalah aktor. Dimana aktor adalah seorang pengguna aplikasi yang dapat menggunakan ataupun merasakan imbas dari adanya aplikasi. Adapun aktor yang terlibat dalam sistem ini adalah *Guest*, *Users* dan *Administrator*. Dimana *users* dapat menjadi *User Customers* atau *Users Vendors*. *User Customers* adalah pengguna dari masyarakat yang dapat melihat daftar *vendor* yang disediakan sistem. Sedangkan *users vendor* adalah pengguna yang merupakan seorang *vendor* atau *wedding organizer* sebagai penyedia informasi mengenai produk atau jasa mereka. *Administrator* adalah pengguna yang mengelola data-data yang ada pada sistem. Adapun identifikasi aktor akan dijelaskan pada tabel 1.

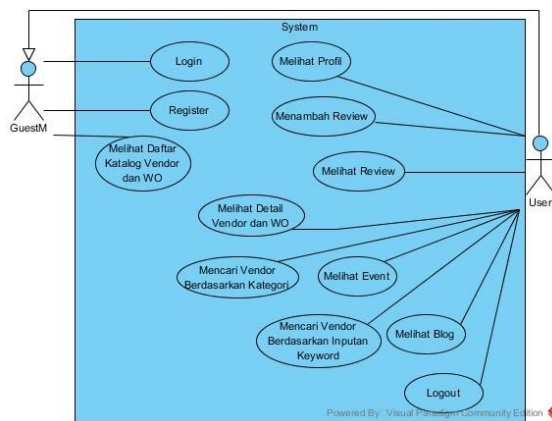
Tabel 1. Identifikasi Aktor

No.	Aktor	Deskripsi
1.	<i>Guest</i>	<i>Guest</i> adalah pengguna yang belum melakukan login pada sistem.
2.	<i>Users</i>	<i>Users</i> adalah pengguna yang sudah melakukan login pada sistem dan dapat melihat daftar <i>vendor</i> yang tersedia.
3.	<i>Vendor</i>	<i>Vendor</i> adalah penyedia informasi produk ataupun jasa yang ditawarkan.
4.	<i>Administrator</i>	<i>Administrator</i> adalah pengguna yang dapat mengelola data yang ada pada sistem.

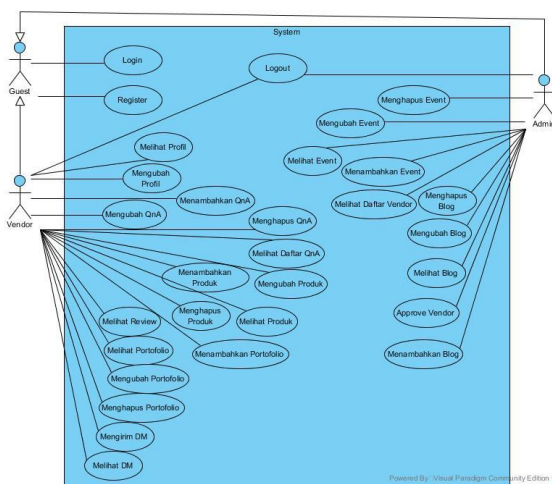
**4.3. Usecase Diagram**

*Usecase* diagram akan menggambarkan interaksi antar pengguna dengan sistem. *Usecase* merupakan penggambaran apa saja yang dapat dilakukan oleh seorang aktor pada sistem. Adapun *usecase* diagram yang dibuat ini

mengacu pada daftar kebutuhan fungsional yang telah dijabarkan sebelumnya. Tujuan adanya *usecase* diagram ini agar pembaca lebih mudah memahami fungsionalitas dari sistem dan apa saja yang dapat dilakukan oleh pengguna terhadap sistem. Adapun *usecase* diagram pada penelitian ini dapat dilihat pada gambar-gambar berikut.



Gambar 2. Usecase Diagram Sistem Mobile



Gambar 3. Usecase Diagram Sistem Website

**5. PERANCANGAN DAN IMPELENTASI**

**5.1. Perancangan**

Tahapan selanjutnya adalah merancang sebuah sistem, sebelum dilakukannya implementasi, agar sistem tidak menjauh dari tujuan dan analisis kebutuhan. Adapun pada penelitian ini, perancangan dibagi menjadi 4 bagian perancangan, yaitu perancangan arsitektur, perancangan data, perancangan algoritma dan perancangan antarmuka.

