

Implementasi Framework MVC dan Prototype Method pada Website Monitoring Pertanian Desa Kalteng

Tomas Leonardo

Teknik Informatika, Fakultas Teknik
Universitas Palangkaraya

*Corresponding author email:
tomasleonardo@it.upr.ac.id

ABSTRAK

Kegiatan kawasan rumah pangan lestari merupakan salah satu kegiatan prioritas nasional dalam mengurangi kemiskinan dan gizi buruk pada wilayah desa yang masuk zona rentan rawan pangan, studi kasus yang dilakukan peneliti adalah area kalimantan tengah yang mana merupakan salah satu area yang cukup terdampak. Pemantauan yang dilakukan ini sangat penting untuk memberikan umpan balik agar dapat dilakukan tindak lanjut terhadap kondisi yang perlu penanganan cepat yaitu permasalahan dilapangan. Berdasarkan observasi yang dilakukan, mekanisme pemantauan perkembangan kegiatan pelaksanaan masih manual dan tidak tersistem secara online sehingga sulit untuk diproses dengan cepat. Berdasarkan permasalahan tersebut maka dilakukan implementasi sistem website yang mampu untuk memudahkan pemantauan kegiatan ini setiap saat. Sistem yang dibuat menerapkan bahasa Pemrograman PHP, Javascript/Ajax, HTML dan MySQL Server dalam Framework Model-View-Controller (MVC) dengan Prototype Development method. Sistem ini juga dapat menjadi sumber referensi untuk penelitian-penelitian serupa tentang bagaimana memetakan dan mengimplementasikan sektor-sektor dan proses kerja framework Model View Controller untuk pengembangan aplikasi web khususnya dalam ruang lingkup E-Government atau sistem monitoring.

Kata Kunci: Model-View-Controller Framework; Prototype Development Method; Web Application.

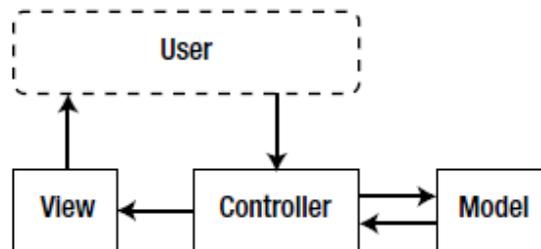
I. Pendahuluan

Kegiatan Kawasan Rumah Pangan Lestari (KRPL) merupakan salah satu kegiatan prioritas nasional dalam mengurangi kemiskinan dan gizi buruk pada wilayah yang masuk zona rentan rawan pangan. Yang dimaksud dengan pemantauan dalam KRPL adalah memastikan kegiatan budidaya berkembang dengan baik secara berkala, berkelanjutan dan tepat waktu mulai dari persiapan, penanaman dan pemeliharaan di kebun bibit, demplot kelompok, pekarangan anggota, kebun sekolah sampai dengan pengolahan hasil pekarangan dengan konsep pangan yang beragam, bergizi seimbang dan aman (B2SA). Pemantauan yang dilakukan pada kegiatan ini sangat penting untuk memberikan umpan balik kepada kelompok dalam bentuk melakukan tindak lanjut terhadap kondisi yang perlu penanganan cepat yaitu permasalahan dilapangan seperti misalkan di suatu desa yang melaksanakan kegiatan KRPL terjadi bencana alam seperti banjir, kekeringan atau serangan bermacam penyakit yang mengakibatkan layu pada tanaman dan tidak cepat dilaporkan. Maka hal tersebut dapat mengakibatkan gagal panen pada kegiatan KRPL tersebut dan dapat mengakibatkan kerugian besar. Berdasarkan observasi studi kasus yang dilakukan, pemantauan perkembangan tidak terpantau secara berkala dan berkelanjutan, yang dikarenakan proses harus dilakukan secara manual dan tidak tersistem secara online. Berdasarkan permasalahan tersebut dilakukan implementasi sistem berbasis website yang mampu untuk memudahkan pemantauan kegiatan ini. Sehingga nantinya diharapkan proses yang ada menjadi terpantau dengan baik dan dapat mempercepat waktu siklus laporan pemantauan. Sistem yang akan dibangun menerapkan framework Model View Controller dengan Prototype Development Method. Hal tersebut dilatarbelakangi oleh pentingnya penerapan sistem yang dapat dimodifikasi dan dimaintenance kedepannya dengan mudah sesuai dengan kebutuhan yang dinamis.

II. Landasan Teori

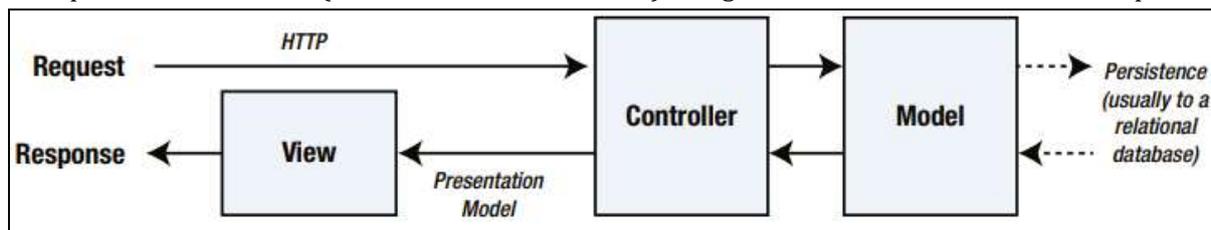
A. Model View Controller (MVC)

Model View Controller adalah pola perancangan perangkat lunak yang dibangun di sekitar interkoneksi tiga tipe komponen utama, dalam bahasa pemrograman seperti PHP seringkali dengan fokus yang kuat pada pemrograman berorientasi objek (OOP). Ketiga tipe komponen tersebut adalah model, view dan controller. Model adalah dimana semua logika bisnis sebuah aplikasi dijaga, view adalah dimana semua elemen antarmuka pengguna dari aplikasi disimpan, controller adalah komponen yang menghubungkan model dan tampilan secara bersamaan. Controller mengisolasi bisnis logika model dari elemen antarmuka pengguna dan menangani bagaimana aplikasi akan merespon interaksi pengguna dalam tampilan (Pitt & Pitt, 2012). MVC banyak digunakan dalam model arsitektur perangkat lunak dan aplikasi web, meningkatkan pemeliharaan dan penggunaan kembali sistem (Lossius et al., 2014).



Gambar 1. Model-View-Controller in a Nutshell (Pitt & Pitt, 2012)

Diketahui jika ada pemisahan sektor view dan komponen controller, perkiraan dari total kode adalah 50% dalam model, 30% dalam view dan 20% di controller. Jika tidak ada pemisahan view dan controller maka estimasi adalah 50% dalam Model, 45% dianggap menjadi view dan 5% menjadi controller (Ollsson et al., 2015). Hal ini membuat MVC lebih mudah di reuse, upgrade atau diadaptasikan ke platform lain karena tingkat ketergantungannya ke sektor View diminimalisir khususnya di Controller. Dalam (Betz et al., 2010) Paten MVC terkait berhubungan dengan aplikasi web dengan framework MVC yang lebih khusus lagi untuk arsitektur dan metode pemrograman yang dapat meningkatkan kinerja dan memudahkan pengembangan dan upaya pemeliharaan. Berikut merupakan analisa dalam (Freeman & Freeman, 2014) mengenai interaksi MVC dalam suatu aplikasi.

