

SISTEM INFORMASI MANAJEMEN PERMINTAAN PERUBAHAN SYSTEM APPLICATION AND PRODUCT (SAP) PADA PT. KRAKATAU INFORMATION TECHNOLOGY

Teguh Sutopo¹, Susy Katarina Sianturi², Gustina³, Munaji⁴

Program Studi S1 Teknik Informatika

Sekolah Tinggi Teknologi Ilmu Komputer Insan Unggul

Jl. Bojonegara No.45, Panggung Rawi, Kec. Jombang, Kota Cilegon, Banten 42412

Email: teguhsutopo@insan-unggul.ac.id, susykatarina@insan-unggul.ac.id,
gustina0881@gmail.com, munaji.id@gmail.com

ABSTRAK

Penelitian ini dilatarbelakangi adanya perubahan dalam pengembangan sistem aplikasi SAP yang merupakan proses yang sering kali terjadi dan tidak dapat dihindari. Perubahan dapat terjadi dalam fase pengembangan ataupun implementasi SAP. Perubahan yang terjadi pada fase pengembangan relatif lebih mudah ditangani dibandingkan yang terjadi saat implementasi. Namun demikian, untuk menjaga konsistensi sistem, setiap perubahan didokumentasikan dan dikendalikan. Permintaan perubahan biasanya dikarenakan laporan masalah yang mengidentifikasi *bug* yang harus diperbaiki, permintaan peningkatan sistem dari pengguna SAP dan integrasi SAP dengan sistem lain.. Tujuan penelitian ini adalah untuk rakayasa *help desk* yang merupakan sebuah sistem informasi manajemen permintaan perubahan aplikasi SAP. Jenis penelitian yang digunakan yaitu *Research & Development*. Pada tahap analisis, data dikumpulkan dengan studi literatur melalui buku-buku dan observasi lapangan. Pada tahap pengembangan aplikasi menggunakan paradigma *waterfall* dan dikembangkan menggunakan PHP framework Laravel. Hasil dari sistem ini diharapkan dapat mengelola permintaan perubahan dengan efektif sesuai dengan kebutuhan penggunaanya.

Kata Kunci: pengembangan sistem, permintaan perubahan, SAP, laravel..

1. Pendahuluan

Pemberian layanan kepada pelanggan difungsikan untuk memuaskan pelanggan dalam menggunakan layanan yang ditawarkan. Pada tahap selanjutnya diharapkan dapat meningkatkan jumlah pelanggan dan pengguna

layanan serta mempertahankan pelanggan yang sudah ada. Peningkatan kualitas layanan Teknologi Informasi (TI) menjadi sebuah kebutuhan dan sebuah tuntutan di setiap instansi perusahaan mengingat peran Teknologi Informasi dan Komunikasi (TIK) yang semakin tinggi dalam peningkatan kualitas layanan TI untuk memenuhi *Good Governance*.

PT Krakatau Information Technology adalah Implementor SAP ERP yang menyediakan layanan *Software as a Service* (SaaS) Aplikasi SAP ERP, adapun *customer* yang menggunakan layanan tersebut adalah Krakatau Steel Group yaitu Krakatau Sarana Infrastruktur, Krakatau Sarana Properti, Krakatau Tirta Industri, Krakatau Bandar Samudra, Krakatau Daya Listrik, dan Krakatau Information Technology sendiri. Divisi SAP adalah divisi yang bertanggung jawab untuk melayani kebutuhan terkait dengan permintaan pengembangan sistem, pembuatan program aplikasi, dan permintaan perubahan program ketika aplikasi SAP sudah di implementasikan. Divisi SAP mempunyai beberapa fungsional, diantaranya *FI (Financial Accounting)*, *CO (Controlling)*, *MM (Material Management)*, *SD (Sales and Distribution)*, *QM (Quality Management)*, *(PM) Plant Maintenance*, *(HCM) Human Capital Management*, *ABAP (Advanced Business Application Programming)* Programmer dan SAP *System Administrator*. Secara umum tahapan-tahapan implementasi SAP ERP dimulai dari *Project Preparation*, *Business Blueprint*, *realization*, *Final Preparation*, *Go Live* dan *Production Support*.

Perubahan dalam pengembangan sistem aplikasi SAP merupakan proses yang sering kali terjadi dan tidak dapat dihindari. Perubahan dapat terjadi dalam fase pengembangan ataupun implementasi SAP. Perubahan yang terjadi pada fase pengembangan relatif lebih mudah ditangani dibandingkan yang terjadi saat implementasi. Namun demikian, untuk menjaga konsistensi sistem, setiap perubahan didokumentasikan dan dikendalikan. Permintaan perubahan

biasanya dikarenakan ditemukannya *bug* yang harus diperbaiki, permintaan peningkatan sistem dari pengguna SAP dan integrasi SAP dengan sistem lain.

Pencatatan permintaan perubahan aplikasi SAP harus dilakukan dengan baik, jika tidak tercatat dengan baik, akan terjadi sulitnya kontrol dalam pengerjaannya. *Programmer / system analyst* sering melupakan tugas yang seharusnya dikerjakan dan harus selesai pada waktu tertentu. Penugasan dilakukan dengan perintah langsung tanpa ada dokumentasi dan catatan kapan waktu pekerjaan itu harus selesai, begitu juga dengan proses *monitoring* permintaan perubahan tersebut tidak dapat di monitor secara *realtime*. Karena masalah tersebut, maka dipandang perlu adanya aplikasi sistem informasi manajemen permintaan perubahan aplikasi SAP sebagai solusi dalam bentuk *help desk*.

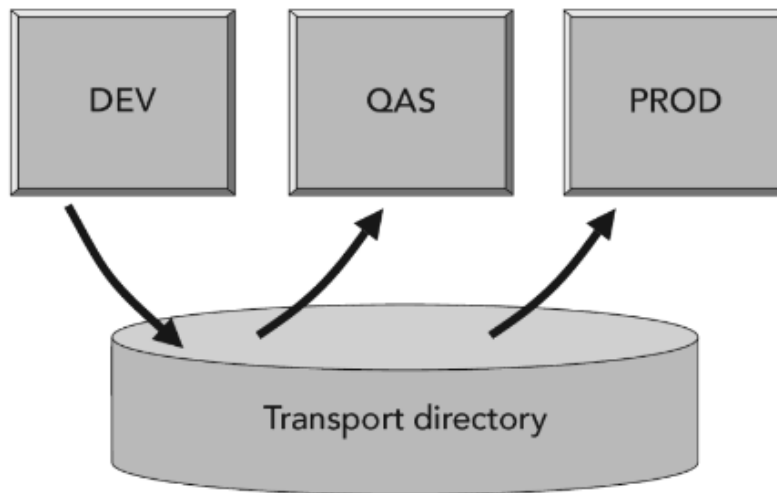
2. Landasan Teori

2.1. SAP System Landscape

System landscape SAP didefinisikan sebagai pengaturan server SAP. Idealnya, di lingkungan SAP, terdapat tiga sistem yang terdiri dari *Development System* (DEV), *Quality Assurance System* (QAS) dan *Production System* (PROD).

Sistem *Development* (DEV) digunakan untuk mengembangkan pengaturan konfigurasi sistem, serta perbaikan khusus menggunakan *abap code*. Perubahan ini secara otomatis direkam dalam direktori transport (*Transport directory*), yang merupakan lokasi data khusus pada sistem DEV. Perubahan tercatat dalam sistem DEV diimpor ke sistem *Quality Assurance* (QAS), tempat konfigurasi dan perbaikan diuji untuk memastikan konfigurasi dan perbaikan tersebut berfungsi dengan baik. jika ada koreksi yang diperlukan, koreksi tersebut dibuat

di sistem DEV dan kemudian di-*transport* lagi ke sistem QAS untuk pengujian. Setelah pengaturan konfigurasi dan program ABAP lulus pengujian di sistem QAS, semua pengaturan, program, dan perubahan di-*transport* ke sistem *Production* (PROD), yaitu sistem yang akan digunakan perusahaan untuk menjalankan proses bisnisnya (Monk, 2013).



