

## Pemanfaatan MVC Pattern Untuk Pembuatan Software Aplikasi Administrasi Desa (Studi Kasus Desa Sukapura)

Suwaldi Mardana<sup>1</sup>, Dr. Kemas Muslim Lhaksana<sup>2</sup>, Eko Darwiyanto, S.T., M.T.<sup>3</sup>

<sup>1,2,3</sup>Fakultas Informatika, Universitas Telkom, Bandung

<sup>1</sup>suwaldim@students.telkomuniversity.ac.id, <sup>2</sup>kemasmuslim@telkomuniversity.ac.id,

<sup>3</sup>ekodarwiyanto@telkomuniversity.ac.id

### Abstrak

Pelayanan Masyarakat merupakan kegiatan yang diperuntukkan untuk melayani masyarakat untuk mempermudah dan membantu masyarakat. Pada era reformasi saat ini masyarakat terbuka dalam memberikan kritikan pada pemerintah dalam pelayanan publik dan juga pada saat ini pelayanan publik desa banyak mendapat sorotan dari masyarakat. Penelitian secara langsung di lapangan kelurahan Desa Sukapura, penggunaan sistem administrasi secara manual yang saat ini digunakan membuat waktu pelayanan relatif lama, ketergantungan terhadap kemampuan komputer tenaga administratif tertentu cukup besar, format surat tidak standar, pencatatan administratif yang kurang baik. Penerapan sistem informasi dan konsep *E-Government* menjadi pertimbangan utama bagi organisasi sektor publik (pemerintah desa) yang melakukan proses penyusunan perencanaan dan pemantauan dan evaluasi hasil pembangunan. Salah satu sistem informasi yang sudah ada yaitu aplikasi desa sundawenang yang akan diterapkan di desa sukapura berbasis *web* dengan menggunakan metode CBSE (*Component-Based Software Engineering*). CBSE adalah pendekatan untuk pengembangan perangkat lunak dan penggunaan kembali komponen-komponen perangkat lunak untuk menggantikan teknik pengembangan dari awal seperti yang dipakai selama ini. *Reusable code* pada *web* desa sundawenang ke desa sukapura memiliki *framework* yang berbeda yaitu dari *framework laravel* ke *framework codeigniter*. Hasil analisis yang didapatkan berupa nilai rata-rata persen untuk penggunaan ulang *code* pada *web* desa sundawenang ke desa sukapura dengan *model* sebesar 0%, *view* 31.9%, *controller* 0%, *database* 100%. Sehingga penggunaan ulang *code* pada *web* desa sundawenang ke *web* desa sukapura tidak dapat digunakan ulang *code* secara penuh.

**Kata kunci :** *Component-Based Software Engineering, Sistem Informasi, E-Government, Codeigniter, Laravel*

### Abstract

*Community Service is an activity intended to serve the community to facilitate and help the community. In the current reform era, the public was open to giving criticism to the government in public servants and also at this time village public services received much attention from the public. Research directly in the village district of Sukapura, the use of a manual administration system which is currently used as a relatively long service time, dependence on the ability of certain administrative personnel computers is large, nonstandard letter formats, poor administrative records. Application of information systems and concepts E-Government is a major consideration for public sector organizations (village governments) that carry out the process of preparing planning and monitoring and evaluating development outcomes. One of the existing information systems is the sundawenang village application which will be implemented in the web-based sukapura village using the CBSE (Component-Based Software Engineering) method. CBSE is an approach to software development and reuse of software components to replace development techniques from the start as used so far. Reusable code on the Sundanese village web to Sukapura village has a different framework, namely from the Laravel framework to CodeIgniter framework. The results of the analysis obtained in the form of an average value of percent for reuse of code on the Sundanese village web to Sukapura village with a model of 0%, view 31.9%, controller 0%, database 100%. So that the reuse of code on the Sundanese village web to the Sukapura village web code cannot be fully reused.*

**Keywords:** *Component-Based Software Engineering, Information System, E-Government, Codeigniter, Laravel*

## 1. Pendahuluan

### 1.1. Latar Belakang

Pelayanan publik merupakan salah satu tanggung jawab dari instansi pemerintah, baik dari pusat, di daerah, maupun di desa. Pada era reformasi saat ini masyarakat terbuka dalam memberikan kritikan pada pemerintah dalam pelayanan publik dan juga pada saat ini pelayanan publik desa banyak mendapat sorotan dari masyarakat. Pelayanan yang berkualitas seringkali mengalami kesulitan untuk dapat dicapai karena seringkali belum mengetahui dan memahami bagaimana cara memberikan pelayanan yang baik, hal ini disebabkan oleh masih rendahnya kemampuan profesional aparat desa. Berdasarkan wawancara dan pemantauan analisis secara langsung

di lapangan kelurahan Desa Sukapura, penggunaan sistem administrasi secara manual yang saat ini digunakan menjadi waktu pelayanan relatif lama, ketergantungan terhadap kemampuan komputer tenaga administratif tertentu cukup besar, format surat tidak standar, pencatatan administratif yang kurang baik. Penerapan sistem informasi dan konsep *E-Government* menjadi pertimbangan utama bagi organisasi sektor publik (pemerintah desa) yang melakukan proses penyusunan perencanaan dan pemantuan dan evaluasi hasil pembangunan. Teknologi Informasi menjadi bagian dari gugusan pembuatan layanan dari pemerintah dan semakin besar pengaruhnya pada organisasi, professional yang berkerja di dalamnya, serta hubungannya dengan publik. Semua rencana kebijakan untuk *E-Government* telah fokus pada isu-isu operasional internal dan dapat memperbaiki administrasi pelayanan publik Desa Sukapura.

Dalam pengembangan suatu *Software*, bagian dalam *code* yang sering mengalami perubahan adalah *user interface*. Beberapa permasalahan dapat muncul ketika suatu aplikasi yang sangat kompleks yang terdiri dari berbagai elemen mencakup data akses, bisnis *logic*, dan presentasi atau tampilan. Masalah tersebut terjadi karena ketergantungan antar elemen yang sangat tinggi, yang pada akhirnya berakibat pada sulitnya proses maintenance. Ketergantungan yang sangat tinggi (*high couple*) ini pula yang menyebabkan suatu elemen tidak dapat di-*Reusable* untuk aplikasi yang lain. Pemisahan antara tampilan (*user interface*) dengan bagian implementasi (*business logic*) dapat diimplementasikan dengan menggunakan arsitektur *Model View Controller* (MVC).[12]

Dalam tugas akhir ini akan memanfaatkan komponen layer *Model*, *View*, dan *Controller* sehingga dapat meningkatkan *flexibility* dan *Reusable*. *Model* merupakan aplikasi objek, *View* merupakan presentasinya, dan *Controller* mendefinisikan bagaimana *user interface* memberikan reaksi dari masukan *user*. Menerapkan solusi yang terjadi di kelurahan desa sukapura dengan *Reusable-code Software* di Desa Sundawenang [1], dengan memanfaatkan MVC *Pattern* dan menerapkan metode *Component Based Software Engineering* (CBSE) ini dapat memaparkan unsur-unsur rancangan pada sistem karena untuk menguji *Pattern* tersebut dengan *framework* yang berbeda yaitu dari *framework* laravel ke *framework codeigniter*. *Component Based Software Engineering* (CBSE) merupakan pendekatan rekayasa perangkat lunak dengan penggabungan komponen. Yang ditawarkan oleh CBSE adalah dalam membangun sebuah aplikasi, penulis tidak perlu membangun semua kebutuhan aplikasi tersebut dari awal, melainkan hanya tinggal menggabungkan bagian-bagian yang tersedia menjadi satu aplikasi yang diinginkan. Semakin terujinya suatu unsur rancangan berarti semakin matangnya unsur tersebut sehingga dapat dikatakan *best practices*, tetapi lebih dari pada itu *Pattern* membuat *Design Software* menjadi lebih efektif, fleksible, resilient dan waktu penyelesaian desain *Software* juga lebih efisien.

Tugas akhir ini bertujuan untuk menganalisis *Reusable code* pada *framework* yang berbeda dari *framework* laravel ke *framework codeigniter* dengan menerapkan MVC *Pattern* menggunakan metode CBSE untuk memaparkan komponen-komponen dalam persentase *Reusable code*.