Pada perancangan arsitektur, akan menjelaskan bagaimana alur dari sebuah fitur

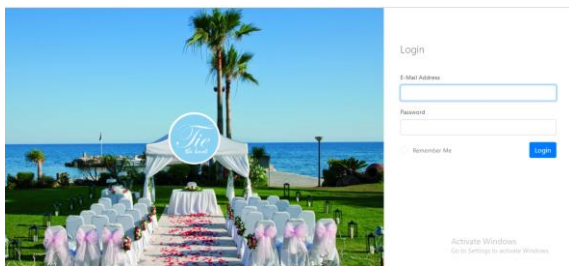
fungsional dari beberapa kebutuhan fungsional. Adapun yang dijabarkan yaitu *sequence* diagram yang menggambarkan alur pesan atau *message* dan waktu hidup sebuah objek pada suatu kebutuhan fungsional, sedangkan *class* diagram menggambarkan struktur sistem dari sudut pandang pendefinisian klas, dimana klas tersebut mengandung atribut dan *method* yang terkandung pada klas tersebut.

Perancangan algoritma menjelaskan algoritma dari sub sistem, bagaimana sebuah fitur dapat diterapkan dengan suatu algoritma tersebut. Merincikan bagaimana suatu sistem mengelola inputan dari *users* dan menjadikan output yang dapat dilihat oleh *user*.

Perancangan basis data dibuat, agar nantinya lebih mudah dalam pengimplementasiannya dan juga agar basis data sesuai dengan kebutuhan sistem dan tidak menjauh dari tujuan utama sistem. Kemudian yang terakhir ialah perancangan antarmuka. Perancangan antarmuka pada sistem katalog *wedding organizer* dan *vendor* ini bertujuan untuk merancang sebuah antarmuka yang efektif untuk sistem dan mudah dioperasikan oleh pengguna. Antarmuka pada sistem ini dirancang agar pengguna dapat merasa nyaman untuk sering menggunakan sistem ini.

**5.2. Implementasi**

Tahap selanjutnya yang perlu dilakukan ialah pengimplementasian sistem. Semua yang telah dijabarkan pada analisis kebutuhan sistem dan perancangan sistem harus ada pada pengimplementasian sistem. Sehingga pengimplementasian sistem harus mengacu pada analisis kebutuhan dan perancangan sistem. Ini berarti implementasi sistem harus sesuai dengan struktur data, algoritma hingga perancangan antarmuka.



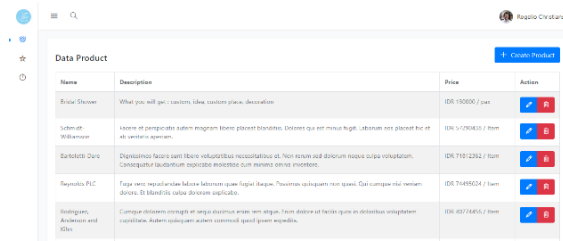
Gambar 4. Implementasi Antarmuka Login pada Website

Pada gambar 4 merupakan hasil implementasi halaman login pada *platform* website, sedangkan pada gambar 5 merupakan

implementasi antarmuka login pada *platform* *mobile* dan gambar 6 merupakan implementasi antarmuka menu utama pada *platform* website.



Gambar 5. Implementasi Antarmuka Login pada Mobile



Gambar 6. Implementasi Antarmuka Menu Utama pada Website

**6. PENGUJIAN**

Setelah dilakukan pengimplementasian sistem, maka tahapan selanjutnya ialah pengujian sistem. Tahap pengujian ini berfungsi untuk mengetahui sistem yang telah dikembangkan pada penelitian ini apakah telah benar-benar memenuhi kebutuhan sistem dan telah sesuai dengan perancangan sistem. Pengujian yang dilakukan pada penelitian ini ialah pengujian unit, pengujian integrasi, pengujian validasi dan juga pengujian *usability*.

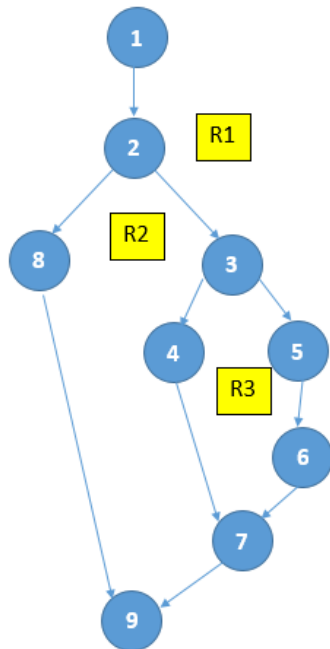
**6.1. Pengujian Unit**

Pengujian unit dilakukan dengan menguji bagian terkecil dari sebuah sistem, yang menghususkan pengujian pada komponen dari perangkat lunak yang telah dikembangkan. Dalam pengujian unit, metode yang digunakan ialah metode pengujian *whitebox* dengan menggunakan teknik pengujian *basis path*. Salah satunya diuji pada *method* "*showVendor()*" pada *class* "*VendorController*" yang algoritma kodenya dapat dilihat pada tabel 2, sedangkan

hasil *flow graph* nya dapat dilihat pada gambar 7 berikut.