Gambar 2.1. SAP System Landscape

Dari penjelasan tersebut, dapat disimpulkan *transport* adalah istilah yang digunakan untuk mengimpor data perubahan dari satu sistem SAP ke sistem lain yang terhubung. Setiap *transport* berisi satu atau lebih perubahan.

2.2. Framework Laravel

Laravel adalah salah satu *framework* PHP terbaik yang dikembangkan oleh Taylor Otwell, proyek laravel dimulai pada April 2011. Laravel adalah sebuah *framework* PHP yang dirilis dibawah lisensi MIT dan dibangun dengan konsep MVC (*Model View Controller*).

Laravel adalah pengembangan *website* berbasis MVP yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan

mengurangi biaya pengembangan awal dan biaya pemeliharaan, serta untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, jelas, dan menghemat waktu (Yudhanto, 2019).

2.3. Basis Data

Basis data (Bahasa Inggris: *database*) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (*query*) basis data disebut manajemen basis data (*Database Management System*, DBMS).

Sistem basis data dipelajari dalam ilmu informasi. Jenis *relational database* adalah jenis basis data yang organisasinya berdasarkan pemodelan data rasional, Hampir semua *relational database* saat ini menggunakan SQL sebagai bahasa *query* dan pemeliharaan basis data. Beberapa contoh *rational database* yang terkenal seperti SQL, PostgreSQL, MongoDB, MariaDB, Oracle Database, SAP HANA, MemSQL, FireBird dan Interbase (Anamisa, 2022).

2.4. Entity Relationship Diagram (ERD)

ERD atau *Entity Relationship Diagram* adalah suatu bentuk diagram yang menjelaskan hubungan antar objek-objek data yang mempunyai hubungan antar relasi. ERD digunakan untuk menyusun struktur data dan hubungan antar data, dan untuk menggambarkan digunakan notasi, simbol, bagan dan lain sebagainya. ERD atau ERM berfungsi sebagai bentuk pemodelan basis data yang kemudian dapat dikembangkan dalam berbagai *project* sistem informasi, contohnya rekayasa *software* (Anamisa, 2022).

Ada sejumlah cara dalam mempresentasikan model data menurut (Fathansyah, 2018), secara umum dikelompokkan ke dalam dua kelompok:

- a. Model logic data berdasarkan OBJEK (*Object Based Logical Model*) yang terdiri dari:
 1. Model Keterhubungan Entitas (*Entity Relationship Model*)
 2. Model Berorientasi Objek (*Object Oriented Model*)
 3. Model Data Semantik (*Semantic Data Model*)
 4. Model Data Fungsional (*Functional Data Model*)
- b. Model logic data berdasarkan Record (*Record Based logical Model*) yang terdiri dari:
 1. Model Relasional (*Relational Model*)
 2. Model Hierarkis (*Hierarchial Model*)
 3. Model Jaringan (*Network Model*)

Proses pembuatan *database* dimulai dengan membuat desain *database*, ERD merupakan salah satu model data yang dapat menggambarkan desain *database* dengan mudah.

3. Metodologi Penelitian

3.1. Metode Pengembangan Sistem

Penelitian ini menggunakan model pengembangan *Waterfall* pada tahapan kegiatannya yang dapat dijelaskan sebagai berikut:

- a. Analisis Spesifikasi Kebutuhan, dimulai dari pengumpulan data melalui *study* pustaka dan menggali informasi sebanyak-banyaknya tentang kebutuhan pengguna yang merupakan kebutuhan fungsional dan non-fungsional sistem.
- b. Perancangan, dibagi menjadi 3 bagian yaitu perancangan model sistem dan perancangan antarmuka-kegrafisan.

- c. Pembuatan Kode Program, menggunakan bahasa pemrograman yang berbasis *Web* yaitu PHP.
- d. Pengujian, dilakukan dengan menggunakan Pengujian *Black Box*, dilakukan untuk memastikan perancangan dan fungsional sistem berjalan sesuai dengan kebutuhan.

3.2. Metode dan Instrumen Pengumpulan Data

Metode pengumpulan data yang digunakan adalah sbb.:

a. Observasi (Pengamatan)

Dilakukan dengan pengamatan langsung dari lokasi yaitu mengamati Divisi SAP PT Krakatau Information Technology dalam menerima permintaan perubahan aplikasi SAP dan menindaklanjuti permintaan perubahan tersebut. Tujuan observasi adalah mendeskripsikan aktivitas-aktivitas yang berlangsung, siapa saja yang terlibat dalam aktivitas-aktivitas tersebut dan memetakan alur sistem informasi sesuai dengan kebutuhan.

b. Studi Pustaka

Yaitu pencarian data dengan membaca dan mempelajari melalui media buku-buku dan sumber-sumber lainnya yang berkaitan.

3.3. Analisa Kebutuhan

Tahapan ini adalah langkah untuk menganalisa kebutuhan pengguna dengan cara berkonsultasi dengan pengguna mengenai kendala, layanan dan tujuan dari sistem yang akan dibangun, dari studi lapangan yaitu dengan cara observasi, melihat langsung proses permintaan perubahan pada aplikasi SAP yang dilakukan oleh pengguna. Juga dilakukan studi literatur melalui buku-buku dan penelitian terdahulu sebagai teori untuk mendapatkan data-data yang dibutuhkan dan mendukung dalam membangun sistem informasi yang

dibangun, sehingga didapatkan analisa kebutuhan fungsional yang harus dimiliki oleh sistem sesuai dengan kebutuhan pengguna.

3.3.1. Kebutuhan Fungsional

Kebutuhan fungsional berisi tentang layanan apa saja yang dilakukan oleh sistem. Diharapkan sistem yang dibangun dapat melakukan fungsi :

Tabel 3.1. Tabel Fungsional Sistem

No.	Fungsional	Keterangan
1.	<i>Login</i>	Sistem mempunyai autentikasi <i>login</i> sebagai keamanan sistem
2.	Master Data	Admin dapat mengelola master data sistem
3.	Permintaan Perubahan	User dapat membuat permintaan perubahan
4.	Perubahan Status	Semua pengguna dapat dapat melakukan perubahan status sesuai dengan aksesnya.
5.	<i>Upload</i>	<i>Administrator</i> , <i>Functional</i> dan Abaper dapat melampirkan dokumen
6.	<i>Download</i>	Semua pengguna dapat mengunduh dokumen
7.	Log	Sistem dapat menyimpan histori perubahan status permintaan perubahan
8.	Notifikasi	Sistem dapat mengirimkan notifikasi saat pembuatan pengguna baru, membuat permintaan perubahan baru dan perubahan status.

3.3.2. Kebutuhan Non Fungsional

a. Kebutuhan Perangkat Keras Komputer

Perangkat keras yang dibutuhkan untuk mengimplementasikan sistem informasi ini adalah sebagai berikut :

Tabel 3.2. Kebutuhan Perangkat Keras Komputer

No	Nama Perangkat Keras	Jenis Perangkat Keras
1.	Processor	2 CPU Intel Skylake, Broadwell, and Haswell, AMD EPYC Rome and EPYC Milan
2.	Ram	1 Gb
4.	Disk	25 Gb

b. Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan untuk mengimplementasikan sistem informasi ini adalah sebagai berikut :

Tabel 3.3. Kebutuhan Perangkat Lunak

No	Jenis Perangkat Lunak	Nama Perangkat Lunak
1.	Sistem Operasi	Debian
2	<i>Web Server</i>	Apache
3.	Bahasa Pemrograman	PHP > 8.0
4.	<i>Database</i>	MariaDB
5.	<i>E-mail</i> Notifikasi	smtp google
6.	Repositori	Git
7.	<i>Depedency Manager</i>	Composer

c. Kebutuhan Infrastruktur

Infrastruktur yang dibutuhkan untuk mengimplementasikan sistem informasi ini adalah sebagai berikut :