## 1.2. Topik dan Batasannya

Pada tugas akhir ini telah dilakukan analisis dan implementasi pembuatan *website* dengan metode CBSE (*Component Based Software Engineering*). Metode ini dipilih karena memiliki kemampuan dalam meningkatkan *reusability* yang efektif dalam pengembangan perangkat lunak berbasis komponen. Pengembangan konsep ini prinsip sistem sebagai hasil komponen-komponen dalam MVC seperti *model*, *view*, dan *controller*. Komponen sebagai entitas-entitas *Reusable* dan upgrading sistem dengan memodifikasi atau mengganti komponennya. Proses CBSE yaitu identifikasi kebutuhan, identifikasi komponen, modifikasi komponen sesuai kebutuhan jika diperlukan, desain arsitektur, identifikasi komponen lagi, dan menggunakan komponen untuk membuat sistem. *Reusable code* pada Web Desa Sundawenang [1] ke Desa Sukapura memiliki *Framework* yang berbeda yaitu dari *Framework* Laravel ke *Framework Codeigniter*. Dengan memanfaatkan kedua *framework* yang berbeda akan dihitung persentase penggunaan ulang *code* pada Web Desa Sundawenang [1] ke Web Desa Sukapura kemudian dikembangkan kembangkan kembali dengan beberapa perubahan *interface* dan fungsi pada *website* baru. Berdasarkan topik yang diangkat, terdapat beberapa rumusan masalah yang diajukan, yaitu :  
Dari permasalahan yang diperoleh, adapun batasan yang dilakukan dalam permasalahan tersebut yaitu sebagai berikut :

1. Aplikasi baru ini melayani surat sebanyak surat diaplikasi lama sebagai berikut : Akta Kelahiran, Surat Kematian, Tempat Tinggal , Perbedaan Identitas, Surat Keterangan Usaha, SKCK, Domisili Paud, Domisili Yayasan, Izin Keramaian, Survai Lapangan, Ket. Status Pernikahan, Ket. Pengakuan Cerai, Domisili Koperasi, Domisili Perusahaan, Domisili Usaha, Ket. Tidak Memiliki Rumah, Ket. Tidak Mampu, Ket. Penghasilan, Ket. Sudah menikah, Ket. Ahli Waris
2. Aplikasi Desa Sundawenang menggunakan Framework Laravel lalu aplikasi Desa Sukapura menggunakan Framework CI (Codeigniter)

## 1.3. Tujuan

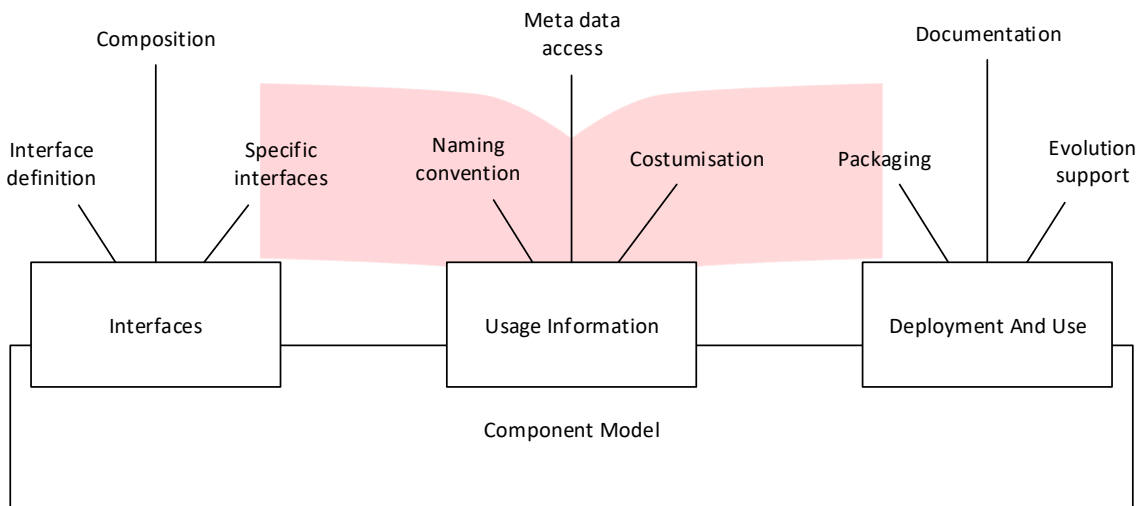
Berdasarkan latar belakang dan rumusan masalah yang telah diuraikan, maka tujuan dari adanya penelitian ini adalah :

1. Menerapkan CBSE (Komponen MVC) terhadap aplikasi Desa Sundawenang ke Desa Sukapura
2. Menguji fungsionalitas aplikasi baru sampai dengan User Acceptent Test
3. Menghitung persentase MVC aplikasi lama dapat dipakai kembali di aplikasi baru

## 2. Studi Terkait

### 2.1. Component Based Software Engineering

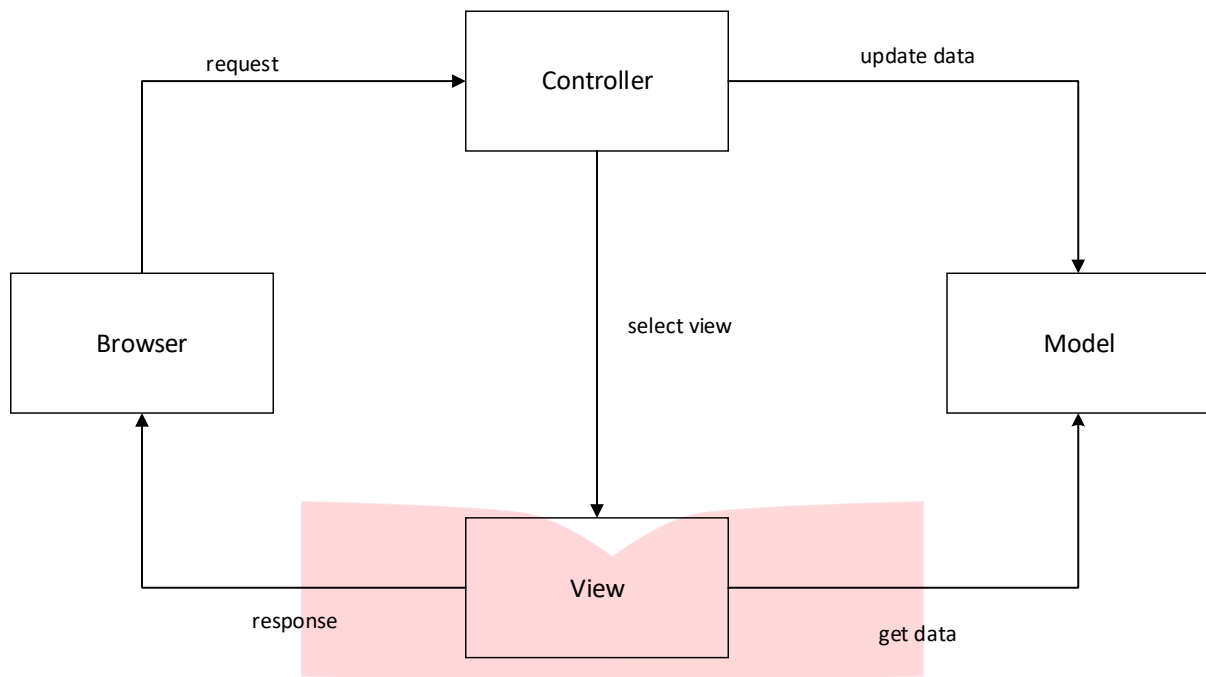
*Component Based Software Engineering* adalah sebuah pendekatan untuk pengembangan perangkat lunak dan penggunaan kembali komponen-komponen perangkat lunak untuk menggantikan teknik pengembangan dari awal seperti yang dipakai selama ini. CBSE ini salah satu paradigma pemrograman yang bertujuan untuk menyempurnakan teknik-teknik pengembangan perangkat lunak yang telah ada.[15]



**Gambar 2.1 Basic Element Of Component Model**

### 2.2. Design Pattern

*Design Pattern* sangat populer di kalangan pengembang dalam rekayasa perangkat lunak dan *Object Oriented Programming* yang merupakan sebuah solusi terhadap masalah pemrograman umum yang sering dihadapi serta berulang kali.[8] Salah satu *Design Pattern* yang sering di bicarakan adalah *Model View Controller (MVC) Pattern*. MVC adalah sebuah konsep yang diperkenalkan oleh penemu Smalltalk (Trygve Reenskaug) untuk meng-  
enkapsulasi data bersama dengan pemrosesan (model), mengisolasi dari proses manipulasi (controller) dan tampilan (view) untuk direpresentasikan pada sebuah user interface Definisi teknis dari arsitektur MVC dibagi menjadi tiga lapisan [7]



Gambar 2.2 MVC Architecture Pattern[7]

### 1. Model

Mewakili suatu objek atau JAVA yang membawa data. Itu juga dapat memiliki logika untuk memperbarui controller jika datanya berubah [8] namun tidak dapat berhubungan langsung dengan bagian *view*.