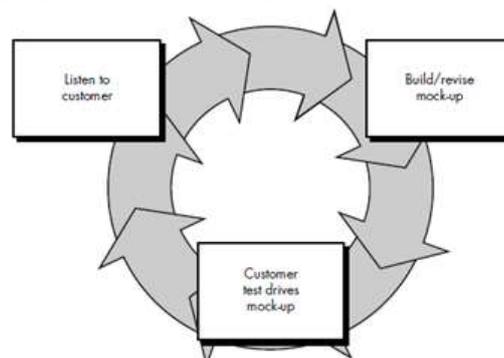


Gambar 2. Model-View-Controller Architecture (Freeman & Freeman, 2014)

Sejumlah besar teknik MVC dan kerangka pengembangan perangkat lunak telah diterapkan dalam beberapa tahun terakhir di bidang rekayasa perangkat lunak untuk menggunakan kembali kode dan aplikasi yang efisien (Sarker & Apu, 2014). Dalam studi kasus MVC (Pop & Altar, 2014), disimpulkan bahwa kerangka kerja MVC yang diimplementasikan mampu mengurangi waktu pengembangan aplikasi web secara drastis, memungkinkan pengembang untuk fokus pada tugas-tugas khusus aplikasi dari pada membuang-buang waktu mencoba menerapkan pola dan praktik yang terkenal. Pola arsitektur MVC sangat penting untuk mengembangkan aplikasi web yang efisien dan berkinerja tinggi (A. Kumar et al., 2023). Kemudian pada studi (Lin & Lin, 2013) juga disimpulkan bahwa NET MVC yang diterapkan pada E-commerce software ternyata tidak hanya berfungsi sebagai separasi model, view dan controller saja namun juga memberikan separasi logika bisnis, lapisan sistem yang kuat, terlepas dari perubahan kedepannya, lapisan model dengan perubahan minimal dan perubahan basis data tidak akan memengaruhi front sistem. Yang bermanfaat bagi anggota-anggota tim yang bekerja secara paralel, sehingga meningkatkan kemampuan penggunaan kembali perangkat lunak.

B. Prototype Method

Model Prototipe menempatkan lebih banyak upaya dalam menciptakan perangkat lunak yang sebenarnya dari pada berkonsentrasi dokumentasi. Dengan cara ini, perangkat lunak yang sebenarnya bisa dirilis terlebih dahulu (Sabale & Dani, 2012). Seringkali pelanggan mendefinisikan serangkaian tujuan umum untuk perangkat lunak namun tidak mengidentifikasi persyaratan input, pemrosesan, atau keluaran rinci. Dalam kasus lain, pengembang mungkin tidak yakin dengan efisiensi suatu algoritma, kemampuan beradaptasi dari sebuah sistem operasi, atau bentuk interaksi manusia atau mesin yang harus dilakukan. Dalam hal ini dan banyak situasi lain, paradigma prototyping mungkin menawarkan pendekatan terbaik. Paradigma prototyping dimulai dengan persyaratan pengumpulan. Pengembang dan pelanggan memenuhi dan menentukan keseluruhan tujuan perangkat lunak, mengidentifikasi apa pun persyaratan yang diketahui, dan garis besar area dimana definisi lebih lanjut adalah wajib. Sebuah "quick design" kemudian terjadi. Desain cepat berfokus pada representasi aspek perangkat lunak yang akan terlihat oleh pelanggan atau pengguna (misalnya, pendekatan input dan format output). Desain cepat mengarah pada konstruksi prototipe. Prototipe dievaluasi oleh pelanggan atau pengguna dan digunakan untuk memperbaiki persyaratan untuk perangkat lunak yang akan dikembangkan iterasi terjadi karena prototipenya disetel untuk memenuhi kebutuhan pelanggan, sementara pada saat bersamaan memungkinkan pengembang untuk lebih memahami apa yang perlu dilakukan. Prototipe berupaya mengurangi risiko proyek yang melekat dengan memecah proyek menjadi segmen yang lebih kecil dan memberikan lebih banyak kemudahan-perubahan selama proses pengembangan. Pengguna terlibat di seluruh proses pengembangan, yang meningkatkan kemungkinan penerimaan pengguna atas implementasi akhir (N. Kumar et al., 2013). Idealnya, prototipe berfungsi sebagai mekanisme untuk mengidentifikasi persyaratan perangkat lunak. Jika prototipe kerja dibuat, pengembang mencoba untuk menggunakan fragmen program yang ada atau alat yang berlaku (generator laporan, pengelola jendela) yang memungkinkan program dihasilkan dengan cepat (Pressman, 2005). Tim yang tangkas dapat mengadopsi best practice terkait penyelarasan antara dokumentasi tertulis dan prototipe, menentukan kapan harus membuat prototipe, dan memastikan keterlibatan tim dalam konstruksinya agar menjadi lebih efisien dan tegas dalam proses pengembangannya (Larrea et al., 2024).



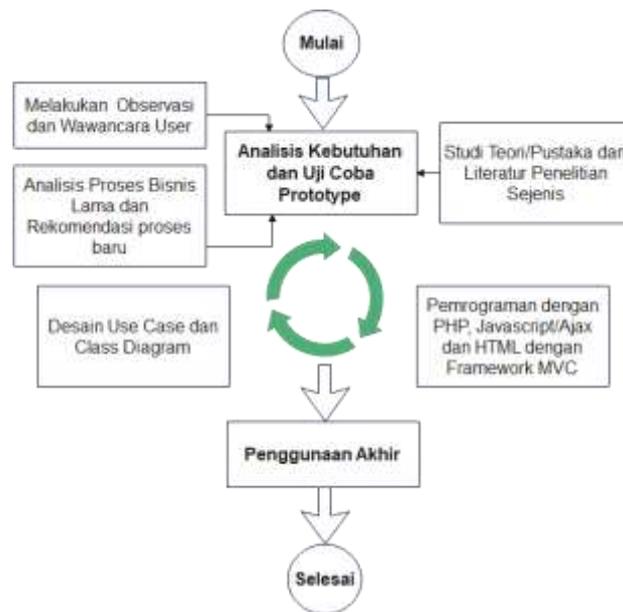
Gambar 3. Paradigma Prototype (Pressman, 2005)

Prototipe juga dapat membantu memahami perilaku produk akhir di awal proses, menurunkan risiko pengembangan produk yang gagal, dan meningkatkan kecepatan pemasaran produk (Yu & Enste, 2019).

III. Metode Penelitian

A. Tahapan Penelitian

Kerangka berpikir dalam suatu penelitian menggambarkan secara singkat namun menyeluruh kerangka pikir penulis terhadap penelitian yang akan dilakukannya. Berikut ini merupakan gambaran kerangka pikir dalam penelitian ini yang ada pada gambar 4 berikut ini.