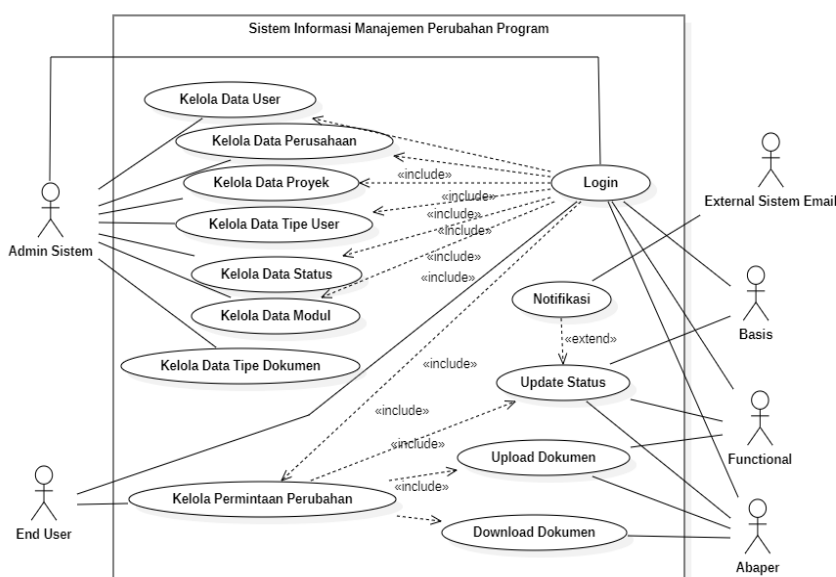
Tabel 3.4. Kebutuhan Infrastruktur

No	Jenis Infrastruktur	Keterangan
1.	Web Hosting	Google Cloud Platform
2.	Address	Alamat IP Public

3.4. Perancangan Model Sistem

A. Use Case Diagram

Berikut ini adalah *use case diagram* pada sistem informasi manajemen perubahan program aplikasi SAP ini.



Gambar 3.1. Use Case Diagram Sistem Usulan

Pada gambar di atas sebelum sistem digunakan oleh *user* lainnya, Admin sistem melakukan input master data terkait sistem seperti master data *user*, master data perusahaan, master data proyek, master data tipe *user*, master

data status *change request*, master data modul dan master data tipe dokumen. *End User* atau yang meminta perubahan program mengkases *login* yang terhubung dengan pengelolaan permintaan perubahan yang digunakan untuk membuat permintaan perubahan dan memperbaharui permintaan perubahan. *Functional* dan Abaper mengakses *login* untuk melakukan perubahan status, dan *upload* dan *download* dokumen terkait permintaan perubahan dan basis hanya melakukan perubahan status pada permintaan perubahan.

a. Definisi Aktor

Definisi aktor mendeskripsikan peranan aktor yang ada pada sistem. Definisi aktor pada sistem ini dapat dilihat pada Tabel 3.14 sebagai berikut.

Tabel 3.5. Tabel Definisi Aktor *Use Case* Sistem Usulan

Aktor	Definisi
<i>End User</i>	Orang yang menggunakan aplikasi untuk membuat permintaan perubahan aplikasi SAP dan memantau permintaan perubahan aplikasi SAP tersebut.
<i>Functional</i>	Orang yang menggunakan aplikasi web untuk upload <i>functional specification</i> , <i>upload testing</i> dokumen dan mengubah status ke tahapan selanjutnya atau sebaliknya.
Abaper	Orang yang menggunakan aplikasi web untuk melakukan perubahan status ke tahap selanjutnya, mendownload <i>functional specification</i> dan <i>upload technical document</i> .
Basis	Orang yang menggunakan aplikasi web untuk Mengubah status ke status status ke tahap selanjutnya
Admin Sistem	Orang yang menggunakan aplikasi web untuk mengelola master data user, master data perusahaan, master data proyek, master data tipe <i>user</i> , master data status, data modul, master data tipe lampiran.
Eksternal Sistem Email	Sistem diluar sistem yang mengirimkan email notifikasi saat pembuatan pengguna baru, pembuatan permintaan perubahan baru dan perubahan status permintaan perubahan.

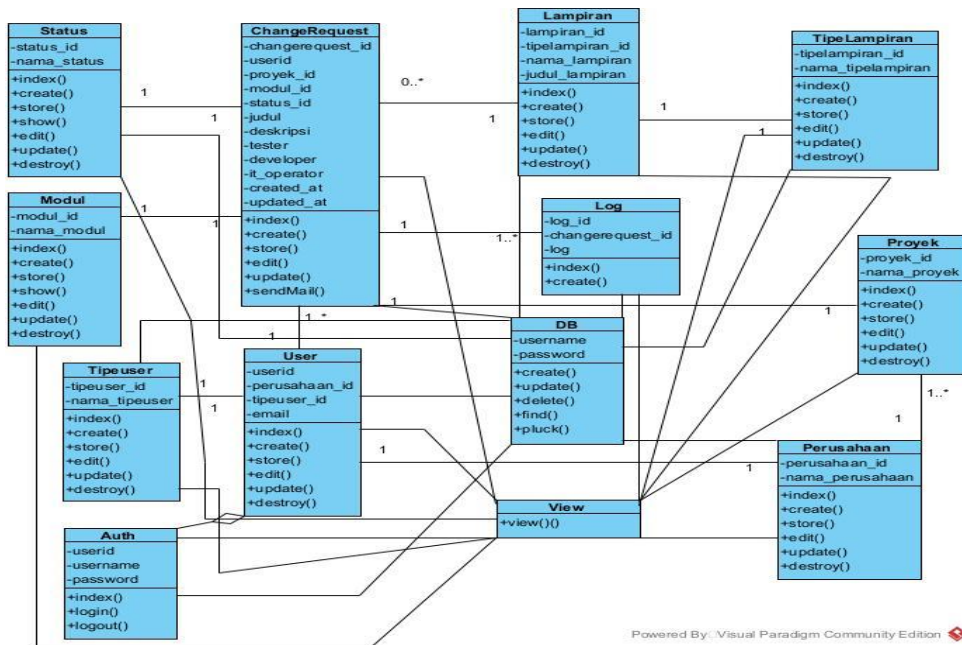
b. Definisi Use Case

Definisi *use case* mendeskripsikan setiap *use case* yang terdapat pada *use case diagram*. Pada tabel berikut merupakan definisi *use case* pada sistem yang dibuat.

Tabel 3.6. Definisi *Use Case* Sistem Usulan

No.	Use Case	Deskripsi
1	<i>Login</i>	Merupakan proses masuk ke aplikasi.
2	Kelola Data <i>User</i>	Merupakan proses mengelola master data pengguna.
3	Kelola Data Perusahaan	Merupakan proses mengelola master data perusahaan.
4	Kelola Data Proyek	Merupakan proses mengelola master data proyek.
5	Kelola Data Tipe Pengguna	Merupakan proses mengelola master data tipe pengguna.
6	Kelola Data Status	Merupakan proses mengelola master data status.
7	Kelola Data Modul	Merupakan proses mengelola master data modul.
8	Kelola Data Tipe Dokumen	Merupakan proses mengelola master data tipe dokumen.
9	Kelola Permintaan Perubahan	Merupakan proses membuat permintaan perubahan
10	<i>Update Status</i>	Merupakan proses untuk merubah status permintaan perubahan ke tahapan selanjutnya atau sebaliknya.
11	<i>Upload</i> Dokumen	Merupakan proses melampirkan dokumen.
12	<i>Download</i> Dokumen	Merupakan proses mengunduh dokumen.
13	Notifikasi	Merupakan proses mengirimkan notifikasi terkait perubahan yang terjadi pada permintaan perubahan program..