### 2. View

Merupakan bagian yang menangani *presentation logic*. Pada suatu aplikasi *web* bagian ini biasanya berupa *file template* HTML, yang diatur oleh *controller*. *View* berfungsi untuk menerima dan merepresentasikan data kepada *user*. Visualisasi data yang berisi model bagian ini tidak memiliki akses langsung terhadap bagian *model* [8]

### 3. Controller

Bertindak pada Model dan tampilan. Ini mengontrol aliran data ke objek model dan memperbarui tampilan setiap kali data berubah. Itu membuat *View* dan *Model* terpisah [8] dan menerima request data dari *user* kemudian menentukan apa yang akan diproses oleh aplikasi.

## 2.3. PHP (HyperText Preprocessor)

PHP (*HyperText Preprocessor*) adalah suatu server-side scripting yang menyatu dengan HTML (*Hyper Text Markup Language*) untuk membuat halaman website yang dinamis. Server-side scripting sendiri berarti suatu pemrograman yang pengeksekusiannya berada di sisi server. Seluruh proses di dalam bahasa pemrograman PHP dilakukan di sebuah server. Proses dilakukan di server tetapi hasil akan ditampilkan di browser. [16]

## 2.5. Unified Modelling Language (UML)

UML adalah kumpulan standar *model* konsep dan notasi yang dikembangkan secara khusus untuk pengembangan objek-orientasi. [14] Ada beberapa jenis *diagram* dalam UML yang dapat digunakan, yaitu:

### 2.5.1. Activity Diagram

*Activity Diagram* merupakan sebuah tipe dari *diagram* alur kerja (*workflow*) yang menggambarkan tentang aktivitas dari pengguna ketika melakukan setiap kegiatan dan aliran sekuensial.

Berikut ini adalah table dari simbol-simbol notasi yang digunakan pada *activity diagram* :

Tabel 2.1 Simbol notasi pada *activity diagram*

No	Simbol	Keterangan
1	<i>Starting Activity</i>	Menunjukkan awal dari suatu proses.
2	<i>Activity</i>	Menunjukkan langkah-langkah aktivitas sistem yang terjadi.

3	<i>Transition Arrow</i>	Untuk menunjuk ke aksi atau proses yang lain. Dilengkapi dengan katakata jika berasal dari <i>decision</i> .
4	<i>Decition Activity</i>	Memiliki sebuah <i>flow</i> masuk dan dua atau lebih <i>flow</i> keluar. Menunjukkan aktivitas yang dapat dipilih.
5	<i>Split</i>	Menotasikan permulaan aksi atau proses paralel yang dapat terjadi dalam suatu urutan atau terjadi secara bersamaan.
6	<i>Join</i>	Semua aksi yang masuk ke <i>join</i> harus telah diselesaikan sebelum proses berlanjut
7	<i>Ending Activity</i>	Menunjukkan akhir dari proses.

### 2.5.2. Class Diagram

Salah satu jenis *class diagram UML* menunjukkan hal-hal dalam pekerjaan *domain user* disebut sebagai *domain model class diagram*. Tipe lain dari notasi *class diagram UML* digunakan untuk membuat desain *class diagram* ketika merancang *class* perangkat lunak. Di *class diagram*, persegi panjang mewakili kelas, dan garis yang menghubungkan persegi panjang menunjukkan asosiasi antara kelas. Dalam persegi panjang (kotak) terbagi dua, bagian atas berisi nama kelas, dan bagian bawah merupakan atribut kelas. Nama kelas selalu diawali dengan huruf kapital, dan atribut nama selalu diawali dengan huruf kecil. *Diagram class* digambarkan dengan menampilkan kelas dan asosiasi antara kelas.

Hubungan di dalam *class diagram* ada tiga, yaitu sebagai berikut:

1. *Aggregation*  
Merupakan hubungan antara objek dengan bagian-bagiannya dimana bagian-bagian tersebut dapat muncul secara terpisah.
2. *Association*  
Merupakan *class* yang merepresentasikan *many-to-many relationship* antara dua *class* lainnya.
3. *Generalization*  
Merupakan suatu super *class* yang menjelaskan *properties* umum kepada kelas-kelas khusus yang disebut dengan *subclass*

### 2.5.3. Sequence Diagram

*System Sequence Diagram* merupakan sebuah *diagram* yang menunjukkan urutan pesan antara aktor eksternal dan sistem selama *use case* atau skenario”

Di dalam *sequence diagram* adalah sebagai berikut :

1. *Actor*  
Merupakan pengguna yang berinteraksi secara langsung dengan sistem.
2. *Input message*  
Merupakan pesan *input* dari pengguna ke dalam suatu sistem.
3. *Returned value*  
Merupakan pesan *output* dari suatu sistem ke pengguna, hasil dari pemrosesan *input*.
4. *Object*  
Merupakan objek-objek yang berinteraksi di dalam *sequence diagram*.
5. *Object lifeline*  
Menggambarkan urutan pesan dari atas ke bawah.

### 2.5.4. Use Case Diagram

*System Sequence Diagram* merupakan sebuah *diagram* yang menunjukkan urutan pesan antara aktor eksternal dan sistem selama *use case* atau skenario

Di dalam *sequence diagram* adalah sebagai berikut:

1. *Actor*  
Merupakan pengguna yang berinteraksi secara langsung dengan sistem.
2. *Input message*

Merupakan pesan *input* dari pengguna ke dalam suatu sistem.

3. *Returned value*

Merupakan pesan *output* dari suatu sistem ke pengguna, hasil dari pemrosesan *input*.

4. *Object*

Merupakan objek-objek yang berinteraksi di dalam *sequence diagram*.

5. *Object lifeline*

Menggambarkan urutan pesan dari atas ke bawah.

## 2.4. Framework Laravel

Laravel merupakan framework PHP yang menekankan pada kesederhanaan dan fleksibilitas pada desainnya. Laravel dirilis dibawah lisensi MIT dengan sumber kode yang disediakan di Github. Sama seperti framework PHP lainnya, Laravel dibangun dengan basis MVC (ModelView-Controller). Laravel dilengkapi command line tool yang bernama "Artisan" yang bisa digunakan untuk packaging bundle dan instalasi bundle. Menurut survey yang dilakukan oleh Sitepoint.com pada Desember 2013 dalam popularitas framework PHP, Laravel menduduki urutan teratas.[16]

## 2.6. Framework CI (Codeigniter)

CodeIgniter adalah sebuah web *Application* framework yang digunakan untuk membangun aplikasi PHP dinamis yang dibangun menggunakan konsep Model View Controller development pattern. CodeIgniter menyediakan berbagai macam library yang dapat mempermudah dalam pengembangan dan termasuk framework tercepat dibandingkan dengan framework lainnya.[16]

## 2.7. Xampp

XAMPP adalah software web server apache yang di dalamnya tertanam server MySQL yang didukung dengan bahasa pemrograman PHP untuk membuat website yang dinamis. XAMPP sendiri mendukung dua system operasi yaitu windows dan Linux. Untuk linux dalam proses penginstalannya menggunakan command line sedangkan untuk windows dalam proses penginstalannya menggunakan interface grafis sehingga lebih mudah dalam penggunaan XAMPP di Windows di banding dengan Linux. Didalam XAMPP ada 3 komponen utama yang di tanam di dalamnya yaitu web server Apache, PHP, dan MySQL.