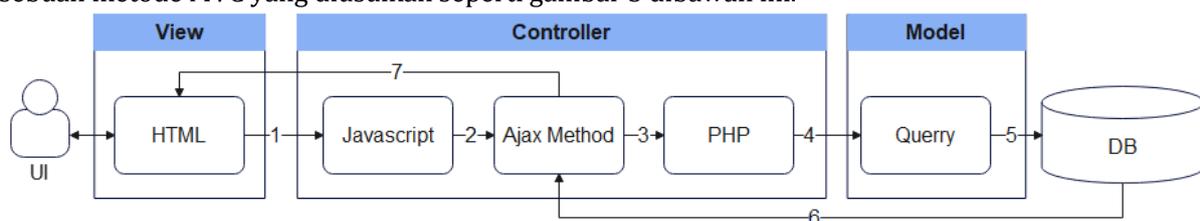


Gambar 4. Kerangka Pikir Penelitian

1. Pada tahapan awal penulis akan fokus untuk melakukan observasi dan wawancara user yang terlibat dalam sistem dan melakukan identifikasi masalah yang dihadapi dilapangan. Selama proses ini penulis akan melakukan kajian literatur dan melakukan analisis terhadap proses bisnis lama agar dapat membuat rekomendasi proses bisnis terbaru dan struktur MVC yang direkomendasikan.
2. Pada tahapan kedua penulis akan fokus menerapkan prototype method dengan membuat sistem prototipe yang dilengkapi dengan desain use case dan class diagram terbaru yang mana sistemnya menerapkan framework MVC. Selama proses ini penulis berkoordinasi dengan user yang terlibat sistem untuk menjamin output yang sesuai. Sehingga akan kembali ketahap 1 jika dibutuhkan.
3. Tahapan akhir adalah penggunaan, pada tahapan ini sistem diimplementasi dan siap digunakan.

B. Metode MVC yang diusulkan

Berdasarkan tinjauan literasi yang dilakukan penulis dari berbagai sumber. Maka dapat dirumuskan sebuah metode MVC yang diusulkan seperti gambar 5 dibawah ini.



Gambar 5. Metode MVC yang diusulkan

Berdasarkan gambar 5 dilakukan pembagian direktori program menjadi 3 sektor utama. View yang menangani komponen dasar dan tampilan utama, controller yang menangani pemrosesan. Kemudian model yang menangani manajemen terhadap database. Berikut ini merupakan penjelasan lebih lanjut bagaimana mekanisme framework ini dapat bekerja.

1. Input

Pada tahapan ini index menerima inputan dari user melalui antarmuka aplikasi. Input yang diterima berupa event yang ditrigger oleh user untuk menjalankan suatu fungsi, namun masih belum memuat variabel ataupun nilai. Yang nantinya nilai tersebut akan diteruskan ketahapan selanjutnya untuk diproses pada validasi input. Index berfungsi sebagai basis komponen dan tampilan utama yang

menjadi media interaksi visual utama bagi pengguna untuk dapat me-mahami informasi sistem yang sedang digunakan.

2. Validation

Pada tahapan ini digunakan fungsi yang menggunakan javascript untuk inputan user untuk menguji validitasnya dengan menerapkan fungsi percabangan. Karena bahasa ini bersifat client side, maka fungsi akan dijalankan secara realtime. Tahap ini berperan sebagai layer error handling agar data yang diinputkan dapat diketahui ketepatannya. Hal ini sangat diperlukan untuk mencegah data korup atau disintegrasikan sistem.

3. Execution

Pada tahapan ini digunakan metode AJAX untuk melakukan request komunikasi terhadap PHP di server. AJAX saat ini adalah istilah yang umum digunakan yang menjelaskan penggunaan XMLHttpRequest (disingkat XHR) API yang didukung semua browser modern. AJAX (Asynchronous JavaScript and XML) adalah teknologi yang memungkinkan aplikasi web untuk bertukar data secara asynchronous ke server tanpa memaksakan pemuatan ulang halaman, yang memungkinkan aplikasi web sisi client yang kaya dan responsive (Adamsen et al., 2018). AJAX menjadi perantara yang memungkinkan bahasa client side pada aplikasi dapat berinteraksi secara realtime terhadap bahasa pada server side. AJAX adalah suatu fungsi yang terdapat di PHP yang memungkinkan terjadinya komunikasi. PHP adalah bahasa pemrograman yang umumnya digunakan dalam pemrograman dinamis, open source, dan platform situs web independen, cepat, dengan banyak fungsi seperti large library dan banyaknya dokumentasi (Medina-Santiago et al., 2017). PHP adalah bahasa pemrograman dengan tingkat keamanan yang sangat baik karena hanya bekerja di sektor server saja. Oleh sebab itu bahasa PHP yang sangat dibutuhkan untuk menangani pemrosesan pada level tingkat tinggi dengan sensitivitas kode program yang sangat mempengaruhi dasar dari aplikasi dapat bekerja dan memproses data.

4. Request

Pada tahapan ini PHP melakukan request MySQL Query untuk melakukan fungsi pengelolaan database yang dibutuhkan. Setelah itu PHP menyiapkan fungsi update tampilan HTML dengan data yang sudah diperbaharui. MySQL adalah Database Management System (DBMS) yang baik, aman, dan cukup populer digunakan dalam pengembangan aplikasi web.

5. Processing

Pada tahapan ini dilakukan proses manajemen database ataupun pengambilan data sesuai dengan request sebelumnya dengan fungsi MySQL Query. Proses ini terjadi di sektor database yang dijalankan oleh MySQL.

6. Return Values

Pada tahapan ini dilakukan pengembalian semua nilai yang telah dilakukan oleh PHP dan MySQL Query sebelumnya kepada variabel AJAX. Sehingga variabel AJAX ini sudah berisi nilai yang memuat informasi terbaru.