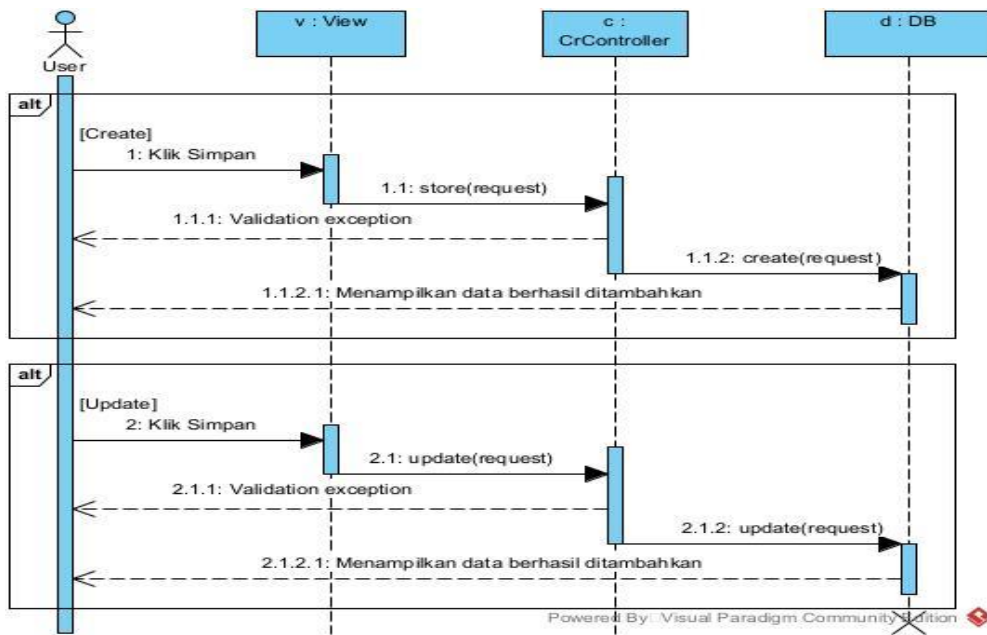
B. Class Diagram



Gambar 3.2. *Class Diagram* Sistem Usulan

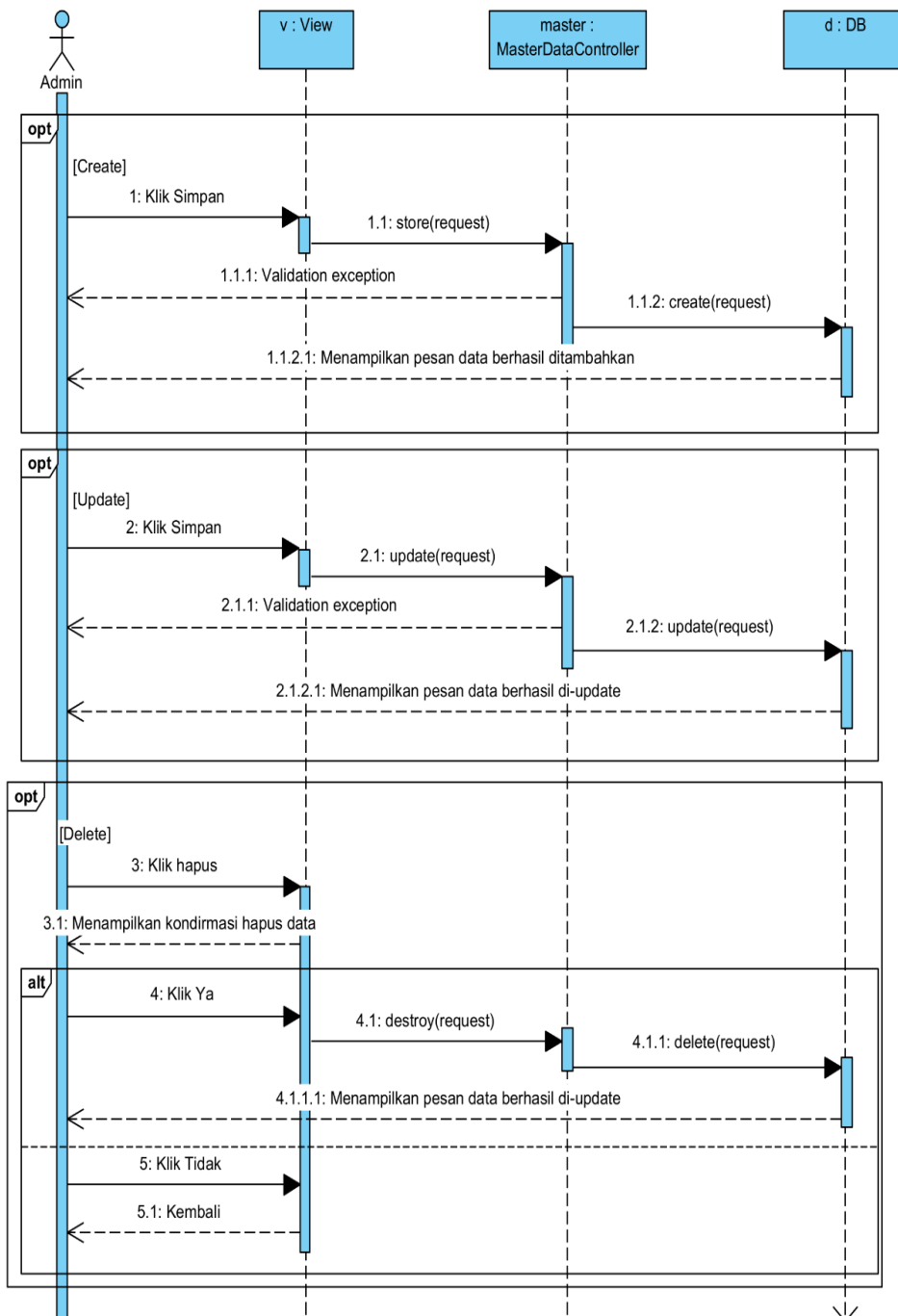
C. Sequence Diagram

a. Sequence Diagram Kelola Permintaan Perubahan



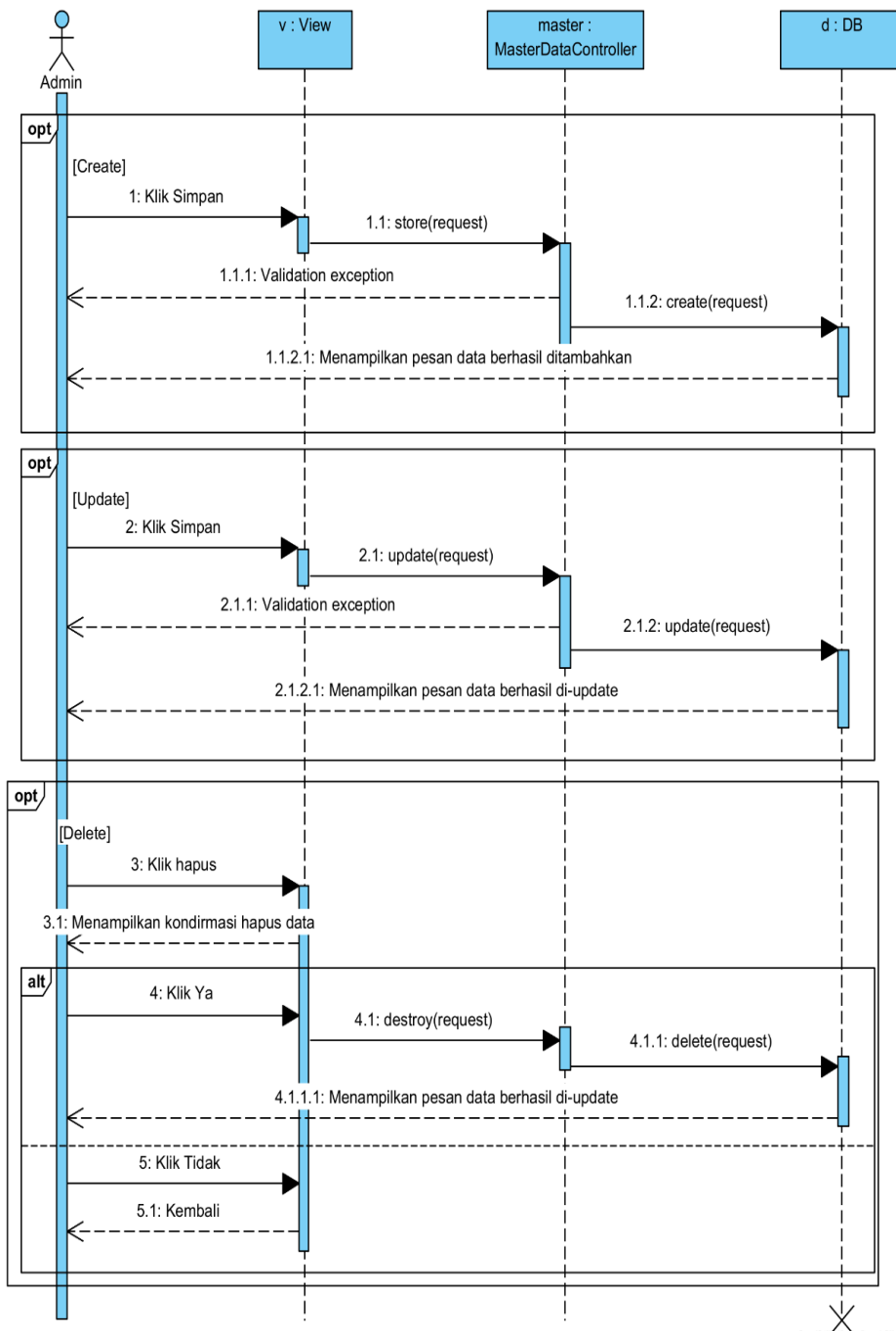
Gambar 3.3. *Sequence Diagram* Kelola Permintaan Perubahan