## 2.7. MySQL

Database MySQL merupakan suatu perangkat lunak *database* yang berbentuk *database relational* atau disebut *Relational Database Management System* ( RDBMS ) yang menggunakan suatu bahasa permintaan yang bernama SQL (*Structured Query Language*). SQL dibagi menjadi tiga bentuk *Query*, yaitu :

### 1. DDL (Data Definition Language)

DDL adalah sebuah metode *Query* SQL yang berguna untuk mendefinisikan data pada sebuah *Database*, *Query* yang dimiliki DDL adalah :

*Create* : Digunakan untuk membuat *Database* dan Tabel

*Drop* : Digunakan untuk menghapus Tabel dan *Database*

*Alter* : Digunakan untuk melakukan perubahan struktur tabel yang telah dibuat, baik menambah Field ( Add ), mengganti nama Field ( Change ) ataupun menamakannya kembali ( Rename ), dan menghapus Field ( Drop ).

### 2. DML (Data Manipulation Language)

DML adalah sebuah metode *Query* yang dapat digunakan apabila DDL telah terjadi, sehingga fungsi dari *Query* DML ini untuk melakukan pemanipulasian *database* yang telah dibuat. *Query* yang dimiliki DML adalah :

*INSERT* : Digunakan untuk memasukkan data pada Tabel *Database*

*UPDATE* : Digunakan untuk perubahan terhadap data yang ada pada Tabel *Database*

*DELETE* : Digunakan untuk Penhapusan data pada tabel *Database*

### 3. DCL (Data Control Language)

DCL adalah sebuah metode *Query SQL* yang digunakan untuk memberikan hak otorisasi mengakses *Database*, mengalokasikan *space*, pendefinisian *space*, dan pengauditan penggunaan *database*. *Query* yang dimiliki DCL adalah :

*GRANT* : Untuk mengizinkan *User* mengakses Tabel dalam *Database*.  
*REVOKE* : Untuk membatalkan izin hak *user*, yang ditetapkan oleh perintah *GRANT*  
*COMMIT* : Menetapkan penyimpanan *Database*  
*ROLLBACK* : Membatalkan penyimpanan *Database*

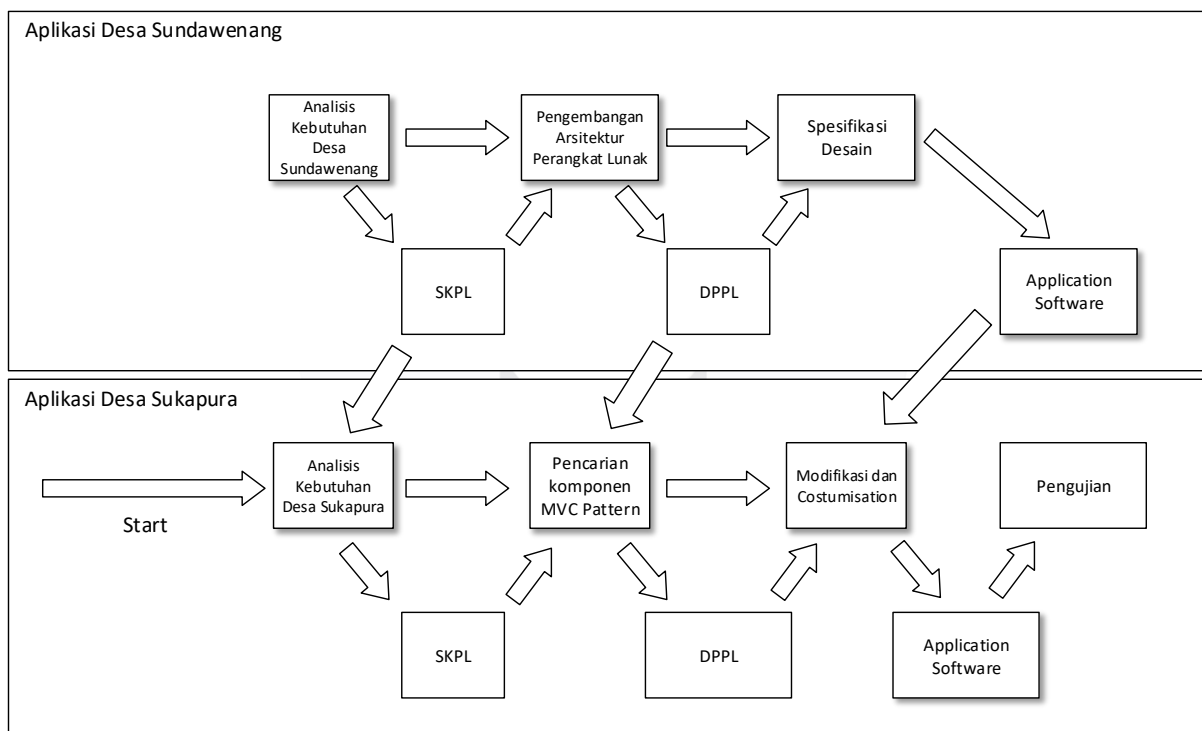
### 2.8. Black-Box Testing

Black Box Testing berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi input dan melakukan penyetelan pada spesifikasi fungsional program. Black Box Testing bukanlah solusi alternatif dari White Box Testing tapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh White Box Testing. Black Box Testing cenderung untuk menemukan hal-hal berikut:[m.Sidi]

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (interface errors).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (performance errors).
5. Kesalahan inisialisasi dan terminasi.

### 3. Sistem yang Dibangun

Berikut adalah penerapan metodologi *Component Based Development*. *Component Based Development* adalah alur pengembangan sebuah sistem dengan menggabungkan komponen-komponen yang telah dikembangkan sebelumnya untuk memfasilitasi pemakaian komponen dalam pembuatan dan perubahan sistem. Tahapan-tahapan penelitian adalah sebagai berikut:



**Gambar 3.1 Component Based Development**

#### 3.1. Analisis Kebutuhan Desa Sundawenang

Pada tahap analisis kebutuhan ini, pengumpulan data menggunakan teknik wawancara kepada pihak Kelurahan Desa Sundawenang dan melakukan wawancara untuk mendapatkan informasi dan kebutuhan sistem seperti *interface* aplikasi dengan mengajukan pertanyaan secara langsung. Penulis menggunakan teknik wawancara semi terstruktur untuk mengetahui dan meneliti gambaran umum Kelurahan Desa Sundawenang dan prosedur-prosedur yang berjalan pada Kelurahan Desa Sundawenang.

#### 3.2. Pengembangan Arsitektur Perangkat Lunak

Pada tahap pengembangan arsitektur perangkat lunak suatu yang bertujuan untuk mengembangkan sistem dan memberikan panduan yang bertujuan untuk menyukseskan proyek pengembangan sistem melalui tahap demi tahap.

### 3.3. Spesifikasi Desain

Menggambarkan bagaimana sistem melakukan instruksi untuk menguji persyaratan tertentu, pengaturan konfigurasi, atau fungsi. Menjelaskan tindakan teknik untuk memastikan keamanan sistem lalu *output* yang dihasilkan oleh sistem.

### 3.4. Application Software

Merupakan aplikasi *web* yang dapat digunakan sesuai dengan sistem yang ada. *Software* ini merupakan penghubung antara aplikasi dan *hardware*.

### 3.5. Analisis Kebutuhan Desa Sukapura

Pada tahap analisis kebutuhan ini, pengumpulan data menggunakan teknik wawancara kepada pihak Kelurahan Desa Sukapura dan melakukan wawancara untuk mendapatkan informasi dan kebutuhan sistem seperti *interface* aplikasi dengan mengajukan pertanyaan secara langsung. Penulis menggunakan teknik wawancara semi terstruktur untuk mengetahui dan meneliti gambaran umum Kelurahan Desa Sukapura dan prosedur-prosedur yang berjalan pada Kelurahan Desa Sukapura

### 3.6. Pencarian Komponen MVC Pattern

Pada pencarian komponen MVC ini akan membuat data *model* untuk merepresentasikan informasi dari *database*, *view* untuk menampilkan data, dan *controller* yang akan menggabungkan keduanya bersama-sama dan menangani tugas lain.