Tabel 2 Pseudocode method : showVendor, class : VendorController

No	Pseudocode
1	get connection
1	get data from webservice
2	if(response success){
3	if(data != null){
4	get data
4	set data dengan adapter
4	tampil data
5	} else {
6	Tampil alert tidak ada
7	data
8	}
8	} else if {
9	Tampil alert jaringan error
	}



Gambar 4 FlowGraph Pengujian showVendor

Dari gambar tersebut dapat dihitung berapa *cyclomatic complexity* (V(G)) yang didapatkan untuk menentukan berapa banyak kasus uji yang dilakukan dalam pengujian unit.

- $V(G) = \text{jumlah region} = 3$
- $V(G) = \text{jumlah edge} - \text{jumlah node} + 2 = 10 - 9 + 2 = 3$
- $V(G) = \text{jumlah predicate node} + 1 = 2 + 1 = 3$

Sehingga didapatkan 3 jalur *independent path* :

- Jalur 1 : 1 - 2 - 8 - 9
- Jalur 2 : 1 - 2 - 3 - 4 - 7 - 9

- Jalur 3 : 1 - 2 - 3 - 5 - 6 - 7 - 9

Hasil pengujian pada method showVendor() setiap jalur pengujiannya dijelaskan pada Tabel 3

Tabel 3 Hasil Pengujian Unit Method “showVendor()” pada class “FragmentVendor”

No. Jalur	Data Pengujian	Expected Result	Result	Status
1	Koneksi dengan response = fail	Menampilkan alert “Jaringan error”	Menampilkan alert “Jaringan error”	Valid
2	Koneksi dengan response = fail Value != null	Menampilkan daftar data vendor	Menampilkan daftar data vendor	Valid
3	Koneksi dengan response = fail Value = null	Menampilkan alert “Belum ada data”	Menampilkan alert “Belum ada data”	Valid

### 6.2. Pengujian Integrasi

Pada pengujian ini difokuskan untuk menguji bagaimana hasil dari desain dan konstruksi arsitektur sistem bisa terintegrasi dan berjalan dengan baik.

Tabel 3 Rencana Pengujian Integrasi

No. Uji	Nama Kelas	Nama Method	Tujuan
1.	FragmentVendor or RetrofitConnection Api	showVendor() getRetrofitInstance() index()	Mengambil data daftar vendor dari webservice

Tabel 4 Hasil Pengujian Integrasi Nomor 1

Nomor Uji	1
Input Pertama	“BASE_URL” : “ <a href="https://tietheknot.000webhostapp.com/api/">https://tietheknot.000webhostapp.com/api/</a> ”
Method dari Kelas RetrofitCon	getRetrofitInstance()

<b>nection</b>	
<b>Output Pertama / Input Kedua</b>	"create()" : "Api.class"
<b>Method dari Kelas FragmentVendor</b>	showVendor()
<b>Output Kedua / Input Ketiga</b>	"@GET" : "vendor"
<b>Method dari Kelas Api</b>	index()
<b>Expected Result</b>	Berhasil koneksi dengan webservice dan mengambil data dari webservice
<b>Result</b>	Berhasil koneksi dengan webservice dan mengambil data dari webservice
<b>Status</b>	Valid

### 6.3. Pengujian Validasi

Tujuan dari dilakukan pengujian validasi ini adalah untuk memastikan fitur-fitur yang ada pada sistem telah memenuhi daftar kebutuhan dan juga sesuai dengan skenario yang telah dibuat. Maka pada pengujian ini, menerapkan metode pengujian *blackbox*, dimana dengan menggunakan metode ini dapat mengetahui kondisi suatu *input* dan *ouput* yang dihasilkan sistem saat menjalankan fitur yang ada yang sesuai dengan kebutuhan.