b. *Sequence Diagram* Kelola Data Pengguna



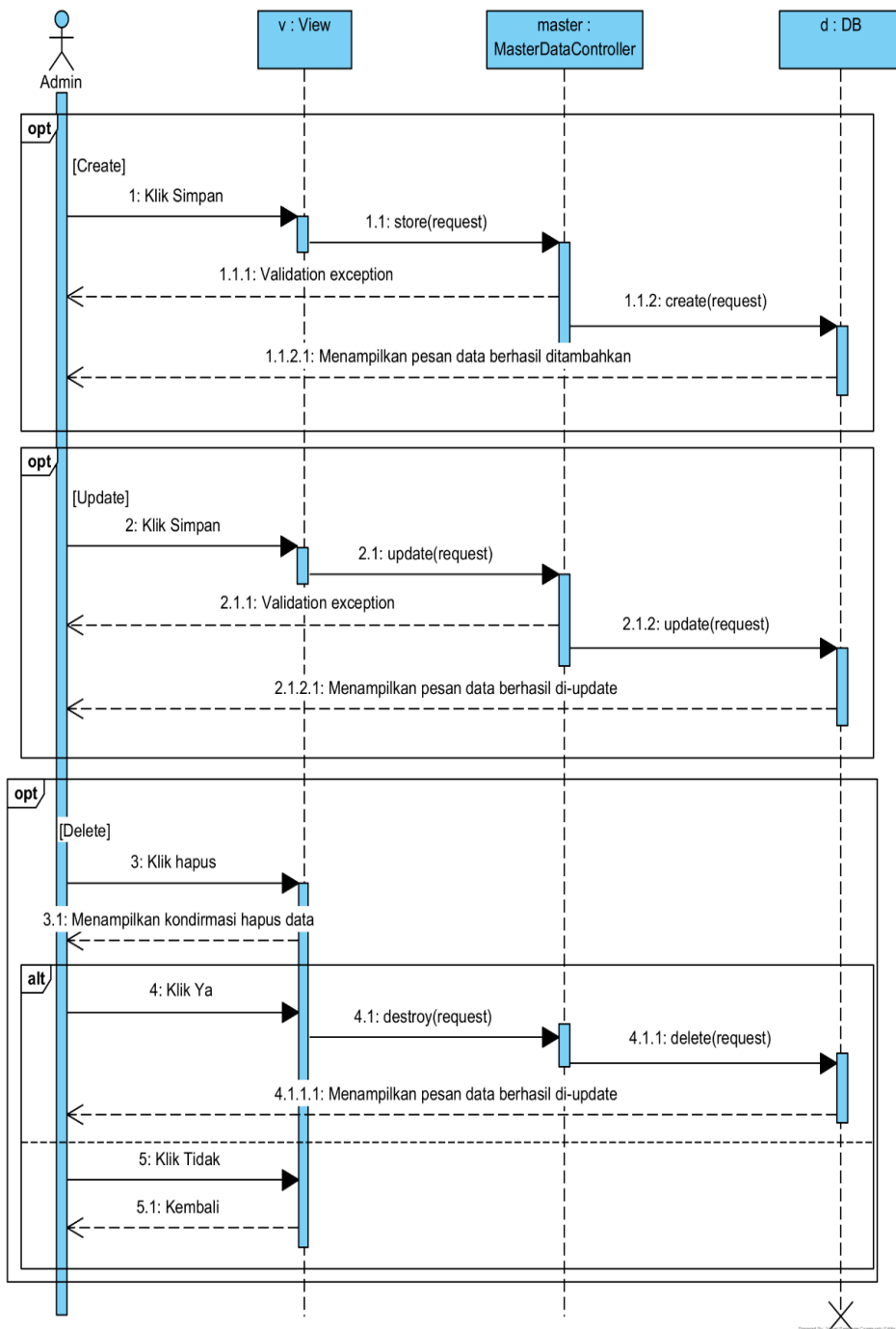
Gambar 3.4. *Sequence Diagram* Kelola Data Pengguna

c. *Sequence Diagram* Kelola Data Perusahaan



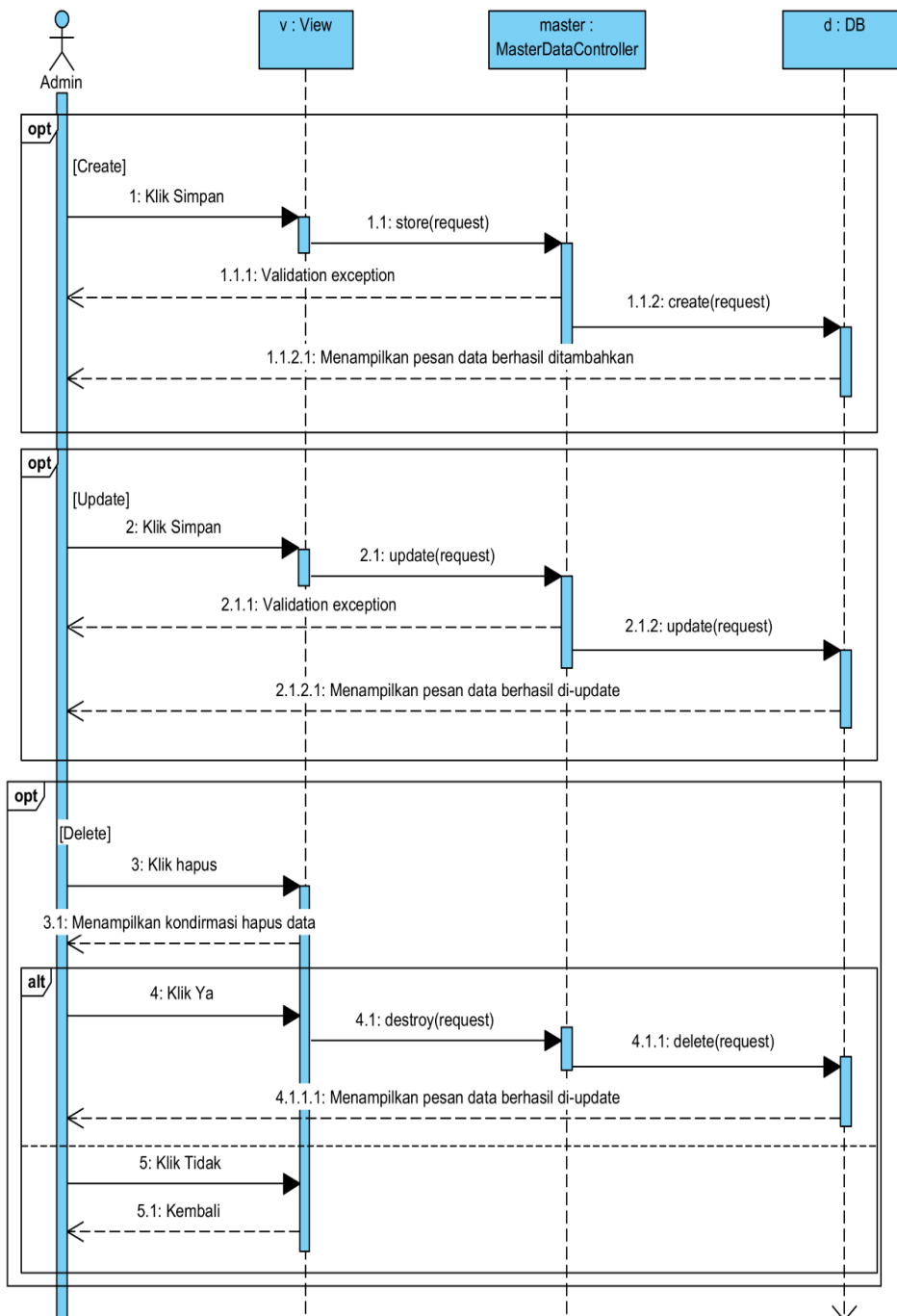
Gambar 3.5. *Sequence Diagram* Kelola Data Perusahaan