### 3.7. Modifikasi dan Customization

Pada tahap ini adalah modifikasi dan customization yang bisa digunakan kembali dengan upgrading sistem dengan memodifikasi atau mengganti komponennya. Proses ini memungkinkan suatu sistem dapat dilakukan *Generalisasi* sehingga pada saat pengembangan *Software* yang sama hanya membutuhkan sedikit modifikasi.

### 3.8. Application Software

Merupakan aplikasi *web* yang dapat digunakan sesuai dengan sistem yang ada. *Software* ini merupakan penghubung antara aplikasi dan *hardware*.

### 3.9. Pengujian

Pengujian ini dilakukan untuk mengetahui adanya kesesuaian antara fungsi-fungsi atau layanan-layanan sistem yang diimplementasikan dengan hasil analisa kebutuhan yang sudah ditentukan pada tahap analisa kebutuhan fungsional maupun non-fungsional. Pengujian juga dilakukan untuk mengetahui apakah sistem tersebut dapat diakses menggunakan beberapa *Software* internet browser. Selain itu dengan dilakukannya pengujian akan dapat diketahui adanya kesalahan-kesalahan dalam proses coding maupun dalam menghasilkan *output* program sehingga kesalahan-kesalahan tersebut dapat diperbaiki. Metode pengujian yang dilakukan adalah white box dan black box.

## 4. Evaluasi

Evaluasi MVC *Pattern* ini dilakukan sebagai bahan pertimbangan *Reusable code* apakah *Reusable code* pada *Web* Desa Sundawenang dapat digunakan ulang semua *code* atau merubah *code* yang sudah ada dengan menambahkan fungsi dalam implementasi yang berbeda. Penilaian *Reusable code* dilakukan dengan mengambil data jumlah baris *code* pada *Web* Desa Sundawenang dan jumlah baris *code* yang digunakan ulang pada *Web* Desa Sukapura, tahapan-tahapan evaluasi MVC *Pattern* sebagai berikut :

### 4.1. Model

#### 4.1.1. User

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
	User	Model_user
Nama Fungsi	<ul style="list-style-type: none"> <li>• Rules</li> <li>• RulesUpdate</li> </ul>	<ul style="list-style-type: none"> <li>• add</li> <li>• getData</li> <li>• edit</li> </ul>



		<ul style="list-style-type: none"> <li>• <i>delete</i></li> </ul>
<b>Peran</b>	<i>Model</i>	<i>Model</i>
<b>Keterangan</b>	<p>Banyak Modifikasi, yaitu melakukan perubahan dengan menambah fungsi yang ada dalam aplikasi sebelumnya dan berbeda dalam implementasi <i>code</i>.</p> <p>Jumlah baris <i>code</i> : 50</p> <p>Jumlah baris <i>code</i> yang di gunakan ulang : 0</p> $\% \text{ reused} = \frac{0}{50}$ <p>Persentase baris <i>code</i> yang digunakan : 0 %</p>	

#### 4.1.2. Aktivitas

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
	-	
<b>Nama Fungsi</b>	-	<ul style="list-style-type: none"> <li>• <i>add</i></li> <li>• <i>getData</i></li> <li>• <i>edit</i></li> <li>• <i>delete</i></li> </ul>
<b>Peran</b>	<i>Model</i>	<i>Model</i>
<b>Keterangan</b>	<p>Banyak Modifikasi, yaitu membuat <i>model log</i> karena diaplikasi desa sundawenang tidak dibuat <i>model</i> untuk aktivitas</p>	

#### 4.1.3. Pajak

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
	Pajak	
<b>Nama Fungsi</b>	<ul style="list-style-type: none"> <li>• <i>Rules</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>add</i></li> <li>• <i>getData</i></li> <li>• <i>edit</i></li> <li>• <i>delete</i></li> </ul>
<b>Peran</b>	<i>Model</i>	<i>Model</i>
<b>Keterangan</b>	<p>Banyak Modifikasi, yaitu melakukan perubahan dengan menambah fungsi yang ada dalam aplikasi sebelumnya dan berbeda dalam implementasi <i>code</i></p> <p>Jumlah baris <i>code</i> : 19</p> <p>Jumlah baris <i>code</i> yang di gunakan ulang : 0</p> $\% \text{ reused} = \frac{0}{19}$ <p>Persentase baris <i>code</i> yang digunakan : 0 %</p>	

#### 4.1.4. Lowongan Pekerjaan

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
	Job	
<b>Nama Fungsi</b>	<ul style="list-style-type: none"> <li>• <i>Rules</i></li> </ul>	<ul style="list-style-type: none"> <li>• <i>add</i></li> <li>• <i>getData</i></li> <li>• <i>edit</i></li> <li>• <i>delete</i></li> </ul>
<b>Peran</b>	<i>Model</i>	<i>Model</i>

<b>Keterangan</b>	Banyak Modifikasi, yaitu melakukan perubahan dengan menambah fungsi yang ada dalam aplikasi sebelumnya dan berbeda dalam implementasi <i>code</i> Jumlah baris <i>code</i> : 20 Jumlah baris <i>code</i> yang di gunakan ulang : 0 $\% reused = \frac{0}{20}$ Persentase baris <i>code</i> yang digunakan : 0 %
-------------------	---

#### 4.1.5. Produk

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
	Product	
<b>Nama Fungsi</b>	<ul style="list-style-type: none"> <li>• <i>Rules</i></li> <li>• photo</li> </ul>	<ul style="list-style-type: none"> <li>• add</li> <li>• getData</li> <li>• edit</li> <li>• delete</li> </ul>
<b>Peran</b>	<i>Model</i>	<i>Model</i>
<b>Keterangan</b>	Banyak Modifikasi, yaitu melakukan perubahan dengan menambah fungsi yang ada dalam aplikasi sebelumnya dan berbeda dalam implementasi <i>code</i> Jumlah baris <i>code</i> : 24 Jumlah baris <i>code</i> yang di gunakan ulang : 0 $\% reused = \frac{0}{24}$ Persentase baris <i>code</i> yang digunakan : 0 %	

#### 4.1.6. Aspirasi

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
	Aspiration	
<b>Nama Fungsi</b>	<ul style="list-style-type: none"> <li>• <i>Rules</i></li> </ul>	<ul style="list-style-type: none"> <li>• add</li> <li>• getData</li> <li>• edit</li> <li>• delete</li> </ul>
<b>Peran</b>	<i>Model</i>	<i>Model</i>
<b>Keterangan</b>	Banyak Modifikasi, yaitu melakukan perubahan dengan menambah fungsi yang ada dalam aplikasi sebelumnya dan berbeda dalam implementasi <i>code</i> Jumlah baris <i>code</i> : 22 Jumlah baris <i>code</i> yang di gunakan ulang : 0 $\% reused = \frac{0}{22}$ Persentase baris <i>code</i> yang digunakan : 0 %	

#### 4.1.7. Pengaduan

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
	Report	
<b>Nama Fungsi</b>	<ul style="list-style-type: none"> <li>• <i>Rules</i></li> </ul>	<ul style="list-style-type: none"> <li>• add</li> <li>• getData</li> <li>• edit</li> <li>• delete</li> </ul>
<b>Peran</b>	<i>Model</i>	<i>Model</i>

<b>Keterangan</b>	<p>Banyak Modifikasi, yaitu melakukan perubahan dengan menambah fungsi yang ada dalam aplikasi sebelumnya dan berbeda dalam implementasi <i>code</i></p> <p>Jumlah baris <i>code</i> : 17</p> <p>Jumlah baris <i>code</i> yang di gunakan ulang : 0</p> <p><math>\% reused = \frac{0}{17}</math></p> <p>Persentase baris <i>code</i> yang digunakan : 0 %</p>
-------------------	---