Tabel 5 Hasil Pengujian Validasi Melihat Daftar Katalog Vendor dan WO

<b>Kode Kebutuhan</b>	TTK-FM-04
<b>Nama Kasus Uji</b>	Melihat daftar katalog vendor
<b>Prosedur</b>	Mengakses menu daftar vendor Terkoneksi internet
<b>Expected Result</b>	Sistem menampilkan daftar vendor dan WO dalam bentuk daftar
<b>Result</b>	Sistem menampilkan daftar vendor dan WO dalam bentuk daftar
<b>Status</b>	Valid

### 6.3. Pengujian Usability

Pengujian ini dilakukan dengan cara memberikan kesempatan beberapa pengguna, yang kemudian menjadi seorang responden, untuk mencoba mengoperasikan sistem katalog

vendor dan WO ini. Peneliti hanya memberikan penjelasan, bagaimana cara kerja sistem ini dan tujuan dibuatnya sistem ini tanpa memberikan petunjuk penggunaannya. Sehingga pengguna, mencari sendiri fitur-fitur yang ada pada sistem. Kemudian, pengguna diberikan sebuah kuesioner dengan meliputi beberapa pertanyaan, menyangkut *usability* dari sistem. Adapun hasil pengujian yang didapat seperti pada tabel – tabel berikut.

Tabel 6 Perhitungan kuesioner dan Hasil Rekapitulasi *Usability Scale* pada *Vendor* dan *WO*

Responden	Skor SUS
R1	65
R2	77,5
R3	77,5
R4	70
R5	57,5
R6	85
R7	77,5
R8	80
R9	70
R10	70
R11	70
Total Skor	800
Rata-Rata	72,73

Tabel 7 Perhitungan kuesioner dan Hasil Rekapitulasi *Usability Scale* pada *Users*

Responden	Skor SUS
R1	90
R2	72,5
R3	67,5
R4	77,5
R5	57,5
R6	65
R7	85
R8	75
R9	77,5
R10	80
Total Skor	747,5
Rata-Rata	74,75

Pada tabel 6 dan tabel 7 merupakan hasil perhitungan dan rekapitulasi dari pengujian *usability* dengan menggunakan *System Usability Scale* (SUS) yang dilakukan pada pihak *vendor* dan *WO*, juga pada pengguna (*users*). Adapun cara perhitungan menggunakan SUS pada perhitungan kuesioner di atas ialah pada



pertanyaan nomor 1,3,5,7 dan 9 yang merupakan pertanyaan bertipe positif, sedangkan pada pertanyaan nomor 2,4,6,8 dan 10 yang merupakan pertanyaan bertipe negatif. Kemudian, dari 11 responden pada pihak *vendor* dan *WO* yang telah diuji, didapatkan hasil total skor SUS sebesar 800, dengan mendapatkan rata-rata skornya sebesar 72,73. Juga didapatkan total skor SUS sebesar 747,5 pada pengujian *usability* dengan 10 responden dari pihak pengguna (*users*), yang kemudian diambil rata-ratanya, dengan cara membagi total skor SUS dengan jumlah total respondennya, sehingga didapatkan skor rata-rata sebesar 74,75. Dimana nilai rata-rata skor dari kedua pihak aktor yang telah diuji, dapat dimasukkan ke dalam kategori “Acceptable”, yang artinya sistem telah dapat diterima oleh pengguna dan mudah untuk dioperasikan.

## 7. KESIMPULAN DAN SARAN

### 7.1. Kesimpulan

Dari hasil penelitian yang dilakukan maka dapat ditarik beberapa kesimpulan, bahwa Sistem Katalog *vendor* dan *wedding organizer* dapat menyediakan informasi mengenai *vendor* maupun *wedding organizer*. Adapun sistem ini dibagi menjadi 2 *platform*, yaitu pada *platform mobile* dan *platform website*. Dimana pada *platform website* terdapat beberapa fitur seperti melihat, menambahkan, mengubah dan menghapus data produk atau jasa yang dimiliki *vendor* dan melihat *review* yang diberikan oleh pengguna. Kemudian, informasi yang telah dikelola oleh *vendor* ini, akan ditampilkan pada *platform mobile*, yang dapat memudahkan pengguna untuk mengakses informasi tersebut. Sehingga sistem dapat menampilkan informasi *vendor* ataupun *wedding organizer* yang sesuai dengan kebutuhan pengguna. Kemudian dengan menggunakan RESTful *webservice* ini dapat mempermudah komunikasi data antar beda *platform* pada sistem dengan menggunakan format data JSON, sehingga data lebih mudah dikelola pada kedua *platform* yang berbeda tersebut. Dapat juga ditarik hasil pengujian *usability* sistem Katalog *vendor* dan *wedding organizer* yang menghasilkan rata-rata skor SUS sebesar 72,73 pada pengujian yang dilakukan pada 11 pihak *vendor* dan *WO* juga didapatkan rata-rata skor SUS sebesar 74,75 pada pengujian pada pihak pengguna (*users*). Sehingga dengan skor tersebut sistem ini dapat dimasukkan ke

dalam kategori “acceptable”, dimana sistem yang dikembangkan telah diterima oleh pengguna dan mudah dalam pengoperasiannya.