d. *Sequence Diagram* Kelola Data Proyek



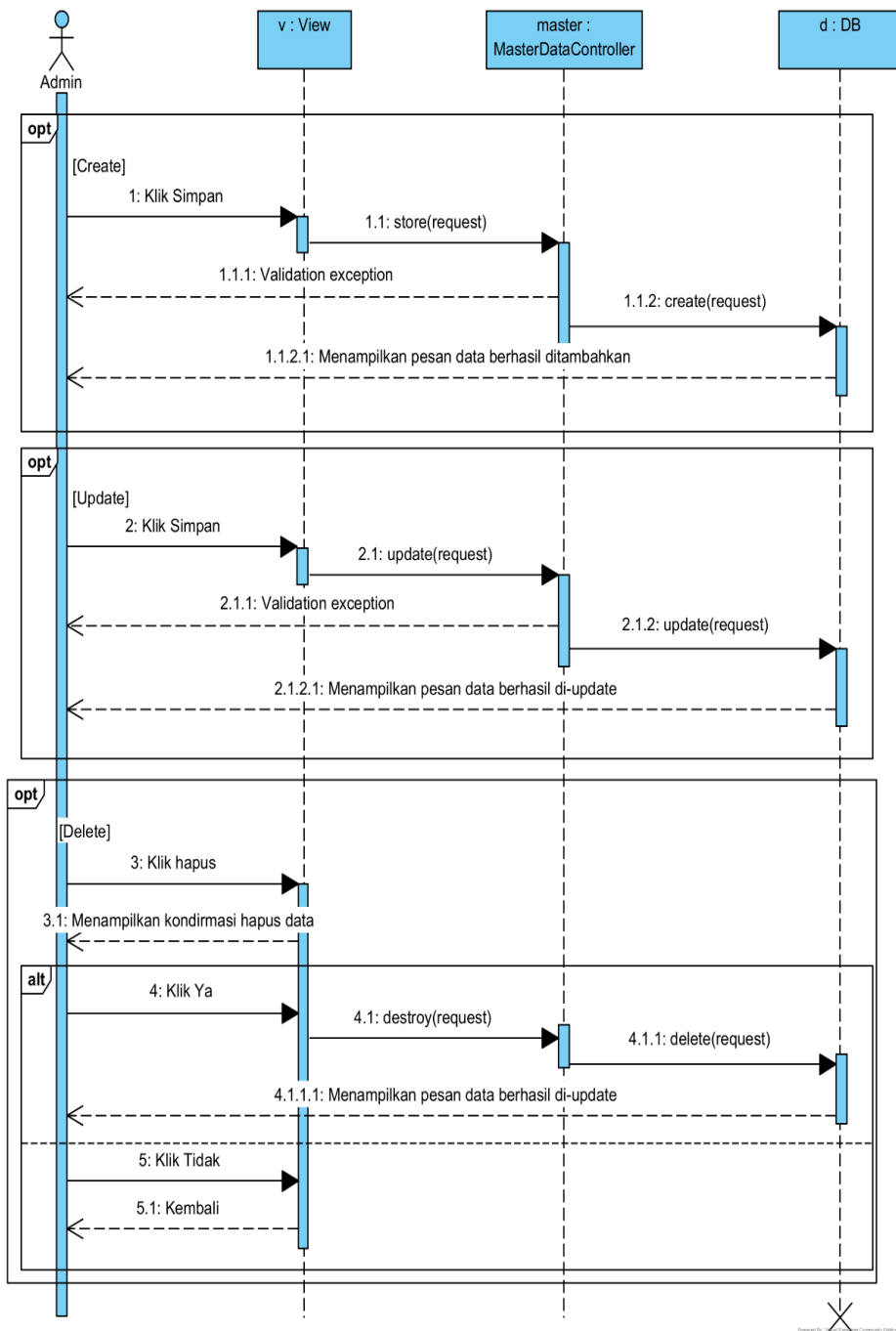
Gambar 3.6. *Sequence Diagram* Kelola Data Proyek

e. *Sequence Diagram* Kelola Data Status



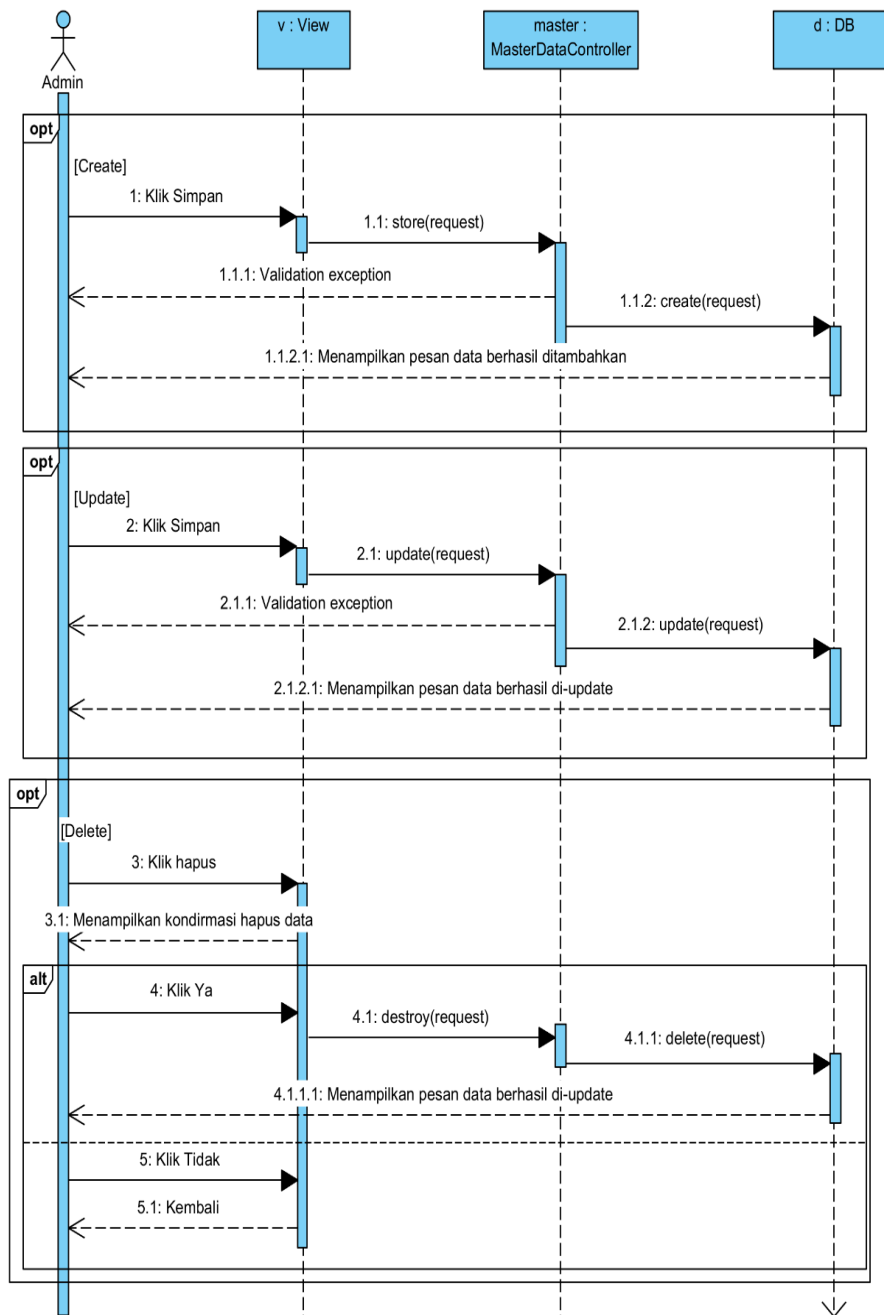
Gambar 3.7. *Sequence Diagram* Kelola Data Status