7. Updating

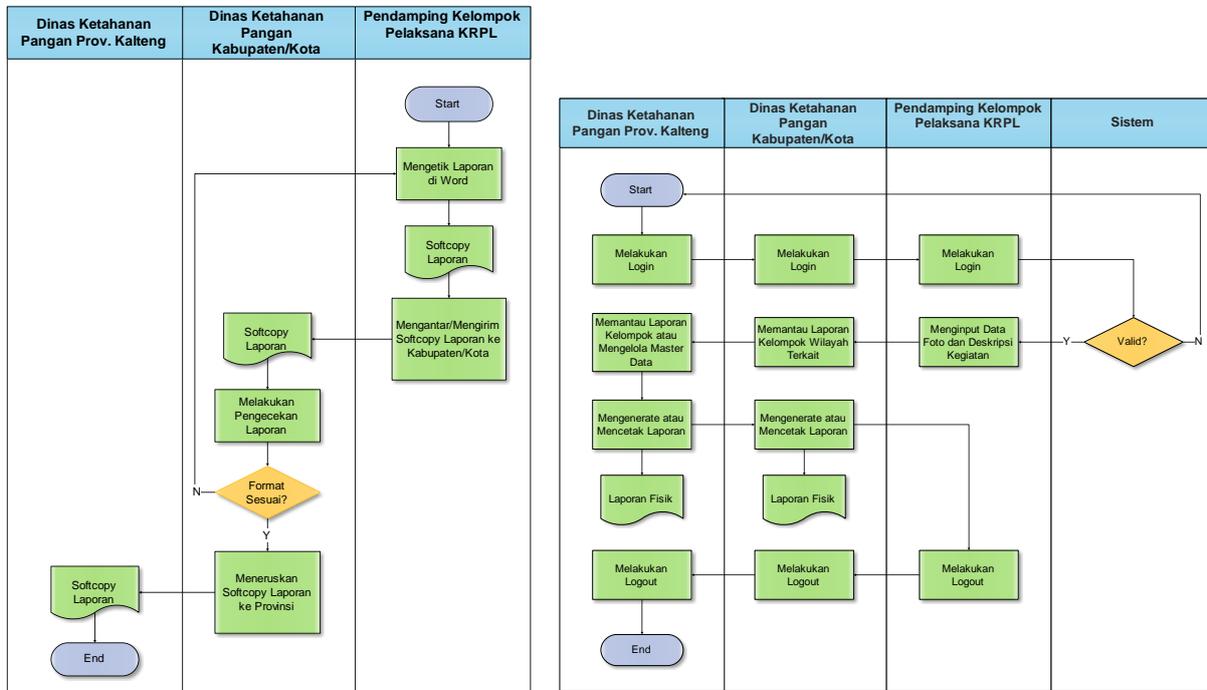
Pada tahapan ini fungsi AJAX melakukan penarikan konten yang lama sehingga digantikan dengan konten terbaru berdasarkan variabel AJAX sebelumnya.

C. Analisis Kebutuhan Sistem

Berdasarkan observasi dan wawancara yang dilakukan, diketahui proses yang dilakukan kurang efisien efektif dikarenakan tidak ada sistem yang dapat melakukan manajemen terhadap semua kegiatan pemantauan KRPL ini. Sehingga sangat perlu untuk dibangun sistem yang dapat memantau dan melaporkan kegiatan secara online dan real time agar kegiatan menjadi terpantau dengan baik. Berikut merupakan analisis kebutuhan sistemnya.

1. Flowchart sistem baru berdasarkan sistem lama.

Pada gambar 6 berikut ini dijelaskan lebih rinci bagaimana bisnis proses yang terjadi pada sistem yang sedang berjalan dengan menggunakan flowchart. Dapat dilihat bahwa aliran proses masih manual, kurang termanajemen dengan baik dan cenderung masih lambat dari segi kecepatan siklus karena ada user yang tidak dapat terlibat langsung ke sistem sampai user yang lain telah menyelesaikan siklusnya. Disisi lain adanya kemungkinan kesalahan format akan memperlama waktu siklus.

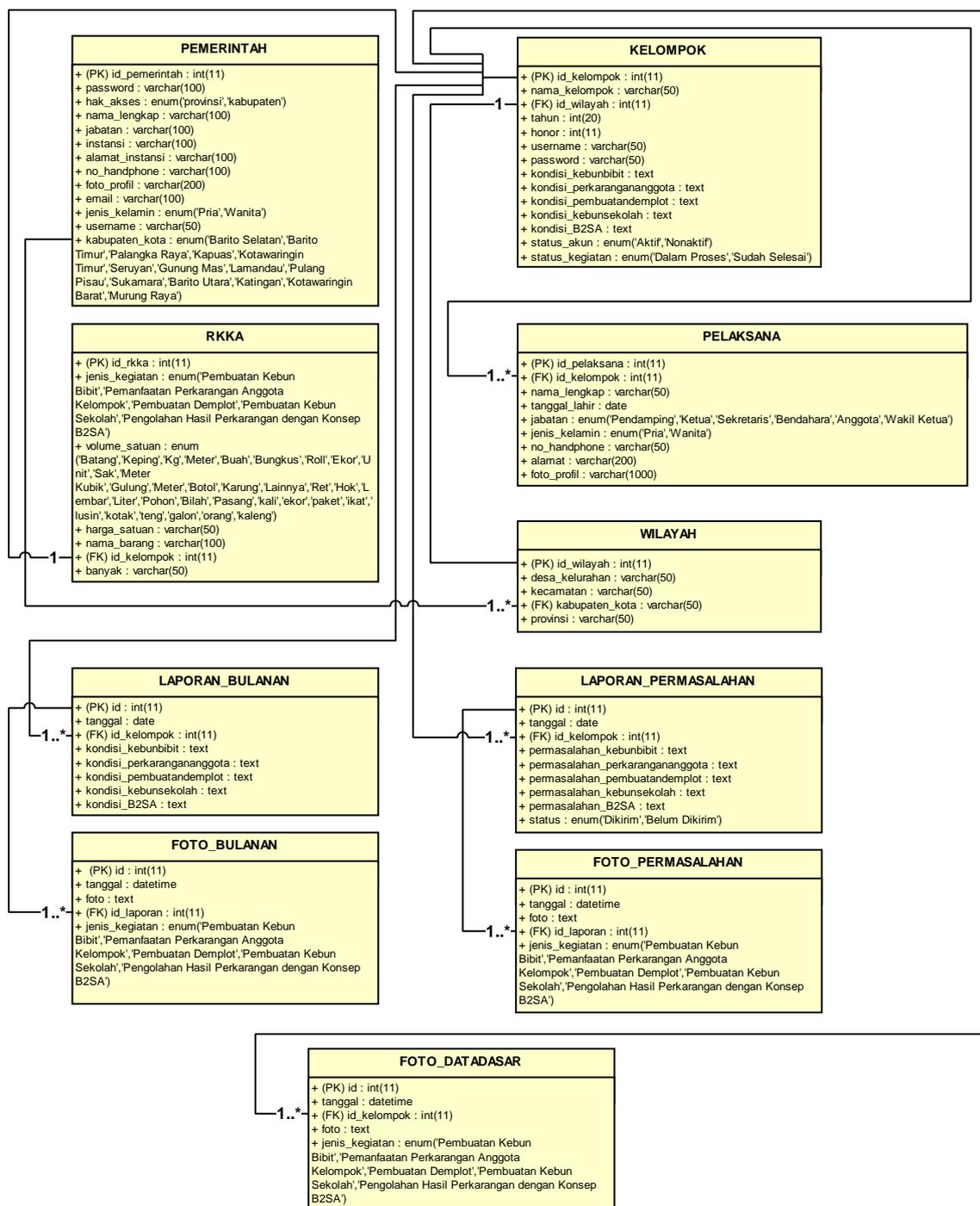


Gambar 6. Komparasi Flowchart sistem lama ke sistem terbaru

Pada gambar 6 juga dijelaskan bagaimana bisnis proses yang terjadi dengan menggunakan sistem yang terbaru yang direkomendasikan oleh penulis. Yang mana penyimpanan data lebih termanajemen dengan baik dan aliran proses dapat berjalan dengan cepat secara realtime dengan bantuan sistem. Dikarenakan semua user dapat terlibat dalam sistem secara bersamaan tanpa harus menunggu user lain. Disisi lain kesalahan format juga tidak akan terjadi dikarenakan sistem telah menyiapkan aturan format yang valid, sehingga user hanya perlu mengisi data saja. Sehingga waktu siklus hanya tergantung dari kapan pendamping kelompok menginputkan foto atau deskripsi kegiatannya.