#### 4.1.8. Layanan

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
		<ul style="list-style-type: none"> <li>• Service</li> </ul>
<b>Nama Fungsi</b>	<ul style="list-style-type: none"> <li>• <i>Rules</i></li> <li>• AhliWaris</li> <li>• DomisiliPaud</li> <li>• DomisiliYayasan</li> <li>• IzinRameRame</li> <li>• Kelahiran</li> <li>• Kematian</li> <li>• KetAhliWaris</li> <li>• KetDomisiliKoperasi</li> <li>• KetDomisiliPerusahaan</li> <li>• KetDomisiliUsaha</li> <li>• KetPengakuanCerai</li> <li>• KetPenghasilan</li> <li>• KetStatusPernikahan</li> <li>• KetSudahMenikah</li> <li>• KetTidakMampu</li> <li>• KetTidakMemilikiRumah</li> <li>• PerbedaanIdentitas</li> <li>• Service</li> <li>• SurvaiLapangan</li> <li>• TempatTinggal</li> </ul>	<ul style="list-style-type: none"> <li>• getDictionary</li> <li>• tambah</li> <li>• edit</li> <li>• get</li> <li>• getDetail</li> <li>• delete</li> </ul>
<b>Peran</b>	<i>Model</i>	<i>Model</i>
<b>Keterangan</b>	<p>Banyak Modifikasi, yaitu melakukan perubahan dengan menambah fungsi yang ada dalam aplikasi sebelumnya dan berbeda dalam implementasi <i>code</i></p> <p>Jumlah baris <i>code</i> : 86</p> <p>Jumlah baris <i>code</i> yang di gunakan ulang : 0</p> <p><math>\% reused = \frac{0}{86}</math></p> <p>Persentase baris <i>code</i> yang digunakan : 0 %</p>	

#### 4.1.9. Pengaturan

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
		Pengaturan

<b>Nama Fungsi</b>	-	<ul style="list-style-type: none"> <li>• add</li> <li>• getData</li> <li>• edit</li> <li>• delete</li> </ul>
<b>Peran</b>	<i>Model</i>	<i>Model</i>
<b>Keterangan</b>	<p>Banyak Modifikasi, yaitu melakukan perubahan dengan menambah fungsi yang ada dalam aplikasi sebelumnya dan berbeda dalam implementasi <i>code</i></p> <p>Jumlah baris <i>code</i> : 13</p> <p>Jumlah baris <i>code</i> yang di gunakan ulang : 0</p> <p><math>\% reused = \frac{0}{13}</math></p> <p>Persentase baris <i>code</i> yang digunakan : 0 %</p>	

## 4.2. View

### 4.2.1. Login

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
		login.blade
<b>Peran</b>	<i>View</i>	<i>View</i>
<b>Keterangan</b>	<p>Banyak Modifikasi, yaitu melakukan perubahan dalam UI/UX pada aplikasi sebelumnya</p> <p>Jumlah baris <i>code</i> : 57</p> <p>Jumlah baris <i>code</i> yang di gunakan ulang : 0</p> <p><math>\% reused = \frac{0}{57}</math></p> <p>Persentase baris <i>code</i> yang digunakan : 0 %</p>	

### 4.2.2. Sidebar

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
		main.blade
<b>Peran</b>	<i>View</i>	<i>View</i>
<b>Keterangan</b>	<p>Banyak Modifikasi, yaitu melakukan perubahan dalam UI/UX pada aplikasi sebelumnya</p> <p>Jumlah baris <i>code</i> : 97</p> <p>Jumlah baris <i>code</i> yang di gunakan ulang : 39</p> <p><math>\% reused = \frac{39}{97}</math></p> <p>Persentase baris <i>code</i> yang digunakan : 40,2 %</p>	

### 4.2.3. Home

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
		Home.blade
<b>Peran</b>	<i>View</i>	<i>View</i>
<b>Keterangan</b>	<p>Banyak Modifikasi, yaitu melakukan perubahan dalam UI/UX pada aplikasi sebelumnya</p> <p>Jumlah baris <i>code</i> : 25</p> <p>Jumlah baris <i>code</i> yang di gunakan ulang : 0</p> <p><math>\% reused = \frac{0}{25}</math></p> <p>Persentase baris <i>code</i> yang digunakan : 0 %</p>	

### 4.2.4. User

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
		users.blade
<b>Peran</b>	<i>View</i>	<i>View</i>

<b>Keterangan</b>	Banyak Modifikasi, yaitu melakukan perubahan dalam UI/UX pada aplikasi sebelumnya Jumlah baris <i>code</i> : 53 Jumlah baris <i>code</i> yang di gunakan ulang : 27 $\% reused = \frac{27}{53}$ Persentase baris <i>code</i> yang digunakan : 50,9 %
-------------------	--

## 4.2.5. Aktivitas

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
	Activity_log.blade	log
<b>Peran</b>	View	View
<b>Keterangan</b>	Banyak Modifikasi, yaitu melakukan perubahan dalam UI/UX pada aplikasi sebelumnya Jumlah baris <i>code</i> : 51 Jumlah baris <i>code</i> yang di gunakan ulang : 10 $\% reused = \frac{10}{51}$ Persentase baris <i>code</i> yang digunakan : 19,6 %	

## 4.2.6. Pajak

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
	pajak.blade	Pajak
<b>Peran</b>	View	View
<b>Keterangan</b>	Banyak Modifikasi, yaitu melakukan perubahan dalam UI/UX pada aplikasi sebelumnya Jumlah baris <i>code</i> : 54 Jumlah baris <i>code</i> yang di gunakan ulang : 27 $\% reused = \frac{27}{54}$ Persentase baris <i>code</i> yang digunakan : 50,9 %	

## 4.2.7. Lowongan Pekerjaan

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
	job.blade	LowonganPekerjaan
<b>Peran</b>	View	View
<b>Keterangan</b>	Banyak Modifikasi, yaitu melakukan perubahan dalam UI/UX pada aplikasi sebelumnya Jumlah baris <i>code</i> : 86 Jumlah baris <i>code</i> yang di gunakan ulang : 5 $\% reused = \frac{5}{42}$ Persentase baris <i>code</i> yang digunakan : 11,9 %	

## 4.2.8. Produk

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
	product.blade	Produk
<b>Peran</b>	View	View
<b>Keterangan</b>	Banyak Modifikasi, yaitu melakukan perubahan dalam UI/UX pada aplikasi sebelumnya Jumlah baris <i>code</i> : 43 Jumlah baris <i>code</i> yang di gunakan ulang : 5 $\% reused = \frac{5}{43}$ Persentase baris <i>code</i> yang digunakan : 11,6 %	

## 4.2.9. Aspirasi

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
	aspiration.blade	Aspirasi

<b>Peran</b>	<i>View</i>	<i>View</i>
<b>Keterangan</b>	Banyak Modifikasi, yaitu melakukan perubahan dalam UI/UX pada aplikasi sebelumnya Jumlah baris <i>code</i> : 51 Jumlah baris <i>code</i> yang di gunakan ulang : 12 $\% reused = \frac{12}{51}$ Persentase baris <i>code</i> yang digunakan : 23,5 %	

#### 4.2.10. Pengaduan

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
		report.blade
<b>Peran</b>	<i>View</i>	<i>View</i>
<b>Keterangan</b>	Banyak Modifikasi, yaitu melakukan perubahan dalam UI/UX pada aplikasi sebelumnya Jumlah baris <i>code</i> : 34 Jumlah baris <i>code</i> yang di gunakan ulang : 17 $\% reused = \frac{17}{34}$ Persentase baris <i>code</i> yang digunakan : 50 %	

#### 4.2.11. Layanan

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
		service.blade
<b>Peran</b>	<i>View</i>	<i>View</i>
<b>Keterangan</b>	Banyak Modifikasi, yaitu melakukan perubahan dalam UI/UX pada aplikasi sebelumnya Jumlah baris <i>code</i> : 198 Jumlah baris <i>code</i> yang di gunakan ulang : 77 $\% reused = \frac{77}{198}$ Persentase baris <i>code</i> yang digunakan : 38,8 %	

#### 4.2.12. Pengaturan

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
		pengaturan.blade
<b>Peran</b>	<i>View</i>	<i>View</i>
<b>Keterangan</b>	Banyak Modifikasi, yaitu melakukan perubahan dalam UI/UX pada aplikasi sebelumnya Jumlah baris <i>code</i> : 22 Jumlah baris <i>code</i> yang di gunakan ulang : 19 $\% reused = \frac{19}{22}$ Persentase baris <i>code</i> yang digunakan : 86.3 %	