f. *Sequence Diagram* Kelola Data Modul



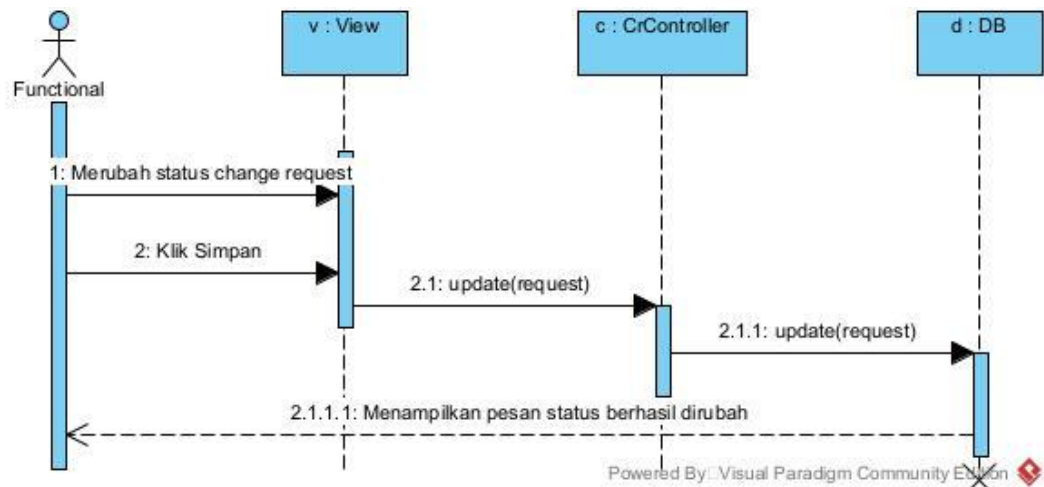
Gambar 3.8. *Sequence Diagram* Kelola Data Modul

g. *Sequence Diagram* Kelola Data Tipe Dokumen



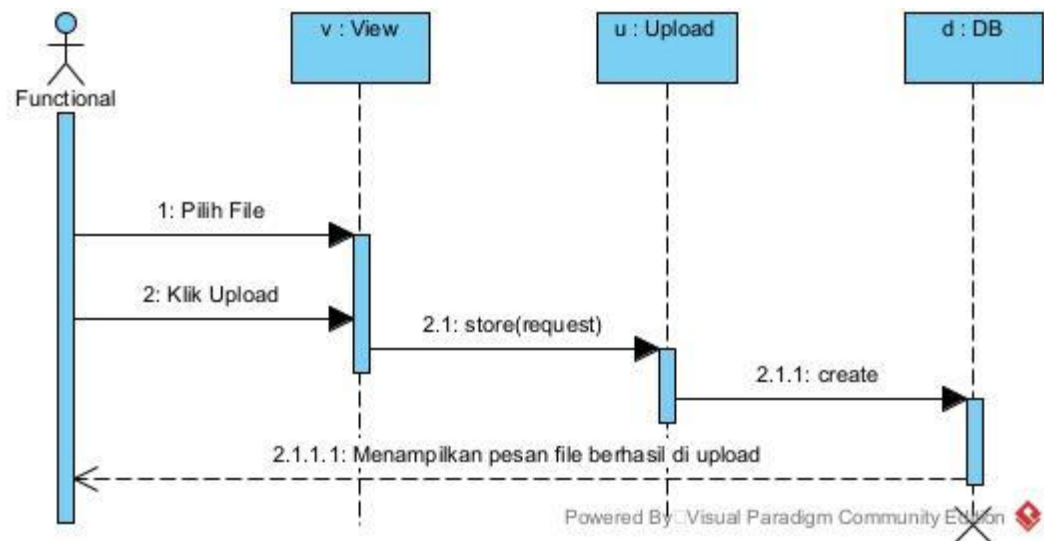
Gambar 3.9. *Sequence Diagram* Kelola Data Tipe Dokumen

h. *Sequence Diagram Update Status*



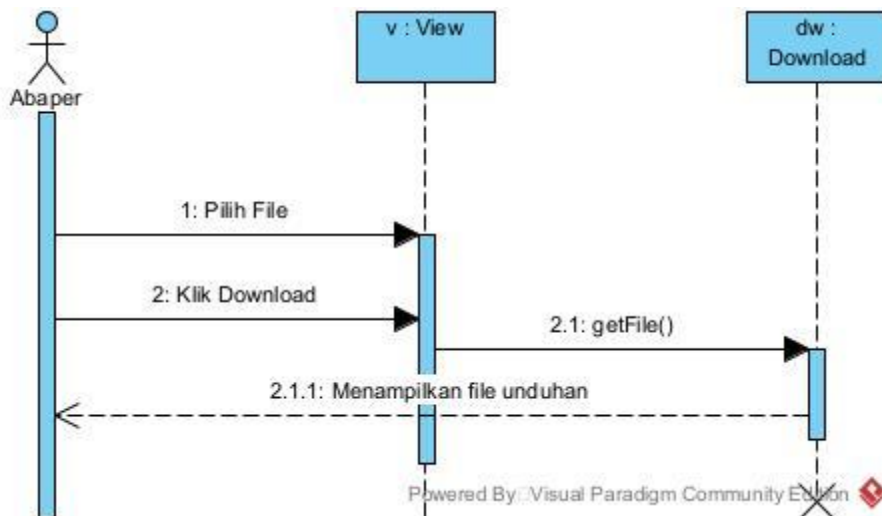
Gambar 3.10. *Sequence Diagram Update Status*

i. *Sequence Diagram Upload Dokumen*



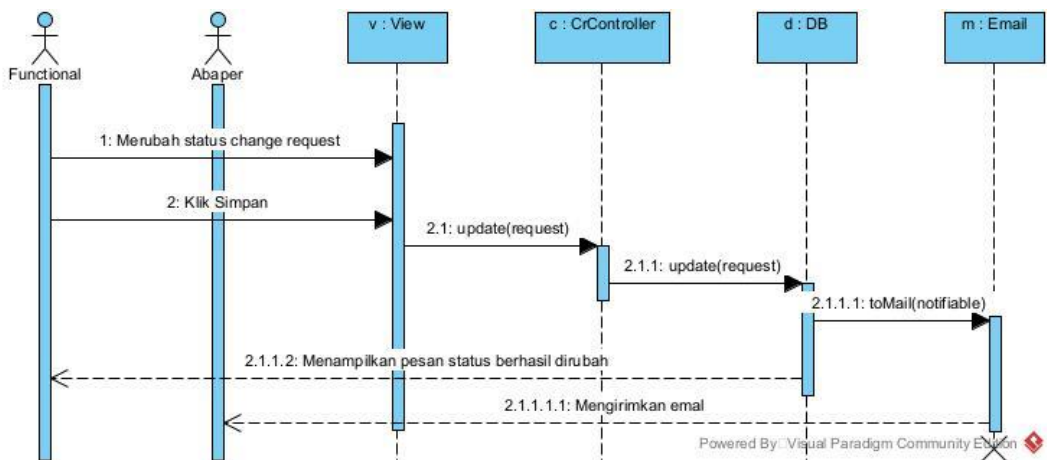
Gambar 3.11. *Sequence Diagram Upload Dokumen*

j. *Sequence Diagram Download Dokumen*



Gambar 3.12. *Sequence Diagram Download Dokumen*

k. *Sequence Diagram Notifikasi*



Gambar 3.13. *Sequence Diagram Notifikasi*

4. Hasil Dan Pembahasan

4.1. Deskripsi Hasil

Pada hasil pengujian terdapat tabel pengujian yang berfungsi untuk menyimpulkan apakah sistem berhasil dan sesuai dengan pengujian atau tidak. Pengujian menggunakan metode *Black Box* menggunakan teknik *Equivalence Partitioning*. Adapun hasil pengujian ditunjukkan pada Tabel 4.4 berikut ini.