### 7.1. Saran

Berdasarkan pengembangan sistem Katalog *Vendor* dan *WO* di ruang lingkup Kota Malang, dapat dilakukan pengembangan lebih lanjut, baik penambahan fitur maupun penambahan fungsi yang lain. Saran yang dapat diberikan ialah seperti sistem dapat diperluas, tidak hanya di ruang lingkup Kota Malang, sistem dapat menampilkan peta pada detail *vendor*, sehingga pengguna dapat lebih mudah dalam menemukan tempat *vendor* tersebut, sistem agar dapat memberikan rekomendasi *vendor* yang bisa digunakan pada pengguna, dengan pengguna menginputkan anggaran dana, tema yang diinginkan dan kota yang diinginkan dan juga sistem agar selalu dilakukan *maintenance* dan evaluasi secara berkala agar sistem tetap berjalan dengan baik.

## DAFTAR PUSTAKA

- A. S, R. & Shalahuddin, M., 2016. *Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. 4 penyunt. Bandung: Informatika Bandung.
- Anjani, R., 2017. *Wedding News*. [Online] Available at: <https://wolipop.detik.com/read/2017/08/06/155643/3588185/854/kasih-sudah-bayar-rp-41-juta-dekorasi-pernikahan-ini-malah-mengecewakan> [Diakses 9 Agustus 2018].
- Badan Pusat Statistik, 2015. *Table Dinamis Nikah, Talak dan Cerai, Serta Rujuk, 2012-2015*. [Online] Available at: <https://www.bps.go.id/linkTableDinamis/view/id/893> [Diakses 8 Agustus 2018].
- Cahyo, F. I. D., 2018. RANCANG BANGUN APLIKASI PENENTUAN VENDOR PERNIKAHAN UNTUK USAHA WEDDING ORGANIZER MENGGUNAKAN METODE SMARTER BERBASIS WEB. p. 62.
- Eason, J., 2015. *An Update on Eclipse Android Developer Tools*. [Online] Available at: <https://android-developers.googleblog.com/2015/06/an->

- [update-on-eclipse-android-developer.html](#)  
[Diakses 12 Agustus 2018].
- Fielding, R. T., 2000. Architectural Styles and the Design of Network-based Software Architectures. *Irvine*.
- Google Developers, 2018. *Fitur Android Studio*. [Online] Available at: <https://developer.android.com/studio/features?hl=id> [Diakses 12 Agustus 2018].
- Goulet, C., Riddell, J. L. & Goulet, T., 2012. *FabJob Guide to Become a Wedding Planner*. s.l.:BookBaby.
- KBBI, 2018. [Online] Available at: <https://kbbi.web.id/vendor> [Diakses 12 Agustus 2018].
- Khabib Mustofa, A. N., 2012. PERBANDINGAN ANTARA “BIG” WEB SERVICE DENGAN RESTFUL. *Jurnal Informatika, Universitas Petra*, pp. 8-16.
- Nur, P. A., 2018. Pengembangan Sistem Informasi 3 Pilar Dalam Penyelesaian Perkara. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, pp. 3525-3532.
- Oracle, 2013. *What Are RESTful Web Service*. [Online] Available at: <https://docs.oracle.com/javaee/6/tutorial/doc/gijqy.html> [Diakses 9 Agustus 2018].
- Pressman, R. S., 2008. *Software Engineering: A Practitioner's Approach*. 8th penyunt. s.l.:Mc Graw Hill.
- Rizky, S., 2011. *Konsep Dasar Rekayasa Perangkat Lunak*. 1st penyunt. Jakarta: Prestasi Pustaka.
- Robertson, J. & Robertson, S., 2012. *Mastering the Requirements Process: Getting Requirements Right*. 3rd penyunt. s.l.:Addison-Wesley Professional.
- Rohman, A., 2014. *Mengenal Framework "Laravel" (Best PHP Frameworks For 2014)*. s.l.:ilmuti.org.
- Rohman, F. R., 2018. Pengembangan Perangkat Lunak Aplikasi Monitoring Klimatologi Menggunakan Metode RESTful Web service Berbasis Android (Studi Kasus : Stasiun Klimatologi Karangploso Malang). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, pp. 2017-2023.