### 4.3. Controller

#### 4.3.1. User

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
		<i>UsersController</i>
<b>Nama Fungsi</b>	<ul style="list-style-type: none"> <li>• index</li> <li>• create</li> <li>• store</li> <li>• show</li> <li>• update</li> <li>• delete</li> </ul>	<ul style="list-style-type: none"> <li>• user</li> <li>• addUser</li> <li>• editUserSimpan</li> <li>• edituser</li> <li>• hapusUser</li> </ul>

<b>Peran</b>	<i>Controller</i>	<i>Controller</i>
<b>Keterangan</b>	<p>Sedikit Modifikasi, yaitu melakukan perubahan terhadap fungsi yang ada dalam aplikasi sebelumnya tetapi tidak merubah tujuan dari fungsi tersebut hanya cara implementasi <i>codenya</i> saja yang berbeda.</p> <p>Jumlah baris <i>code</i> : 82</p> <p>Jumlah baris <i>code</i> yang di gunakan ulang : 0</p> <p><math>\% reused = \frac{0}{82}</math></p> <p>Persentase baris <i>code</i> yang digunakan : 0 %</p>	

#### 4.3.2. Aktivitas

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
		<i>ActivityLogController</i>
<b>Nama Fungsi</b>	<ul style="list-style-type: none"> <li>• index</li> </ul>	<ul style="list-style-type: none"> <li>• log</li> </ul>
<b>Peran</b>	<i>Controller</i>	<i>Controller</i>
<b>Keterangan</b>	<p>Sedikit Modifikasi, yaitu melakukan perubahan terhadap fungsi yang ada dalam aplikasi sebelumnya tetapi tidak merubah tujuan dari fungsi tersebut hanya cara implementasi <i>codenya</i> saja yang berbeda</p> <p>Jumlah baris <i>code</i> : 15</p> <p>Jumlah baris <i>code</i> yang di gunakan ulang : 0</p> <p><math>\% reused = \frac{0}{15}</math></p> <p>Persentase baris <i>code</i> yang digunakan : 0 %</p>	

#### 4.3.3. Pajak

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
		<i>PajakController</i>
<b>Nama Fungsi</b>	<ul style="list-style-type: none"> <li>• index</li> <li>• create</li> <li>• store</li> <li>• show</li> <li>• <i>update</i></li> <li>• <i>delete</i></li> </ul>	<ul style="list-style-type: none"> <li>• pajak</li> <li>• addPajak</li> <li>• editPajakSimpan</li> <li>• editPajak</li> <li>• hapusPajak</li> </ul>
<b>Peran</b>	<i>Controller</i>	<i>Controller</i>
<b>Keterangan</b>	<p>Sedikit Modifikasi, yaitu melakukan perubahan terhadap fungsi yang ada dalam aplikasi sebelumnya tetapi tidak merubah tujuan dari fungsi tersebut hanya cara implementasi <i>codenya</i> saja yang berbeda</p> <p>Jumlah baris <i>code</i> : 89</p> <p>Jumlah baris <i>code</i> yang di gunakan ulang : 0</p> <p><math>\% reused = \frac{0}{89}</math></p> <p>Persentase baris <i>code</i> yang digunakan : 0 %</p>	

#### 4.3.4. Lowongan Pekerjaan

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
		<i>JobController</i>
<b>Nama Fungsi</b>	<ul style="list-style-type: none"> <li>• index</li> <li>• create</li> <li>• store</li> <li>• show</li> <li>• <i>update</i></li> <li>• <i>destroy</i></li> </ul>	<ul style="list-style-type: none"> <li>• lowonganpekerjaan</li> <li>• addLowonganPekerjaan</li> <li>• editLowonganPekerjaanSimpan</li> <li>• editLowonganPekerjaan</li> <li>• hapusLowonganPekerjaan</li> </ul>

<b>Peran</b>	<i>Controller</i>	<i>Controller</i>
<b>Keterangan</b>	<p>Sedikit Modifikasi, yaitu melakukan perubahan terhadap fungsi yang ada dalam aplikasi sebelumnya tetapi tidak merubah tujuan dari fungsi tersebut hanya cara implementasi <i>codenya</i> saja yang berbeda</p> <p>Jumlah baris <i>code</i> : 129</p> <p>Jumlah baris <i>code</i> yang di gunakan ulang : 0</p> $\% \text{ reused} = \frac{0}{129}$ <p>Persentase baris <i>code</i> yang digunakan : 0 %</p>	

#### 4.3.5. Produk

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
		<i>ProductController</i>
<b>Nama Fungsi</b>	<ul style="list-style-type: none"> <li>• index</li> <li>• create</li> <li>• store</li> <li>• show</li> <li>• <i>update</i></li> <li>• destroy</li> </ul>	<ul style="list-style-type: none"> <li>• produk</li> <li>• addProduk</li> <li>• editprodukSimpan</li> <li>• editproduk</li> <li>• hapusproduk</li> </ul>
<b>Peran</b>	<i>Controller</i>	<i>Controller</i>
<b>Keterangan</b>	<p>Sedikit Modifikasi, yaitu melakukan perubahan terhadap fungsi yang ada dalam aplikasi sebelumnya tetapi tidak merubah tujuan dari fungsi tersebut hanya cara implementasi <i>codenya</i> saja yang berbeda</p> <p>Jumlah baris <i>code</i> : 126</p> <p>Jumlah baris <i>code</i> yang di gunakan ulang : 0</p> $\% \text{ reused} = \frac{0}{126}$ <p>Persentase baris <i>code</i> yang digunakan : 0 %</p>	

#### 4.3.6. Aspirasi

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
		<i>AspirationController</i>
<b>Nama Fungsi</b>	<ul style="list-style-type: none"> <li>• index</li> <li>• create</li> <li>• store</li> <li>• show</li> <li>• <i>update</i></li> <li>• destroy</li> <li>• status</li> </ul>	<ul style="list-style-type: none"> <li>• aspirasi</li> <li>• addAspirasi</li> <li>• aspirasiEdit_Simpan</li> <li>• aspirasiEdit</li> <li>• aspirasiHapus</li> <li>• aspirasiAccepted</li> </ul>
<b>Peran</b>	<i>Controller</i>	<i>Controller</i>
<b>Keterangan</b>	<p>Sedikit Modifikasi, yaitu melakukan perubahan terhadap fungsi yang ada dalam aplikasi sebelumnya tetapi tidak merubah tujuan dari fungsi tersebut hanya cara implementasi <i>codenya</i> saja yang berbeda</p> <p>Jumlah baris <i>code</i> : 124</p> <p>Jumlah baris <i>code</i> yang di gunakan ulang : 0</p> $\% \text{ reused} = \frac{0}{124}$ <p>Persentase baris <i>code</i> yang digunakan : 0 %</p>	

#### 4.3.7. Pengaduan

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
		<i>ReportController</i>



<b>Nama Fungsi</b>	<ul style="list-style-type: none"> <li>• index</li> <li>• create</li> <li>• store</li> <li>• show</li> <li>• <i>update</i></li> <li>• destroy</li> </ul>	<ul style="list-style-type: none"> <li>• pengaduan</li> <li>• addPengaduan</li> <li>• editpengaduansimpan</li> <li>• editpengaduan</li> <li>• hapuspengaduan</li> </ul>
<b>Peran</b>	<i>Controller</i>	<i>Controller</i>
<b>Keterangan</b>	<p>Sedikit Modifikasi, yaitu melakukan perubahan terhadap fungsi yang ada dalam aplikasi sebelumnya tetapi tidak merubah tujuan dari fungsi tersebut hanya cara implementasi <i>codenya</i> saja yang berbeda</p> <p>Jumlah baris <i>code</i> : 117</p> <p>Jumlah baris <i>code</i> yang di gunakan ulang : 0</p> $\% \text{ reused} = \frac{0}{117}$ <p>Persentase baris <i>code</i> yang digunakan : 0 %</p>	