Tabel 4.1. Tabel Pengujian *Black Box*

Nilai	Skenario Uji	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
Login				
<i>Valid</i>	Memasukan <i>username</i> dan <i>password</i> yang sesuai	Sistem akan menampilkan halaman <i>dashboard</i>	Berhasil masuk ke menu <i>dashboard</i>	Berhasil
<i>Invalid</i>	Memasukan <i>username</i> dan <i>password</i> yang tidak sesuai	Sistem akan menampilkan kesalahan validasi	Menampilkan kesalahan validasi " <i>username</i> atau <i>password</i> salah"	Berhasil
Tambah Data Tipe Pengguna				
<i>Valid</i>	Mengisi semua <i>field</i> pada <i>form</i> tambah data tipe pengguna	Sistem akan menampilkan peringatan berhasil	Menampilkan pesan "data berhasil ditambahkan"	Berhasil
<i>Invalid</i>	Mengosongkan semua <i>field</i> pada <i>form</i> tambah data tipe pengguna	Sistem akan menampilkan peringatan kesalahan	Menampilkan kesalahan validasi " <i>field</i> wajib di isi"	Berhasil
Tambah Data Perusahaan				
<i>Valid</i>	Mengisi semua <i>field</i> pada <i>form</i> tambah data perusahaan	Sistem akan menampilkan peringatan berhasil	Menampilkan pesan "data berhasil ditambahkan"	Berhasil
<i>Invalid</i>	Mengosongkan semua <i>field</i> pada <i>form</i>	Sistem akan menampilkan	Menampilkan kesalahan validasi " <i>field</i> wajib di isi"	Berhasil

	tambah data perusahaan	peringatan kesalahan		
Tambah Data Modul				
<i>Valid</i>	Mengisi semua <i>field</i> pada <i>form</i> tambah data modul	Sistem akan menampilkan peringatan berhasil	Menampilkan pesan "data berhasil ditambahkan"	Berhasil
<i>Invalid</i>	Mengosongkan semua <i>field</i> pada <i>form</i> tambah data modul	Sistem akan menampilkan peringatan kesalahan	Menampilkan kesalahan validasi " <i>field</i> wajib di isi"	Berhasil
Tambah Data Status				
<i>Valid</i>	Mengisi semua <i>field</i> pada <i>form</i> tambah data status	Sistem akan menampilkan peringatan berhasil	Menampilkan pesan "data berhasil ditambahkan"	Berhasil
<i>Invalid</i>	Mengosongkan semua <i>field</i> pada <i>form</i> tambah data status	Sistem akan menampilkan peringatan kesalahan	Menampilkan kesalahan validasi " <i>field</i> wajib di isi"	Berhasil
Tambah Data Proyek				
<i>Valid</i>	Mengisi semua <i>field</i> pada <i>form</i> tambah data proyek	Sistem akan menampilkan peringatan berhasil	Menampilkan pesan "data berhasil ditambahkan"	Berhasil
<i>Invalid</i>	Mengosongkan semua <i>field</i> pada <i>form</i> tambah data proyek	Sistem akan menampilkan peringatan kesalahan	Menampilkan kesalahan validasi " <i>field</i> wajib di isi"	Berhasil
Tambah Data Tipe Lampiran				
<i>Valid</i>	Mengisi semua <i>field</i> pada <i>form</i> tambah data tipe lampiran	Sistem akan menampilkan peringatan berhasil	Menampilkan pesan "data berhasil ditambahkan"	Berhasil
<i>Invalid</i>	Mengosongkan semua <i>field</i> pada <i>form</i>	Sistem akan menampilkan	Menampilkan kesalahan validasi " <i>field</i> wajib di isi"	Berhasil

	tambah data tipe lampiran	peringatan kesalahan		
Tambah Data Pengguna				
<i>Valid</i>	Mengisi semua <i>field</i> pada <i>form</i> tambah data pengguna	Sistem akan menampilkan peringatan berhasil	Menampilkan pesan "data berhasil ditambahkan"	Berhasil
<i>Invalid</i>	Mengosongkan semua <i>field</i> pada <i>form</i> tambah data pengguna	Sistem akan menampilkan peringatan kesalahan	Menampilkan kesalahan validasi " <i>field</i> wajib di isi"	Berhasil
Tambah Data Permintaan Perubahan				
<i>Valid</i>	Mengisi semua <i>field</i> pada <i>form</i> tambah data permintaan perubahan	Sistem akan menampilkan peringatan berhasil	Menampilkan pesan "data berhasil ditambahkan"	Berhasil
<i>Invalid</i>	Mengosongkan semua <i>field</i> pada <i>form</i> tambah data permintaan perubahan	Sistem akan menampilkan peringatan kesalahan	Menampilkan kesalahan validasi " <i>field</i> wajib di isi"	Berhasil
Upload Dokumen				
<i>Valid</i>	Mengisi semua <i>field</i> pada <i>form</i> <i>upload</i> dokumen	Sistem akan menampilkan peringatan berhasil	Menampilkan pesan "data berhasil ditambahkan"	Berhasil
<i>Invalid</i>	Mengosongkan semua <i>field</i> pada <i>form</i> <i>upload</i> dokumen	Sistem akan menampilkan peringatan kesalahan	Menampilkan kesalahan validasi " <i>field</i> wajib di isi"	Berhasil
Perubahan Status Permintaan Perubahan				
<i>Valid</i>	Pengguna sesuai dengan akses perubahan status	Sistem akan merubah status permintaan perubahan	Menampilkan perubahan status dan mengirimkan notifikasi	Berhasil

<i>Invalid</i>	Pengguna tidak sesuai dengan akses perubahan status	Sistem akan menampilkan peringatan kesalahan	Menampilkan kesalahan validasi	Berhasil
----------------	---	--	--------------------------------	----------

5. Kesimpulan

Dari hasil dan pembahasan yang telah dijelaskan pada bab-bab sebelumnya, dapat disimpulkan bahwa:

1. Sistem informasi manajemen permintaan perubahan aplikasi SAP pada PT Krakatau Information Technology berbasis *website* yang dapat digunakan untuk menangani pembuatan permintaan perubahan dan sekaligus memantau pekerjaan yang berkaitan dengan permintaan perubahan berhasil dibangun menggunakan *php framework laravel*.
2. Sistem informasi manajemen permintaan perubahan aplikasi SAP pada PT Krakatau Information Technology berbasis *website* sehingga dapat diakses oleh penggunanya melalui internet sehingga proses pengelolaan permintaan perubahan aplikasi SAP dapat dilakukan dengan lebih efektif.

6. DAFTAR PUSTAKA

- Anamisa, D. R. (2022). *Dasar Pemrograman WEB Teori dan Implementasi : HTML,CSS,Javascript,Bootstrap,Codelgniter*. Malang: Media Nusa Creative (MNC Publishing).
- Fathansyah. (2018). *Basis Data*. Bandung: Informatika Bandung.
- Monk, E. F. (2013). *Concepts in Enterprise Resource Planning*. London: Cengage Learning.

Yudhanto, Y. &. (2019). *Mudah Menguasai Framework Laravel*. Jakarta: Elex Media Komputindo.