2. Class Diagram

Class Diagram digunakan untuk menjelaskan dan memetakan bagaimana struktur basis data yang akan diterapkan sistem yang akan dibangun. Sehingga akan memudahkan untuk melakukan analisa entitas yang terlibat, struktur data yang dimanajemen dan juga bagaimana relasi antar tabel. Setelah penulis menganalisa studi kasus yang terjadi maka dirancang skema class diagram yang menggambarkan kebutuhan struktur database pada gambar 7 berikut ini.



Gambar 7. Pemetaan sistem baru dengan Class Diagram

Pada gambar 7 tersebut dapat dipahami bahwa sistem ini akan menggunakan 10 tabel dengan detail struktur data yang dapat dilihat pada gambar tersebut. Pada gambar tersebut juga terlihat relasi antar tabel yang digambarkan dengan menggunakan kardinalitas. Berikut merupakan penjelasan lebih lengkap mengenai relasi yang terjadi.

1. Relasi Tabel Pemerintah dan Wilayah

Pada tabel tersebut dapat dilihat bahwa tabel pemerintah berperan sebagai tabel parent dan wilayah sebagai tabel child. Hal tersebut dapat dilihat dari penggunaan foreign key referensi pada field kabupaten/kota pada tabel wilayah dengan kardinalitas 1..* (one or many). Yang menandakan bahwa 1 pemerintah dapat memantau banyak wilayah desa.

2. Relasi Tabel Kelompok dan Pelaksana

Pada tabel tersebut dapat dilihat bahwa tabel kelompok berperan sebagai tabel parent dan pelaksana sebagai tabel child. Hal tersebut dapat dilihat dari penggunaan foreign key referensi pada field `id_kelompok` pada tabel pelaksana dengan kardinalitas 1..* (one or many). Yang menandakan bahwa 1 kelompok dapat memiliki banyak pelaksana.

3. Relasi Tabel Kelompok dan RKKA

Pada tabel tersebut dapat dilihat bahwa tabel kelompok berperan sebagai tabel parent dan RKKA sebagai tabel child. Hal tersebut dapat dilihat dari penggunaan foreign key referensi pada field `id_kelompok` pada tabel RKKA dengan kardinalitas 1 (only one). Yang menandakan bahwa 1 kelompok dapat memiliki paling banyak 1 RKKA.

4. Relasi Tabel Wilayah dan Kelompok

Pada tabel tersebut dapat dilihat bahwa tabel wilayah berperan sebagai tabel parent dan kelompok sebagai tabel child. Hal tersebut dapat dilihat dari penggunaan foreign key referensi pada field `id_wilayah` pada tabel kelompok dengan kardinalitas 1 (only one). Yang menandakan bahwa 1 kelompok bertugas hanya pada 1 wilayah desa.

5. Relasi Tabel Kelompok dan Laporan_Bulanan

Pada tabel tersebut dapat dilihat bahwa tabel kelompok berperan sebagai tabel parent dan laporan_bulanan sebagai tabel child. Hal tersebut dapat dilihat dari penggunaan foreign key referensi pada field `id_kelompok` pada tabel laporan_bulanan dengan kardinalitas 1..* (one or many). Yang menandakan bahwa 1 kelompok dapat membuat banyak laporan bulanan.

6. Relasi Tabel Kelompok dan Laporan_Permasalahan

Pada tabel tersebut dapat dilihat bahwa tabel kelompok berperan sebagai tabel parent dan laporan_permasalahan sebagai tabel child. Hal tersebut dapat dilihat dari penggunaan foreign key referensi pada field `id_kelompok` pada tabel laporan_permasalahan dengan kardinalitas 1..* (one or many). Yang menandakan bahwa 1 kelompok dapat membuat banyak laporan permasalahan.

7. Relasi Tabel Kelompok dan Foto_Datadasar

Pada tabel tersebut dapat dilihat bahwa tabel kelompok berperan sebagai tabel parent dan foto_datadasar sebagai tabel child. Hal tersebut dapat dilihat dari penggunaan foreign key referensi pada field `id_kelompok` pada tabel foto_datadasar dengan kardinalitas 1..* (one or many). Yang menandakan bahwa 1 kelompok dapat memiliki banyak foto didalam laporan datadasar.

8. Relasi Tabel Laporan_Bulanan dan Foto_Bulanan

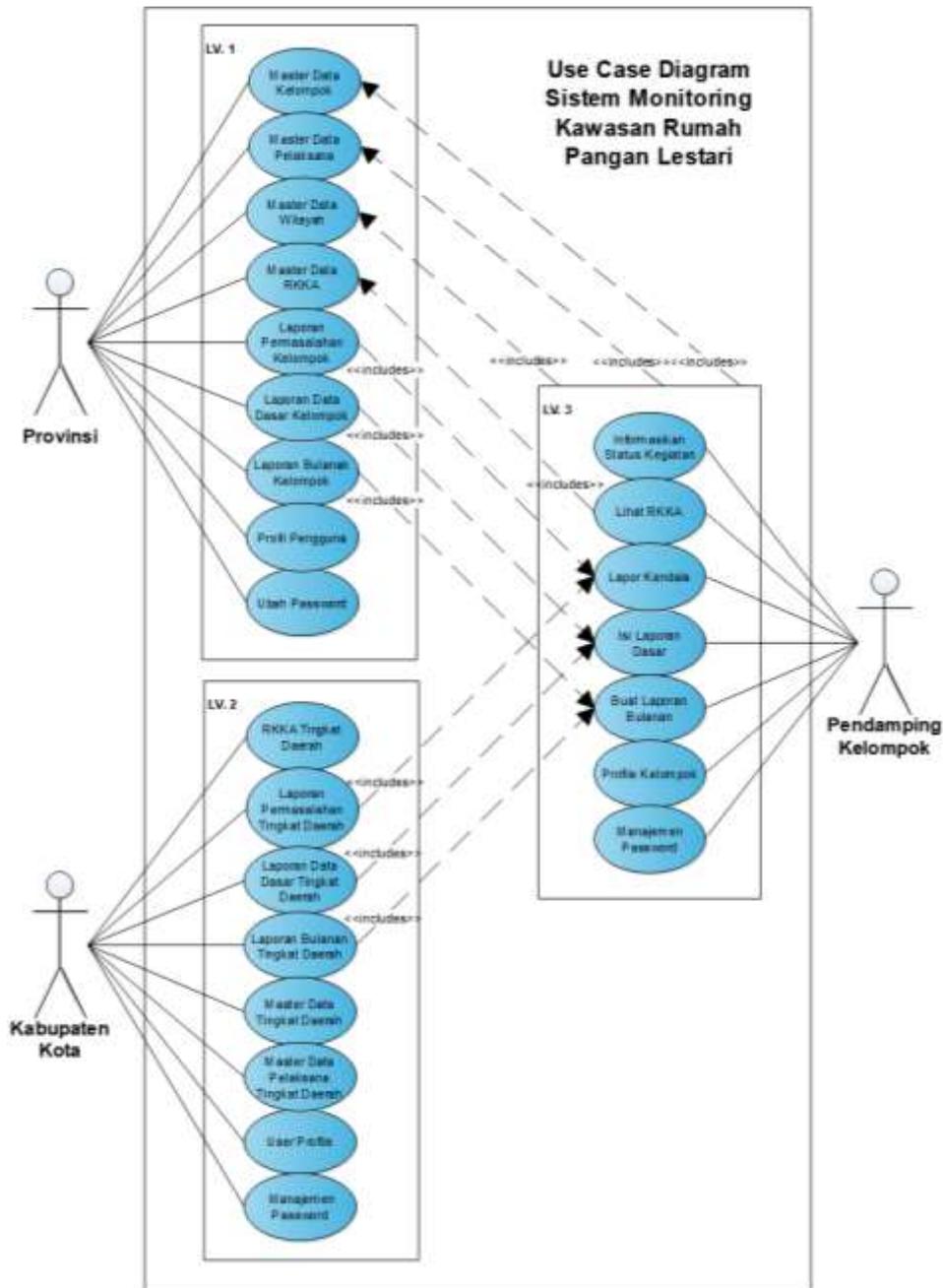
Pada tabel tersebut dapat dilihat bahwa tabel laporan_bulanan berperan sebagai tabel parent dan foto_bulanan sebagai tabel child. Hal tersebut dapat dilihat dari penggunaan foreign key referensi pada field `id_laporan` pada tabel foto_bulanan dengan kardinalitas 1..* (one or many). Yang menandakan bahwa sebuah laporan bulanan dapat memuat banyak foto.