#### 4.3.8. Layanan

	<b>Web Desa Sundawenang</b>	<b>Web Desa Sukapura</b>
<b>Nama Kelas</b>	<ul style="list-style-type: none"> <li>• <i>ServiceController</i></li> <li>• <i>DomisiliPaudController</i></li> <li>• <i>DomisiliYayasanController</i></li> <li>• <i>IzinRameRameController</i></li> <li>• <i>KelahiranController</i></li> <li>• <i>KematianController</i></li> <li>• <i>KetAhliWarisController</i></li> <li>• <i>KetDomisiliKoperasiController</i></li> <li>• <i>KetDomisiliPerusahaanController</i></li> <li>• <i>KetDomisiliUsahaController</i></li> <li>• <i>KetPengakuanCeraiController</i></li> <li>• <i>KetPenghasilanController</i></li> <li>• <i>KetStatusPernikahanController</i></li> <li>• <i>KetSudahMenikahController</i></li> <li>• <i>KetTidakMampuController</i></li> <li>• <i>KetTidakMemilikiRumahController</i></li> <li>• <i>PerbedaanIdentitasController</i></li> <li>• <i>SkckController</i></li> <li>• <i>SurvaiLapanganController</i></li> <li>• <i>TempatTinggalController</i></li> <li>• <i>UsahaController</i></li> </ul>	<i>Master_controller</i>
<b>Nama Fungsi</b>	<ul style="list-style-type: none"> <li>• index</li> <li>• create</li> <li>• store</li> </ul>	<ul style="list-style-type: none"> <li>• layanan</li> <li>• tambahLayanan_proses</li> <li>• editLayanan_proses</li> </ul>

	<ul style="list-style-type: none"> <li>• show</li> <li>• edit</li> <li>• <i>update</i></li> <li>• destroy</li> </ul>	<ul style="list-style-type: none"> <li>• hapusLayanan</li> <li>• accepting_layanan</li> <li>• cetakLayanan</li> </ul>
<b>Peran</b>	<i>Controller</i>	<i>Controller</i>
<b>Keterangan</b>	<p>Sedikit Modifikasi, yaitu melakukan perubahan terhadap fungsi yang ada dalam aplikasi sebelumnya tetapi tidak merubah tujuan dari fungsi tersebut hanya cara implementasi <i>codenya</i> saja yang berbeda</p> <p>Jumlah baris <i>code</i> : 132</p> <p>Jumlah baris <i>code</i> yang di gunakan ulang : 0</p> $\% \text{ reused} = \frac{0}{132}$ <p>Persentase baris <i>code</i> yang digunakan : 0 %</p>	

#### 4.3.9. Pengaturan

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
		Pengaturan <i>Controller</i>
<b>Nama Fungsi</b>	<ul style="list-style-type: none"> <li>• Index</li> <li>• <i>update</i></li> </ul>	<ul style="list-style-type: none"> <li>• pengaturan</li> <li>• editPengaturan</li> </ul>
<b>Peran</b>	<i>Controller</i>	<i>Controller</i>
<b>Keterangan</b>	<p>Sedikit Modifikasi, yaitu melakukan perubahan terhadap fungsi yang ada dalam aplikasi sebelumnya tetapi tidak merubah tujuan dari fungsi tersebut hanya cara implementasi <i>codenya</i> saja yang berbeda</p> <p>Jumlah baris <i>code</i> : 23</p> <p>Jumlah baris <i>code</i> yang di gunakan ulang : 0</p> $\% \text{ reused} = \frac{0}{23}$ <p>Persentase baris <i>code</i> yang digunakan : 0 %</p>	

#### 4.4. Database

Nama Kelas	Web Desa Sundawenang	Web Desa Sukapura
		sdwn.sql
<b>Nama Fungsi</b>	-	-
<b>Peran</b>	<i>Database</i>	<i>Database</i>
<b>Keterangan</b>	<p>Digunakan kembali secara keseluruhan</p> <p>Jumlah tabel : 39</p> <p>Jumlah baris table yang digunakan ulang : 39</p> $\% \text{ reused} = \frac{39}{39}$ <p>Persentase baris <i>code</i> yang digunakan : 100 %</p>	

#### 5. Kesimpulan

Berdasarkan analisis bab sebelumnya didapatkan rata-rata penggunaan ulang *code web* desa sundawenang ke desa sukapura pada *model* sebesar 0%, *view* 31.9%, *controller* 0%, *database* 100%. Pada *model* dan *controller* terlihat tidak memiliki penggunaan *code* ulang dari desa sundawenang ke desa sukapura dibandingkan dengan *view* dan *database*. *Framework* yang digunakan di desa sundawenang yaitu laravel lalu untuk desa sukapura yaitu CI (*Codeigniter*) dengan perbedaan kedua *framework* tersebut pada bagian *model* dan *controller* harus menggunakan penggunaan kembali *code* dengan cara full modifikasi dikarenakan cara implementasi yang berbeda pada kedua *framework* tersebut. Pada bagian *view*, penggunaan *framework* frontend (CSS, HTML) menggunakan *framework* yang sama yaitu bootstrap3 sehingga ada beberapa *code* yang dapat digunakan kembali, lalu untuk bagian *database* menggunakan ulang *code* kembali secara keseluruhan tanpa ada modifikasi karena structure *database* dan jenis *database* yang digunakan sama yaitu MySQL. Oleh karena itu dapat disimpulkan penggunaan ulang *code* pada *web* desa sundawenang ke *web* desa sukapura tidak dapat digunakan ulang *code* secara penuh tetapi masih dengan alur yang sama dan ada beberapa komponen yang dapat digunakan kembali seperti *view* dan *database*.

**Daftar Pustaka**

1. *Software Aplikasi Desa Sundawenang*
2. Aditi Dubey, H. K. (2015). *Reusability Types and Reuse Metrics: A Survey*. Punjab: Lovely Professional Univers.
3. Artem Syromiatnikov, D. W. (n.d.). *A Journey Through the Land of Model-View-Design Patterns*. V`axj`o, Sweden: Linnaeus University.
4. Benneth Christiansson, L. J. (n.d.). *Component-Based Software Development Life Cycles* .
5. Christos Malliarakis, M. S. (n.d.). *Designing Educational Games for Computer Programming: A holistic Framework*.
6. Felix Bachmann, L. B.-D. (2000). *Technical Concepts of Component-Based Software Engineering*.
7. Hidayat Arief, S. B. (2012). *Penerapan Arsitektur Model View Controller (MVC)*. Dalam Rancang Bangun Sistem Kuis Online Adaptif.
8. Iqbal H. Sarker, K. A. (2014). *MVC Architecture Driven Design and Implementation of Java Framework for Developing Desktop Application*. Department of Computer Science & Engineering, Chittagong University of Engineering & Technology (CUET).
9. Jianhu Zheng, Y. F. (2014). *A Unified Modeling Language-Based Design and Application for a Library Management Information System* . BULGARIAN ACADEMY OF SCIENCES .
10. Letkowski, J. (2014). *Doing database design with MySQL* . Western New England University.
11. M. Sidi Mustaqbal, R. F. (2015). *PENGUJIAN APLIKASI MENGGUNAKAN BLACK BOX TESTING BOUNDARY VALUE ANALYSIS (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN)* . Universitas Widyatama.
12. Medha Kalelkar, D. K. (2014). *Implementation of Model-View-Controller Architecture Pattern for Business Intelligence Architecture* . International Journal of Computer Applications .

13. Meyer, B. (n.d.). *Object Oriented Software Construction Second Edition*. Santa Barbara: Interactive Software Engineering Inc.
14. Petre, M. (n.d.). *UML in Practice*. Milton Keynes, UK : The Open University .
15. Reenskaug, T. (1979). *THING-MODEL-VIEW-EDITOR an Example from a planningsystem*.
16. Ruli Erinton, R. M. (2017). *ANALISIS PERFORMASI FRAMEWORK CODEIGNITER DAN LARAVEL MENGGUNAKAN WEB SERVER APACHE* . Bandung: Universitas Telkom.
17. Sharanjit Singh, A. S. (2016). *Component Based Software Engineering*. International Research Journal of Engineering and Technology (IRJET).
18. Sommerville, I. (2004). *Component-based software engineering*.
19. Vicky Bumbak, P. K. (2011). *An Approach for Reusability Measurement of Object-Oriented Code Based Upon CK Metrics* . Hisar, India: University of science and technology.