9. Relasi Tabel Laporan_Permasalahan dan Foto_Permasalahan

Pada tabel tersebut dapat dilihat bahwa tabel laporan_permasalahan berperan sebagai tabel parent dan foto_permasalahan sebagai tabel child. Hal tersebut dapat dilihat dari penggunaan foreign key referensi pada field `id_laporan` pada tabel foto_permasalahan dengan kardinalitas 1..* (one or many). Yang menandakan bahwa sebuah laporan permasalahan dapat memuat banyak foto.

3. Use Case

Setelah melakukan analisa Class Diagram, akhirnya dapat disimpulkan bahwa kebutuhan fitur-fitur aplikasi pada sistem ini dapat dijelaskan menggunakan diagram Use Case. Pada gambar 8 berikut ini menjelaskan fitur apa saja yang disediakan bagi pengguna sistem dengan tingkat hak akses provinsi, kabupaten/kota dan kelompok yang telah sukses melakukan login ke sistem.



Gambar 8. Pemetaan fitur dengan Use Case Diagram

Pada gambar 8 tersebut di tingkat pengguna provinsi terdapat 4 fitur master data yang digunakan sebagai penyimpanan utama bagi sistem untuk mengelola atau memantau data kelompok, data pelaksana, data RKKK dan data wilayah seluruh kegiatan yang ditandai dengan fitur-fitur tersebut di include oleh level kelompok. Kemudian juga terdapat 3 fitur laporan yaitu laporan permasalahan, laporan datadasar dan laporan bulanan yang meng include fitur laporan di level kelompok. Fitur-fitur tersebut digunakan untuk melakukan pemantauan laporan setiap kelompok. Kemudian juga disediakan fitur profile pengguna dan ubah password untuk menyediakan kelengkapan dan keamanan bagi akun pengguna. Pendamping kelompok juga dapat menginformasikan ke pengguna dengan tingkatan diatas agar dapat mengetahui status kegiatan kelompok apakah masih berjalan atau sudah selesai. Pembahasan lebih detail mengenai masing-masing fitur ini dapat dilihat pada bagian hasil dan implementasi fitur program. Untuk tingkat kabupaten/kota hanya dapat melakukan pemantauan saja tanpa pengelolaan dan cakupan wilayah juga dibatasi berdasarkan regional masing-masing.

IV. Hasil Dan Pembahasan

1. Login User

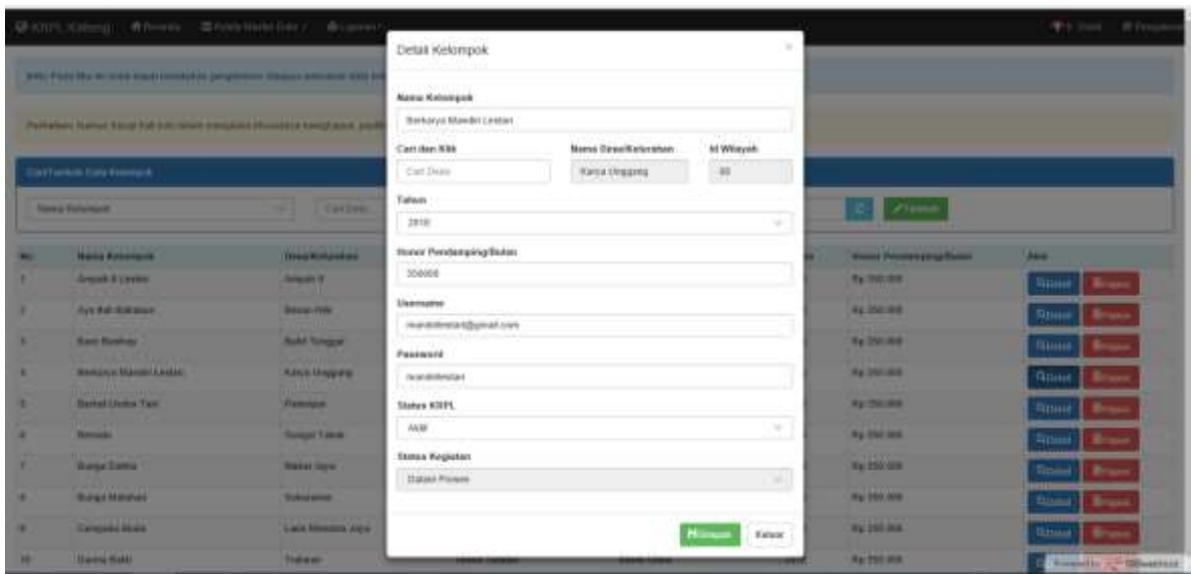
Pada gambar 9 dapat terlihat halaman login untuk mengakses sistem. Halaman ini berfungsi untuk melakukan validasi pengguna yang akan mengakses web. Pengguna ini adalah pengguna dengan hak akses provinsi yang mana bertugas sebagai pemantau dan pengelola master data dalam sistem, kabupaten/kota sebagai pemantau per wilayah kabupaten masing-masing, dan kelompok yang bertugas sebagai pelaksana lapangan dalam kegiatan KRPL.



Gambar 9. Login

2. Master Data Kelompok

Pada gambar 10 dapat terlihat gambar halaman master data kelompok. Fitur ini menunjukkan semua kelompok KRPL dari seluruh periode atau tahun. Data kelompok ini dapat dicari, ditambah, diubah dan dihapus sesuai dengan keperluan dari administrator provinsi. Namun untuk saat ini data tidak dapat dihapus untuk alasan keamanan. Data ini menunjukkan identitas dan informasi lengkap dari setiap kelompok seperti nama dan lokasi kerja kelompok. Fitur ini juga tersedia bagi administrator kabupaten/kota. Namun untuk level administrator kabupaten/kota tidak diijinkan untuk dapat melakukan penambahan, pengubahan atau penghapusan data kelompok. Kemudian data kelompok yang ditampilkan hanya bersifat regional. Yang mana hanya pada wilayah kabupaten/kota wilayah terkait saja.

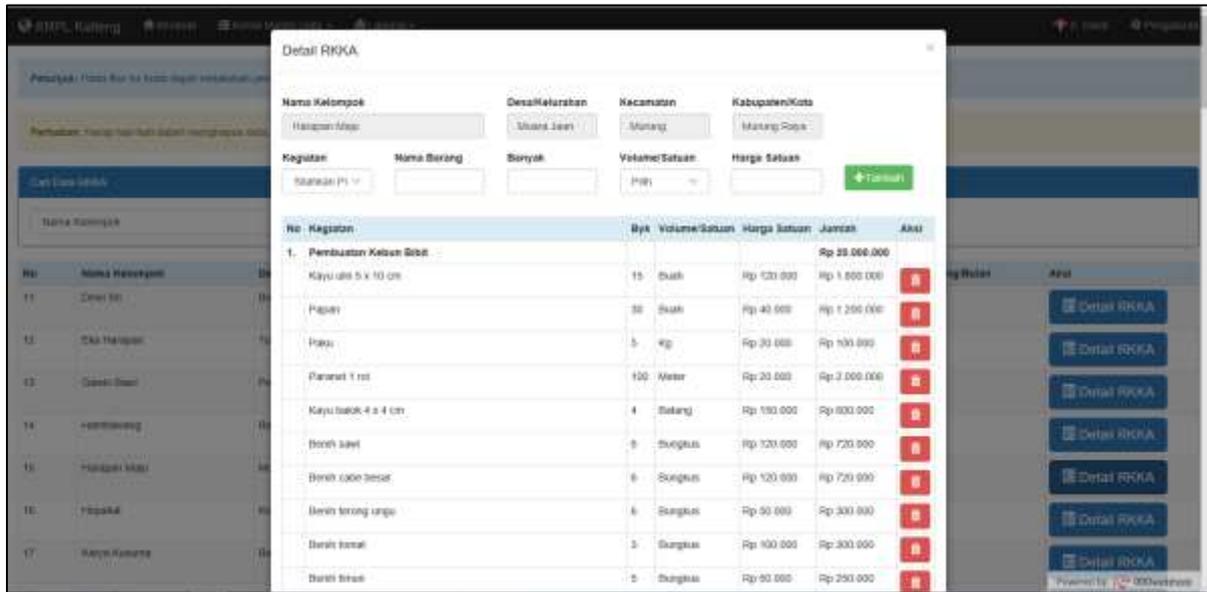


Gambar 10. Master Data Kelompok

3. Master Data RKKA Kelompok

Dapat dilihat pada gambar 11 bahwa Data RKKA tersebut dapat dikelola sesuai dengan kebutuhan administrator provinsi kapan saja. Data RKKA ini adalah semua data barang dan biaya yang dikeluarkan. Yang mana sudah diusulkan dan disetujui untuk dibelanjakan dalam rangka melakukan

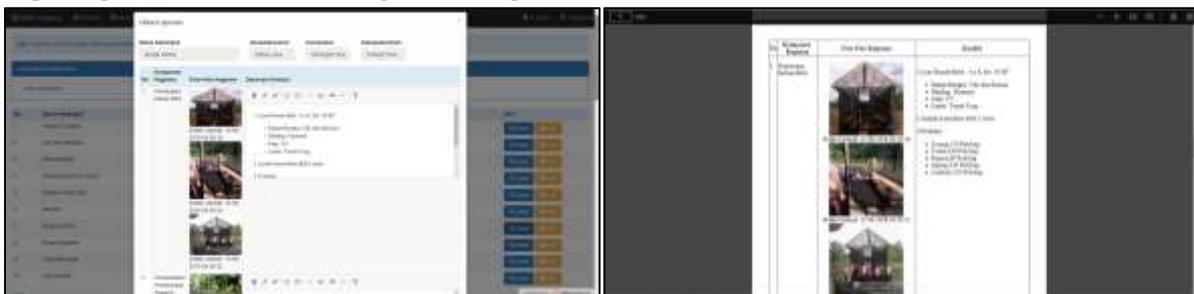
kegiatan pelaksanaan kegiatan KRPL di wilayah desa masing-masing. Data barang ini dapat dikelola dengan mudah menggunakan panel pengelolaan seperti yang terlihat pada gambar tersebut. Fitur ini juga tersedia bagi administrator kabupaten/kota. Namun untuk level administrator kabupaten/kota tidak diijinkan untuk dapat melakukan penambahan atau penghapusan data kelompok. Kemudian data kelompok yang ditampilkan hanya bersifat regional. Yang mana hanya pada wilayah kabupaten/kota wilayah terkait saja. Sedangkan proses data yang terjadi adalah aplikasi web menampilkan semua data RKKA kelompok terpilih untuk dapat dikelola.



Gambar 11. Detail RKKA Kelompok

4. Laporan Data Dasar Kelompok

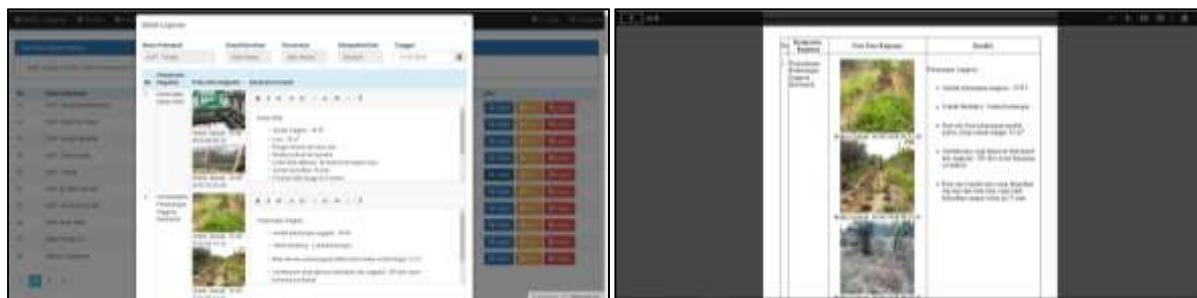
Pada gambar 12 dapat terlihat gambar halaman laporan data dasar kelompok. Pada fitur ini administrator provinsi dapat memantau pembuatan peralatan dasar atau modal dasar lainnya yang dilaksanakan suatu kelompok untuk membangun dasar dari kawasan rumah pangan lestari. Sedangkan proses data yang terjadi adalah aplikasi web melakukan operasi join tabel kelompok dan wilayah untuk menampilkan semua data kelompok dan wilayah kelompok tersebut bertugas agar dapat dipilih untuk melakukan generate laporan.



Gambar 12. Laporan Data Dasar Kelompok

5. Laporan Bulanan Kelompok

Pada gambar 13 dapat terlihat gambar halaman laporan bulanan kelompok. Pada fitur ini administrator provinsi dapat memantau perkembangan dari segi produksi yang mulai dihasilkan dari dasar kawasan rumah pangan lestari yang telah dibangun oleh setiap kelompok pada data dasar sebelumnya. Laporan ini formalnya dilaporkan secara rutin per bulan. Sedangkan proses data yang terjadi adalah aplikasi web melakukan operasi join tabel laporan bulanan, kelompok dan wilayah untuk menampilkan semua data laporan dengan informasi kelompok pemilik laporan dan wilayah kelompok tersebut bertugas agar dapat dipilih untuk melakukan generate laporan.



Gambar 13. Laporan Bulanan Kelompok

6. Laporan Permasalahan Kelompok.

Pada gambar 14 yang merupakan halaman laporan permasalahan kelompok. Pada fitur ini administrator provinsi dapat memantau permasalahan yang dialami kelompok dilapangan setiap saat. Permasalahan ini formalnya adalah permasalahan yang non teknis dan tidak terprediksi. Misalkan dikarenakan bencana alam seperti banjir, kekeringan atau wabah seperti penyakit tanaman dan lainnya. Permasalahan dapat dijelaskan per komponen kegiatan. Sedangkan proses data yang terjadi adalah aplikasi web melakukan operasi join tabel laporan permasalahan, kelompok dan wilayah untuk menampilkan semua data laporan dengan informasi kelompok pemilik laporan dan wilayah kelompok tersebut bertugas agar dapat dipantau melalui laporan permasalahan.



Gambar 14. Laporan Bulanan Kelompok

V. Kesimpulan dan Saran

Berdasarkan studi kasus yang telah dilakukan oleh penulis, akhirnya dapat dihasilkan sistem aplikasi berbasis web yang sudah dapat berjalan dengan normal dan sesuai dengan harapan. Aplikasi web ini menerapkan penggunaan paradigma dasar framework Model View Controller (MVC) yang bekerja dengan cara melakukan segmentasi sektor sistem menjadi 3 bagian utama. Yaitu View sebagai media interaksi visual, Controller sebagai pusat pengendali sistem dan Model sebagai pusat manajemen basis data. Sistem ini menerapkan MVC dikarenakan kemudahannya untuk upgrade fungsi-fungsi baru dan skalabilitas cakupan wilayah KRPL kedepannya. Dari segi maintenance framework MVC juga memberikan kemudahan dibandingkan dengan framework sistem yang biasa. Sistem yang diterapkan ternyata juga telah mampu memberikan kemudahan pemantauan kelompok pelaksana KRPL di Provinsi Kalimantan Tengah oleh Dinas Ketahanan Pangan Provinsi Kalimantan Tengah dan seluruh Kabupaten/Kota di wilayah terkait. Secara umum juga dapat menjadi sumber referensi bagi penelitian-penelitian serupa tentang gambaran bagaimana memetakan dan mendesign, lalu mengimplementasikan sektor-sektor dan proses kerja framework Model View Controller untuk pengembangan aplikasi web khususnya dalam ruang lingkup E-Government atau sistem monitoring.

Penulis juga menyadari bahwa dalam studi kasus yang dilakukan ini mungkin terdapat ketidaksempurnaan yang dilakukan dalam hal perancangan framework atau pemahaman dari paradigma MVC. Sehingga penulis sangat terbuka terhadap masukan kedepannya.

Daftar Rujukan

- Adamsen, C. Q., Møller, A., Alimadadi, S., & Tip, F. (2018). Practical AJAX race detection for JavaScript web applications. *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 38–48.
- Betz, K., Leff, A., & Rayfield, J. T. (2010). *Methods and apparatus for reducing the number of server interactions in network-based applications using a dual-MVC approach*. Google Patents.
- Freeman, A., & Freeman, A. (2014). *Pro asp. net mvc 5 platform*. Springer.
- Kumar, A., Pandey, S. K., Prakash, S., Singh, K. U., Singh, T., & Kumar, G. (2023). Enhancing Web Application Efficiency: Exploring Modern Design Patterns within the MVC Framework. *Proceedings of International Conference on Computational Intelligence and Sustainable Engineering Solution, CISES 2023*, 43–48. <https://doi.org/10.1109/CISES58720.2023.10183582>
- Kumar, N., Zadgaonkar, A. S., & Shukla, A. (2013). Evolving a new software development life cycle model SDLC-2013 with client satisfaction. *International Journal of Soft Computing and Engineering (IJSCE)*, 3(1), 2231–2307.
- Larrea, V., Silveira, M. S., & Da Silva, T. (2024). The use of prototypes as a tool in Agile software development. *Proceedings of the ACM Symposium on Applied Computing*, 842–849. <https://doi.org/10.1145/3605098.3636064>
- Lin, Y. J., & Lin, S. (2013). Research on E-Commerce Software Framework of MVC Mode Based on .NET. *Applied Mechanics and Materials*, 380, 2645–2648.
- Lossius, T., De La Hogue, T., Wolek, N., & Baltazar, P. (2014). Demo: Using Jamama's MVC features to design an audio effect interface. *Proceedings - 40th International Computer Music Conference, ICMC 2014 and 11th Sound and Music Computing Conference, SMC 2014 - Music Technology Meets Philosophy: From Digital Echos to Virtual Ethos*, 1792–1795. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84908876778&partnerID=40&md5=b665e43f814a1cdbc62d64030497d85a>
- Medina-Santiago, A., Cisneros-Gómez, A., Melgar-Paniagua, E. M., Nango-Sólis, G. B., Moreno-López, E. A., Castellanos-Morales, M. E., Cantoral-Díaz, D. B., & Blanco-Gonzalez, L. M. (2017). Web application development by applying the MVC and table data gateway in the annual program budget management system. *International Journal of Advanced Computer Science and Applications*, 8(2).
- Ollsson, T., Toll, D., Wingkvist, A., & Ericsson, M. (2015). Evolution and evaluation of the model-view-controller architecture in games. *2015 IEEE/ACM 4th International Workshop on Games and Software Engineering*, 8–14.
- Pitt, C., & Pitt, C. (2012). CodeIgniter: MVC. *Pro PHP MVC*, 349–365.
- Pop, D.-P., & Altar, A. (2014). Designing an MVC model for rapid web application development. *Procedia Engineering*, 69, 1172–1179.
- Pressman, R. S. (2005). *Software engineering: a practitioner's approach*. Palgrave macmillan.
- Sabale, R. G., & Dani, A. R. (2012). Comparative study of prototype model for software engineering with system development life cycle. *IOSR Journal of Engineering*, 2(7), 21–24.
- Sarker, I. H., & Apu, K. (2014). Mvc architecture driven design and implementation of java framework for developing desktop application. *International Journal of Hybrid Information Technology*, 7(5), 317–322.
- Yu, F., & Enste, B. (2019). Smart Prototyping. In *The Routledge Companion to Innovation Management* (pp. 237–265). <https://doi.org/10.4324/9781315276670-12>